Robust Indoor Positioning with Lifelong Learning

Zhuoling Xiao

Kellogg College

University of Oxford

A thesis submitted for the degree of

Doctor of Philosophy

December 2014

This thesis is dedicated to my family. Without their support and love I would never be who and where I am today. This work is also dedicated to Linda who has always believed in me and offered great help and love throughout the process. I could not have accomplished as much as I have without her support and understanding.

Acknowledgements

Foremost, I would like to express my sincerest gratitude to my supervisor Dr. Niki Trigoni for the continuous support of my research throughout the years as her Ph.D student. Her motivation, enthusiasm, and immense knowledge have made my Ph.D years a thoughtful and rewarding journey. I have been extremely fortunate to have such a great supervisor whose guidance benefits me not only during the years as a Ph.D student, but the rest of my life as well.

My sincerest gratitude also goes to Dr. Andrew Markham for his selfless time and creative thinking. He is incredibly helpful and inspiring to my research. His rigorous scientific approaches and profound knowledge have helped shape the excellent atmosphere for doing research in our laboratory.

I am also thankful for the excellent feedback from examiners of my reports during these years, Prof. Nando de Freitas, Prof. Ralf Hinze, Prof. Daniel Kroening, and Dr. Ivan Martinovic. I would like to extend my gratitude to Dr. Phil Blunsom for his excellent lectures, interesting discussions, and generous help. Finally, I would like to thank Prof. Jeff Frolik for his prompt help and insightful comments on our joint papers.

I would like to thank the enthusiastic researchers around, particularly Dr. Andrew Symington, Dr. Muzamil Hussain, Dr. Zhengyu Wang, Dr. Bangdao Chen, Dr. Hongkai Wen, Dr. Traian Abrudan, Dr. Lei Xu, Fenglin Liao, Yujiao Zhou, Bo Tang, Xin Zhou, Savvas Papaioannou, and Ronnie Clark, for the helpful discussions on research problems.

I would also like to thank China Scholarship Council who has offered me scholarship to support my study. The Department of Computer Science at University of Oxford has also provided support to complete my study.

Finally, I owe my deepest gratitude to my family and friends for their support over the last three years. I have been so fortunate to have such a group of individuals around without whom I can never finish this thesis.

Abstract

Indoor tracking and navigation is a fundamental need for pervasive and context-aware applications. However, no practical and reliable indoor positioning solution is available at present. The major challenge of a practical solution lies in the fact that only the existing devices and infrastructure can be utilized to achieve high positioning accuracy.

This thesis presents a robust indoor positioning system with the lifelong learning ability. The typical features of the proposed solution is low-cost, accurate, robust, and scalable. This system only takes the floor plan and the existing devices, e.g. phones, pads, etc. and infrastructure such as WiFi/BLE access points for the sake of practicality. This system has four closely correlated components including, non-line-of-sight identification and mitigation (NIMIT), robust pedestrian dead reckoning (R-PDR), lightweight map matching (MapCraft), and lifelong learning. NIMIT projects the received signal strength (RSS) from WiFi/BLE to locations. The R-PDR component converts the data from inertial measurement unit (IMU) sensors ubiquitous in mobile devices and wearables to the trajectories of the user. Then MapCraft fuses trajectories estimated from the R-PDR and the coarse location information from NIMIT with the floor plan and provides accurate location estimations. The lifelong learning component then learns the various parameters used in all other three components in an unsupervised manner, which continuously improves the the positioning accuracy of the system.

Extensive real world experiments in multiple sites show how the proposed system outperforms state-of-the art approaches, demonstrating excellent sub-meter positioning accuracy and accurate reconstruction of tortuous trajectories with zero training effort. As proof of its robustness, we also demonstrate how it is able to accurately track the position regardless of the users, devices, attachments, and environments. We believe that such an accurate and robust approach will enable always-on background localization, enabling a new era of location-aware applications to be developed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

How much time do we spend indoors: working, eating, shopping - at the office, at home, at shopping centers, at airports, at train stations, etc.? The Report to Congress on Indoor Air Quality by the U.S. Environmental Protection Agency has revealed that people spend over 90 percent of their time in indoor environments [1]. Not surprisingly, most people may have similar experience of being overwhelmed by the complicated indoor structure of modern buildings and trying hard to find the destination. A lot more may have wasted a large amount of time finding their friends in public areas like shopping malls, airports, etc. All these problems, can be addressed by knowing the current locations of people within the indoor environments - called indoor positioning. Generally, an indoor positioning system is a solution to locate objects or people inside a building. Whereas GPS is the *de facto* solution for outdoor positioning, indoor positioning still remains an open problem.

This chapter presents a brief introduction to indoor positioning, including illustrative motivating applications, existing problems in this area, dominant challenges for a practical indoor positioning solution, and the outline of the proposed solution. We also introduce the key contributions and publications of this thesis.

The structure of this chapter is as follows. Sec. 1.1 introduces the motivation and key objectives of this thesis. The research problem and the corresponding challenges are formulated and discussed in Sec. 1.2 and Sec. 1.3, respectively. Sec. 1.4 presents the system architecture of the proposed system. The key contributions of this thesis are introduced in Sec. 1.5. Sec. 1.6 lists the resulting publications and Sec. 1.7 describes the organization of the whole thesis.

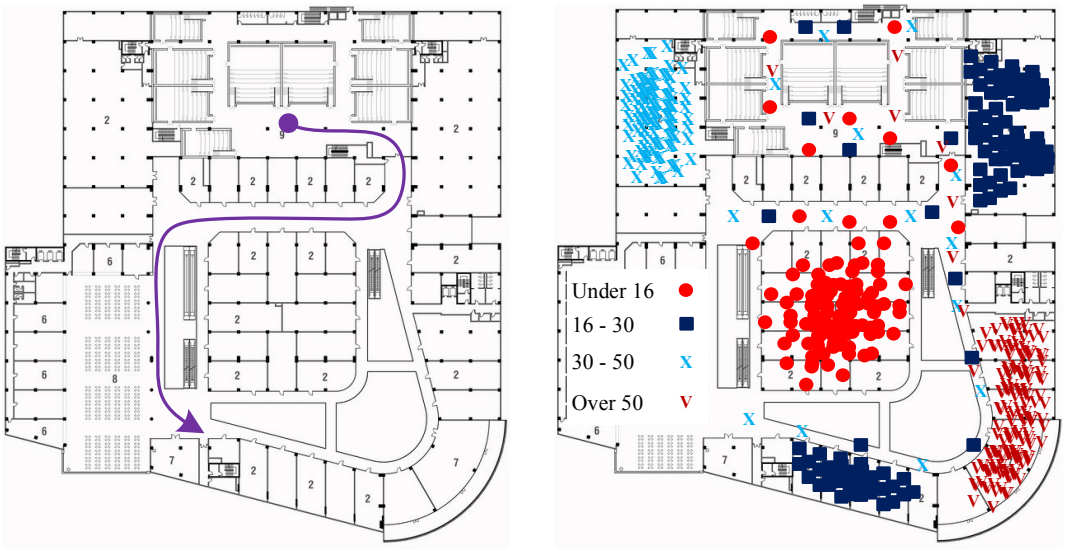
1.1 Motivation

The demand for location-based services (LBS) has been significantly driven by the increasing availability of mobile devices in recent years. As essential context-aware services, LBS deliver services to people leveraging the spatial context of the user. As people spend most of the times indoor, the demand for LBS in indoor environments, e.g. seamless indoor tracking of friends/family and coupon alerts, watching video and surfing the web with relevant location-specific content, indoor navigation, location-based advertising, etc. has grown consistently. The provision of LBS in indoor environments is made possible only when accurate indoor positioning is achieved.

Indoor positioning is a fundamental need for pervasive and context-aware smartphone applications. It opens up a realm of opportunity for businesses and individuals to communicate with people and things around them. With indoor positioning, a large number of useful and interesting things can now happen in indoor environments: indoor navigation, location sharing, social networking, augmented reality, proximity based messaging, workforce location and many more. Typical application scenarios can be exhibition centers, museums, hotels, hospitals, and shopping centers. Below we present some typical examples of application scenarios where indoor positioning is both necessary and useful.

For example, indoor positioning can be important to both visitors and the organizers of exhibitions. Visitors can use the indoor positioning service to navigate to exhibits of interest, to find the facilities in the exhibition center with little effort, and to browse all exhibition-specific offers and vouchers of interest quickly. Meanwhile, the organisers can then analyze the up-to-date needs of visitors and the trends of market. These market data are of tremendous value to the related industry and people of interest as well. It can not only help the industry understand the market trend, but also help the exhibition center to organize the next exhibition more efficiently.

In places like hotels, entertainment complexes, casinos, etc., indoor positioning can not only help people navigate through the confusing indoor structures to their destinations, but also enable enhanced customer service and family tracking. The customer service center can quickly send staff closest to where they are needed and provide service to customers at their current location. For example, it would be extremely attractive for customers to just click the screen before a drink finds him/her by the swimming pool. For families taking vacations, they can easily find each other or the children by sharing locations between them, which is very useful in terms of



(a) Navigation in shopping centers (b) Occupancy analysis in shopping centers

Figure 1.1: The use of indoor positioning in shopping centers.

knowing the locations of their children and quickly reaching them to keep the children safe.

In shopping centres and chain stores, the indoor positioning system can help customers navigate through shops and find items of interest, as shown in Fig. 1.1(a). This gives a competitive advantage to shops, which are targeted to customers on the move, such as those located in airports and other transport hubs. There is also a more general need for all big chains to be able to track customers for advertising purposes. Customers could opt to share their trajectories with the shop owners for a discount, leading to opportunities for location-based advertising. Application specific needs here pertain to the very high accuracy needed to locate people and lead them to find specific items. This is particularly difficult given the dynamics of the environment movement of goods, staff and customers and the density of people in a shop, which makes the camera-based navigation very difficult. More importantly, the shops inside the shopping malls can further analyze the occupancy of their shops, e.g. in terms of ages, as shown in Fig. 1.1(b). This data can be of tremendous value to the shop owners because they are able to have clear understanding of which commodity the customers buy the most, like the most (visit frequently without buying), dislike the most (only a few people visit that section) etc. Then the shop owners can accordingly adjust their selling strategy and maximize their profit.

In hospital scenarios, the indoor positioning system would be useful for patients to navigate through the hospital to get to an appointment, or for people to visit

a friend / relative at hospital; the idea here is to be able to visit someone at the hospital without finding out the specific wing, floor and room. The person who wants to receive a visit from their friend simply pushes a button to share their location with their friend, who can then easily navigate through the hospital to find them. It might also be useful for monitoring ambulatory patients who are encouraged to move around the hospital after an operation, but still require mobile devices for monitoring vital signs and their location if an alarm is raised.

Visitors of museums can use the indoor positioning service to design their travelling route inside the museum to see the most pieces of work using the least time. The museum can provide visitors with context specific information and additional media by knowing their current locations. In addition, for those who are really constrained by time, they can easily approach the art works of their interest through the shortest path without looking around the museum.

Motivated by the huge potential market, academic and industrial labs have been working on indoor positioning solutions for years. Techniques using radio, acoustic signals, magnetic signals, visible light, and other modalities have been proposed to address the indoor positioning problem. In addition, a wide spectrum of algorithms from trilateration, spatial/temporal fingerprinting, inertial tracking, data fusion to cooperative localization and simultaneous localisation and mapping have been developed in recent years for indoor positioning.

Industry has shown great interest in this problem space, because of the large number of location-based services that can be developed once location information is available. These range from location-based advertising, to smart places check-in and augmented reality. The timeliness and industrial interest in indoor localisation is evident by the recent *Accurate Mobile Indoor Positioning Industry Allliance (In-Location)*, which was formed in August 2012 to create *solutions offering high accuracy, low power consumption, mobility, implementability and usability*[1]. There are also other applications that would benefit from low cost, low power tracking solutions such as wearable sensors and smart shoes.

Despite the intensive research and commercial significance in this area, no clear solution has as yet emerged for indoor positioning. The ultimate objective of an indoor positioning system is to provide continuous, reliable and accurate positioning on smartphone class devices. Below we define the problem to be solved, assumptions, and key requirements for an indoor positioning solution to be widely adopted, in the same way as GPS in outdoor environments.

[1]Nokia press release, August 2012

1.2 Problem Definition

Given the research interests and commercial significance of indoor positioning, the research question in this thesis is naturally ***how to design a practical indoor positioning system.***Here the positioning can indicate both continuous tracking — the positioning system provides a continuous trajectory of the object being tracked, or on-demand localization — the positioning system can provide location estimates on demand. The focus of this thesis is continuous tracking but the proposed system also has the NIMIT component for on-demand localization in Chapter 3. Before the design of the positioning system, we need to first identify what makes a practical indoor positioning system, including reasonable assumptions and key objectives.

The main assumption we make is the existence of indoor maps. First of all, positioning does not make any sense to users if they are not provided with a map telling their locations with respect to the map. Therefore, maps are an indispensable part of positioning, anyway. Furthermore, with large companies (like Google) being interested in expanding mapping services indoors, we believe that indoor maps will be increasingly available for use, especially those of large commercial / public buildings, such as shopping centres, hospitals, airports, and so on. A map can be viewed in the broadest sense as a spatial graph which provides constraints. At the simplest level this takes the form of a floor plan of a building. This constrains the allowable motion of a user - people cannot walk through walls and can only enter a room through a door. Other maps (meta-maps essentially) provide additional constraints or features, such as the positions of access points, radio fingerprints, signal strength peaks or distorted geomagnetic fields.

Regarding the key objectives, we claim that a practical indoor positioning solution that can be widely adopted in most application scenarios should be low-cost, accurate, robust, lightweight, and scalable.

- **Low-cost:** The cost of existing indoor positioning solutions comes from two aspects: infrastructure deployment and site survey. Infrastructure deployment installs additional infrastructure like iBeacon, UWB transceivers, etc. which serve as the landmarks for positioning service. Taking into account the number of devices needed for positioning services, the infrastructure deployment which requires initial investment and continuous expensive maintenance, is impractical for most indoor environments. Therefore, instead of requiring special-purpose infrastructure, a good solution should piggyback on existing systems and devices widely available, such as WiFi access points, the earth's magnetic field, mobile

(a) Existing infrastructure in a typical building (b) Sensors available in a typical smartphone

Figure 1.2: Existing infrastructure and devices that can be utilized for indoor positioning.

devices, etc.. Fig. 1.2 shows existing infrastructure in a typical building and various sensors available in a typical smartphone. In addition, the site survey which takes signal signatures in known locations to provide reference points for location estimation is usually achieved manually. The labor cost contributes greatly to the total cost of the positioning system, especially for those techniques that require continuous manual maintenance. Therefore, a practical system should avoid labor-intensive site survey.

- **Accurate:** The accuracy of a measurement system is the degree of closeness of measurements of a quantity to that quantitys true value. In a positioning system the accuracy is usually measured as the distance between the estimated locations and the true locations. The root-mean-square error (RMSE) is a frequently used measure as it represents the sample standard deviation of the differences between estimated locations and true locations.The RMSE aggregates the magnitudes of the estimation errors for various times into a single measure. It is a suitable measure because it provides a quadratic loss function, squaring and subsequently averaging the various errors (prior to applying the square root), which gives considerably more weight to larger errors than smaller ones. This is desirable for location-based services (such as navigation), which can tolerate small errors, but could fail in the presence of large errors. The value of a positioning system largely relies on how accurately it can report the user's location. In most applications like indoor navigation, real time social networks, etc., an accuracy of 0-3 meters is usually acceptable. Understanding the behavior of customers in shops typically requires sub-meter accuracy. Different

sites may have different infrastructure available subject to varying levels of interference affecting achievable accuracy. A good solution is expected to report the most accurate location estimation using information currently available in that application scenario.

- **Robust:** The robustness measures whether the system can work equally well in different conditions, which is another fundamental requirement for positioning systems. It is hard to broadly apply solutions that can work well in some settings but not in others. A good solution should be robust to different conditions including devices, environments, sensor observations, infrastructure, interference, users, user habits, etc.

- **Lightweight:** The computational complexity of a positioning system largely determines whether the positioning system can run on mobile devices or only on the server. The problem with server-based positioning systems is that they can support only a limited number of users simultaneously due to the massive data communication incurred between mobile devices and the server. Therefore, a truly practical positioning system should be sufficiently lightweight to run as a standalone application on each mobile device.

- **Scalable:** The scalability of a positioning solution is its ability to be deployed in new environments or to handle a growing number of positioning objects in a capable manner. It is a crucial metric for a positioning system because it actually determines the size of the market it can be deployed into. Given the additional cost in terms of money and time needed to deploy extra infrastructure, solutions that rely on extra infrastructure are usually less scalable than solutions that are infrastructure-free or using existing infrastructure. Similarly, systems that need extensive site survey effort are less scalable compared with systems requiring no site survey.

In summary, the research question that will be addressed in this thesis is **how to design a practical indoor positioning system that is low-cost, accurate, robust, lightweight, and scalable**. Taking into account what has been defined as practical, these objectives raise significant research challenges discussed below.

1.3 Research Challenges

To build a truly practical indoor positioning system which is low-cost, accurate, robust, lightweight, and scalable, the challenges are significant. The major challenges are the limitation of infrastructure and devices available, and the variations in available infrastructure, users, devices, users, and usage patterns.

Infrastructure limitations: The low-cost objective requires the positioning system to utilize only existing infrastructure designed for other purposes like communication instead of localization. The typical problem with existing infrastructure is coverage and reliability. For example, WiFi access points deployed within a building may not necessarily cover every corner of the whole building, which degrades the performance of localization methods using only WiFi signals. Other signals like the magnetic distortions of the earth are useful for positioning in metal-rich indoor environments, but are less informative in environments without metallic objects necessary to create the magnetic variations.

Device limitations: The low-cost objective also poses a requirement on the devices available for use by the positioning system. It is neither practical nor efficient to ask people to carry extra devices along only for positioning purposes. Therefore, the best candidate devices for positioning should be what people already have in their pocket, e.g. mobile phones, pads, or wearable sensors. However, one big challenge with these devices is the availability and reliability of sensor data obtained. Take the inertial measurement unit (IMU) widely available in mobile devices as an example. The motion of the device, as captured by the IMU, actually provides little information about the motion of the user because the user may just shake the hand without really moving forward. Unfortunately, the signals captured by the IMU when the user is walking (which contributes to the locomotion of the user) are nearly identical to the signals when the user is actually shaking their hands (which does not contribute anything to the locomotion of the user). Another typical example is the use of WiFi signals for positioning. Typical mobile devices can only collect the received signal strength (RSS) from RF signals such as WiFi and Bluetooth. The challenge of using RSS for positioning is that the collected RSS is not always reliable. The reason lies in the well-known multi-path effect, especially in non-line-of-sight (NLOS) conditions. Therefore, to use noisy RSS in the absence of existing fingerprinting database, we need to identify and mitigate the non-line-of-sight conditions using only RSS. This is very challenging because it is likely that we get the same RSS from a NLOS location near the transmitter and a LOS location far from the transmitter.

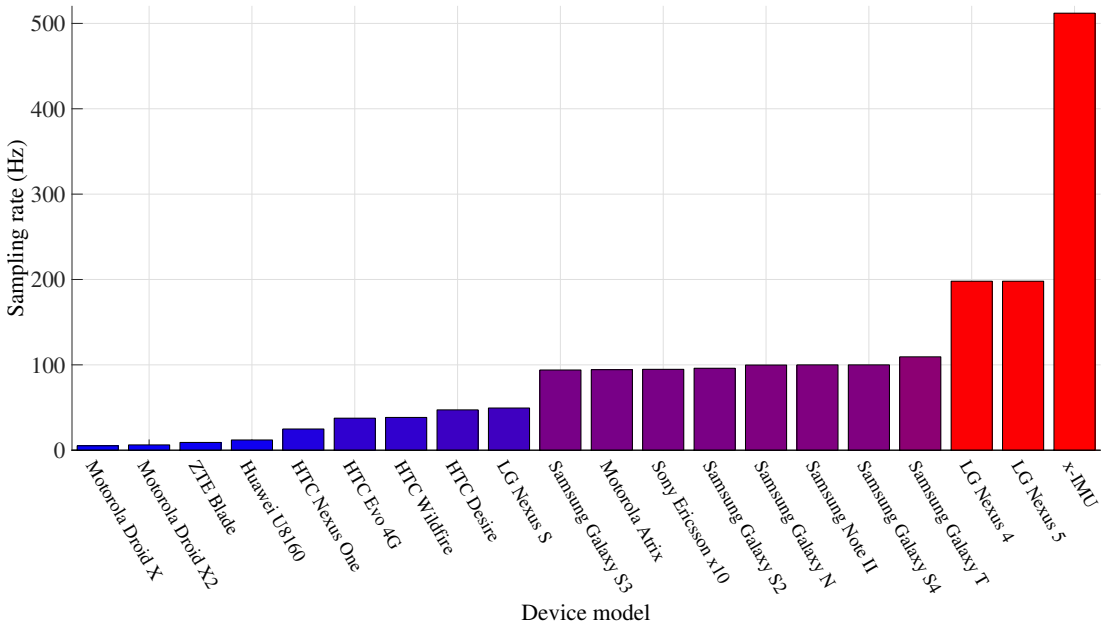


Figure 1.3: Accelerometer sampling rate of different device models. It is observed that the sampling rates differ significantly, ranging from 10 Hz to 500 Hz.

Environment variability: For a positioning system to be accurate and scalable, it should be able to accommodate various infrastructure available in different environments. For example, we may find WiFi available in one site and Bluetooth Low Energy (BLE) in another. There may be other types of infrastructure available for use in different sites, such as visible light, vision, power line, generated magnetic field, visual landmarks, etc. Different modalities vary significantly in terms of usage model, coverage, reliability, etc. Therefore, to build an accurate positioning system, we need to flexibly fuse all these different infrastructure and pertinent fingerprinting databases into the system, which poses great challenges to the flexibility and robustness of the system.

Device variability: It is particularly challenging to account for device variation in order to build a robust indoor positioning system. Sensors on different mobile devices, such as mobile phones, pads, and wearable sensors, differ significantly in terms of price, availability, functionality, quality (accuracy, reliability, etc.) and performance. The IMU sensors are typical examples to illustrate the difference. As shown in Fig. 1.3, the sampling rate of the IMU sensors ranges from 5 Hz (Motorola Droid X) to 512 Hz (x-IMU). The quantization precision also differs significantly. Another example involves the WiFi scanners of different devices. The RSS received by different devices from the same access points in the same location can differ up to 15 dBm. In addition, the variance of the RSS samples can also be very different for different devices.

Users and usage pattern variability: The ability to work similarly well with users of different features (height, age, gender, etc.) and usage patterns (different attachments of mobile devices, different usage habits, etc.) is the fundamental requirement of robustness. This is especially important and challenging for inertial sensor based tracking such as pedestrian dead reckoning. Since the displacement of the user can only be inferred from arbitrarily mounted mobile devices, the various features of the user and usage patterns have great impact on the displacement estimation. An illustrative example is the impact of the height of the user on the step length. The locomotion of users of different height can be very different even if we observe nearly identical acceleration signals from the handheld mobile devices. In addition, it is especially challenging to track the heading of the user if the mobile device is held differently, e.g. when the mobile device is being rotated while walking. A universal positioning system that works for individuals with different features and usage patterns is a key component.

1.4 Proposed System

In this thesis, we propose LL-Tracker[2], an indoor positioning system that is low-cost, accurate, robust and scalable. The block diagram of the system architecture is shown in Fig. 1.4. The input of the system includes the floor plan and sensor data from sensors widely available in various mobile devices. Typical examples of these sensors are IMU sensors (accelerometer, magnetometer, and gyroscope) and WiFi/BLE scanner. Other sensors such as barometer, light sensors, microphone, near field communication (NFC) sensors, etc. available in some mobile devices can also be utilized in this system. The proposed indoor positioning system comprises four main components:

- Non-line-of-sight (NLOS) identification and mitigation
- Robust pedestrian dead reckoning (R-PDR)
- Lightweight map matching
- Lifelong learning

The output of the system is the estimated location of the user, and possibly signal maps associated with the floor plan provided.

[2]Lifelong Learning Enabled Tracking System

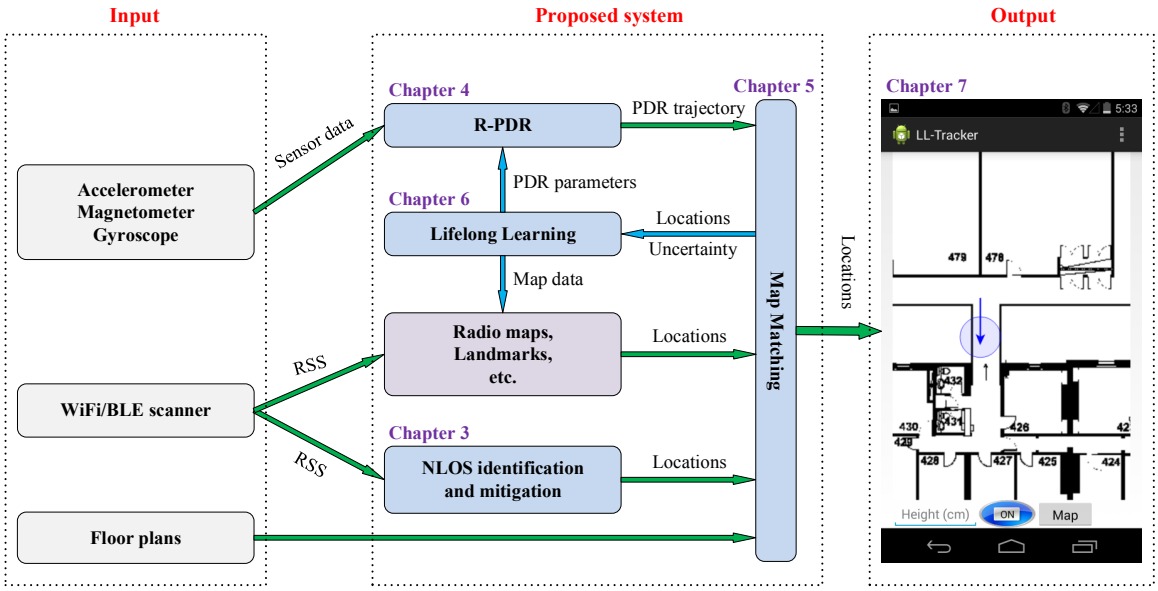


Figure 1.4: System architecture of the proposed system.

Data flows in the system as follows. Two typical inputs are taken into account in the system architecture: the received signal strength (RSS) from WiFi/BLE scanners and the IMU data comprised of acceleration, magnetic field, and angular velocity obtained from IMU sensors. The RSS from various access points can be either passed to a radio map if available or to the NLOS identification and mitigation component to estimate coarse user locations. These coarse locations then flow to the map matching component. The other input, IMU data, are fed to the R-PDR component which estimates the PDR trajectory – a sequence of <distance, heading> pairs – and feeds the PDR trajectory into the map matching component. The map matching component then estimates and reports the location of the user, and provides the estimated trajectory and uncertainty metrics to the lifelong learning component. The lifelong learning component learns various maps such as radio maps, landmarks, etc. associated with this floor plan to build/improve these maps, and PDR parameters to improve the PDR accuracy. Note that the lifelong learning component forms a feedback loop in the proposed system, which can finetune the crucial parameters in the system and therefore guarantees the robustness and accuracy of the positioning system. The core components are discussed in detail below.

1.4.1 Non-line-of-sight Identification and Mitigation

Recall that typical mobile devices can only collect the received signal strength (RSS) from RF signals such as WiFi and Bluetooth. The challenge of using RSS for position-

ing is that the collected RSS is not always reliable. The reason lies on the well-known multi-path effect, especially in non-line-of-sight (NLOS) conditions. Therefore, to use noisy RSS in the absence of existing fingerprinting database, we need to identify and mitigate the non-line-of-sight conditions using only RSS. However, this task is quite challenging because the same RSS could be received from a NLOS location near the transmitter and a LOS location far from the transmitter.

NLOS identification and mitigation techniques developed to date have been primarily investigated for ultra-wide band (UWB) signals. Some general NLOS mitigation algorithms that can be possibly applied to WiFi signals have impractical assumptions that the LOS anchors outnumber NLOS anchors.

Given the fact multiple RSS received at the same location can provide much more information than a single RSS which tells little about the LOS/NLOS conditions, this component explores various features from multiple RSS collected at the same location and compare the difference in LOS and NLOS conditions. It is observed that the distributions of RSS samples collected in LOS and NLOS conditions are different, which can be leveraged to distinguish between LOS/NLOS conditions and estimate the transmitter-receiver distance. Then this component develops various algorithms based on machine learning and hypothesis testing to identify the LOS/NLOS conditions and mitigating the impact of NLOS conditions on distance estimation. These algorithms differ in terms of assumptions, training cost, identification/mitigation accuracy.

1.4.2 Robust Pedestrian Dead Reckoning

Recall that only mobile devices like smartphones and smartwatches are practical for indoor positioning, the pedestrian dead reckoning (PDR) based on these mobile devices is very challenging. Unlike the foot-mounted PDR where the velocity of the device is frequently reset using zero velocity update, handheld PDR can hardly find chances to reset and calibrate IMU sensors, which leads to the accumulation of errors. The variations of users and usage patterns, e.g. various motion patterns of users and different placements of the device, further embarrass the PDR on handheld devices by blurring the boundaries between inertial signals with and without useful locomotion.

Existing PDR systems on mobile devices comprise four main steps: a motion recognition algorithm to distinguish between different motion modes like hand swinging, texting, etc., a step detector to detect steps from the inertial measurements, a step length estimation algorithm to estimate the step length of the pedestrian, and

a heading estimator to determine the walking heading of the user. However, the existing algorithms are limited to either fixed placement or orientation of the device.

To address these problems, this thesis exploits the nature of bipedal motion and the physical constraints of human walking, and proposes robust pedestrian dead reckoning (R-PDR). Unlike existing systems which attempt to recognize different placements through sensor data, R-PDR instead simply determines whether the motion of one or both legs impact the measurements. This comprehensive way of classifying human motions can accommodate all possible motions made by human walking and greatly facilitate the step detection because acceleration signal and the device orientation of each step have strong patterns which are easy to recognize. In addition to the comprehensive motion classification, we also leverage the fact that gyro sensors can be trusted within a short period of time to accurately estimate the gravitational vector, which thereby corrects the long-term drift of gyro sensors and accurately estimate the device orientation. The fact that the walking speed of humans should be the same before and after each step to calibrate the sensor drift. This has been demonstrated to be very effective in removing sensor drift and improving the step detection accuracy. The proposed R-PDR has similar steps like existing PDR algorithms. However, it is intended for use in arbitrary device placement and orientation.

1.4.3 Lightweight Map Matching

PDR has the benefit that it provides continuous trajectories that are accurate within a short period of time. However, there are two challenges in putting PDR into practice. One challenge is that the drift of PDR can be significant in long run. The other challenge is that it requires external information to obtain the global locations. These two challenges can be addressed by matching the trajectory into the floor plan or radio maps, called map matching.

Existing map matching algorithms employ Bayesian filters, e.g. hidden markov models (HMMs), particle filters, etc. to match the PDR trajectory with the floor plan or radio maps. However, Bayesian filters have limitations in modelling temporally or spatially correlated sensor observations. To improve the tracking accuracy, it is crucial to keep track of the sensor bias, especially for low-cost IMU sensors on mobile phones. In addition, the computational complexity of Bayesian filters is usually high. To make a practical positioning system, the map matching algorithm should be lightweight enough to run on mobile devices.

Therefore, a good map matching algorithm for indoor positioning should be lightweight and flexible to model temporally/spatially correlated observations. To meet these

requirements, this thesis proposes MapCraft, a map matching algorithm using conditional random fields (CRFs) to fuse floor plans, various radio maps, and sensor observations together. The CRF model is particularly well suited to this sequential problem because it allows us to flexibly define *feature functions* that capture the extent to which observations support states and state transitions, given map constraints. The location estimation of MapCraft is achieved by Viterbi decoding over a sparse graph which is lightweight in terms of memory usage ($< 5MB$) and execution time (< 10 ms).

1.4.4 Lifelong Learning

Given the variations of devices, users, and usage patterns, parameters concerning PDR and map matching often require context-specific tuning. However, it is not likely for each user to finetune various parameters before positioning actually takes place. The way to address this challenge is to learn these parameters in an unsupervised manner, rather than manually tune them or assume them as input.

Little work so far has investigated this research challenge. Even the state-of-the-art work only proposes the learning of signal maps and simple PDR parameters. The major limitation of existing learning approaches lies on the fact that no comprehensive learning framework has been proposed that can be applied to the learning of parameters used in PDR and map matching.

To this end, this thesis exploits the interplay between the PDR and the map matching components. Instead of the unidirectional data flow which feeds PDR output into a map matching technique, the proposed lifelong learning component establishes a reverse link between the two components. It augments the functionality of the map matching component to learn context-specific parameters including maps (radio maps, landmarks, etc.) to build additional maps for positioning, and PDR parameters fed back to the R-PDR component to improve its robustness and accuracy. The idea of lifelong learning is broadly applicable to improve the performance of any PDR implementation that requires parameter tuning.

1.5 Key Contributions

Having briefly discussed the proposed system architecture and each individual component, we summarize the key contributions of this thesis as follows:

1. Non-line-of-sight identification and mitigation to improve the location estimation using WiFi/BLE signals.

- *Novel methods of NLOS identification and mitigation*: The NLOS identification and mitigation leverages only RSS measurements from WiFi/BLE signals, which greatly improves the potential for RSS-based localization.

- *Novel feature extraction*: The features extracted from collected RSS measurements are shown to be effective in LOS/NLOS discrimination.

- *Effective NLOS identification*: Two classifiers based on machine learning and a third based on hypothesis testing to identify the NLOS conditions, which are shown to perform well even in environments other than these used for training.

- *Effective NLOS mitigation*: Two regressors based on machine learning and a third which fuses hypothesis testing with propagation models to accurately estimate the transmitter-receiver distances in NLOS conditions.

- *Extensive real world experiments*: This thesis presents the results of several empirical studies in different environments and show the benefits of the proposed techniques in terms of improving position accuracy compared to existing methods.

2. Robust pedestrian dead reckoning (R-PDR) which enables the pedestrian tracking with unconstrained mobile devices. Specifically, three novel capabilities are proposed to make the algorithm robust to user, device and phone placement variability, including:

- *Comprehensive motion classification*: The proposed novel comprehensive way of classifying human motion exploits the physical nature of the bipedal motions and thereby accommodates all possible motions. This classification greatly facilitates the motion recognition and step detection.

- *Rotationally invariant orientation tracking*: The proposed rotationally invariant orientation tracker can accurately track the orientation of the mobile device regardless of device rotations, which then builds the foundation for other steps of robust pedestrian dead reckoning.

- *Accurate acceleration estimation*: Leveraging the physical constraints of human walking, the proposed acceleration estimation technique further improves the projection of acceleration in the coordinate frame of the earth, which makes it possible to identify whether a motion corresponds to a real step.

- *Extensive real world experiments*: This thesis shows how we can greatly increase the accuracy of PDR systems by combining this suite of techniques in extensive real world tests with unconstrained placements.

3. Lightweight map matching technique to flexibly fuse different information including floor plan, radio maps, PDR trajectories, and location estimated from WiFi/BLE signals.

- *Lightweight map matching*: The proposed map matching technique enables computationally efficient, real-time map-matching using sensor data from multiple sources and a floor plan.

- *Accurate indoor tracking with noisy trajectories*: We demonstrate excellent performance even in the presence of bias, noise and distortions. As an extreme case, we show accurate tracking can be obtained from gyro-free dead reckoning.

- *Extensive real world experiments*: We achieve high tracking accuracy in multiple environments (office, museum, market) with the proposed map matching technique. It outperforms existing map matching techniques even without training.

4. Lifelong learning framework which leverages the map matching technique and establishes a feedback loop to tune PDR parameters or build radio/distortion maps.

- *Robust lifelong learning framework*: We propose the general principle and framework of lifelong learning to ease the effort of parameter tuning, which is then applied to different learning scenarios.

- *Illustrative learning examples*: We applied the learning technique to the learning of sensor bias, heading bias, step length, and environment, which yield good learning results.

- *Extensive real world experiments*: We have demonstrated that the proposed lifelong learning algorithm is very effective in learning the PDR parameters and radio/distortion maps, which significantly improves the the tracking accuracy.

5. Extensive real world experiments to test the accuracy and robustness of the whole system.

- *Extensive real world experiment design*: We designed and performed extensive real world experiments in three different sites.

Table 1.1: Relation between chapters and publications.

Chapters	3	4	5	6
Publications	[1], [2]	[3]	[4], [5]	[6]

- *Comparative accuracy evaluation*: We compare the proposed indoor positioning system with other state-of-the-art end-to-end indoor positioning systems.
- *Extensive robustness evaluation*: We evaluate the robustness of the proposed system against various context variations, including user variation, device variation, attachment variation, and environment variation.

1.6 Publications

The list below contains published work that fully, or partially relates to the contents of this thesis. The technical chapters (Chapters 4 to 6) are closely related to these publications. Specifically, the major dependence relations between the technical chapters of this thesis and the major publications are described in Table 1.1.

[1] **Zhuoling Xiao**, Hongkai Wen, Andrew Markham, and Niki Trigoni, "Non-line-of-sight identification and mitigation using received signal strength," *IEEE Transactions on Wireless Communications*, 2015.

[2] **Zhuoling Xiao**, Hongkai Wen, Andrew Markham, Niki Trigoni, Phil Blunsom, and Jeff Frolik: "Identification and mitigation of non-line-of-sight conditions using received signal strength," in *Proc. 9th IEEE International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob'13)*, (Lyon, France), pp. 667-674, 2013.

[3] **Zhuoling Xiao**, Hongkai Wen, Andrew Markham, and Niki Trigoni, "Robust pedestrian dead reckoning (R-PDR) for arbitrary mobile device placement". in *Proc. 5th International Conference on Indoor Positioning and Indoor Navigation (IPIN'14)*, (Busan, Korea), pp. 1-10, 2014.

[4] **Zhuoling Xiao**, Hongkai Wen, Andrew Markham, and Niki Trigoni, "Lightweight map matching for indoor localization using conditional random fields," in *Proc. 13th International Conference on Information Processing in Sensor Networks (IPSN'14)*, (Berlin, Germany), pp. 131-142, 2014. (**Best Paper award**)

[5] **Zhuoling Xiao**, Hongkai Wen, Andrew Markham, and Niki Trigoni, "Indoor tracking using undirected graphical models," *IEEE Transactions on Mobile Computing*, 2015.

[6] **Zhuoling Xiao**, Hongkai Wen, Andrew Markham, and Niki Trigoni, "Robust Indoor Positioning with Lifelong Learning," *IEEE Journal on Selected Areas in Communications*, 2015.

1.7 Thesis Organization

The remainder of the thesis is organized as follows. Chapter 2 introduces the related work of this research. Chapter 3 develops various techniques to identify and mitigate the non-line-of-sight conditions. Chapter 4 proposes the robust pedestrian dead reckoning (R-PDR) algorithm. Chapter 5 formulates the flexible map matching framework that can accommodate various sensor observations and map constraints. Chapter 6 presents the lifelong learning framework and examples on how to learn different parameters in an unsupervised manner. Chapter 7 discusses the implementation of the indoor positioning system, extensively evaluates its performance, and compares it with state-of-the-art competing approaches. For the sake of readability, this chapter only evaluates the performance of the whole positioning system while the performance of individual components is evaluated in their corresponding chapters (3 - 6). Chapter 8 summarizes the whole thesis and discusses ideas for future work.

Chapter 2

Background

The idea of locating people and objects in indoor environments dates back to the 1990s when the Real Time Locating System (RTLS) [2] was deployed in hospitals to track equipment like wheel chairs, and warehouses to track merchandise, etc. To track objects and people, RTLS initially deployed active RF tags on the target. Later WiFi tags were deployed on objects being tracked and the locations were estimated from at least three nearby access points via triangulation. Similar systems like the Active Badges [3] were also deployed in the 1990s to track people and objects in indoor environments.

The recent few years have witnessed a dramatic trend of increase in the research and commercialization of indoor positioning solutions. Allied Business Intelligence, Inc. has forecasted the indoor localization market to reach $4 billion in 2018. The major role of indoor positioning in venues like exhibition centers, shopping malls, hotels, airports, and hospitals is to provide location-based content and services to mobile device users. A large number of events on indoor positioning, both academic and industrial, happened in the last few years. Figs. 2.1 and 2.2 show some typical and well-known academic and industrial events on indoor positioning in recent years.

As a result of the great interest in indoor positioning from academia and industry, a wide variety of indoor positioning solutions have been proposed. This chapter first surveys existing end-to-end indoor positioning solutions, both academic and industrial, in Sec. 2.1, and then focuses the discussion on the related work for practical indoor positioning solutions, including Pedestrian dead reckoning (PDR) in Sec. 2.2, RSS-based positioning in Sec. 2.3, and data fusion in Sec. 2.4. Since further improvement to the positioning accuracy can be achieved by the learning of various parameters involved, this chapter also discusses the related work in the parameter learning in Sec. 2.5. Finally, Sec. 2.6 summarizes this chapter and briefly discuss the limitations of existing techniques.

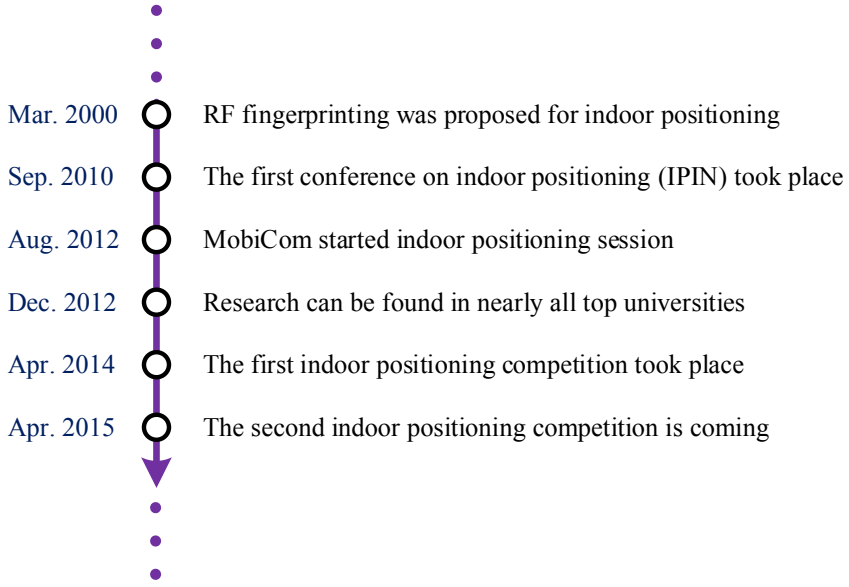


Figure 2.1: Some typical academic events on indoor positioning.

2.1 Indoor Positioning Solutions

Existing end-to-end indoor positioning solutions fall into two categories: infrastructure-based and infrastructure-free. The infrastructure-based category needs to deploy additional infrastructure such as RF tags, Bluetooth Low Energy (BLE) tags, ultra-wide-band (UWB) transceivers, and cameras, which are then used as anchors to estimate the location of the mobile objects. In contrast, the infrastructure-free category does not require any extra infrastructure or only uses existing infrastructure widely available like WiFi access points. Typical indoor positioning solutions are shown in Fig. 2.3.

Infrastructure-based solutions are generally more accurate and robust than infrastructure free solutions because the additional infrastructure is properly designed for localization purposes. However, the deployment of extra infrastructure adds an extra cost in both deployment and maintenance. In contrast, infrastructure-free solutions are low-cost and ubiquitous.

2.1.1 Infrastructure-based Indoor Positioning Systems

Typical infrastructure-based solutions include, among others, RF tags [4–12], BLE tags [13–15], UWB transceivers [16–25, 25–36], and artificial magnetic fields [37–47].

The early stage indoor positioning system in 1990s uses RF tags, such as RTLS [2], Active Badges [3], etc. These tags are deployed in both buildings (anchors) and the

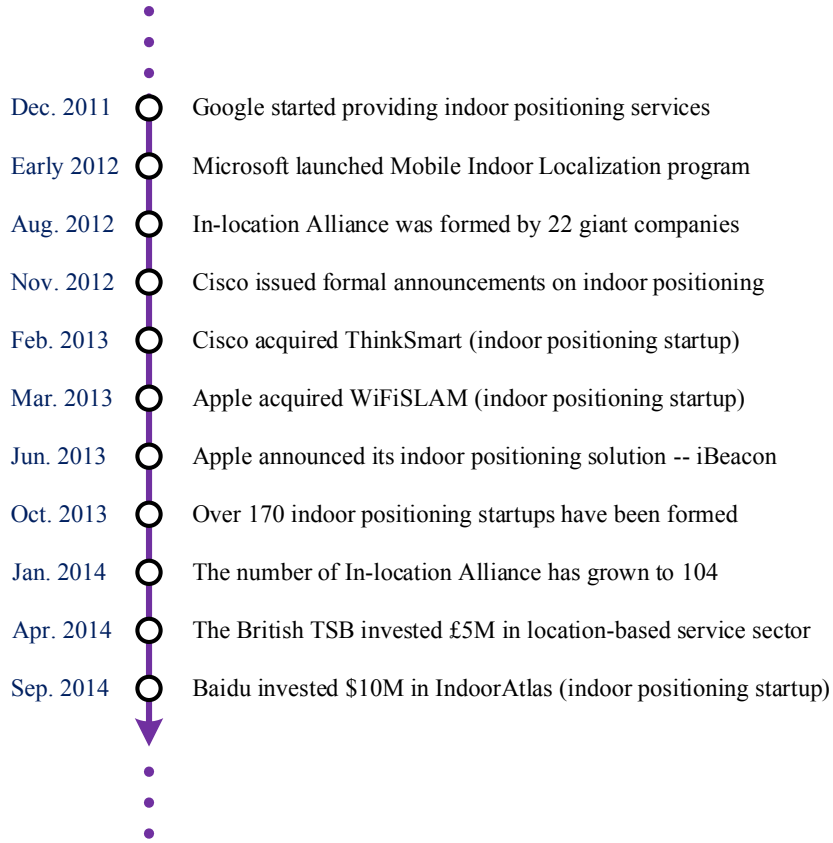


Figure 2.2: Some major industrial events on indoor positioning.

objects been tracked (agents). The location of an agent is then inferred from its proximity to anchors whose locations are recorded when deployed. Other positioning techniques based on RF tags include trilateration [7, 8, 11], angle of arrival [4], time of arrival [6, 11]. Localization accuracy can be improved using Bayesian filtering [9] or machine learning [10]. Passive tags are usually deployed on objects being tracked, but recently the performance of semi-passive tags has also been studied [12]. The advantage of positioning using RF tags is that they are not as expensive as their counterparts like UWB and they can also be easily deployed and used. The major disadvantage lies in the sensing range of RF tags which is usually very short and thus a large number of tags need to be deployed before positioning can take place.

Started by Nokia Inc. since 2006, positioning with Bluetooth (Low Energy) quickly became popular after Apple Inc. announced its indoor positioning solution – iBeacon [13] in June 2013. iBeacon is a proximity-based positioning solution using BLE tags instead of RF tags. However, the underlying technique is also proximity, similar to RF tags. Due to the significant impact of Apple Inc. in IT industry over the world, the iBeacon solution has been adopted in a number of shops in North America

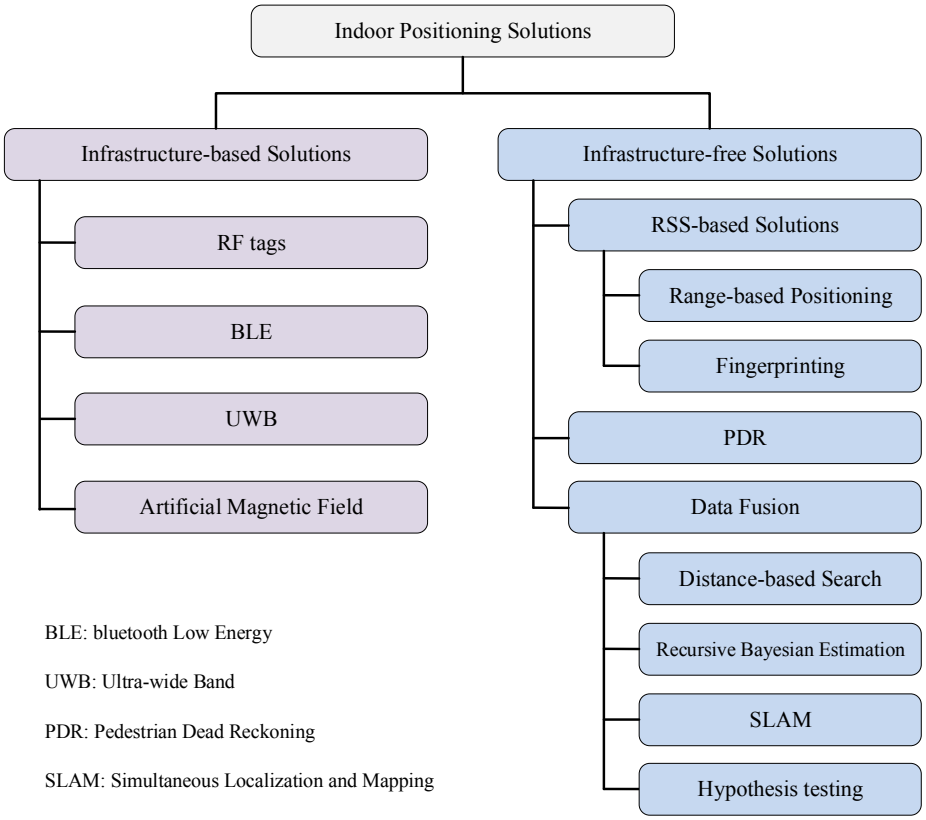


Figure 2.3: Typical indoor positioning solutions.

and China. Compared with standard RF tags, BLE tags have longer transmission range and higher price.

Due to the high temporal resolution of UWB signals, it is possible to capture the time of arrival between two objects close to each other [31–36]. UWB-based indoor positioning utilizes this feature of UWB signals to measure the distance between various transmitters and the receiver, and then estimate the location of the receiver using trilateration. It requires one of the most expensive infrastructure and devices among all techniques discussed in this thesis. However, UWB-based positioning systems can usually provide up to centimeter accuracy [18–20, 22] or even millimeter accuracy [23, 29], making it suitable for special positioning scenarios that need extreme accuracy like emergency and rescue [28]. The typical problem that UWB-based solutions encounter is that distance estimation performance is rather poor in conditions when some obstacles lie in between the transmitter and the receiver, called non-line-of-sight conditions [30]. Therefore, it requires quite high density of transmitters. Its performance also needs to be improved in places like corners where the difference between the direct path and the various reflected paths is not significant [48].

The potential of artificial magnetic fields in positioning has been studied for over a decade [38, 42]. The ability of the magnetic field to penetrate obstacles like rocks and walls, due to its extremely large wave length (km level), makes it widely researched and applied to wild animal tracking [49–51] and indoor tracking as well [37, 39–41, 43–47]. The typical magnetic field transmitter consists of three orthogonal coils which respectively generate magnetic fields (X, Y, and Z). Note that the receiver has identical coil structure as the transmitter (X', Y', and Z'). Then the receiver is able to get nine signals from the transmitter which can determine the 3D positions of the receiver relative to the transmitter with a hemispherical ambiguity. The ambiguity can be solved by either constraining the areas of positioning or using additional transmitters [51]. Given the fact that the magnetic field has extremely fast attenuation with an equivalent path loss exponent of six in free space [52], the accuracy of positioning systems based on artificial magnetic fields can be centimeter level regardless of non-metal obstacles. However, metallic objects in indoor environments can greatly distort the received magnetic fields and result in significant errors. In addition, due to its large attenuation factor, it is difficult to increase the transmission range (currently 20 meters at most) without substantially amplifying the transmission power, which might be a problem for the health of people involved [53].

Other signals like Ultrasonic [54] and infrared provide alternative options for localization. But these signals are generally vulnerable to external interference ubiquitous in indoor environments. In addition, due to the short transmission range and non-line-of-sight problem, a large number of such sensors need to be deployed to provide full coverage in a building. Gu et. al [55] and Sana [56] comprehensively surveyed numerous indoor localization systems and made comparisons between their cost and performance. Other existing localization algorithms of this category including TOA, TDOA, etc. were investigated and their advantages and disadvantages discussed in [57].

2.1.2 Infrastructure Free Indoor Positioning Systems

Generally speaking, there are three types of infrastructure free (including using existing infrastructure) indoor positioning system, namely pedestrian dead reckoning (PDR), RSS-based positioning, and the fusion of previous two. In this section, we only give a high level overview of these techniques with more details coming later.

PDR does not rely on external infrastructure but instead estimates the current position using the previous position and the estimated displacements since last update. However, PDR cannot provide location without extra information, e.g. the

start point. In addition, the drift of inertial sensors used in PDR, especially low-cost inertial sensors, accumulates quickly, which leads to significant positioning error within a short period of time.

In comparison, RSS-based positioning uses the signal strength of either the ubiquitous magnetic field of the earth or the WiFi access points widely available to estimate the position. It can provide absolute location estimate without long term drift problem. However, it usually takes large effort to first survey the RSS of the site before the positioning actually takes place. In addition, the accuracy of RSS-based positioning is usually compromised.

Fortunately, these two techniques are complementary in terms of error behavior, if we take a closer look. Therefore, to address the shortcomings of both techniques, a typical way is using the location estimated from RSS-based positioning as periodic corrections to compensate for the long term drift of PDR. As a result, a hybrid indoor positioning system can be derived by fusing PDR and RSS-based positioning. The fusion can also take into account other external information information sources like the radio map or the floor plan to help improve the positioning accuracy. This is probably the most popular and promising indoor positioning technique which has attracted most attention from researchers from both academia and industry.

In what follows, we will introduce the related work of these three types of practical infrastructure free indoor positioning system in detail in Secs. 2.2, 2.3, and 2.4.

2.2 Pedestrian Dead Reckoning

Pedestrian dead reckoning utilizes measurements from inertial sensors, including accelerometer, magnetometer, and gyroscope, to derive the current location and orientation in a cumulative manner. For foot-mounted IMU, this drift can be addressed by zero velocity update (ZUPT) which resets the velocity and the filter every time the foot hits the ground, as shown in Fig. 2.4 (source [58]). Therefore, the displacement could be obtained from double integration of the accelerometer readings and the orientation is determined by the acceleration direction and rotation vector provided by the gyroscope.

Unlike foot-mounted inertial sensors, it is infeasible to infer the displacement from double integration for general mobile devices. To address this issue, four typical steps are then involved in pedestrian dead reckoning with handheld devices, including motion recognition, step detection, step length estimation, and orientation track-

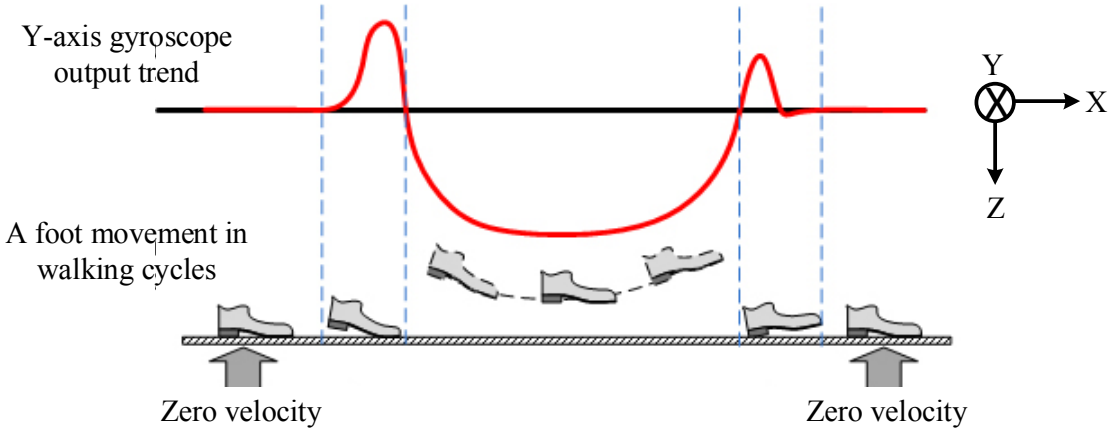


Figure 2.4: A foot movement in normal walking cycle. Two zero velocity moments can be observed at the beginning and end of each stride. These zero velocity moments can be utilized to reset the accumulated double integration error of acceleration signals.

ing/heading estimation. Various algorithms have been developed to address these four issues, especially with hand held mobile devices like smartphones.

2.2.1 Step Detection

A wide variety of algorithms have been developed to identify steps from inertial signals so far [59]. Typical step detection algorithms have been developed to accurately identify specific events for data segmentation, for instance, peaks, zero crossing, spectral analysis, and auto/cross correlations, etc.

- *Peak detection*: The toe-off and heel-strike events are usually associated with sharp changes in vertical or the magnitude of acceleration which are the targets for typical peak detection algorithms [60–62]. However, each heel-strike could come with multiple peaks, especially when the mobile device is in a swinging hand or attached low, e.g. a trousers pocket, which significantly increases the algorithm complexity.

- *Zero-crossings*: Another way of identifying the cyclic patterns in acceleration is the detection of zero crossings [63–65]. This algorithm is simple to implement but the accuracy is not satisfactory due to the false positive zeros crossings from numerous events other than walking.

- *Spectral analysis*: This category attracts the most attention recently due to its robustness [66–69]. Algorithms of this category first convert the acceleration signal to frequency domain using different algorithms like Fourier trans-

form [66, 67], short time Fourier transform [68], or wavelet transform [69], and then the dominant peak of the signal in frequency domain is identified as the step frequency. Windowed acceleration signal (magnitude) is common choice for people working with spectral analysis [67]. As a result, the online step detection would make this algorithm very difficult and computationally intensive. The computation complexity of this category is the most intensive compared with other step detection algorithms discussed in this section.

- *Auto/cross correlations*: Since each individual human has a surprisingly consistent walking pace [70], the strong periodicity of the acceleration signal from such walking behavior makes it possible to extract the cyclic pattern with mean-adjusted autocorrelation of the sensor data sequence or cross correlation with prepared templates – sample sequence of sensor data from training phase [71].

These algorithms are mostly applied to acceleration or angular velocity signal, or their combinations [69, 72]. To handle unconstrained device placement, a number of techniques have been proposed, such as motion recognition which attempts to classify different motion patterns (such as hand swinging, texting, running, etc.) based on waveform features [73, 74].

The most crucial underlying assumption that existing algorithms make is that people do not generate periodical acceleration signals while not walking, such as a hand swing without the user really moving forward. This is because the two actions (a step and a hand swing) can actually generate an identical acceleration signal patterns, especially when the step detection algorithms are applied to the magnitude of the acceleration [75]. The situation would be exacerbated with less constrained motions like crawling, stumbles, side-steps, shuffles, etc. [59], which pose great challenge to motion recognition algorithms.

2.2.2 Step Length Estimation

Unlike the foot-mounted inertial navigation system [76], double integration of the acceleration to get the step length is not applicable to handheld devices, reasons being: 1) zero velocity update (ZUPT) is not available for handheld devices because only the part under the ankle can be regarded as (almost) static when one foot is planted on the floor during walking period while it is not common behavior for people to put a handheld device under the ankle; 2) the output of the low-cost accelerometer on mobile devices are too noisy to produce any sensible results through double integration.

To address the step length estimation problem, the simplest approach is to make it a constant [77] because pedestrians have surprisingly constant step length with a natural walking pace [70]. However, when walking with others, people are likely to adjust their natural walking pace, which leads a significant change of the step length [59].

The most common approaches people are using to estimate the step length is a linear [69, 78] or nonlinear [61, 62] model to relate the step length to variables like pedestrian height, step frequency, acceleration variance, etc. These models are easy to implement and thus widely adopted in practice.

An alternative of the linear/nonlinear model is regression. Machine learning regression techniques like support vector machine [74, 75], neural networks [79] are applied based on various features extracted from the sensor data. Some other systems also dynamically tune the step length by optimizing the trajectories with the aid of maps constraints [77, 80, 81].

2.2.3 Orientation Tracking

Orientation tracking (heading estimation), tackles the problem of continuously estimating the orientation of the mobile device, relative to North. Typical orientation tracking algorithms with inertial data are based on indirect Kalman filters [82] or variants of Kalman filters like extended Kalman filters (EKF) [83] or Unscented Kalman filters (UKF) [84]. The standard method of tracking the orientation with Kalman filters is to update the orientation using the angular velocity from gyroscope and compensate for the long term drift of orientation tracking from noisy gyro data with the magnetic field from magnetometer or gravity vector from accelerometer.

However, the majority of research on orientation tracking in the context of indoor positioning is to determine the heading of the pedestrian rather than the heading of the device. Most research therefore assumes that the position where the mobile device is attached is fixed and known, thereby making the assumption that the heading of the pedestrian is always consistent with the heading of the mobile device [81]. This is not always true. To address this problem, Zee [80] attempted to estimate the difference between the user's heading and the phone's heading by looking at the second harmonic of the acceleration. They demonstrated that it is either completely absent or is extremely weak in the direction perpendicular to the user's walk, and is conversely dominant in the direction parallel to the user's walk. However, headings estimated with this approach can be easily polluted with noise. In addition, this approach cannot estimate the precise heading of the user but two ambiguous headings

which are 180° apart instead. A more recent work, WalkCompass [85] samples the acceleration signal during the segment of a step with maximum deceleration. After removing noise and rotating samples to a stable coordinate system, it averages them to obtain the walking direction in 3D. It then projects the estimated direction to the horizontal *walk* plane. This approach works for various device placements, but is sensitive to noise.

Another challenge in orientation tracking is the long term drift of inertial sensors especially gyro sensors. Though in existing work the magnetic field of the earth can mostly compensate long term yaw errors, it has little correction power in roll and pitch estimation. This is because the gravity vector can be used only when the device is static [86].

Another research problem regarding orientation tracking is how to determine the initial orientation of the device. The most common way of getting the initial orientation is by using the acceleration (assuming the device is static) and the magnetic field of the earth [66]. However, the magnetic field of the earth is significantly distorted in indoor environments [87], making the initial orientation less reliable and useful. Therefore, people also use extra constraints like maps correct the heading bias caused the magnetic distortion [81].

2.3 RSS-based Positioning

RSS-based positioning is a popular positioning method in indoor environments due to the wide availability of wireless APs, and the benefit of not having to install and maintain special-purpose infrastructure. Generally speaking, RSS of WiFi signals are mostly used. But further techniques have also been developed to deal with heterogeneous wireless clients [88] or to exploit additional features of the environment, e.g. FM signals [89], sound, light and color [90].

The RSS-based approaches first create a signal map by associating locations on the floor plan with certain location-specific RSS signature, and later infer the location of the user by finding the closest match between the received signature and the signatures from the signal map. The signal map may be derived from propagation models or location-RSS databases, leading to two typical ways of using the RSS as an indicator of the current location: range-based positioning and fingerprinting.

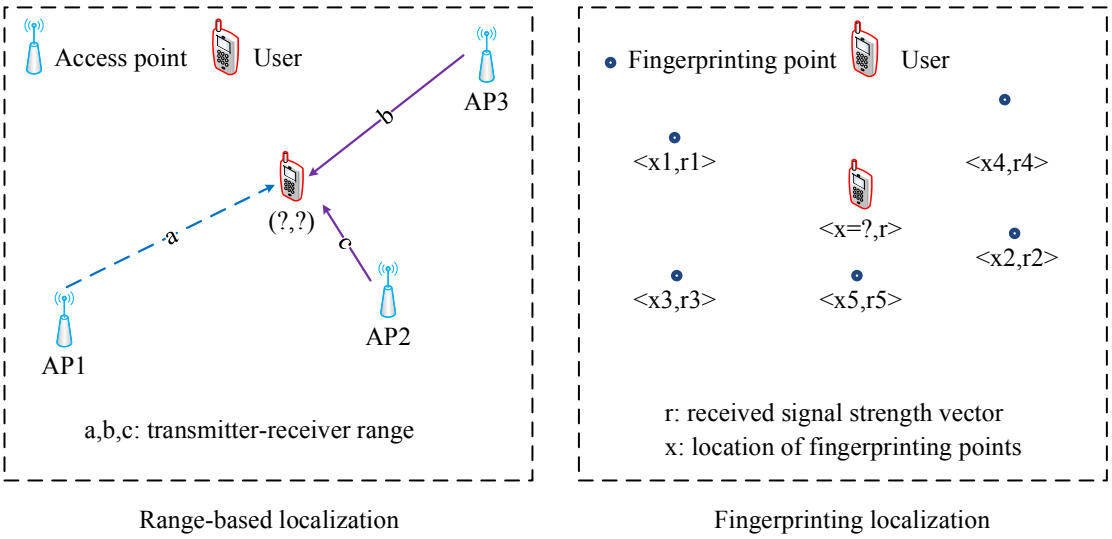


Figure 2.5: Range-based positioning and fingerprinting.

2.3.1 Range-based Positioning

The range-based positioning approaches first estimate distances between the various transmitters and the receiver using a propagation model established prior to the localization process and then apply triangulation or trilateration to estimate the location of the user, as shown in Fig. 2.5. EZ [91] is a typical example using WiFi signals in this category. However, there are some accuracy limits for localization using WiFi signal strength, especially some fundamental limitations which are not likely to be transcended, e.g., the median localization error of 3 meters and 97th percentile of 9 meters [92].

To overcome this barrier, various methods have been proposed to improve the distance estimation accuracy. The work in [93] proposes to use multiple regression analysis using least-square estimation to calculate the distance between the access points and the user. The approach in [94] averages multiple RSS from two frequencies and different power levels with weights which are linearly proportional to the RSS themselves.

To further improve the distance estimation accuracy, automatic learning of the propagation parameters rather than manual setting of the path loss parameters has also been proposed. For example, Li et. al [95] estimated the path loss parameters from multiple RSS with non-linear least square approaches to minimize the loss function. Xie et. al [96] first trained the propagation model with a small number of fingerprinting locations and then applied the propagation model to estimate the distance between the access points and the user. Barsocchi et. al [97] adopted mutual

training between access points to achieve the automatic calibration of the propagation model. In this work two different training methods were used in each access point to obtain the propagation parameter of all other access points.

Generally speaking, the accuracy of current range-based localization approaches is not satisfactory because RSS is detrimentally affected by multi-path effects including reflection, refraction, and diffraction, especially in non-line-of-sight (NLOS) conditions when the received signal contains no direct line-of-sight (LOS) component. Although many indoor localization approaches and indoor propagation models have been proposed to analyze or mitigate the influence of multi-path effects [36, 98, 99], the mere analysis of multi-path effects without considering the LOS/NLOS conditions is insufficient.

The use of LOS/NLOS information can greatly improve the performance of the localization of people and objects inside buildings or in urban landscapes. In general, for the same distance, the RSS in LOS conditions can be over a hundred times stronger than the RSS in NLOS conditions (as shown in Sec. 3.1), which can result in very large distance estimation errors when employing simple propagation models. Furthermore, since multi-path can impact both LOS and NLOS conditions, accurate path loss models are difficult to ascertain in practice even if NLOS conditions are identified. Therefore, other approaches beyond simple propagation models, e.g., regression, should be employed to estimate distance.

NLOS identification and mitigation techniques developed to date have been primarily investigated for ultra-wide band (UWB) signals [100–106]. Due to the high temporal resolution of UWB signals, the LOS component can be readily identified and extracted from the received signal. The commonly used techniques in NLOS identification of UWB signals are hypothesis testing [103,107] and machine learning [30,106] based on features from the received UWB signals. Features including range estimates [107] (range and error distribution), channel statistics [100–103] (RMS delay spread, mean/excess delay, amplitude), and position estimate [104] (ray tracing with map data) of LOS components differ greatly from those of NLOS components. However, with much narrower bandwidths, typical mobile WiFi devices can only report RSS data rather than these detailed features.

NLOS mitigation was achieved in [100,108,109] by minimizing the weighted residual of estimated distances. Chen [108] minimized the distance estimation residual by selecting a best subset from available access points (AP). Authors in [100, 109] improved this algorithm to reduce the computation complexity and speed the convergence by using only three distance measurements instead of all available distance

measurements. However, these techniques are not suitable for environments with few LOS APs and many NLOS APs. A recent generic NLOS mitigation technique [110] applicable to all wireless systems, tries to reduce the distance estimation errors in NLOS conditions using convex programming. But this approach requires the availability of at least 50% LOS samples, which is also often unrealistic.

2.3.2 Fingerprinting

Fingerprinting approaches work in two stages: radio map construction and location estimation. The radio map construction stage builds the radio map in the form of < location, RSS > pairs which relate the locations in the physical map to RSS signatures. Note that the RSS is a vector rather than a single value to make it unique for each specific location. Then the location estimation stage estimates the location of the user by comparing the received RSS with the recorded RSS already in the database.

Three types of fingerprinting approaches are most widely used, including deterministic frameworks like weighting estimation [111, 112], probabilistic frameworks such as maximum likelihood or density distance [113], and Bayesian filtering approaches, e.g. Kalman filters [114–118]. A comparative survey of current fingerprinting approaches is presented in [119].

RADAR [111] is an example of a deterministic framework which uses naive regression like nearest neighbour or K nearest neighbours to derive the unknown location. However, this work has actually started the fingerprinting branch in the RSS-based localization. After that, researchers have proposed a number of variants from the original fingerprinting approach.

HORUS [113] has gained the most reputation in fingerprinting by providing an elegant probabilistic framework to account for the hardware variations of mobile devices. Some other techniques were also proposed, e.g. nonparametric statistical procedures [120] for diagnosis of the fingerprinting model, and the modified LCA machine learning algorithm [121], to estimate the missing location information.

Applying fingerprinting approaches to the magnetic field of the earth is a natural branch of fingerprinting methods [122–130, 130–139]. The magnetic fields in modern buildings arise from both natural and man-made sources, such as electric power systems, and industrial devices [140]. The metallic objects ubiquitous in indoor environments, like the steel in concrete structures, power lines, etc. distort the ambient magnetic field and cause useful variations. As a result, the local anomalies of the magnetic field can be recorded (map construction) and then applied to localize objects

(location estimation) [141, 142]. However, given the fact that the magnetic variations largely depend on the environments, the accuracy of the positioning system also varies in different environments. To increase the robustness of positioning, a series of magnetic field readings across a time interval are usually collected and then used to compare with the signal map database [87, 143].

The main disadvantage of these methods is that they require labour intensive surveying of the environment to generate radio maps. To address this problem, there have been a number of simultaneous localisation and mapping efforts recently that aim to automatically build the radio map, by fusing RSS with motion sensor data, as discussed in detail below.

2.4 Data Fusion

All PDR algorithms based on inertial sensors have one problem in common: the fact that over time, the estimated position drifts further away from the true position. A well established approach to correcting the long-term drift in PDR is to use map constraints, such as those encoded in floor plans, radio RSS maps and magnetic distortion maps. This is a reasonable assumption given the wide availability of maps (especially floor plans), and the intensive effort invested by major players, such as Google, Microsoft, Apple and Qualcomm, in indoor mapping. Data fusion discussed in this thesis includes both the fusion of various sensor data and various constraints like radio maps and floor plans. The data fusion can be implemented in a number of ways.

Existing data fusion algorithms are widely available in various tracking systems in both outdoor, e.g. road networks and indoor environments. The techniques involved are distance-based search [144–151], recursive Bayesian estimation, e.g. variable-order hidden markov models (HMMs) [152–156], particle filters [80,81], and hypothesis testing [157–160]. These algorithms work very well in road networks but their performance is degraded in indoor environments because of the complicated mesh structure for possible movements.

2.4.1 Distance-based Search

The distance-based search algorithms first define distance between points [144, 145], arcs [146, 147, 149] or even areas [148], then evaluate the distance between the observation and all legal points/arcs/areas, and finally select the closest match as the output. Extensions of these algorithms may also select some critical location samples

and match the critical samples before the non-critical samples [150] or accommodate simultaneous multi-track matching [151]. No statistical estimation is involved in this category.

2.4.2 Recursive Bayesian Estimation

Recursive Bayesian networks model the conditional dependence structure between observation and state variables using directed graphical models [161]. It is often assumed that the sensor measurements are conditionally independent of each other given the state, i.e. $p(Z_i|S_i) = p(Z_i[1]|S_i)p(Z_i[2]|S_i)$. This is the basis of the Naive Bayesian model. The extension of this model to a sequence of states linked through transition probabilities leads to recursive Bayesian models, such as HMMs. The joint probability distribution between all state and observation variables is decomposed into a product of conditional distributions.

The three key terms that form the basis of that decomposition are: 1) the *transition model* $p(S_i|S_{i-1})$, which describes our knowledge of how likely it is to transit from S_{i-1} to S_i; 2) the *observation model* $p(Z_i|S_i)$, which expresses how likely it is to observe Z_i when the state is S_i; and 3) a prior distribution $p(S_0)$ over states at time 0. For simplicity, let's assume that an observation Z_i at step i consists of two sensor measurements $Z_i[1]$ and $Z_i[2]$.

The joint probability distribution between states and observations can be expressed as a product of conditional probability distributions:

$$p(S_{0:T}, Z_{0:T}) = p(S_0)p(Z_0|S_0)\prod_{i=1}^{T}[p(S_i|S_{i-1})p(Z_i|S_i)] \tag{2.1}$$

where $S_0, \ldots, S_T$ are the variables representing the real states of a system (e.g. the locations of a person) over a time horizon $0, \ldots, T$, and $Z_0, \ldots, Z_T$ are the observation variables over the same time period.

The filtering and smoothing problems within this framework can be expressed as finding $p(S_{t+k}|Z_{1:t})$, where $k = 0$ for filtering and $k < 0$ for smoothing. The problem of indoor localisation is either cast as a filtering problem, or as a smoothing problem if the user can tolerate some delay, or as the inference problem of finding the most likely trajectory $S_{0:T}$ given a time window's observations $Z_{0:T}$. , i.e. $\text{argmax}_S p(S_{0:T}|Z_{0:T})$). The latter is equivalent to finding S that maximizes the joint distribution computed above, i.e. $\text{argmax}_S p(S_{0:T}, Z_{0:T})$.

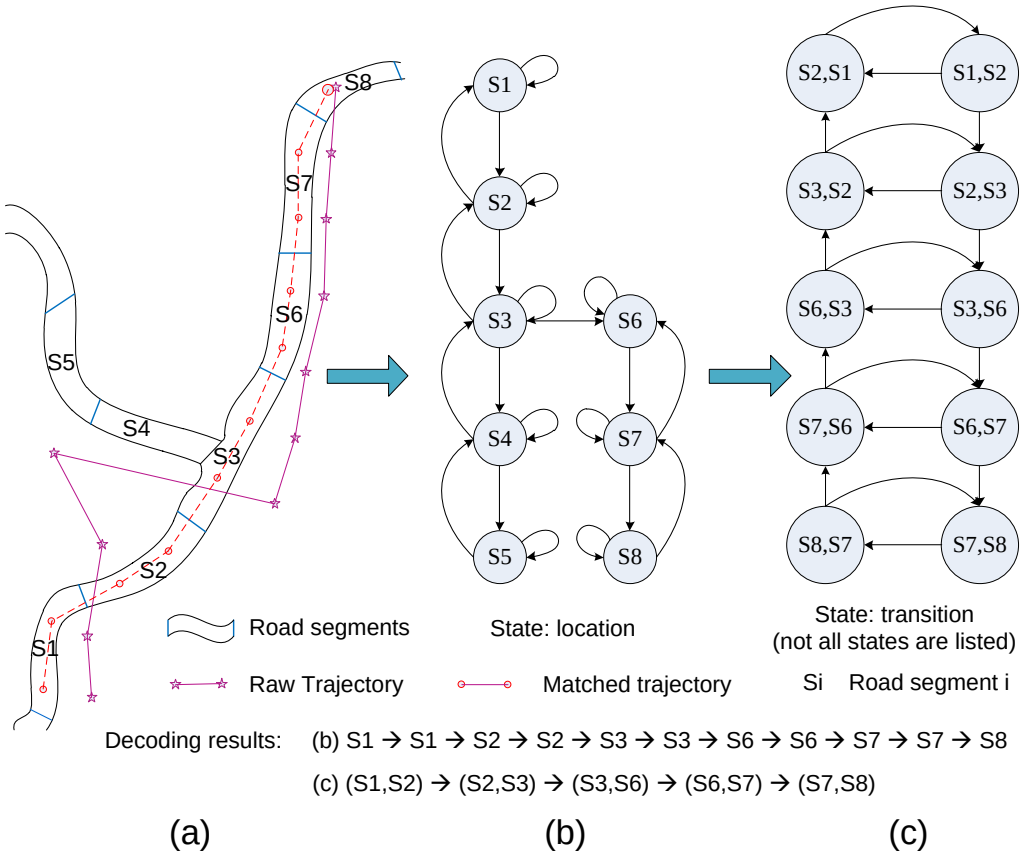


Figure 2.6: Hidden Markov Model. (a) shows the map and raw trajectory. (b) demonstrates states based on location (first order HMM). (c) shows states based on transitions between pairs of locations (second order HMM).

2.4.2.1 Hidden Markov Models

Hidden Markov Models (HMMs) are a special class of a recursive Bayesian model where state variables are discrete, and the transition model $p(S_i|S_{i-1})$ is a matrix. When observations are also discrete, the observation model is also a matrix, otherwise (semi-continuous HMMs) it is typically represented as a Gaussian or mixture of Gaussians. HMMs have been widely used for map matching and location estimation, both outdoors to match the frequently-sampled GPS data with low noise from vehicles to road networks [152–154, 156] as well as indoors [162, 163].

HMMs usually are used as first order HMMs where states represent user locations. The directed graphical model for this is shown in Fig. 2.6(b). If a user is in a particular state, they can transit to a neighbouring location or remain in the same state. The problem with these approaches [152, 153] lies in the fact that the low-degree intersections (intersections without too many outgoing roads) always have

higher transition probabilities than high-degree intersections, which may break the inference process because the low-degree intersections always have higher probabilities to be selected. To address this issue, VTrack [154] extends the HMM and introduces an absorbing state (called dead-end state in the paper) to make sure that all meaningful transitions (from one intersection to another, not to the absorbing state) in the whole graph have equal transition probabilities, which corrects the model bias caused by inappropriate graph assumptions [152, 153] in inference process. To reduce the energy consumption of power-hungry GPS and WiFi radios, CTrack [155] further extends VTrack to a two-pass HMM that sequences cellular fingerprints directly without converting them to geographic coordinates and fuses data from low power consumption sensors including accelerometer and compass. The general weakness of first order HMM-based approaches used in indoor environments is that it can only express how observations depend on individual user locations, whereas inertial data obviously depend on transitions between locations.

One possible extension of HMMs to accommodate this need is to add control variables to the model, which leads to an extension often called input-output HMM or, in the field of robotics, Markov Localization [164]. Control variables have been used in robotics to show how motor commands influence the transition to the next state. The transition model $p(S_i|S_{i-1})$ is thus refined into a process/action model $p(S_i|A_{i-1}, S_{i-1})$ where A_{i-1} is the control/inertial variable at step $i-1$.

Instead of modelling the locations, an alternative is to model the transitions between pairs of locations as the state itself, as shown in Fig. 2.6(c), which is equivalent to a second-order HMM, e.g. AutoWitness [156]. This enables AutoWitness to further take into account the displacements of objects obtained from inertial sensors.

A limitation of these HMM-based models is that they do not take into account correlations between nearby inertial observations, for example correlated magnetometer bias due to the metal disturbances in the earth's magnetic field.

2.4.2.2 Particle Filters

Particle filters, another typical implementation of recursive Bayesian filters, have also been used for data fusion [80, 81, 165–167]. The key idea of particle filters is to approximate the distribution $p(S_{t-1}|Z_1, \ldots, Z_{t-1})$ by a set of particles. In each round, these are first moved according to the transition model, their weights are then updated according to the observation model, and particles are then re-sampled according to their weights. The likelihood of each trajectory being correct is calculated and trajectories which are unlikely or impossible (for example, crossing a wall) are

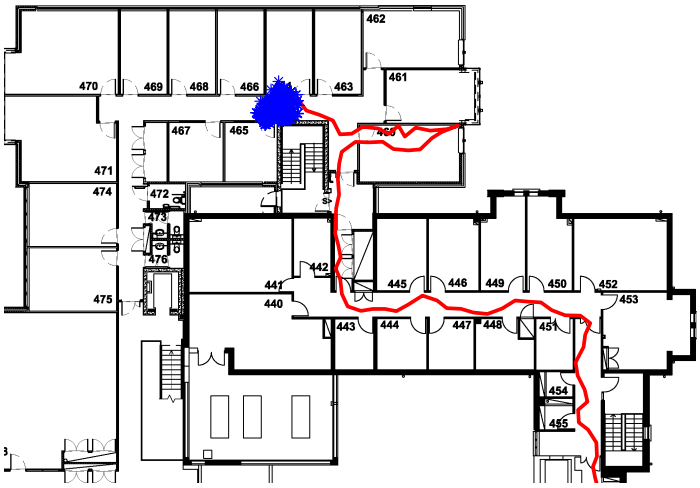


Figure 2.7: Particle filter approach to map-matching, showing how constraints from the map control the distribution of the particles.

culled, as shown in Fig. 2.7. Locations which are more likely act as seeds for the next iteration of the algorithm, through a resampling step. Over time, the particles typically converge to the most likely position of the user.

The use of particle filters in indoor positioning has drawn much attention in recent years. Li et. al [81] proposed an indoor localization method using displacements derived from the phone's inertial sensors. A particle filter is then employed to search a legal trajectory that 1) does not violate the physical constraints of the floor plan, and 2) best supports the raw PDR trajectory. This approach reports a positioning error of 2.4 m. Almost at the same time, Zee [80] introduced an augmented filter to match the raw trajectories obtained from smartphones to the indoor floor plan with a reported accuracy of around 1 m. However, these two approaches were largely tested only in corridors rather than rooms, similar to the application scenario of road networks.

The data fusion system in [168] uses an adaptive mixture particle filter to accommodate both WiFi and inertial data. The interesting aspect in this paper is that the author performed semi-automatic fingerprinting with a controlled robot, which is not fully automatic and the fingerprints collected by the robot are apparently different from the signal strength received by a device held by a person because the human body has significant impact on the RSS.

The major limitation of the particle filter-based approaches is that the model only has local information in the area covered by particles, making it prone to observation errors. Another major issue of the particle filter is the computation time, as a large number of particles are typically required to ensure good estimation of the continuous

probability distribution, especially when dealing with noisy inertial data and large maps.

2.4.2.3 Kalman Filters

Kalman filters are an alternative approach which considers the location of a user as a continuous variable. Variants of Kalman filters, e.g. Extended or Unscented Kalman Filter (e.g. [169]) are usually used when the motion model is not linear. Kalman filters have also been widely used in outdoor environments to fuse inertial trajectories and GPS readings with map information [170]. Sensor fusion is typically performed in two steps. First, the Kalman Filter estimates the position of a moving node after taking into account its previous position and the inertial sensor data. Then, RSS data are used to update the node's position, abiding by map constraints. Similar to first order HMMs, Kalman Filters are effective when radio signal strengths are periodically sampled from various access points, and fused with inertial data; however, their performance deteriorates when we solely make use of inertial sensors and the building's floor plan. An example is the RSS-based localization and tracking approach using sigma-point Kalman filter with WiFi RSS data, infrared motion sensors and foot switches [171].

2.4.3 Simultaneous Localization and Mapping

Simultaneous localization and mapping (SLAM) is the problem of constructing or updating a map of an unknown environment while simultaneously keeping track of the location of the agent in that environment [172–174]. Although SLAM was firstly and largely researched and experimented in robotics, it has also been applied to indoor positioning. The SLAM techniques can be viewed as map generators providing optional input to other algorithms, e.g. radio fingerprint maps [172], or organic landmark maps [91, 175].

The SmartSLAM algorithm [176] simultaneously builds the indoor floor plan and fingerprinting map with smart phones. Similarly, Kim et. al proposed an interesting automatic fingerprinting system based on smart phones [177, 178]. They assumed a central server but without the map information and used only the inertial sensor data to build the global fingerprinting map with a starting point. In addition, the existing fingerprinting map could in return correct the drift errors of users who arrive later. However, as shown in Fig. 2.8, without proper corrections, even the foot-mounted pedestrian dead reckoning system with professional IMU sensors has large drift error without correction information from other data sources.

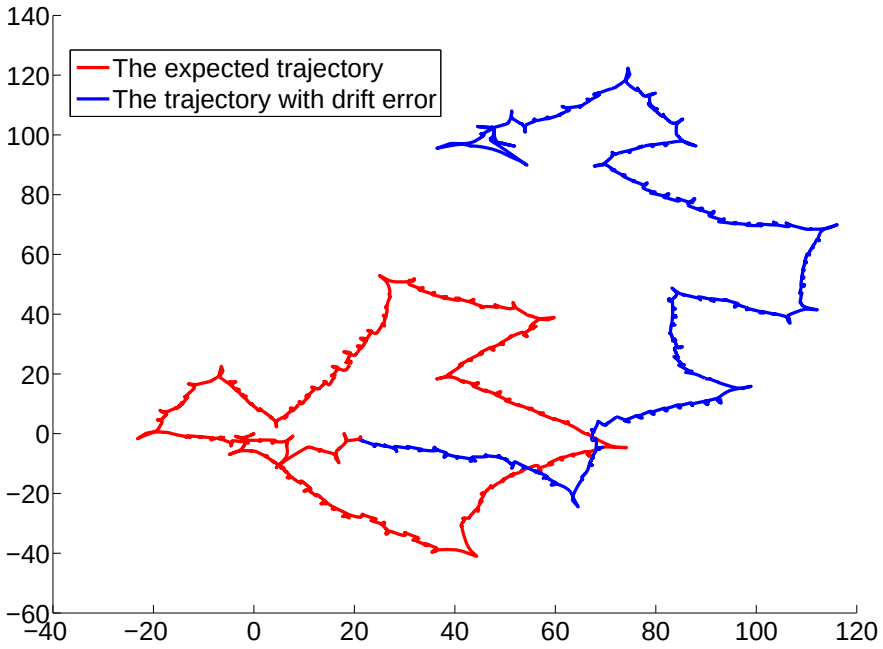


Figure 2.8: The drift error of dead reckoning without correction.

To reduce the cost of radio map construction, interpolations using Gaussian processes [172] or its variant [179] were developed which only use a small set of fingerprinting data to construct the radio map of the whole area. However, radio maps generated in this way have issues with accuracy.

2.4.4 Hypothesis Testing and Others

Hypothesis testing first models each outgoing road of a given intersection as a hypothesis [158, 159], builds the corresponding test functions, and then evaluates the values of the test functions given various hypotheses. Therefore it is not suitable for real time or mobile device applications. Similar approaches have also been proposed, e.g. voting-based map matching [160] and fuzzy logic map matching [157], with the underlying algorithm similar to multi-hypothesis testing. The problem with hypothesis testing-based approaches is that they are usually very expensive in terms of time and space complexity due to the large number of hypotheses to process in real world implementations.

An interesting system proposed in [180] used only the accelerometer to build the fingerprinting database. It first converts the floor plan into a high dimensional structure where the distance between any two points in the map is the walking distance

instead of the real distance. Then the number of steps is recorded by the accelerometer between any two fingerprinting points. When enough fingerprinting data has been collected to cover the whole experimental site, the fingerprinting trajectories could form another high dimensional space structure. The automatic fingerprinting could be accomplished by mapping the reference points in the two high-dimensional structures. Unfortunately, the shape matching algorithm here could only be used when the different shapes are highly distinguishable. Therefore, this approach could only be used in corridor level localization when the corridor structure is unique and does not work in open space settings like rooms or halls.

The autonomous construction algorithm of the access points location map in [181] adopted the multi-dimensional scaling technique. They explore the autonomous construction of the fingerprinting map using the access points location map. But this method makes the assumption of an accurate propagation model, which is extremely challenging in indoor environments.

2.5 Parameter Learning

Similar to humans who become better at finding their whereabouts as they visit the same environment multiple times, so should an indoor positioning algorithm as it runs repeatedly in the same context. Therefore, a well designed positioning algorithm should be able to learn the environments and contexts, which improves the positioning performance of either other users or the same user in the future in the same environment.

The idea of learning from the environment is not entirely new in indoor positioning. An illustrative example is the implementation of map matching using a simultaneous localisation and mapping approach, such as WiFiSLAM [179]. This approach is a typical implementation of the well-known GraphSLAM algorithms [182] from robotics in indoor positioning. It first records the signal fingerprints when the user first visits a location, and then detects whether the user comes back to the same location again by comparing the recorded fingerprints with the recently received fingerprints. Once the user visits the same location again (loop closure), it corrects all the trajectory history between the two visits and builds the signal database along the trajectory. The construction of the signal database is actually the learning of the environment. Typical SLAM algorithms are usually offline to detect loop closure while some online SLAM algorithms are proposed, such as online SLAM [183], DP-SLAM [184] and FootSLAM [185]. Online SLAM [183] is capable of distinguishing

the static and dynamic parts of the environments while FootSLAM [185] combines an online particle filter with a probabilistic map. DP-SLAM aims to achieve truly simultaneous localization and mapping without landmarks. DP-SLAM is reported to be accurate enough that no special loop closing techniques are required in most cases by maintaining a large number of maps in an efficient manner. DP-SLAM makes only a single pass over the sensor data. DP-SLAM works by maintaining a joint probability distribution over maps and robot poses using a particle filter.

The learning could also take place after map matching. The map matching usually presents trajectories with good accuracy, based on which the signal database either the radio map or the magnetic distortion map [87] can be constructed. Rai et al [80] proposes the learning of the radio map using Bayesian filter-based map matching. In principle, the more a user visits an environment, the more accurate the inferred radio maps become, and the better the performance of map matching.

Apart from the learning of environments, the learning of other parameters, like the step length in PDR, has also been studied. Li et. al [81] proposed an adaptive step length estimation algorithm which learns the step length of the user using particle filtering. However, due to the constraints of the map matching algorithms based on particle filters, the learning of the step length does not significantly improve the positioning accuracy.

As mentioned earlier in this section, the idea of learning various parameters is not entirely new in indoor positioning. But the existing learning techniques have two limitations. The first limitation is the inaccuracy of the learned parameters. Since the accuracy of parameters learned largely depends on the accuracy of the underlying localization systems, the inaccuracy of existing localization system mentioned above has significant impact on the performance of the learning algorithms. The second limitation lies in the fact that existing learning techniques only cover some high level parameters like the radio fingerprinting map and step length. Some lower level while more important parameters such as the sensor bias and the initial heading bias are not investigated.

2.6 Discussion

This chapter began with a discussion on existing end-to-end indoor positioning systems. Two general types of indoor positioning system, infrastructure-based and infrastructure free, were reviewed. For the more practical infrastructure free case, people have developed pedestrian dead reckoning, RSS-based positioning, and fusion of

these two to improve the positioning accuracy. This chapter also discussed the related work for parameter learning techniques to further improve the positioning accuracy. Table 2.1 presents a comparison between typical positioning systems in each category in terms of practicality, as defined in Sec. 1.2.

Table 2.1: Comparison of typical indoor positioning systems in terms of cost, accuracy, robustness, computation, and scalability.

System	Cost	Accuracy	Robustness	Computation	Scalability
RF tags [4, 12]	High	1-3 m	Low	Low	Very low
iBeacon [13]	High	1-3 m	Low	Low	Very low
UWB [32, 36]	Very high	0-1 m	Low	High	Very low
AMF [40, 45]	High	0-2 m	High	High	Very low
EZ [91]	Low	1-5 m	Very low	Low	Low
RADAR [111]	Low	3-10 m	Low	Low	Low
HORUS [113]	Low	1-5 m	Medium	Very high	Low
Zee [80]	Low	2-8 m	Low	High	Medium
VTrack [154]	Low	2-8 m	Low	High	Medium
AutoWitness [156]	Low	2-8 m	Medium	Very high	Medium
WiFiSLAM [179]	Very low	2-5 m	Low	High	High

Although many algorithms have been developed to cover various aspects of indoor positioning as discussed in this chapter, they still have limitations that prevent them from being put into real world applications. The first limitation is that no existing pedestrian dead reckoning using general handheld devices is robust to context-specific variations, e.g. device variation, user variation, attachment variation, and environment variation. The second limitation is the lack of non-line-of-sight identification and mitigation algorithms using just RSS from WiFi/BLE signals to provide accurate location estimation in the absence of radio map. The third limitation lies in the fact that existing data fusion approaches are both computational expensive and unable to flexibly fuse different sensor observations and various constraints together (particle filter, HMMs, etc.). Given their limited battery and computation power, only lightweight data fusion algorithms can practically run on mobile devices. The last limitation is the manual setting of various PDR and fusion parameters which actually requires context-specific tuning. There are some other minor limitations but these four limitations are the major factors that prevent the indoor positioning from being widely adopted.

To address these limitations, this thesis proposes four components, including NLOS identification and mitigation with only RSS measurements, robust pedestrian

dead reckoning with mobile devices, lightweight map matching component, and life-long learning. The next chapter discusses the use of WiFi/BLE signals to perform localization. For these signals, fingerprinting approaches are generally preferred with existing radio maps. However, in most scenarios the construction of radio map is too expensive in terms of both time and cost. Therefore, approaches that work without radio maps are also required, especially with the absence of inertial data. The next chapter focuses on how to optimize the positioning accuracy using radio signals by means of non-line-of-sight identification and mitigation.

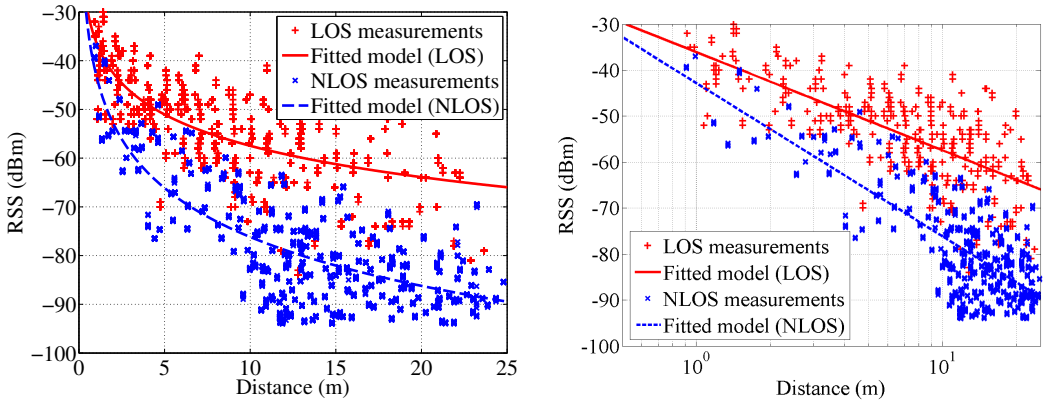
Chapter 3

Non-line-of-sight Identification and Mitigation

Indoor wireless systems often operate under non-line-of-sight (NLOS) conditions that can cause ranging errors for location-based applications. As such, these applications could benefit greatly from NLOS identification and mitigation techniques. These techniques have been primarily investigated for ultra-wide band (UWB) systems, but little attention has been paid to WiFi systems, which are far more prevalent in practice.

This chapter proposes approaches based on machine learning to address the NLOS identification and mitigation problems using multiple received signal strength (RSS) measurements from WiFi signals. Extensive experiments in various indoor environments show that our techniques can distinguish between LOS/NLOS conditions with an accuracy of around 95%. Furthermore, the presented techniques improve distance estimation accuracy between a transmitter-receiver pair by 60% as compared to state-of-the-art NLOS mitigation techniques. Finally, improvements in distance estimation accuracy of 50% are achieved even without environment-specific training data, demonstrating the practicality of our approach to real world implementations.

The structure of this chapter is as follows. Sec. 3.1 formulates the problem and introduces the system architecture of the proposed approaches. Sec. 3.2 presents the feature extraction schemes. Sec. 3.3 proposes three algorithms to distinguish between LOS and NLOS conditions. Sec. 3.4 develops strategies to mitigate the effect of NLOS conditions. Sec. 3.5 evaluates the proposed algorithms in various settings. Sec. 3.6 describes the impact of the proposed algorithms on the positioning system. Sec. 3.7 summarizes this chapter and discusses limitations and ideas for future works.



(a) Linear (b) Log-distance

Figure 3.1: A large number of RSS measurements were collected in a variety of settings (different access points, distances and environments) in both LOS and NLOS conditions. The propagation model was derived using least square approximations.

3.1 System Architecture

In indoor environments, there are usually many WiFi access points (AP) which are continuously transmitting packets including beacons and user data that can be received with mobile wireless devices. The received signal strength (RSS) of these packets not only determines link quality (LQ) but can also be explored for localization of mobile devices. However, in indoor environments it is very common for the direct path between the transmitter (e.g. AP) and receiver (e.g. mobile devices) to be obstructed by objects such as walls and doors, due to which the receiver is unable to directly 'see' the transmitter. We define this as the non-line-of-sight (NLOS) condition. On the contrary, if the receiver can directly 'see' the transmitter, we have line-of-sight (LOS) conditions.

In multi-path environments it is possible to measure the same RSS for different distances, obstacles, geometries and transmission powers, as shown in Fig. 3.1. Although a single RSS sample tells us little about the transmitter-receiver distance and LOS/NLOS conditions, a set of T ($T >> 1$) temporal RSS samples from a certain receiver position L, denoted by $< P^{(1)}, \cdots, P^{(T)} >$, can provide us with much more information. Given these RSS measurements, we aim to infer whether the transmitter and the receiver are in LOS or NLOS conditions ($b = 1/-1$), called NLOS **identification**, and estimate the transmitter-receiver distance (d), called NLOS **mitigation**.

The proposed framework has two major stages: training and testing, as shown in Fig. 3.2. The training stage consists of the following four steps. 1) An exten-

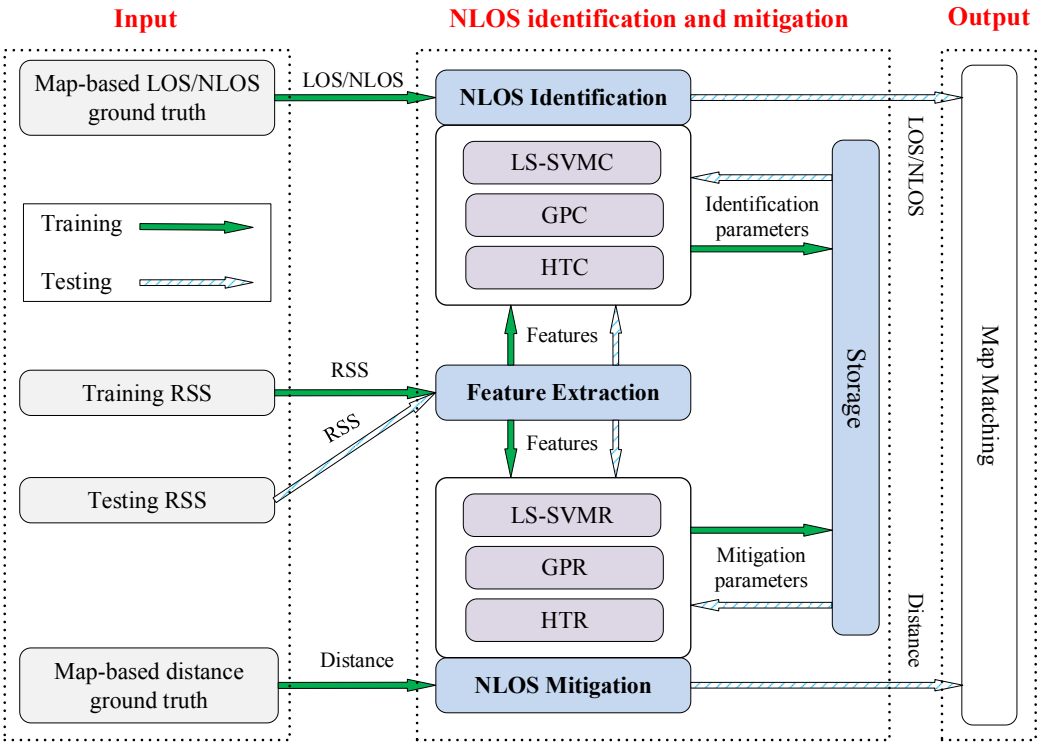


Figure 3.2: The system architecture of the proposed NLOS identification and mitigation system.

sive indoor measurement campaign is performed to collect training data, including RSS, and receiver locations which are later used to derive the transmitter-receiver distances and LOS/NLOS conditions with the aid of the floor plan. Suppose we have obtained temporal RSS samples $< P_k^{(1)}, \cdots, P_k^{(T)} >$, LOS/NLOS conditions b_k, and transmitter-receiver distance d_k from N locations L_k $(k = 1, \cdots, N)$ which can be used to train the identification and mitigation models. 2) Then a set of M features including the mean, χ^2 goodness of fit, etc., denoted with $< x_k^{(1)}, \cdots, x_k^{(M)} >$ are extracted from the RSS data, which are identified to be effective in distinguishing LOS and NLOS conditions and estimating transmitter-receiver distances even in NLOS conditions. 3) Then the parameters of NLOS identification/mitigation algorithms are learned and stored from these features, together with the LOS/NLOS conditions ground truth, and transmitter-receiver distances with the aid of the floor plan. Note that the parameters learned from training are different for each algorithm proposed and are specified in detail in Secs. 3.3 and 3.4.

The testing stage has three steps. 1) A new set of RSS data, denoted with $< P_j^{(1)}, \cdots, P_j^{(T)} >$, are collected in a new location L_j. 2) The same set of features $< x_j^{(1)}, \cdots, x_j^{(M)} >$ are extracted from the RSS set $< P_j^{(1)}, \cdots, P_j^{(T)} >$. 3) Then with

identification/mitigation parameters from the training stage, the proposed NLOS identification/mitigation algorithms predict the LOS/NLOS conditions ($b_j = 1/-1$) and the transmitter-receiver distance (d_j). Note that the LOS/NLOS condition from NLOS identification is only explicitly used in some of the considered mitigation algorithms, specifically in HTR, but not in LS-SVMR and GPR, as discussed in detail in Sec. 3.4.

In this study we propose and compare three distinct NLOS identification algorithms based on machine learning techniques and hypothesis testing, namely least square support vector machine classifier (LS-SVMC), Gaussian processes classifier (GPC), and hypothesis testing classifier (HTC). LS-SVMC has a low computational cost compared with other machine learning techniques while GPC has slightly higher identification accuracy. We will show that after the model parameters have been trained from the measurement campaign, the two machine learning algorithms can also make accurate predictions in a different environment site (Sec. 3.5). To reduce the training phase, which is expensive for most applications in terms of training data collection, we also propose HTC which only requires a much less intensive training phase in term of time and space complexity. The only training needed for HTC is a least square estimation of the mean and variance of all RSS measurements. We also present three algorithms to perform NLOS mitigation including least square support vector machine regressor (LS-SVMR), Gaussian processes regressor (GPR), and hypothesis testing regressor (HTR). LS-SVMR and GPR can explicitly predict the transmitter-receiver distances even in NLOS conditions. However, the hypothesis testing algorithm cannot predict distances directly; it first uses the Hypothesis Testing Classifier (HTC) to infer LOS/NLOS conditions, and then selects the appropriate radio propagation model for distance estimation depending on the LOS/NLOS outcome.

3.2 Feature Extraction

In this section, through observation of multiple RSS samples in LOS and NLOS conditions, we identify several key features from the collected RSS measurements useful for identifying NLOS conditions, including the mean, standard deviation, kurtosis, skewness, Rician K factor, and χ^2 goodness of fit, etc. Each feature is derived from a set of T (e.g., $T = 10 - 20$) RSS samples $[P^1, \cdots, P^{(T)}]$ collected at a particular location over a short period of time (e.g., 1 second) from one AP. All NLOS identification and mitigation algorithms in this study are developed based on these features.

Mean and Standard Deviation (μ, σ_s) from RSS data can help NLOS identification with features below.

Kurtosis ($\mathcal{K}$) measures the peakedness of the probability distribution. Generally, RSS measurements in LOS conditions are more centralized than samples in NLOS conditions because the dominant LOS signal is much stronger in terms of energy.

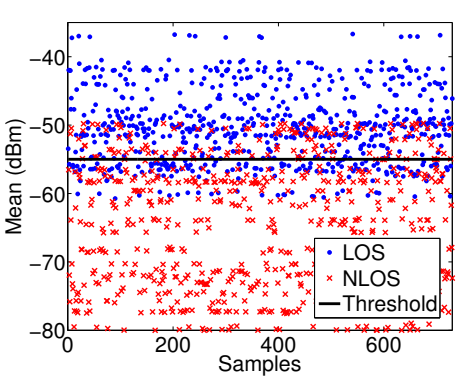
Skewness ($\mathcal{S}$) measures the asymmetry of the probability distribution. The skewness of Rayleigh distribution is a constant (approx. 0.63) which is generally larger than the skewness of Rician distribution. In other words, the LOS measurements should have lower skewness than the NLOS measurements.

Rician K Factor (K_r) is defined as the ratio between the power in the direct path and the power in other scattered paths $K = \nu^2/(2\sigma^2)$ [186]. Existing theoretical and empirical studies have shown that there is a link between the Rician K factor and the presence of LOS conditions [187]. Specifically, in NLOS conditions where no direct path exists, the Rician K factor is expected to be close to zero.

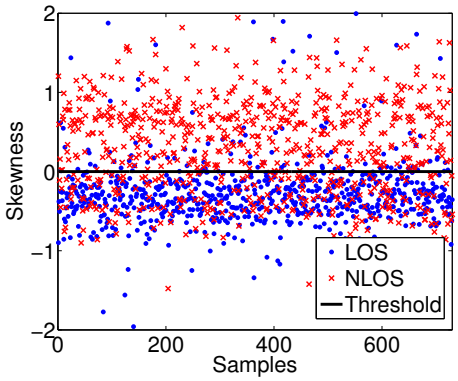
χ^2 Goodness of Fit (χ^2) measures the distance between RSS measurements and the underlying Rician distribution. Compared with other scattered signals, the LOS signal reacts minimally to the environments, which leads to different empirical distributions and thus different χ^2 in LOS and NLOS conditions.

The log-mean ($\hat{d}$) is designed primarily for NLOS mitigation. We convert the RSS to logarithmic space because there is a linear relationship between the logarithmic distance and RSS.

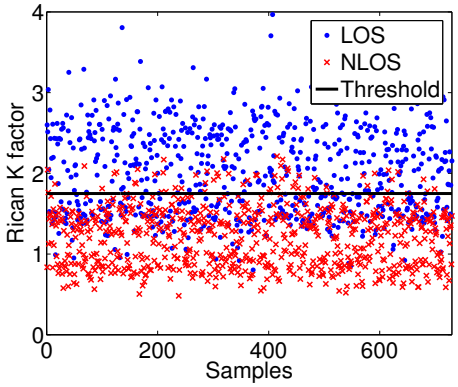
Feature Distributions are also derived for hypothesis testing from theory and empirical data. Our data indicates that $\mathcal{K}$ follows closely log-normal distribution, the same as the Kurtosis of UWB signals [100]. In addition, we also find μ, $\mathcal{S}$, K_r, and χ^2 follow closely the Gaussian distribution. Our preliminary experiments show that σ_s does not improve hypothesis testing accuracy and hence we do not discuss its distribution. Fig. 3.3 illustrates the values of some typical features in different locations under both LOS and NLOS conditions. It is observed that the mean (Fig. 3.3(a)), the skewness (Fig. 3.3(b)), and the Rician K factor (Fig. 3.3(c)) each can individually achieve an identification accuracy of around 70 percent alone with a single threshold estimated using least squares. By fusing features, we can achieve an identification accuracy of around 95 percent with techniques discussed below.



(a) Mean (misclassification 26.46%)



(b) Skewness (misclassification 29.92%)



(c) Rician K factor (misclassification 28.79%)

Figure 3.3: Illustration of how different features can distinguish between LOS and NLOS conditions.

3.3 NLOS Identification

The training data for the NLOS identification and mitigation including the RSS, NLOS conditions, and distances were collected in the corridors of our experimental site shown in Fig. 3.4. We recorded the RSS from all access points marked in Fig. 3.4 at locations distributed every 1 meter in the corridors. At each location, the LOS/NLOS condition and the distance to each access point were later derived from the recorded locations with the aid of the map. Details of the experiment setup and how we collected data can be found in Sec. 3.5. Based on the set of features extracted from the RSS, the task here is to decide whether a given set of RSS samples corresponds to LOS or NLOS conditions.

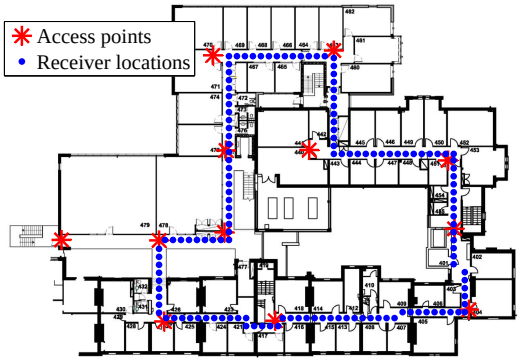


Figure 3.4: Map of the office site ($65m \times 45m$). Access points and receiver locations are marked.

3.3.1 Least Square Support Vector Machine Classifier (LS-SVMC)

Recall that our algorithm is designed for use in mobile devices. As such, the quality of generalization and ease of training are the two highest priorities in the selection of machine learning algorithms. Therefore, we propose using the Support Vector Machine (SVM), a supervised machine learning algorithm that can be used as a classifier to separate data sets with different features and whose capabilities in these two aspects is generally better than other machine learning approaches [188].

Given a set of N training items $\{\boldsymbol{x}_k, b_k\}_{k=1}^{N}$ where $\boldsymbol{x}_k$ is the kth input consisting of a subset of features described in Sec. 3.2 and $b_k \in \{-1, 1\}$ indicates LOS/NLOS conditions ($b_k = 1$ LOS and $b_k = -1$ NLOS), linear machine learning algorithms are designed to separate the data set in the following form.

$$b(\boldsymbol{x}) = \text{sign}\left[\boldsymbol{w}^T \varphi(\boldsymbol{x}) + w_0\right], \tag{3.1}$$

in which sign is the signum function, $\boldsymbol{w}$ and w_0 are weight parameters learned from the training data using optimization Eqn. (3.3) discussed below, and $\varphi(\cdot)$ is the pre-determined feature mapping function. Since the LOS/NLOS RSS measurements are not linearly separable as shown in Fig. 3.3, we use a Gaussian radial basis function (RBF) to get a better result than a linear feature mapping [189]:

$$\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x_k}) = \varphi(\boldsymbol{x})^T \cdot \varphi(\boldsymbol{x_k}) = \exp\left[-\frac{||\boldsymbol{x} - \boldsymbol{x_k}||_2^2}{2\sigma^2}\right], \tag{3.2}$$

where σ^2 is the hyperparameter learned from the training data using Eqn. (3.3).

To avoid the quadratic programming problem of SVM, LS-SVM [190] is used in this study, which simplifies the optimization to learn the weights $\boldsymbol{w}$, w_0, and the

penalty of misclassification $\boldsymbol{e}$ as follows.

$$\begin{aligned} \underset{\boldsymbol{w},w_0,\boldsymbol{e},\sigma^2}{\operatorname{argmin}} \quad & \frac{||\boldsymbol{w}||^2}{2} + c\frac{1}{2}\sum_{k=1}^{N} e_k^2 \\ \text{s.t.} \quad & b_k\left[\boldsymbol{w}^T\varphi(\boldsymbol{x}_k) + w_0\right] = 1 - e_k, \quad \forall k, \end{aligned} \tag{3.3}$$

where c is the weighting factor that controls the tradeoff between training error and model complexity. It has been proven that the optimization problem in Eqn. (3.3) is a linear programming problem [190], which can be solved with its Lagrangian dual and Karush-Kuhn-Tucker (KKT) conditions [191].

After we have ascertained the model parameters w_0, σ^2 from the training data with Eqn. (3.3), the prediction of LS-SVMC is given by

$$b(\boldsymbol{x}) = \operatorname{sign}\left[\sum_{k=1}^{N} \lambda_k b_k \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x_k}) + w_0\right], \tag{3.4}$$

where λ_k is the Lagrange multiplier, and $\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x_k})$ is the kernel function presented in Eqn. (3.2).

3.3.2 Gaussian Processes Classifier (GPC)

Gaussian processes have gained much interest in recent years [192]. Their explicit formulation makes it possible to both make probabilistic predictions and infer accurate model hyper-parameters, which provides precise trade-off between data fitting and smoothing. In addition, their low computational complexity make Gaussian processes suitable for mobile devices with small data sets [193], such as our application.

To develop classifiers based on Gaussian processes, we implement least-square classification which ignores the discreteness of the target value and treats it as a regression problem. Given the training data $\{\boldsymbol{x}_k, b_k\}_{k=1}^N$, which are the same as for LS-SVMC, the LOS/NLOS condition for a given data point $\boldsymbol{x}$ is estimated as

$$b(\boldsymbol{x}) = \operatorname{sign}\left[\boldsymbol{w}^T\varphi(\boldsymbol{x}) + n\right], \tag{3.5}$$

where the weights $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_p)$ and the measurement noise $n \sim \mathcal{N}(0, \sigma_n^2)$. Given a training set $\{\boldsymbol{x}_k, b_k\}_{k=1}^N$ where $\boldsymbol{x}_k$ is the kth input consisting of a feature subset described in Sec. 3.2 and b_k is the ground truth LOS/NLOS condition of the kth sample, rather than search for weights $\boldsymbol{w}$ as in LS-SVMC, we find that:

$$b \sim \mathcal{N}(\boldsymbol{0}, \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x'}) + \sigma_n^2 \boldsymbol{I}_n), \tag{3.6}$$

where $\boldsymbol{I}_n$ is the identity matrix and the covariance function or kernel function $\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x_k}) = \boldsymbol{\varphi(x)^T \Sigma_p \varphi(x_k)}$. Note that it is usually assumed that the mean of the Gaussian processes observation is zero everywhere.

With noisy observations, the joint distribution of the observed target values and the function values under the Gaussian prior can thus be written as

$$\begin{bmatrix} \boldsymbol{b} \\ \boldsymbol{b^*} \end{bmatrix} \sim \mathcal{N}\left(\boldsymbol{0}, \begin{bmatrix} \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x^*}) + \sigma_n^2 \boldsymbol{I} & \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x^*}) \\ \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x^*}) & \boldsymbol{k}(\boldsymbol{x^*}, \boldsymbol{x^*}) \end{bmatrix}\right), \tag{3.7}$$

where $(\boldsymbol{x}, \boldsymbol{b})$ is the training set and $(\boldsymbol{x^*}, \boldsymbol{b^*})$ is the test set. Then the conditional joint posterior distribution on the observation $p(\boldsymbol{b^*}|\boldsymbol{b})$ is Gaussian with mean and variance:

$$\mathbb{E}[\boldsymbol{b^*}|\boldsymbol{b}] = \boldsymbol{k}(\boldsymbol{x^*}, \boldsymbol{x})(\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{b}, \tag{3.8}$$

$$\mathrm{cov}[\boldsymbol{b^*}|\boldsymbol{b}] = \boldsymbol{k}(\boldsymbol{x^*}, \boldsymbol{x^*}) - \boldsymbol{k}(\boldsymbol{x^*}, \boldsymbol{x})(\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x}) + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x^*}). \tag{3.9}$$

Note that the mean prediction is a linear combination of kernel functions, each one centered on a training point, that is

$$\mathbb{E}[\boldsymbol{b^*}|\boldsymbol{b}] = \sum_{i=1}^{n} \alpha_i \boldsymbol{k}(\boldsymbol{x_j}, \boldsymbol{x^*}), \tag{3.10}$$

where α_i is the ith element of the vector $(\boldsymbol{k}(\boldsymbol{x_j}, \boldsymbol{x^*}) + \sigma_n^2 \boldsymbol{I})^{-1} \boldsymbol{b}$ and $\boldsymbol{x_j}$ is a vector representing the jth feature of the training data set.

The kernel function used for this study is as follows:

$$\boldsymbol{k}(\boldsymbol{x}, \boldsymbol{x^*}) = \sigma_0^2 \exp\left(-\frac{|\boldsymbol{x} - \boldsymbol{x^*}|_2^2}{2l}\right) + \sigma_n^2 \boldsymbol{x}^T \boldsymbol{x^*}, \tag{3.11}$$

where the hyper-parameters σ_0 and l are the amplitude and length scale which can be learned from the training data by minimizing the negative log marginal likelihood with respect to the hyper-parameters [192].

The least-square classification is simple and effective, but generally the misclassification rate is slightly higher than other approaches, such as neural networks [194]. Other approximation methods like Laplace approximation [195,196] can provide more accurate predictions, but these approaches are very computationally expensive and thus not appropriate for mobile devices.

3.3.3 Hypothesis Testing Classifier (HTC)

The HTC can identify NLOS conditions with only least square approximation of the mean and variance of features. Compared with LS-SVMC and GPC which are

accurate but expensive in terms of training cost, HTC skips the expensive training phase at the cost of degraded performance and flexibility.

To identify the LOS/NLOS conditions, we employ the well-known likelihood ratio test where the two competing hypotheses (LOS/NLOS) are defined as

$$\begin{aligned} H_l: &\quad h \le h_t, \quad \text{LOS conditions } (b=1), \\ H_n: &\quad h > h_t, \quad \text{NLOS conditions } (b=-1). \end{aligned} \tag{3.12}$$

That is, what we need is a proper function h and a threshold h_t to identify the NLOS conditions.

Recall that only the five features μ, $\mathcal{K}$, $\mathcal{S}$, K_r, and χ^2 in Section 3.2 help improve the identification accuracy in hypotheses testing. Denote the five features as $x^{(i)}, i = 1, \cdots, 5$, and the joint distribution of M $(1 \le M \le 5)$ features with $p(x^{(1)}, \cdots, x^{(M)}|H)$. Since the joint distribution requires the convolution of the PDFs, the computation complexity could be extraordinarily high. To make the algorithm practical, we adopt a suboptimal solution which assumes the variables are independent. Then h is defined as

$$h = \frac{p(x^{(1)}, \cdots, x^{(M)}|H_l)}{p(x^{(1)}, \cdots, x^{(M)}|H_n)} = \prod_{i=1}^{M} \frac{p(x^{(i)}|H_l)}{p(x^{(i)}|H_n)}, \tag{3.13}$$

where $p(x^{(i)}|H_l)$ and $p(x^{(i)}|H_n)$ are the distributions of feature $x^{(i)}$ in LOS and NLOS conditions, and the threshold $h_t = 1$.

To provide evidence for the validity of the suboptimal solution, we first learn joint distributions $p(\mu, \mathcal{K}, K_r|H_l)$ and $p(\mu, \mathcal{K}, K_r|H_n)$ from 29,700 sets of features extracted from all RSS we collected in the experiments discussed in Sec. 3.5. Then we count the cases when the learned joint distributions without independence assumption and the suboptimal solution make opposite decisions, e.g., $\prod_{i=1}^{M} \frac{p(x^{(i)}|H_l)}{p(x^{(i)}|H_n)} < 1$ when $\frac{p(x^{(1)},\cdots,x^{(M)}|H_l)}{p(x^{(1)},\cdots,x^{(M)}|H_n)} > 1$ or the other way around. We observed only 2.02% of such events, which justifies the use of the suboptimal solution, especially in light of its significantly reduced computation requirements.

3.4 NLOS Mitigation

In this section, we present three algorithms, including LS-SVMR, GPR, and HTR, to accurately estimate the transmitter-receiver **distances** in LOS and NLOS conditions. In addition, we also make comparisons between the cost and performance of these algorithms in the context of our application.

3.4.1 Least Square Support Vector Machine Regressor (LS-SVMR)

The training data of the LS-SVMR are various subsets of features extracted from the RSS samples (input) and the estimated transmitter-receiver distances (output), denoted with $\{\boldsymbol{x}_k, d_k\}_{k=1}^N$. The regressor is very similar to the classifier on the optimization problem. The regressor is a function from $\mathbb{R}^n$ to $\mathbb{R}$.

$$d(\boldsymbol{x}) = \boldsymbol{w}^T \varphi(\boldsymbol{x}) + w_0, \tag{3.14}$$

where $d(\boldsymbol{x})$ is the estimated distance with input $\boldsymbol{x}$ using LS-SVMR.

The distances between the support vectors and the separating hyperplane are maximized by Eqn. (3.15).

$$\begin{aligned} \underset{\boldsymbol{w},w_0,\boldsymbol{e}}{\operatorname{argmin}} \quad & \tfrac{||\boldsymbol{w}||^2}{2} + c\tfrac{1}{2}\textstyle\sum_{k=1}^N e_k^2 \\ \text{s.t.} \quad & d_k = \boldsymbol{w}^T \varphi(\boldsymbol{x}_k) + w_0 + e_k, \quad \forall k. \end{aligned} \tag{3.15}$$

where c is also the weighted factor as in Eqn. (3.3) and e_k is the regression penalty learned from the optimization problem (3.15). Similar to the classification optimization problem in Eqn. (3.3), the regression problem in Eqn. (3.15) can also be solved by standard optimization tools.

3.4.2 Gaussian Process Regressor (GPR)

As we use least-square classification in GPC, the regression for Gaussian processes is almost identical to the classification in Sec. 3.3.2 with a single difference in the training and testing output. The output in the classification is the LOS/NLOS labels while for regression it is the transmitter-receiver distance. Therefore, given the same training data as in LS-SVMR for Gaussian processes regression $\{\boldsymbol{x}_k, d_k\}_{k=1}^N$, the linear regression model with Gaussian noise for regression is

$$d(\boldsymbol{x}) = \boldsymbol{w}^T \varphi(\boldsymbol{x}) + n, \tag{3.16}$$

where $\boldsymbol{w} \sim \mathcal{N}(\boldsymbol{0}, \Sigma_p)$ and the measurement noise $n \sim \mathcal{N}(0, \sigma_n^2)$. The rest of the regression is identical to the process from Eqns. (3.6) to (3.11) in GPC with continuous distance $\boldsymbol{d}$ in place of discrete labels $\boldsymbol{b}$.

Compared to LS-SVMR, the training phase of GPR is more expensive because it takes all training and testing samples which might be thousands of samples as a joint Gaussian distribution and predicts the output as a conditional mean rather than only maximizing the distances between the supporting vectors and the separating hyperplane in LS-SVMR.

3.4.3 Hypothesis Testing Regressor

In general, NLOS mitigation is achieved by regression. However, since hypothesis testing yields binary decision, we use different propagation models based on the decision. Many propagation models have been developed so far. For instance, the log-normal propagation model:

$$P(d)[\text{dBm}] = P(d_0) + 10\gamma \log \frac{d}{d_0} + WAF + X_\sigma, \tag{3.17}$$

in which $P(d)$ is the path loss in a location d meters away from the AP, d_0 is the reference distance, γ is the distance power loss coefficient, WAF is the wall attenuation factor which accounts for both real walls in the environment and other minor obstacles like the cases of the mobile devices or access points, and X_σ is a zero-mean Gaussian distributed random variable with variance σ^2 to account for shadowing.

NLOS mitigation is achieved by incorporating HTC and such propagation models. For LOS and NLOS conditions the parameters in Eqn. (3.17) (especially γ and σ) can vary greatly thus producing significant errors in distance estimation should inappropriate parameters be used. Therefore, we first identify the LOS/NLOS conditions using HTC and then estimate the transmitter-receiver distances with different propagation models for LOS and NLOS conditions. Specifically, we use different propagation parameters including γ, σ, and WAF, approximated by least-squares from the training data, for LOS and NLOS conditions. For instance, incorporating the LOS/NLOS conditions into Eqn. (3.17) makes HTR as follows.

$$P(d) = \begin{cases} P(d_0) + 10\gamma^L \log \frac{d}{d_0} + WAF^L + X_\sigma, & H = H_l, \\ P(d_0) + 10\gamma^N \log \frac{d}{d_0} + WAF^N + X_\sigma, & H = H_n. \end{cases} \tag{3.18}$$

The incorporation of LOS/NLOS conditions with other propagation models is similar.

Compared to LS-SVMR and GPR, HTR only requires least-square approximation of the propagation parameters (γ, σ, and WAF) and feature distribution parameters (mean, variance) from training data. However, the reduced training phase and the independence assumption between features will degrade the distance estimation performance.

3.5 Evaluation

This section demonstrates and evaluates the proposed algorithms through extensive data collection at different times and places. To make the evaluation reliable, RSS

samples are collected during different periods of the day to account for time variability, and in three experimental sites with different floor plans to account for environment variability.

3.5.1 Experimental Setup

Sites: The proposed NLOS identification and mitigation algorithms were evaluated in three different real-world settings, namely an office, a basement, and an attic. The site shown in Fig. 3.4 is the 4th floor of a multi-storey office building with a stone and brick construction, reinforced with metal rebars. The majority of RSS samples in this site were collected along the corridors surrounded by rooms. The site shown in Fig. 3.10 is a basement where the RSS samples were collected in both corridors and rooms which are surrounded by soil and rocks. Fig. 3.11(a) shows the layout of the attic test location. All access points and locations where RSS samples are collected in three experiment sites are marked in Figs. 3.4, 3.10, and 3.11(a).

Devices and Implementations: To make the proposed approaches more readily implementable for localization, different mobile devices, including Acer Aspire One ZG5 running Ubuntu 3.2.0 and Huawei U8160 mobile phones running Android 2.3.3, were utilized in the experiments. To account for the various relative antenna orientations that may occur between transmitters and receivers, the mobile device always kept face up and oriented parallel to the trajectories taken so the relative orientation changed with movement during the data collection. Measurements from over ten different mobile devices were fused in the experiments to account for hardware variations between these mobile devices.

Ground truth: To provide accurate ground truth, numbered labels were placed along corridors and within rooms on a 1 meter grid where experiments were conducted. We recorded the current time and label number manually when the mobile device reached a certain label. The labels are then mapped to locations on the floor plan to obtain the ground truth of LOS/NLOS conditions and transmitter-receiver distances. With the settings of access point locations and data collection locations, the transmitter-receiver distances vary from 0.8 meters to 25 meters with maximum discretization step of 1m.

Data collection: We implemented two approaches to collect RSS samples: passive and active. With the passive approach, a mobile device scans for beacons transmitted by the various APs and then records RSS values. Given that the default beacon interval is 102.4 ms (or 51.02 ms for some APs), it is possible to collect 10-20 samples from each AP per second. APs can be configured to send beacons as rapidly as 1000

Hz, but this involves reconfiguring infrastructure, which one does not necessarily have access to. In comparison, during the active approach, a mobile device broadcasts probe requests and obtains RSS samples from probe responses. The active approach collects data much more quickly (can be over 1000 Hz) than the passive approach at the cost of higher energy consumption on the mobile device. In practice, one can switch between the passive approach and the active approach according to the motion of the user, found by monitoring the change in the mean per AP RSS. We used the active approach when the RSS changed rapidly and the passive approach otherwise. We collected 50 to 1000 samples at each given location. Since the numbers of LOS and NLOS samples differ in various scenarios, we collect half of the samples in LOS conditions and the other half in NLOS conditions for training and testing.

3.5.2 Data Sets and Training

The accuracy of NLOS identification techniques can be easily decreased by interference from people walking around and other signal noise. Although people around may not block the LOS signal, they may block and absorb other components of the received WiFi signal which leads to the variation of the measurement distribution. Moreover, from the long-term perspective of practical use, it is impossible to avoid interference from people.

Therefore, we have two separate categories of RSS samples in the data sets to account for the interference from people. The first group of samples was collected during nights when there were few people walking around to absorb and block the WiFi signal (called *static environment* hereafter). The other group of samples was collected during busy office hours when there were many people walking around the corridors, which interfere with the RSS measurements (called *dynamic environment* hereafter). Each of the two groups contains approximately 1500 sample sets, each of which is composed of 1000 RSS samples (3,360,000 RSS samples in total). We divided each sample set into subsets according to the sample size, as discussed in the next subsection, and extract features from each subset. As noted, half of the sample sets in each group were taken from LOS conditions and the other half from NLOS conditions.

After the database of measurements was built, we tested the accuracy of the proposed algorithms using five-fold cross validation. Specifically, we randomly divide all the RSS measurements collected in the office site into five data sets. Then the algorithms were trained with four datasets and tested with the remaining dataset. The training and testing processes were repeated five times until each data set has

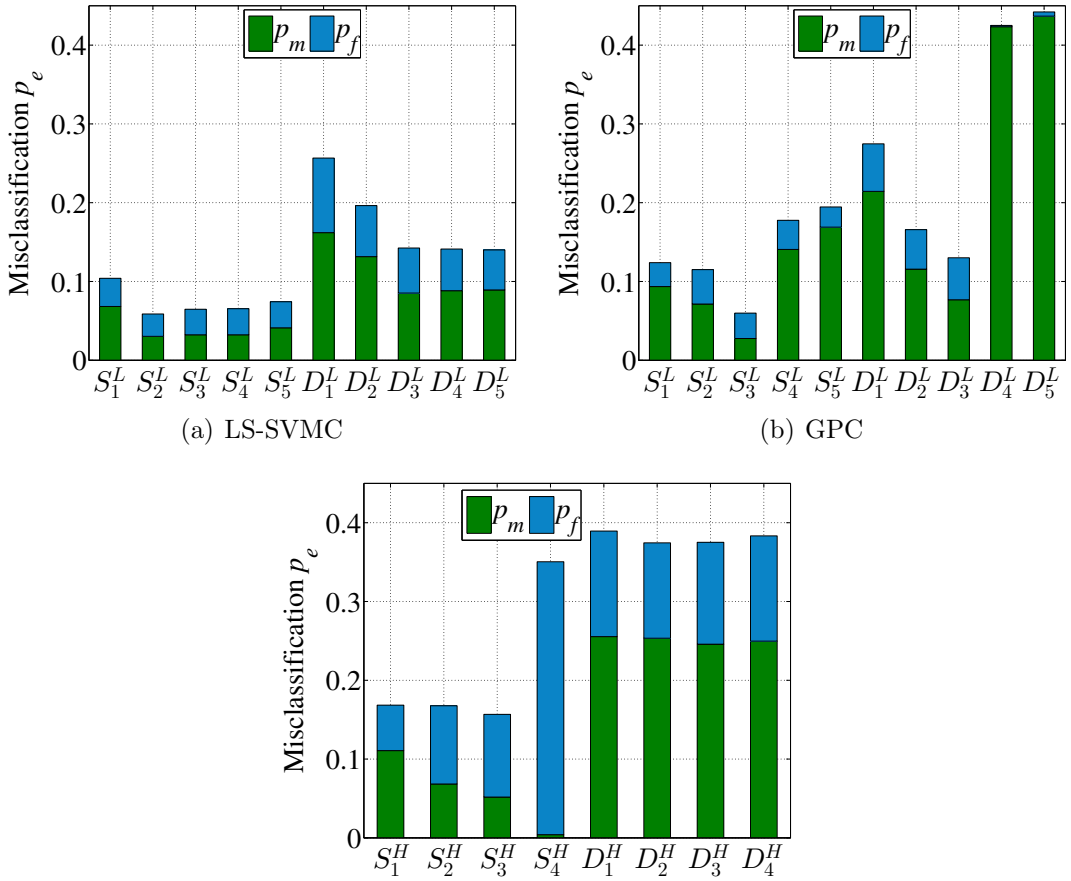
Table 3.1: Features in different feature subsets.

Subsets	μ	σ_s	$\mathcal{K}$	$\mathcal{S}$	K_r	χ^2
S_1^L	✓					
S_2^L	✓				✓	
S_3^L	✓				✓	✓
S_4^L	✓	✓			✓	✓
S_5^L	✓	✓	✓	✓	✓	
D_1^L	✓					
D_2^L	✓			✓		
D_3^L	✓	✓		✓		
D_4^L	✓		✓	✓	✓	
D_5^L	✓		✓	✓	✓	✓
S_1^H	✓					
S_2^H	✓			✓		
S_3^H	✓		✓		✓	
S_4^H	✓		✓		✓	✓
D_1^H				✓		
D_2^H			✓	✓		
D_3^H	✓		✓	✓		
D_4^H	✓		✓	✓	✓	

been tested. Then we calculate the mean accuracy as the performance metric of the algorithm on the whole database. The performance of the classifiers and regressors are discussed in the next subsection.

3.5.3 Testing

In this subsection we discuss the performance of the proposed NLOS identification and mitigation algorithms with both the training data and testing data from the office site in Fig. 3.4. For a given number of features, e.g., three, we performed cross validation with all $\binom{7}{3}$ combinations of the features introduced in Sec. 3.2. The feature subset with the lowest misclassification rate and distance estimation errors are presented in Tables 3.1 and 3.2. We denote the feature subsets with lowest misclassification rate for machine learning techniques (LS-SVMC and GPC) in static and dynamic environments with S_i^L and D_i^L $(i = 1, \cdots, 5)$, respectively. The feature subsets with lowest misclassification rate for HTC in static and dynamic environments are denoted with S_i^H and D_i^H $(i = 1, \cdots, 4)$, respectively. The feature subsets with lowest distance estimation errors for machine learning techniques are denoted with S_i^R $(i = 1, \cdots, 5)$.



(a) LS-SVMC (b) GPC

(c) HTC

Figure 3.5: Missed detection probability (p_m), false alarm probability (p_f), and overall misclassification probability (p_e) of the proposed algorithms, showing the impact of different feature sets.

3.5.3.1 NLOS Identification

The performance of the algorithms is measured in terms of missed detection probability p_m (deciding LOS when the RSS samples are from NLOS conditions), false alarm probability p_f (deciding NLOS when the RSS samples are from LOS conditions), and overall misclassification probability $p_e = p_m + p_f$.

Fig. 3.5 shows the misclassification rates of the proposed techniques in static and dynamic environments using varying number of features shown in Table 3.1. We can observe that the misclassification rates of LS-SVMC in the static environment is far better than those in the dynamic environment. The best identification errors for LS-SVMC, GPC, and HTC are 0.0648, 0.0599, and 0.1568 in static environments, and 0.1401, 0.1301, and as high as 0.3744 in dynamic environments. The major reason for the poor performance of HTC is that HTC is equivalent to a linear classifier while the

majority of LOS/NLOS conditions are not linearly separable, as shown in Figs. 3.1 and 3.3, especially with nonlinear features like χ^2 or in dynamic environments, which explains the poor performance of S_4^H and feature sets in dynamic environments.

From Figs. 3.5(a) and 3.5(b), K_r appears in most feature sets in static environments, which proves K_r to be indicative for NLOS identification in static environment. The reason is straightforward: K_r measures the difference between a Rician distribution (LOS condition) and a Rayleigh distribution (NLOS condition). However in dynamic environments, K_r is no longer an essential feature, which indicates that the interference actually blurs the demarcation line between distributions in LOS and NLOS conditions. Instead, the skewness appears in each data set and thus becomes the most crucial feature.

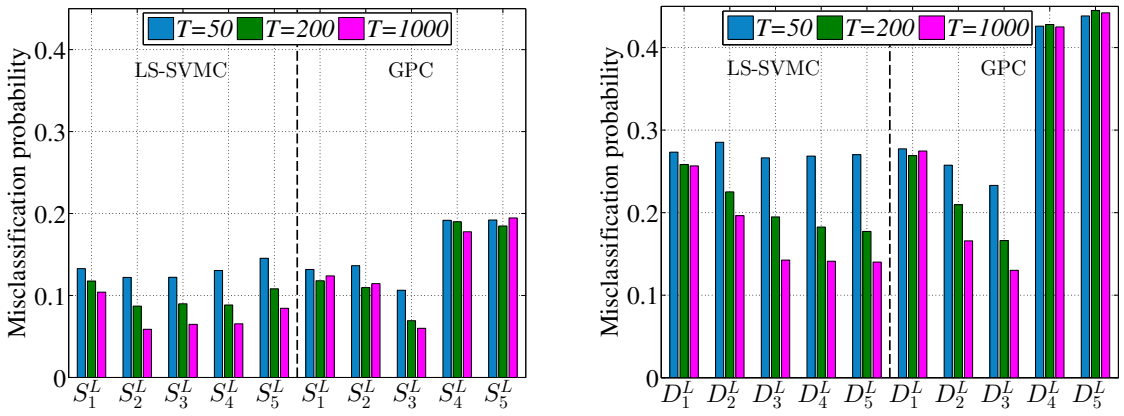
Fig. 3.6 compares the misclassification rates of LS-SVMC and GPC using different sample sizes in static and dynamic environments. In both static and dynamic environments, the misclassification rates are the lowest when the sample size is 1000 and the largest when the sample size is 50, which indicates that the number of samples collected at each location also impacts the identification accuracy. The reason for the impact of sample size is that a larger number of samples can reduce the influence of noisy RSS samples, which leads to a more precise fit of the samples to a distribution and thus a more accurate result.

The number of packets exchanged during the reception of a standard text email is on average around 30, including the overhead packets like beacons, handshake, and handoff. Based on the above experiments the number of RSS samples from these packets would be sufficient for the proposed technique to provide an acceptable NLOS identification accuracy. An email with picture attachments contains hundreds of MAC layer packets which can make the identification very accurate without any change to the existing protocol stacks or infrastructure.

Summary: The identification accuracy of LS-SVMC and GPC can be up to around 95% with training phase while HTC can greatly reduce the training phase at the cost of lowering the NLOS identification accuracy to 85%. For all three identification techniques, the sample size has impact on the accuracy.

3.5.3.2 NLOS Mitigation

In this subsection, we will discuss the accuracy improvement of distance estimation with NLOS mitigation techniques compared with conventional propagation models.



(a) Static environments (b) Dynamic Environments

Figure 3.6: Overall misclassification probability in static (a) and dynamic (b) environments for different sample sizes. Features of different sizes are consistent with Table 3.1.

Standard Propagation Model (SPM): This is the first strawman algorithm used for comparison. We used all training RSS data including both LOS and NLOS conditions to estimate γ and WAF in Eqn. (3.17) with least squares. Note that SPM does not differentiate between LOS from NLOS conditions.

Breakpoint Propagation Model (BPM): This is the second benchmark algorithm for comparison [197]. This propagation model incorporates the breakpoint phenomenon [198] into the log-distance propagation model, which makes the breakpoint propagation model [197]. The breakpoint distance can be determined by the first Fresnel zone [199] and takes the typical value of 7m for WiFi signals between APs and mobile devices in this study [200]. Similarly, we also use least squares approximation to derive other BPM parameters such as γ and WAF.

Least Square Support Vector Machine Regression (LS-SVMR): This is the first proposed algorithm. Instead of estimating the distance with only the mean of RSS measurements in the propagation models, LS-SVMR takes into account more features of the RSS measurements to estimate the distances.

Gaussian Processes Regression (GPR): This is the second algorithm proposed. GPR uses the same feature sets as LS-SVMR.

Hypothesis Testing Regression (HTR): This is the third proposed algorithm. It combines HTC, which identifies LOS/NLOS conditions, with the variant of SPM described in Eqn. (3.18) for distance estimation.

The distance estimation errors of various models are shown in Fig. 3.7. It is observed that HTR only improves by around 20% compared to BPM. In comparison,

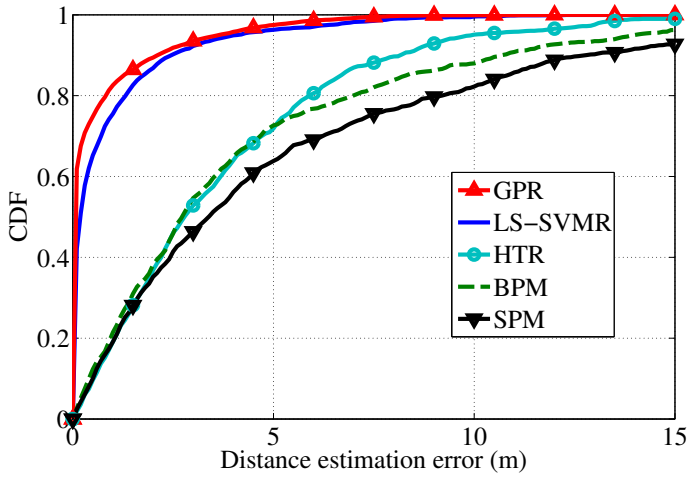
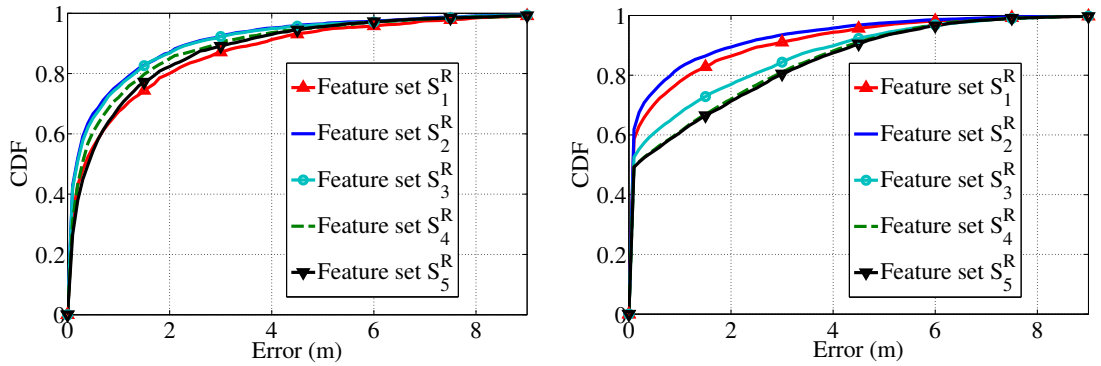


Figure 3.7: CDF of various distance estimation methods.



(a) LS-SVMR (b) GPR

Figure 3.8: Comparison of distance estimation performance between LS-SVMR and GPR.

both LS-SVMR and GPR greatly outperform other models with the mean error from 6.61m for SPM to 0.86m for LS-SVMR and 0.82m for GPR.

Fig. 3.8 compares the distance estimation performance of LS-SVMR and GPR with different feature sets. Subsets with different features are shown in Table 3.2. It is observed that GPR works slightly better than LS-SVMR with the best feature set (S_2^R), while its performance rapidly drops with other feature sets. The impact of limited training on the distance estimation performance is shown in Fig. 3.9. We can observe that the distance estimation accuracy can be improved with more training data. This trend is apparent especially when the percentage of training data is low (less than 40%). Note that there is no overlap between the training and testing data.

It is observed from Figs. 3.5 and 3.8 that the identification accuracy and distance estimation errors of LS-SVMC/LS-SVMR are more stable with more noisy features

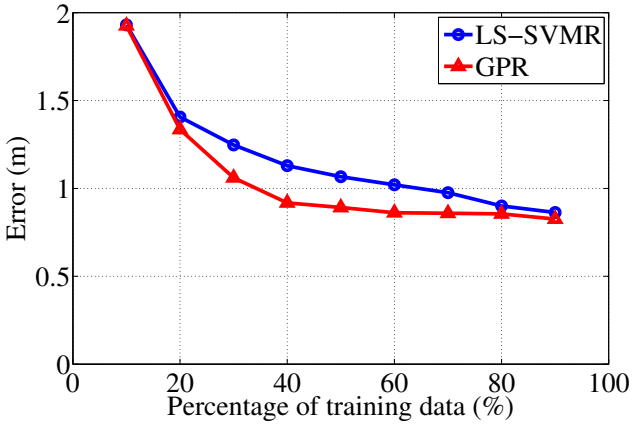


Figure 3.9: Impact of limited training.

Subsets	Features						
	μ	σ_s	$\mathcal{K}$	$\mathcal{S}$	K_r	χ^2	$\hat{d}$
S_1^R	✓						
S_2^R	✓						✓
S_3^R	✓					✓	✓
S_4^R	✓				✓	✓	✓
S_5^R	✓	✓			✓	✓	✓

Table 3.2: Feature subsets for LS-SVMR/GPR

than GPC/GPR. As spline smoothing techniques, both LS-SVM and GP predict a new target by reweighting the vector of training targets to arrive at a weight vector which is then used to form an average over the correlations between the new input point and the training points. The difference between the two methods lies in the reweighting factors for the targets. In our implementation of GPC/GPR, all features share the same length scale to reduce the training computational complexity, which also reduces the feature selection ability of GPC/GPR. Therefore, the performance of GPC/GPR is degraded when noisy features are included. To increase the feature selection ability and make GPC/GPR robust to noisy features, we can implement automatic relevance determination (ARD) in the parameter learning, which, intuitively, learns different length scales for different features.

It is also observed that HTC/HTR always performs worse than LS-SVMC/LS-SVMR and GPC/GPR. The underlying reasons are three-fold. As noted, HTC is equivalent to a linear classifier while the LOS/NLOS conditions are not linearly separable using only RSS. In addition, the identification errors of HTC still have impact on HTR because HTR is the combination of HTC and propagation models. Moreover, the log-distance propagation model used in HTR is only a rough two-parameter ap-

proximation of the actual RSS-distance relation, as shown in Fig. 3.1. On the contrary, LS-SVMR and GPR can learn a best model in the least square sense which can describe the RSS-distance relation much better than a simple log-distance model. Even with only one feature (e.g., the RSS mean), LS-SVMR and GPR perform much better than the HTR – the combination of HTC and the log-distance propagation model. Finally, LS-SVMR and GPR take more useful features into account for distance estimation than the propagation model used in HTR, which further helps improve the distance estimation accuracy.

In addition, Fig. 3.8 shows that the performance of the regressor does not always improve as we add more features. Only informative features like the mean μ and log-mean $\hat{d}$ (please refer to Sec. 3.2 for details) can improve the regression accuracy while noisy features like σ_s could degrade the regression performance. That is why we can observe that the feature set S_2^R that includes both informative features (μ and $\hat{d}$) always performs better than any feature set with more than these two features.

Summary: LS-SVMR and GPR can greatly improve the distance estimation accuracy to around 0.86m as opposed to over 6.6m with conventional propagation models. In addition, with a comparatively much shorter training phase, HTR can achieve an accuracy of 3.5m.

3.5.4 Validation

This subsection validates the performance of the proposed algorithms in a different experiment site (basement site) than from where the training data was collected (office site). It is crucial for practical algorithms to work in different sites without per-site training because training is highly labour intensive.

To test the robustness of the two machine learning based NLOS identification and mitigation algorithms over different sites, we use the same set of parameters trained from the static environments in the office site in Fig. 3.4 to identify and mitigate the NLOS conditions in the basement site as shown in Fig. 3.10.

The overall misclassification rates of both the LS-SVMC and GPR with the best 3-feature set $S_3^L(\mu, K_r, \chi^2)$ from Fig. 3.5 are around 0.091 when tested in the basement site, as opposed to 0.065 when tested in the office site. In terms of average distance estimation error, the mean distance estimation error is 2.41m for LS-SVMR and 2.33m for GPR in the robustness testing, as opposed to 0.86m and 0.82m when they were trained and tested in the office site. The observed errors still far outperform the propagation models by over 60% improvement over SPM.

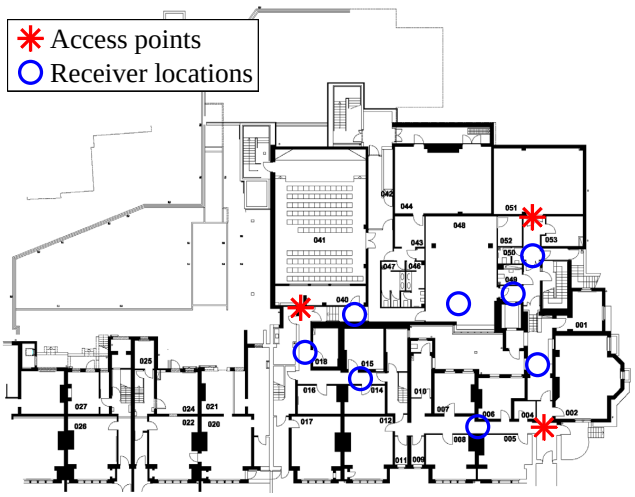


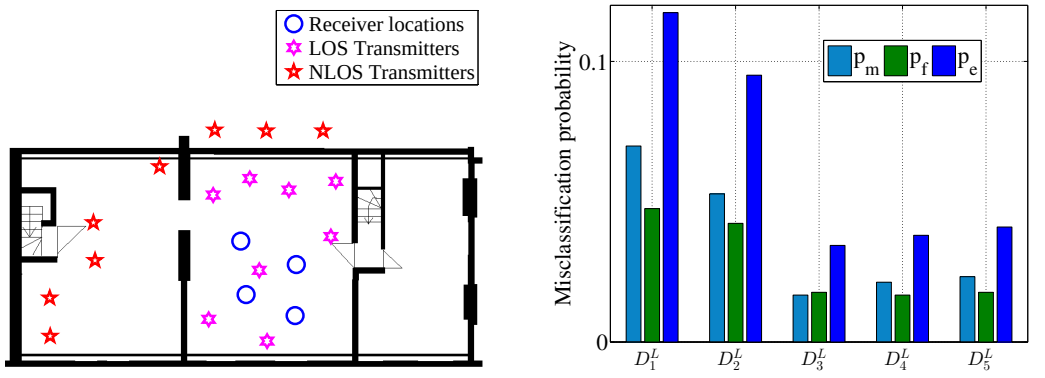
Figure 3.10: Map of the basement site ($55m \times 40m$). Access points are marked in the maps.

From the aforementioned experiments, the "general applicability" of the training parameters for machine learning based NLOS identification and mitigation algorithm makes the training phase very easy. Therefore, although the training phase of the machine learning based method is time-consuming and costly, we do not have to train the parameters every time we plan to use them. Instead, we could use the same parameters from other applications concerning the NLOS identification and mitigation algorithms. In addition, since RSS is a simple measurement that every WiFi device can provide, potentially we can have a huge amount of crowd-sourced data. Based on such data, it would be possible that an unsupervised learning algorithms such as Expectation Maximization (EM) [201] could help in learning the algorithm parameters.

Summary: In a different environment from where the training parameters are obtained, LS-SVMC and GPC can also obtain over 90% accuracy in NLOS identification. Meanwhile, LS-SVMR and GPR can achieve an accuracy 60% higher than traditional distance estimation approaches.

3.5.5 Robustness

It is important that the proposed NLOS identification and mitigation algorithms take into account the impact of different signal properties such as the transmission power level and signal frequency. The goal of the following experiments was to test whether our algorithms are robust against such changes.



(a) The attic site ($17m \times 7m$) (b) The misclassification rates of LS-SVMC

Figure 3.11: Experiment site and misclassification rate of LS-SVMC in dynamic environment in testing the robustness of our system against various signal properties.

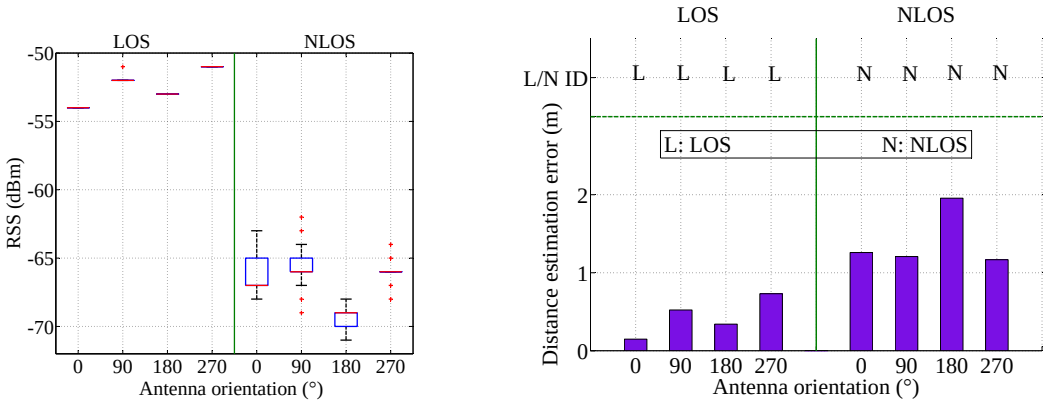
Fig. 3.11(a) shows the experiment scenario of testing the impact of different WiFi signal properties on our NLOS identification algorithms. Four receivers were put inside a room together with eight LOS transmitters which are evenly distributed within the room. Meanwhile, one group of five NLOS transmitters are put in a neighbouring room connected by an open door while the other NLOS group of three transmitters are put outside the room where the transmitters and receivers are separated by a wall built with concrete building materials. Both LOS and NLOS transmitters are set to work with different transmission power levels ranging from 0 dBm to 20 dBm. In addition, some of the LOS and NLOS transmitters are set to transverse all available WiFi channels (hop to the next legal frequency band periodically) to test the influence of signal frequency on our system.

Transmission Power Level

The misclassification rate for LS-SVMC in this experiment is shown in Fig. 3.11(b). It is observed that the same feature sets in dynamic environment from Fig. 3.5 give a misclassification rate as low as 0.0344 for feature set D_3^L $(\mu, \sigma_s, \mathcal{S})$, which further demonstrates the effectiveness of our algorithm. The proposed algorithm performs better in this experiment than in Sec. 3.5.3 as the extracted RSS features are able to discriminate better the conditions when the transmitter is in the same room as the receiver (LOS) and when the transmitter is in a different room (NLOS).

Signal Frequency

The experiment results show that the switch from one channel to another does not impact our algorithm. The reason is that we are collecting a large number of RSS data, we can "average out" small scale multipath effects which might be more



(a) The raw RSS

(b) Identification and mitigation accuracy

Figure 3.12: The impact of antenna orientation on the NLOS identification and mitigation algorithms.

apparent in some channels than others.

Antenna Orientation

We also evaluate the impact of antenna orientations on the proposed algorithms. Fig. 3.12(a) shows the mean and 75 percentile of the RSS measurements we collected in LOS and NLOS conditions with the same transmitter-receiver distance but different antenna orientations. It is observed that antenna orientations have impact on the RSS, especially in NLOS conditions. However, from Fig. 3.12(b) we have demonstrated that GPC can identify the LOS/NLOS conditions with 100% accuracy (please see the "L/N ID" at the top of the plot). In addition, GPR can also predict nearly identical distances with different antenna orientations.

Summary: The experimental results show that the proposed NLOS identification and mitigation algorithms are robust against the change of transmission power level, signal frequency, and antenna orientations.

3.6 Impact on Positioning System

With the NLOS mitigation techniques in Sec. 3.5, we can derive the locations of the mobile devices with trilateration [202]. Fig. 3.13 compares the trajectories estimated from different localization approaches. All trajectories presented are estimated from the same raw RSS measurements. The trajectories are estimated from (a) ground truth, (b) SPM, (c) Guvenc et al. [100], (d) HTR, (e) Fingerprinting (HORUS) [113], (f) BPM, (g) Nawaz et al. [110], and (h) LS-SVMR. As the performance of GPR is similar to LS-SVMR, it is not included here.

Figs. 3.13 shows the raw trajectories generated by different algorithms. It is apparent that the frequent outliers observed in these trajectories can significantly degrade the performance of the corresponding algorithms. To overcome this outlier problem, a simple particle filter with 2000 particles has been introduced to filter out the outliers and smooth the trajectories, as shown in Fig. 3.14.

The particle filter updates the particles with randomly generated distance d and walking directions θ. With the assumption that human walking speed is less than 3 m/s, the update of the particle is

$$\begin{aligned} x(t) &= x(t-1) + d\cos(\theta), \\ y(t) &= y(t-1) + d\sin(\theta), \end{aligned} \tag{3.19}$$

where d is uniformly sampled from $[0, 3]$ and θ is uniformly sampled from $[-\pi, \pi]$.

The weight of each particle is determined by assuming the location estimation from trilateration has Gaussian noise. The importance resampling strategy is then applied to resample particles according to their weights. The current location is then reported as the weighted mean of all particles.

It is observed from Fig. 3.14 that the approach proposed by Guvenc et al. [100], which selects the AP subset with minimum weighted residual, works fairly well in our experiments[1]. In addition, recall that the generic NLOS mitigation approach proposed by Nawaz et al. in [110] tries to reduce the estimation errors by assuming that the number of LOS APs is greater than the number of NLOS APs. In our experiments this algorithm results in poor performance at many locations (RMSE up to 10 meters) where NLOS APs outnumber LOS APs.

Fig. 3.14 also shows that there is only a slight improvement in accuracy for the propagation model derived from hypothesis testing results, compared with the generic NLOS identification and mitigation approaches proposed in existing work [100, 110]. The inability of a simple propagation model to capture major features of different complicated indoor environments results in this phenomenon.

We can also see from Fig. 3.14 that the localization system based on LS-SVMR improves the localization accuracy by 60% compared with the trajectories estimated with the state-of-the-art NLOS mitigation algorithms, which greatly increases the potential of using WiFi-based localization in practical settings.

[1]Since the mean excess delay and RMS delay spread measurements are not available from WiFi signals, we use the best feature set in our experiments (μ, $\mathcal{K}$, and K_r) instead which are calculated from the directly measurable RSS.

We also evaluated the positioning experiment settings including AP and receiver locations, attenuation factor distributions in this study with Cramer Rao Lower Bounds (CRLB) [203]. The path loss based theoretical CRLB of the positioning error is 1.31m, almost the same as the positioning error of the proposed machine learning approaches (approx. 1.30m), which proves the effectiveness of the proposed methods. The major reasons why the positioning error of the proposed approaches is slightly smaller (less than 1%) than CRLB are a) the assumption of CRLB that each AP has the same distribution for attenuation factor γ is not true in practice, especially in indoor environments, and b) the particle filter removes outliers and improves the positioning accuracy.

In addition, we implemented the well-known fingerprinting approach HORUS [113] with the same training and testing data as the proposed techniques. It is observed from Figs. 3.14(b) and 3.14(h) that the accuracies of HORUS and LS-SVMR/GPR are comparable. However, one major advantage of the proposed approaches is that LS-SVMR/GPR can be used in different environments (as shown in Sec. 3.5.4), which means the proposed techniques do not require per-site training or extra user effort. On the contrary, it is well-known that per-site training is a must for fingerprinting approaches because different sites have different access points and links between the locations and RSS from different access points.

Summary: The proposed NLOS mitigation algorithms can greatly improve the distance estimation accuracy and thus improve the RSS-based localization accuracy by over 60% compared with the state-of-the-art NLOS mitigation algorithms. Performance is shown to be comparable with site-specific fingerprinting approaches.

3.7 Discussion

This chapter has proposed three algorithms to address the problem of NLOS identification and mitigation using only WiFi RSS measurements. To our knowledge, this is the first time NLOS identification and mitigation are conducted using only RSS from real experiments with mobile devices. The proposed algorithms can not only accurately identify NLOS conditions, but also greatly improve the transmitter-receiver distance estimation. In addition, the proposed algorithms are robust to changes in environments and signal properties, and conditions without any LOS anchors. That is, algorithms can be trained in one environment and be utilized in another. More specifically, using a combination of mean, Rician K factor and χ^2 goodness of fit along

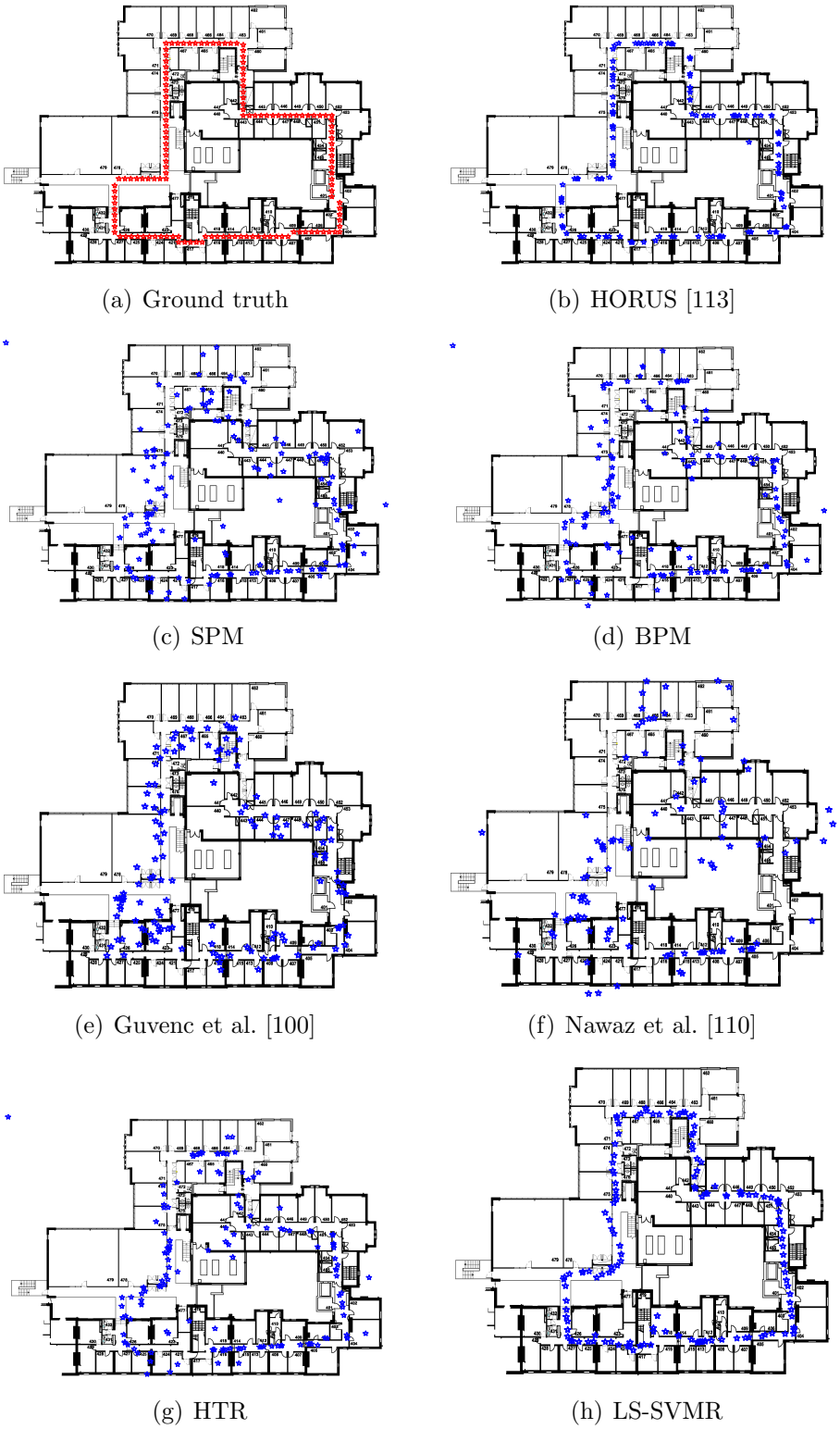
with LS-SVMC or GPC algorithms, proved to provide low (less than 10 percent) classification errors in new environments. Moreover, a combination of features such as mean, Rician K factor, and log-mean with LS-SVMR or GPR significantly improved RSS-based localization error over existing techniques. The high accuracy of NLOS identification and mitigation using generic model parameters makes the training for individual environments unnecessary, which makes the algorithms, to some degree, generic to a variety of user locations and greatly improves the potential use of our algorithms in real world applications. Note that though the proposed approaches have been tested in different environments, these environments have similar characteristics they are all medium sized environments of width less than 100 meters, and they include small- to mid-sized rooms, relatively narrow corridors, and small people density fluctuations. In drastically different settings, it is possible that the proposed and competing algorithms exhibit different performance. For example, one might observe that typical line of sight conditions in large open spaces are actually classified as non line of sight conditions because of people movement. Similarly, in buildings with very high fluctuations of people density (e.g. lecture theatres) one may expect frequent changes between LOS and NLOS conditions between the same transmitter and receiver nodes. We expect, however, our results to be fairly reproducible in typical mid-sized office buildings similar to the ones studied in this chapter.

While designed for implementation on mobile devices, the limitation of the proposed algorithms is the costly training phase. From the user's perspective, they are unlikely to collect all the training data before they can actually use the system. However, to reduce or even eliminate the training phase requires some other information, either from the environments like the map and the location of access points or from the existing algorithm database like the training data acquired in other buildings. Therefore, our future work will incorporate such information to develop online learning algorithms or unsupervised machine learning algorithms to identify the LOS/NLOS conditions. Alternatively, future models could also be initialized with our existing data sets and then adapted to new environments. In addition, the proposed techniques in this chapter have not accounted for some other interesting problems, e.g., the automatic switching between static and dynamic settings. These problems also present interesting directions for future work.

Note that the proposed approaches have achieved comparable performance with fingerprinting approaches which are generally preferable when the signal map is available. However, the construction of signal maps is very labor intensive, which hinders the wide adaption of fingerprinting approaches. The proposed approaches in this

chapter provide an accurate alternative with the absence of signal maps and other information like inertial sensor data.

Recall that for a practical indoor positioning solution, only existing devices such as smartphone and infrastructure like WiFi/BLE signals can be utilized. Specifically, IMU and WiFi/BLE are the most potential tools to aid the indoor positioning. Having discussed how to make the best from WiFi/BLE signals with the absence of various radio maps, the next chapter discusses the proposed robust pedestrian dead reckoning (R-PDR) component. Specifically, it first proposes new capabilities added to PDR system and then describes the renewed R-PDR steps involved to get the PDR trajectories.



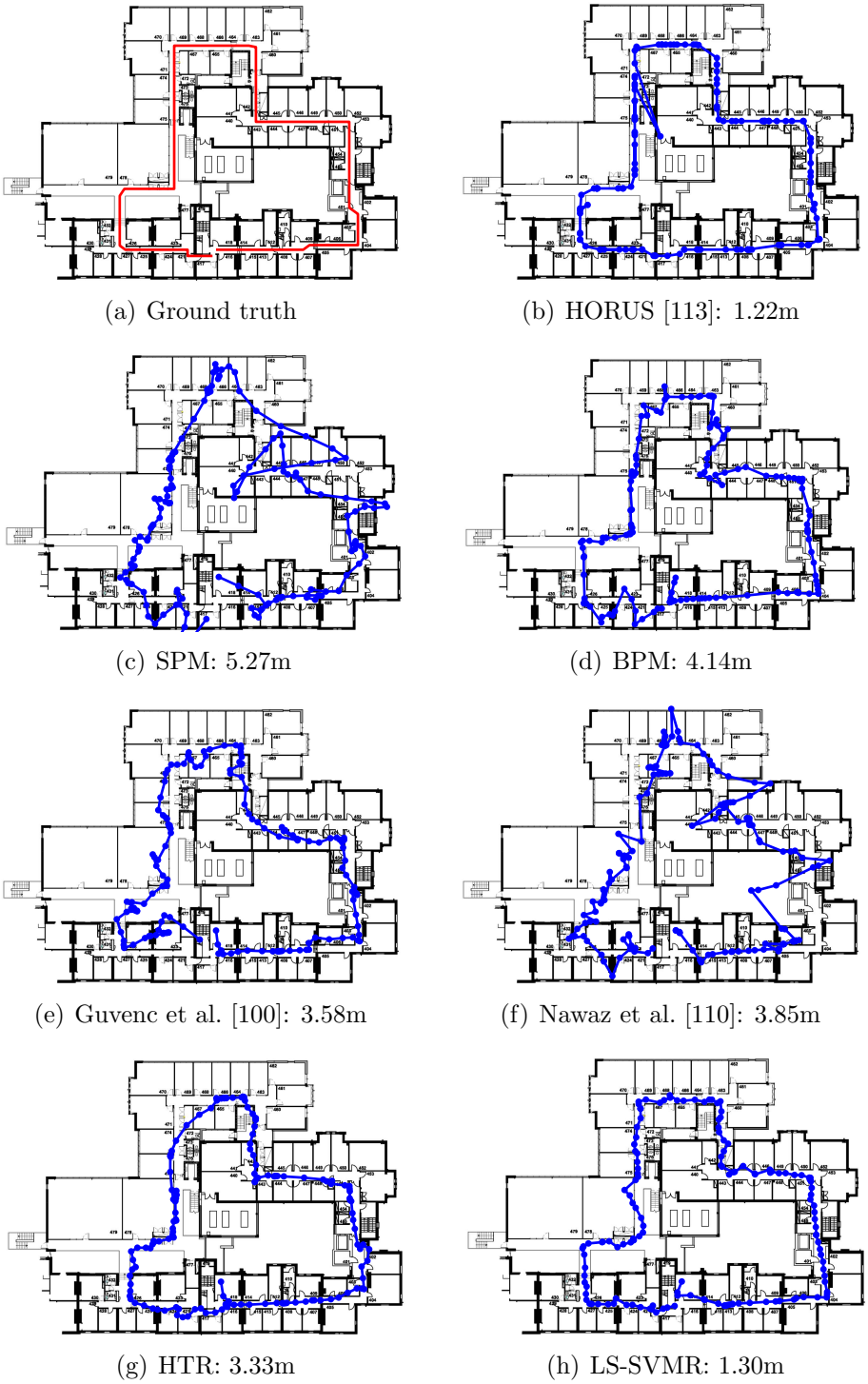
(a) Ground truth (b) HORUS [113]

(c) SPM (d) BPM

(e) Guvenc et al. [100] (f) Nawaz et al. [110]

(g) HTR (h) LS-SVMR

Figure 3.13: Discrete locations estimated by different algorithms: (a) Ground truth, (b) Fingerprinting (HORUS) (c) SPM, (d) BPM, (e) Guvenc et al., (f) Nawaz el al., (g) HTR, and (h) LS-SVMR.



(a) Ground truth (b) HORUS [113]: 1.22m

(c) SPM: 5.27m (d) BPM: 4.14m

(e) Guvenc et al. [100]: 3.58m (f) Nawaz et al. [110]: 3.85m

(g) HTR: 3.33m (h) LS-SVMR: 1.30m

Figure 3.14: Filtered trajectories along with the RMS errors: (a) Ground truth, (b) Fingerprinting (HORUS) (c) SPM, (d) BPM, (e) Guvenc et al., (f) Nawaz el al., (g) HTR, and (h) LS-SVMR.

Chapter 4

Robust Pedestrian Dead Reckoning

Pedestrian dead reckoning (PDR), especially on smartphones, is likely to play an increasingly important role in indoor tracking and navigation, due to its low cost and ability to work without any additional infrastructure. A challenge however, is that positioning, both in terms of step detection and heading estimation, must be accurate and reliable, even when the use of the device is varied in terms of placement (e.g. handheld or in a pocket) or orientation (e.g. holding the device in either portrait or landscape mode). Furthermore, the placement can vary over time as a user performs different tasks, such as making a call or carrying the device in a bag. A second challenge is to be able to distinguish between true steps and other periodic motions such as swinging an arm or tapping when the placement and orientation of the device is unknown. If this is not done correctly, then the PDR system typically overestimates the number of steps taken, leading to a significant long term error.

This chapter presents a fresh approach, robust PDR (R-PDR), based on exploiting how bipedal motion impacts acquired sensor waveforms. Rather than attempting to recognize different placements through sensor data, we instead simply determine whether the motion of one or both legs impacts the measurements. In addition, we formulate a set of techniques to accurately estimate the device orientation, which allows us to very accurately (typically over 99%) reject false positives. We demonstrate that regardless of device placement, we are able to detect the number of steps taken with >99.4% accuracy. R-PDR thus addresses the two main limitations facing existing PDR techniques.

The organization of this chapter is as follows. Sec. 4.1 proposes the system architecture of the R-PDR system. Sec. 4.2 presents three novel capabilities of R-PDR which make it accurate and robust even in the context of long term tracking. Sec. 4.3 explains how R-PDR steps exploit these novel capabilities. Sec. 4.4 evaluates the performance of the proposed R-PDR system and its impact on indoor positioning with

extensive real world experiments. Sec. 7.4 summarizes this chapter with discussions on the applicability of the proposed algorithm.

4.1 System Architecture

The main motivation of our work is to build an indoor tracking system that is robust to noise introduced by IMU sensors and to context variability. Note that PDR algorithms typically include tasks for motion mode recognition, step detection, step length estimation and step heading estimation. In accurately accomplishing these tasks, we have addressed the following key challenges:

Motion classification: Despite a plethora of work on recognising different motion modes, to our knowledge, there is still a lack of a comprehensive classification scheme that captures the wide range of human motion and device attachment. Moreover, solutions that capture many motion modes are typically complex, since they have to include a large number of motion-specific optimisation rules, especially for solving the problem of step detection.

Device orientation estimation: The performance of the PDR algorithm largely depends on the ability to accurately track the orientation of the mobile device. This task is particularly challenging in the context of unconstrained (mobile/wearable) devices embedded with low-cost IMU sensors. Generally, the gyro drift and magnetic disturbances, especially in indoor environments, are the major obstacles to accurate orientation estimation. Addressing this problem is essential, because orientation information is used by nearly all tasks of the PDR algorithm (Sec. 4.2).

Acceleration long/short term drift: Another major challenge is the correction of short-term and long-term drift in the acceleration signal. The noise that distorts the acceleration signal is a major issue because it creeps into all four tasks of the PDR algorithm, as will be explained in Sec. 4.3.

Fig. 4.1 shows the proposed system architecture that was designed to address the challenges discussed above. The sensor data from IMU sensors, including acceleration, magnetic field, and the angular velocity, are fed into the R-PDR component. The proposed R-PDR is equipped with three novel capabilities: Firstly, it features a novel motion classification scheme, which only includes two classes, but captures a wide spectrum of human motion and device placement combinations. Secondly, R-PDR features a novel mechanism for estimating device orientation which is robust to device rotations. Thirdly, leveraging simple observations about human motion,

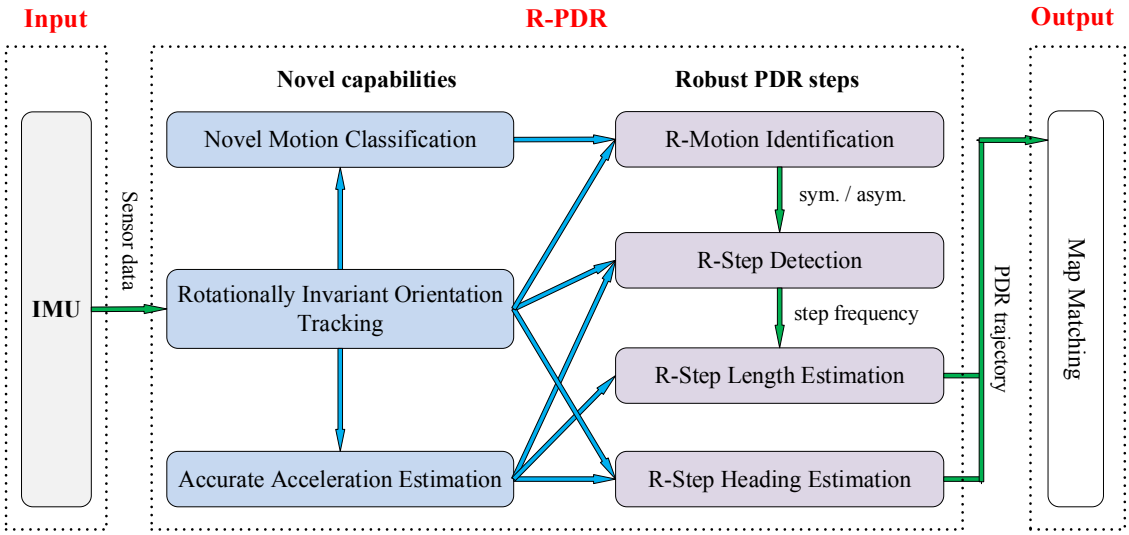


Figure 4.1: System architecture of R-PDR.

R-PDR provides a robust mechanism for correcting short-term and long-term acceleration bias. Sec. 4.2 first describes in detail the three new capabilities of R-PDR, while Sec. 4.3 explains how they become instrumental to reinforcing its four steps.

4.2 Novel Capabilities

We are now in a position to drill down to the three novel capabilities of R-PDR, starting with a novel scheme devised to address the current challenges in classifying human motion.

4.2.1 Novel Motion Classification

Existing work has typically built more and more elaborate models to recognize different classes of motion [72, 74], including phoning, texting, hand swinging, static, etc. [69]. However, these motion modes are only a subset of all possible human motion modes, and suffice in controlled experiment settings. To make the motion mode recognition work in unconstrained real world settings, a large number of new motion modes has to be defined, leading to a steep rise if we aim to count all possible motion modes of human beings.

We take a fresh stance, based on fundamentals of bipedal motion and resultant sensor coupling for arbitrarily placed devices. Walking motion consists of two phases, the stance phase (when the foot is in contact with the ground) and the swing phase (when the foot is moving forward in an arc), which each leg undertakes in a rhythmic,

offset pattern. When walking, the arms also typically swing back and forth. The motion of the legs and arms couples through the body, and depending on where the sensor is placed, it will either pick up equal and symmetric forces from both sides of the body, or asymmetric forces from only one half of the body, with the other side of the body weakly coupling to the sensor. An example of symmetric motion would be making a call whilst walking, as the smartphone would be held relatively rigidly against the head and thus would be impacted by the movements of both legs equally. Conversely, holding the device in one's hand and swinging the arms would lead to strongly asymmetric waveforms. We argue that by exploiting these basic observations, the task of classification devolves to a simple binary decision, rather than the need for a classifier with multiple categories.

Therefore, instead of separating motion modes according to their differences, we take the approach of grouping them together as much as possible using the features they have in common and define a very simple classification scheme consisting of only two modes: symmetric and asymmetric. Informally, the symmetric mode is the class of motions that are impacted in a similar manner by the movement of the left and right leg; the asymmetric mode includes all other types of motion.

Examples of typical symmetric motion modes include:

- Texting: The mobile device is held in front of the user while walking;
- Phoning: The mobile device is held close to the head while walking;
- Heavy handbag: The mobile device is put in a heavy bag (so the hand of the pedestrian is not swinging) while walking;
- Shirt pocket: The mobile device is put in a shirt pocket while walking;
- Static: The mobile device is not moving, regardless of placement.

Similarly, some typical asymmetric motion modes include:

- Hand Swinging: The mobile device is held in a swinging hand while walking;
- Trouser pocket: The mobile device is put in a trouser pocket (front or back) while walking;
- Belt mounted: The mobile device is fastened to the belt while walking;
- Handshaking: The mobile device is (possibly periodically) shaken or tapped while not moving.

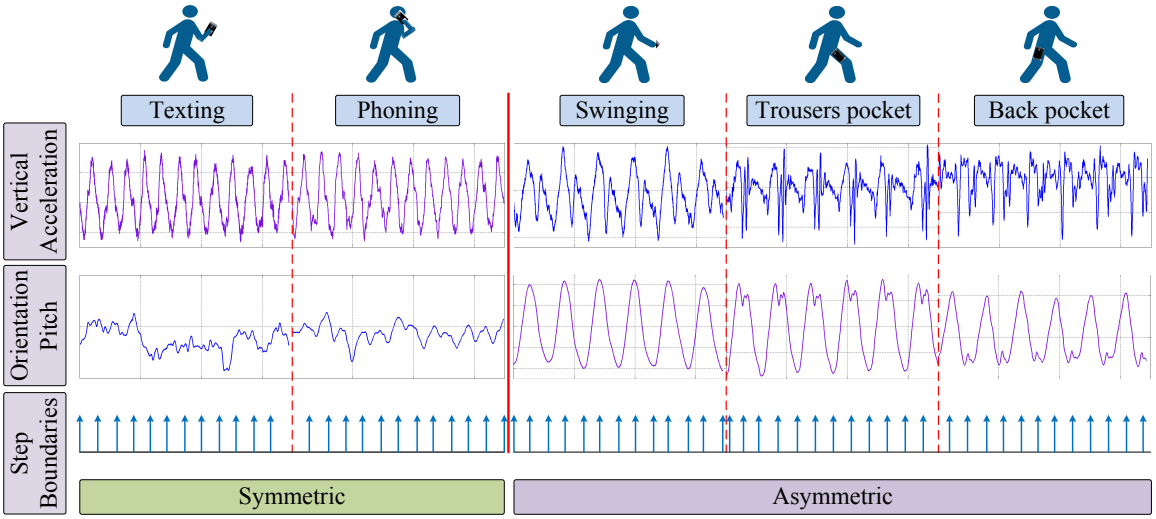


Figure 4.2: Acceleration and orientation signals of some typical motions. Note how symmetric motions have strong periodicity in acceleration, whereas asymmetric motion has strong periodicity in pitch.

Fig. 4.2 illustrates a few examples of symmetric and asymmetric motion modes. Notice that all symmetric motion modes have similar periodic acceleration signals, with a period corresponding to a single step, while all asymmetric motion modes have similar periodic orientation patterns, with a period corresponding to two steps. The benefits of classifying motions into symmetric and asymmetric are three fold. First, it is a comprehensive scheme that leaves no motion out. Second, the motion recognition step itself is a very easy task, which requires simple cross correlation with sine waves (Sec. 4.3). Lastly, the fact that symmetric motions have periodic acceleration signals while asymmetric motion modes has periodic orientation signals greatly facilitates the task of step detection also discussed in Sec. 4.3.

4.2.2 Rotationally Invariant Orientation Estimation

Being able to accurately track the device's orientation is the foundation for accurate long term inertial tracking. This is because most PDR subtasks, such as step detection and user heading estimation, are based on knowing the orientation of the device. However, determining device orientation remains a major issue for inertial tracking with unconstrained devices, especially with low-cost IMU sensors ubiquitous in mobile devices and wearable sensors. Generally, the gyro drift (and to a lesser degree, magnetic distortions from the environment) is the major obstacle for accurate orientation tracking.

Most existing work uses Kalman filters, e.g. indirect Kalman filters [82], extended Kalman filters (EKF) [83] and Unscented Kalman filters (UKF) [84], to track device orientation with angular velocities from gyroscope to periodically update the orientation. In this study, UKF is employed to track device orientation due to its optimal performance in nonlinear system. The detail of the UKF used is listed in Appendix A.

The gyro drift in horizontal plane (yaw) can be compensated with measurements of the Earth's magnetic field especially if we have a map of the area's magnetic distortions. However, the gyro drifts in roll and pitch angles are still a pressing problem for devices with motion acceleration [86]. We argue that this problem can be addressed by accurately estimating the gravitational acceleration from the raw accelerometer signal, and using the gravity vector as an observation in the Kalman filter to correct drift in both roll and pitch.

Currently there are two dominant approaches to estimating the gravity component of accelerometer measurements. The first approach, proposed by Mizell et al. [204], calculates the mean over a fixed length window to remove dynamic acceleration and estimate gravity as the static component. This approach is elegant and simple but is not suitable for mobile devices used by pedestrians. The major reason is that a change in orientation introduces a considerable lag in accurate gravity estimation. To reduce the lag, more recent work [205] reduces the length of the window, with a consequent decrease in accuracy. The second approach, recently proposed by Hemminki et. al [206], places additional constraints over allowable sampling instants for gravity estimation. Specifically, this approach estimates the direction of the gravitational vector only when the variation of the acceleration signals over a window is below a certain threshold. This reduces the lag introduced by the first approach, but also suffers from two fundamental limitations. First, this approach does not always lead to enough samples for estimating the gravitational vector, as it only works when the mobile device is static for a period of time. It typically does not work if the user is walking, when accurate estimates of the gravity vector are mostly needed. Secondly, the thresholds of variation in different motion modes are different, making it impractical in real-world settings, as significant tuning is required. A similar approach [207] opportunistically estimates the gravity for the devices attitude estimation when the angular velocity of the device is detected to be less than a threshold.

To address these limitations, we propose a novel algorithm that can accurately estimate the gravity vector without lag and operates continually. The key property that it exploits is that gyro sensors have low drift and high reliability over a short time window, and thus can provide accurate measurements of relative changes in

orientation. As a result, the relative change in orientation of the device within a short time window can be accurately estimated. Given accurate measurements of orientation, the acceleration signals within this window can be rotated to the same coordinate system, e.g. the coordinate system of the device at the last time stamp of the window. After the rotation, the acceleration signals along each axis (X, Y, and Z) can be averaged because they are now in exactly the same coordinate system. Since typical walking behavior consists of periodic linear acceleration and deceleration, they will cancel each other out in the averaging process. As a result, the remaining linear acceleration is tiny compared with the gravitational acceleration, which makes the average of the acceleration an accurate estimate of the gravity vector.

Assume we want to estimate the gravity vector at time t given a window of historical accelerometer data $\boldsymbol{a}_{t-T:t}$ of size T. We first rotate acceleration signals $\boldsymbol{a}_{t-T:t-1}$ to the same orientation as $\boldsymbol{a}_t$, and then estimate the gravity at time t as the mean of the acceleration readings after the rotation. In this way, we address the limitation of the first approach in needing a long window for accurate measurements of device orientation and avoids the constraint of the second in needing the device to be stationary for a length of time. Specifically, the gravity at time t is estimated as

$$g_t = \boldsymbol{R}_t \sum_{\tau=t-T}^{t} \frac{\boldsymbol{R}_\tau^T \boldsymbol{a}_\tau}{T}, \tag{4.1}$$

where $\boldsymbol{a}_\tau$ is the acceleration vector in the coordinate frame of the device, and R_τ is the rotation matrix describing the rotation from the earth coordinate system to the coordinate system of the device at time τ, as obtained from the Kalman Filter. The gravity value g_t is then used to compensate the Kalman Filter, by supplying an accurate observation of the gravity, thus removing the effects of long term gyro drift.The gravity estimation accuracy is robust over window length as long as the window length is bigger than the duration of multiple strides, reason being that the majority of acceleration and deceleration within period of a stride will cancel each other out because the velocities at the beginning and end of each stride are so close that they can be regarded as almost the same, as will be explained in Sec. 4.2.3. Experiments with window lengths of 2–8 seconds all lead to good results.

Experiments have been conducted to compare the proposed algorithm with the state-of-the-art gravity estimation techniques. In these experiments the pedestrian walks normally with the mobile device carried in their hand. The gravity vector is estimated using the proposed and competing approaches and then rotated to the earth's coordinate system according to the current orientation of the device. The

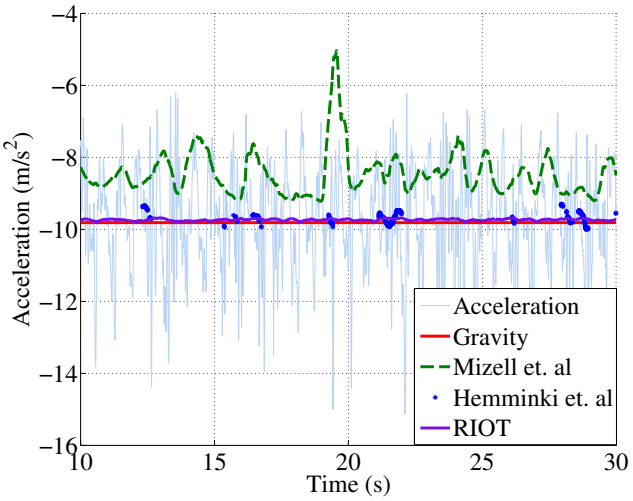


Figure 4.3: The acceleration due to gravity estimated along the Z axis using different techniques. It is observed that Mizell's approach is not robust to orientation change of the device and thus very inaccurate. Hemminki's approach can only generate sparse gravity estimation samples. In comparison, the proposed RIOT algorithm can continuously provide very accurate gravity estimation results.

gravity value, as shown in Fig. 4.3, is then estimated as the acceleration in the Z axis (vertical axis). If the orientation algorithm is working correctly, the gravity vector should only have a component in the vertical axis, and this value should be equal to the true value of gravity. It is observed that Mizell's approach [204] has the largest errors due to its inability to manage changes in orientation of the device. Meanwhile, Hemminki's approach [206] provides very sparse and sometimes inaccurate gravity estimates, because in most cases the motion of the pedestrian makes the variation in acceleration bigger than the threshold. In comparison, the proposed approach offers accurate gravity estimates and works continually. Both the proposed approach and Mizell's approach use a window length of 4 s.

The gravity vector estimates are then fed into the Kalman filter as additional observations (along with magnetometer measurements) to estimate the orientation of the device. We can indirectly see the benefits of the proposed orientation tracking approach as follows. We first rotate every acceleration vector to the earth coordinate system according to the device orientation estimate at that moment. Then the gravity magnitude is simply estimated as the mean of the acceleration in Z axis in a short window. Fig. 4.4 shows the long term estimate of gravity with and without the proposed orientation tracking algorithm in the Z axis. Again, we would expect a good technique to maintain the estimate of the gravity component at a constant -9.81m/s^2, regardless of the orientation of the device or dynamic acceleration changes

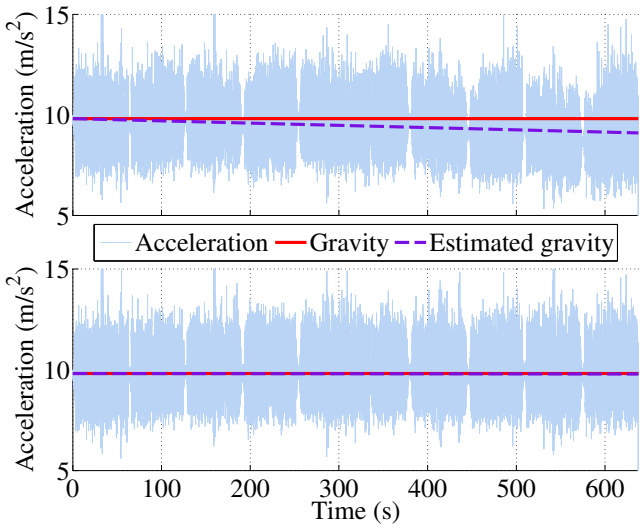


Figure 4.4: Vertical acceleration without (top) and with robust gravity estimation (bottom). It is observed that without the proposed gravity estimation approach the estimated gravity suffers from accumulated drift. In comparison, the proposed gravity estimation approach can continuously estimate very accurate gravity measurements without long term drift.

due to motion. As expected, the drift of device orientation grows quickly without the proposed approach, leading to the gravity value drifting away from the actual. This drift can result in inaccuracy in counting number of steps, as will be explained in Sec. 4.3. However, the drift has been successfully corrected by the proposed orientation tracking approach, which has always kept the estimated gravity value very close to its true value.

Compared with using the raw acceleration vector directly into the Kalman filter, the advantages of applying the averaged gravity vector are apparent. Firstly, device orientation estimated by directly using the raw acceleration vector is significantly biased by the linear acceleration. It might yield correct orientation at rare occasional time stamps when the historic linear acceleration and deceleration exactly cancel each other out but this does not frequently occur. This makes the estimated orientation much less useful for step detection. Furthermore, directly feeding the acceleration to Kalman filter requires every sample of acceleration to be taken into account in the filtering process, which increases the computational complexity significantly and meanwhile reduces the orientation tracking accuracy.

4.2.3 Accurate Acceleration Estimation

Although RIOT can achieve long term orientation tracking of the device, as described in Sec. 4.2.2, it is not sufficient to accurately estimate acceleration on the horizontal plane, which is important for step detection. The major reason is that a very small tilt of the device which cannot be corrected by the robust orientation tracking with the noisy measurements from low-cost IMU sensors, can significantly impact the acceleration signal on the horizontal plane. For instance, a tilt error of only two degrees changes the acceleration reading in the vertical axis from 9.806 m/s^2 to 9.801 m/s^2, but it can offset the measured acceleration on the horizontal plane (either X or Y) by 0.342 m/s^2. To determine whether a step is valid or not (described in detail below in Sec. 4.3.2), the raw acceleration measurements are integrated twice to calculate the displacement of the step. Any orientation offset is significantly amplified through double integration which results in significant errors in displacement estimation. The example two degree tilt error described above can result in a cumulative displacement error of up to 20 metres within 10 seconds.

To address this problem, we have further explored the physical constraints imposed by human walking. It is demonstrated in [70] that humans have surprisingly consistent walking patterns for consecutive steps. Inspired by this phenomenon, we present an algorithm to estimate and correct the orientation error using the repetitiveness in walking patterns, called repetitive pattern update (REPUT). Specifically, we assume that the extrinsic states of the mobile device (velocity, height, etc.) are largely the same after each step (or every stride in the case of asymmetric motion). If we assume that the bias in the orientation drifts slowly and remains constant over the duration of the step[1], our task is to find the inverse rotation matrix $\boldsymbol{R}$ that will compensate the offset, subject to the imposed constraints.

Consider the case of maintaining the velocity of the device as a state that remains constant over a step. Suppose one step starts at time t_0 with velocity $\boldsymbol{v}_0$ and ends at time t_e with velocity $\boldsymbol{v}_e$. If there is no orientation bias, the difference between the initial and terminal velocities should be zero i.e. $\Delta\boldsymbol{v} = \boldsymbol{v}_e - \boldsymbol{v}_0 = 0$.

We can formulate the relation between the initial and terminal velocities as

$$\boldsymbol{v}_e = \boldsymbol{v}_0 + \sum_{t=t_0}^{t_e} \boldsymbol{R} \cdot \boldsymbol{a}_t \cdot \Delta t, \tag{4.2}$$

[1]The start and end of every step or two can be easily detected with an enhanced zero crossing detector as discussed in Sec. 4.3

in which $\boldsymbol{R}$ is the inverse rotation matrix that corrects the orientation drift, $\boldsymbol{a}_t = [a_t^x, a_t^y, a_t^z]$ is the acceleration vector at time t in the earth's coordinate frame, and Δt is the time interval between two consecutive acceleration readings. Then based on our REPUT assumption, we have

$$\Delta \boldsymbol{v} = \boldsymbol{v}_e - \boldsymbol{v}_0 = R \sum_{t=t_0}^{t_e} \cdot \boldsymbol{a}_t \cdot \Delta t = \mathbf{0}, \tag{4.3}$$

which allows us to solve for the unknown rotation matrix.

Other repetitive patterns which impose constraints on orientation drift can also be derived. For instance, if it is known that the device is only used in 2D positioning, then the mobile device has the same height above the ground at the beginning and end of a step. Obviously in 3D tracking this cannot be used without knowledge of the map as a user could be climbing stairs for example, leading to a height difference. This can be formulated as

$$\Delta l = R \sum_{t=t_0}^{t_e} \left(\sum_{\tau=t_0}^{t} a_t^z \Delta \tau \right) \Delta t = 0. \tag{4.4}$$

where a_t^z is the vertical acceleration along the Z axis.

It is possible that these conditions could contradict each other because either the velocity difference $\Delta \boldsymbol{v}$ or distance difference Δl cannot be strictly zero. Therefore, least-square solutions of the counter-drift rotation matrix R should be obtained from simple optimization. Then the drift can be corrected by rotating the acceleration signal with R.

These multiple repetitive constraints can be used to find the optimal rotation matrix through least squares optimization. To test the effectiveness of the REPUT algorithm, we have conducted a simple experiment during which the pedestrian walks normally with device held in a hand. Fig. 4.5 shows the acceleration signals before and after applying the REPUT algorithm. We can observe that the acceleration signals along the X and Y axis are significantly biased without the REPUT algorithm, whereas the offset has been corrected with the REPUT algorithm.

4.3 Proposed R-PDR Steps

Now that we have introduced the key novel capabilities of R-PDR, we are in a position to show how they are employed to design robust versions of the four pedestrian dead reckoning steps: namely R-Motion Identification, R-Step Detection, R-Step Length Estimation, and R-Heading Offset Estimation.

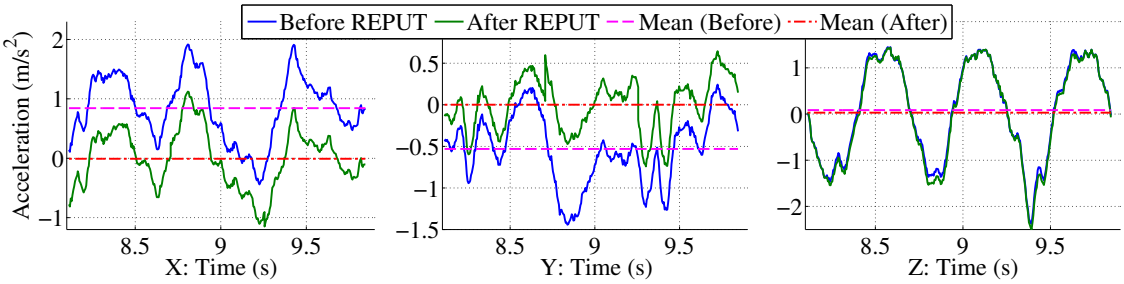


Figure 4.5: Acceleration signals before and after REPUT. According to the walking nature of human beings, the X and Y acceleration signals are expected to have mean values close to zero in a window. However, the mean acceleration drifts significantly away from zero without REPUT.

4.3.1 R-Motion Identification

This step exploits the novel classification scheme, which consists of two motion classes only - symmetric and asymmetric. In addition, it leverages the corrected orientation and acceleration signals in order to identify the motion class.

To determine whether a device placement is leading to symmetric or asymmetric coupling, we examine both the vertical acceleration and the pitch (rotation around the y axis) component of the device orientation. This is the main reason that long term tracking of the orientation of the device is so important and why the RIOT technique introduced in Sec. 4.2.2 is crucial to accurate classification. Based on extensive experimental evidence, we make the novel observation that symmetric motion modes lead to strong periodicity in the vertical acceleration, closely approximating a sinusoidal wave, whereas the pitch waveform is aperiodic. On the other hand, asymmetric motion leads to aperiodicity in vertical acceleration, but strong periodicity in pitch angle. This is shown graphically in Fig. 4.2, illustrating a few examples of symmetric and asymmetric motion modes. Notice that all symmetric motion modes have similar periodic vertical acceleration signals, with a period corresponding to a single step, while all asymmetric motion modes have similar periodic pitch patterns, with a period corresponding to a stride.

Thus, our classifier is based on assessing which one of the vertical acceleration or pitch waveforms has the strongest periodic component. More precisely, we compute the DFT of each signal (pitch and acceleration) over the frequency range that corresponds to human walking, typically between 1.2 Hz and 2.5 Hz. We then use an energy detector to decide whether the device placement is resulting in symmetric (vertical acceleration has stronger periodicity than pitch) or asymmetric (pitch has stronger periodicity than vertical acceleration).

Based on this decision (symmetric/asymmetric), the waveform with stronger periodicity is fed as input to the R-Step Detection component described in detail below.

4.3.2 R-Step Detection

Once the motion mode is identified, the next task of a PDR algorithm is to detect human steps. Typical existing step detection algorithms detect steps by identifying the cyclic patterns in acceleration signal, e.g. peak detection [60–62], zero-crossings [63–65], spectral analysis like Fourier transform [66,67], short time Fourier transform [68], and continuous wavelet transform [69], and auto/cross correlation [71,80]. The aforementioned techniques assume that people generate periodic acceleration signals only when walking but this is not true. Hence, they cannot distinguish between a real step and a simple hand swing without the user really moving forward. This is because the two actions can actually generate very similar acceleration signal patterns. The problem of false positive step detection is worse when applying algorithms that use the magnitude of the acceleration signals [75], and is further exacerbated with less constrained motion modes, like crawling, stumbles, side-steps, and shuffles [59].

To summarise, existing techniques only focus on identifying the boundaries of *step-like* signal segments, referred to as step pattern extraction. In comparison, the proposed R-Step Detection algorithm not only proposes a novel way of performing step pattern extraction, but also introduces an additional classification step for distinguishing whether the extracted step is real or fake. To clarify, a real step is defined as a step with locomotion, as opposed to a fake step without locomotion. An example of a fake step is one detected when a person swings the phone in her hand without actually moving her legs. Details of the novel step pattern extraction and step classification modules are provided below.

Part 1: Step pattern extraction: The proposed approach here aims to identify the boundaries of step-like signal segments in a lightweight online fashion. We leverage our novel motion mode classification scheme to extract steps very accurately with time domain analysis. The approach that we use depends on whether the motion is symmetric or asymmetric. If it is symmetric, we use the acceleration signal for step detection, because it has a simple sine-wave-like pattern in symmetric motion mode. Step boundaries here are detected with an enhanced zero crossing detector similar to [65]. However, as shown in Fig. 4.2, the acceleration patterns of the asymmetric motion modes differ significantly, making it extremely difficult to identify steps with temporal features like peaks or zero crossings. Therefore, if the motion is asymmetric, we identify the orientation signal as the key to step extraction, because the change

of orientation instead of acceleration is highly periodic in asymmetric motion modes, as shown in Fig. 4.2. In this case, step boundaries are the peaks and troughs of the device orientation signal - e.g. first step is from trough to peak, second step is from peak to trough etc. (provided that they are far apart to avoid local fluctuations).

Part 2: Step classification: The step pattern extraction step is not sufficient to distinguish between real and fake steps. We argue that a further classification task is needed, that takes into account informative features of the orientation and acceleration signals. In particular, the horizontal and vertical displacements are identified as informative features in telling whether real locomotion happens to the mobile device. They are derived from the double integration of the acceleration signal during one step (in the x-y plane / z axis resp.). These two features are extremely informative because they can capture the fundamental difference between real and fake steps. In real steps, the horizontal displacement is higher and the vertical displacement is lower than in fake steps. The cross correlation between the acceleration signal and the orientation signal is also identified as an effective feature to reject false positives because the acceleration and orientation signal tend to have a higher correlation while shaking the device than normal walking behaviour. The orientation offset is another feature used for real/false step classification. The orientation offset measures how much the orientation has changed during one step. This feature is proposed to prevent the acceleration generated from making turns from being taken into account in calculating the horizontal and vertical displacement. The above features, including horizontal/vertical displacements, cross correlation value, and orientation offset, are then fed into a simple three nodes decision tree to determine whether the candidate step really corresponds to a true step.

4.3.3 R-Step Length Estimation

Unlike foot-mounted inertial navigation systems [76], double integration of the acceleration to get the step length is not applicable to handheld devices. The zero velocity update (ZUPT) is not available for handheld devices because only the heel can be regarded as (almost) static when one foot is planted on the floor during a walking period.

Various models have been proposed to address the step length estimation problem. The simplest approach is to make it a constant [77] because pedestrians have surprisingly constant step length with a natural walking pace [70]. However, when walking with others, people are likely to adjust their natural walking pace, which leads to a significant change of the step length [59]. The most common approaches used to

estimate the step length are linear [69, 78] or nonlinear [61, 62] models that relate the step length to variables like pedestrian height, step frequency, acceleration variance, etc. These models are easy to implement and thus widely adopted in practice. An alternative of the linear/nonlinear model is regression. Machine learning regression techniques like support vector machine [74, 75] and neural networks [79] have also been applied based on various features extracted from the sensor data.

In this study we use a simple step length model [208], which uses accelerometer data to detect the step frequency (denoted with f_t at time t) which is later used to estimate step length l_t as follows:

$$l_t = h(\alpha f_t + \beta) + \gamma. \tag{4.5}$$

where h is the pedestrian height, α, β, and γ are constants. The accuracy of this model is not satisfactory by itself because step length parameters vary across different users and environments. The proposed R-Step Length Estimation algorithm however benefits from being fed with carefully configured parameter values thanks to a novel feedback loop introduced between R-PDR and Map Matching. This unsupervised lifelong learning approach is describe in detail in Chapter 6.

4.3.4 R-Heading Offset Estimation

To estimate the heading of the pedestrians, most existing work assumes the knowledge of the position where the mobile device is attached, thereby making the assumption that the heading of the pedestrian is always consistent with the heading of the mobile devices [81], which is not true in real world settings. Zee [80] demonstrated that the heading can be estimated by observing the frequency spectrum of a typical walk. Although this is an interesting idea, it suffers from direction ambiguity. WalkCompass [85] samples the acceleration signal during the segment of a step with maximum deceleration. After removing noise and rotating samples to a stable coordinate system, it averages them to obtain the walking direction in 3D. It then projects the estimated direction to the horizontal *walk* plane. This approach works for various device placements, but is sensitive to noise as shown in Sec. 4.4.4.

In this work, instead of averaging like WalkCompass, we fit the straight line that best matches the acceleration readings of a single step in the horizontal plane (X and Y) using simple least squares. We disambiguate between the two headings of the straight line using the first 50% of acceleration samples. If the residual error is small (Fig. 4.6 left and middle) we use the resulting estimate as the offset between device

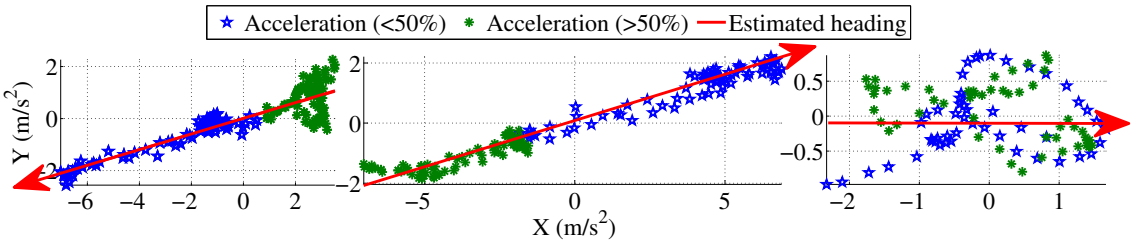


Figure 4.6: Least squares heading offset estimation, showing how the heading of the pedestrians can be estimated from the acceleration signals.

and user heading; otherwise (Fig. 4.6 right), we use the most recent offset that was reliably estimated.

The proposed heading offset estimation approach relies on being able to correctly rotate acceleration data to the earth coordinate system. This requires accurate device orientation estimates, which in turn depends on the quality of gyro data as discussed in Sec. 4.2. Our approach to addressing the issue of long term drift of the gyro sensors, and further improving the accuracy of R-Heading Offset Estimation, is to learn the sensor bias via a novel lifelong learning approach, discussed in Chapter 6.

4.4 Evaluation

In this section, we compare and contrast the performance of the proposed approach with leading PDR approaches for step detection and counting. The competing approaches we compare against are:

- **Peak detection (PD):** The toe-off and heel-strike events are usually associated with sharp changes in vertical or the magnitude of acceleration which are the targets for typical peak detection algorithms [60–62]. However, each heel-strike could result in multiple peaks, especially when the mobile device is in a swinging hand or attached in a low position of the body, e.g. in a trousers pocket. This significantly increases the algorithm complexity. In our implementation of the peak detection approach, we use a low pass filter followed by a peak detector to remove the multiple strikes.

- **Zero-crossing (ZC):** Another way of identifying the cyclic patterns in acceleration is the detection of zero crossings [63–65]. This algorithm is simple to implement, but can suffer from false positives due to zeros crossings from events other than walking.

- **Spectral analysis (STFT and CWT):** This promising technique has attracted a large amount of attention recently due to its robustness [66–69]. Algorithms of this category first convert the acceleration signal to the frequency domain using different algorithms like Fourier transform [66, 67], short time Fourier transform (STFT) [68], or continuous wavelet transform (CWT) [69], and then the dominant peak of the signal in frequency domain is identified as the step frequency. Windowed acceleration signal (magnitude) is common choice for techniques working with spectral analysis [67]. As a result, online step detection makes this algorithm very computationally intensive.

- **Auto/cross correlation (AC):** Since each individual human has a surprisingly consistent walking pace [70], the strong periodicity of the acceleration signal from such walking behavior makes it possible to extract the cyclic pattern with mean-adjusted autocorrelation of the sensor data sequence or cross correlation with prepared templates – such as a sample sequence of sensor data from the training phase [71]. A typical approach in this category is Dynamic Time Warping (DTW) [209]. Zee, a recent approach that we use as comparison, uses normalized auto-correlation (NASC), and can also exploit characteristic waveforms from different device placements [80].

4.4.1 Experimental Setup

For all tests, the ground truth of steps was obtained using foot mounted IMU's (X-IO Technologies Inc.) which are accurately calibrated. Two IMUs were used, one on each foot, to accurately count the number of steps. Ground truth of position was obtained by placing numbered labels along the route at 1-m evenly spaced intervals and using a down facing hand-held camera to note the exact time when the foot crossed the midpoint of the label. For each test, the user was outfitted with 2 IMUs (one on each foot) and 6 Nexus-S smartphones capturing IMU data at various positions of the body. This allows us to perform direct comparisons of accuracy, as an accurate PDR system should report the same number of steps, regardless of device placement.

4.4.2 Overall Detection Accuracy

In this test, we examine the accuracy of overall step detection for various placements for each of the competing algorithms, compared to the ground truth. Data was obtained from an experiment where a mix of daily behaviours such as walking, standing, sitting, sending a text, making a call and typing were performed. This is in contrast

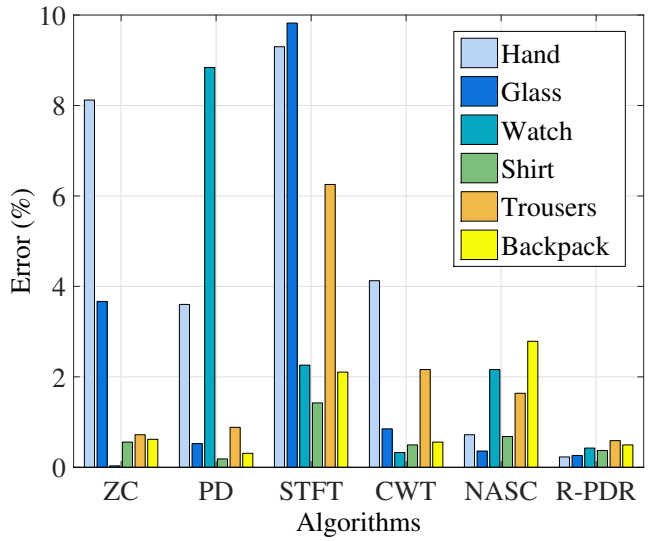


Figure 4.7: Comparison between state-of-the-art step detection algorithms and the proposed approach, R-PDR, in terms of step detection accuracy for various device attachments.

to other work where a limited subset of behaviours (e.g. walking and standing still) are used to demonstrate step detection accuracy. The experiment lasted two hours and a total of 4669 true steps were counted by the two foot mounted IMUs. Fig. 4.7 shows the results for different algorithms and different placements. The detection error rate is defined as $\frac{\text{missed steps + false steps}}{\text{total number of steps}}$. It is interesting to note that for the competing algorithms, they typically attain good performance for some placements, but for one or two placements the accuracy is particularly poor, in some cases as high as 10%. In particular, the handheld placement typically registers the highest errors as repetitive motion such as typing has strong periodicity like a true step. Out of the competing algorithms, the normalized autocorrelation (NASC) algorithm performs reasonably well across the placements, with a maximum error of 3%. The strength of our approach, R-PDR, is apparent from the results, as regardless of placement, errors are consistently below 1%. This experiment using realistic human behaviour demonstrates how existing techniques suffer from sensitivity to device placement, whereas our approach is able to correctly detect the correct number of steps.

4.4.3 False Positive Rejection

In this test, using data from a number of behaviours that are repetitive but do not correspond to forward displacement, we demonstrate the performance of existing algorithms in determining whether a step has actually been taken or not. This is

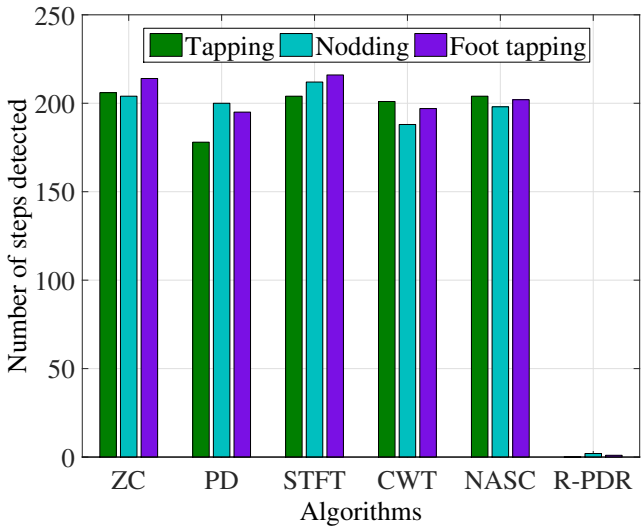


Figure 4.8: Comparison between state-of-the-art step detection algorithms and the proposed approach, R-PDR, in terms of step detection accuracy for various device positions.

shown in Fig. 4.8. Each behaviour, such as tapping, was performed 200 times.

Existing algorithms which are based solely on characteristics of the waveform itself, such as periodicity, are unable to distinguish between behaviours like tapping and true steps caused by walking. This is because they are only designed to detect periodical patterns and do not consider whether a motion, as registered by the sensors, actually corresponds to true forward motion. This is made even more challenging for unconstrained device placements. The suite of techniques presented in R-PDR allows us to evaluate, for each step, the displacement of the user and assess whether or not a step has really been taken. This ability to distinguish between true and false steps enables our algorithm to outperform existing algorithms. It is observed from Fig. 4.8 that the proposed algorithm reports close to 0 real steps while all other algorithms report approximately 200 steps, which demonstrates the effectiveness of the proposed algorithm in rejecting false positive steps.

Experiments have been conducted to evaluate the performance of both accurate step detection and false positive rejection. Fig. 4.9 compares the step detection accuracy of R-PDR against autocorrelation which is reported to have the best overall detection accuracy in existing techniques. It is observed that R-PDR always outperforms autocorrelation for various human motions and placements of devices. The major reason lies in the fact that R-PDR could reject false positives while NASC cannot. Motions like nodding, tapping, typing, and foot tapping can generate step-like signals without really moving forward. As a result, NASC significantly over counts

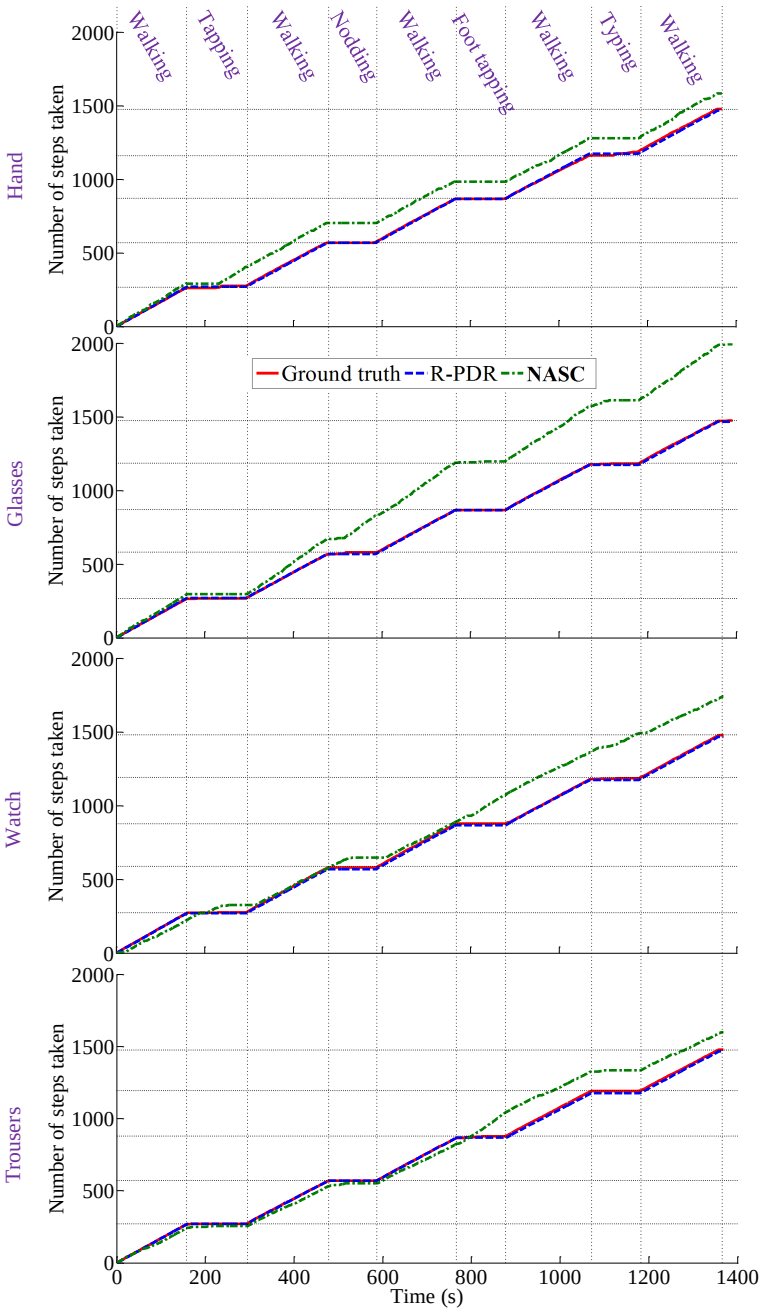


Figure 4.9: Typical human motions and the corresponding step detection accuracy for R-PDR and NASC.

the number of steps, which, inevitably leads to significant errors in tracking.

4.4.4 Heading Estimation

Recall that most existing work assumes the knowledge of the position where the mobile device is attached, thereby making the assumption that the heading of the

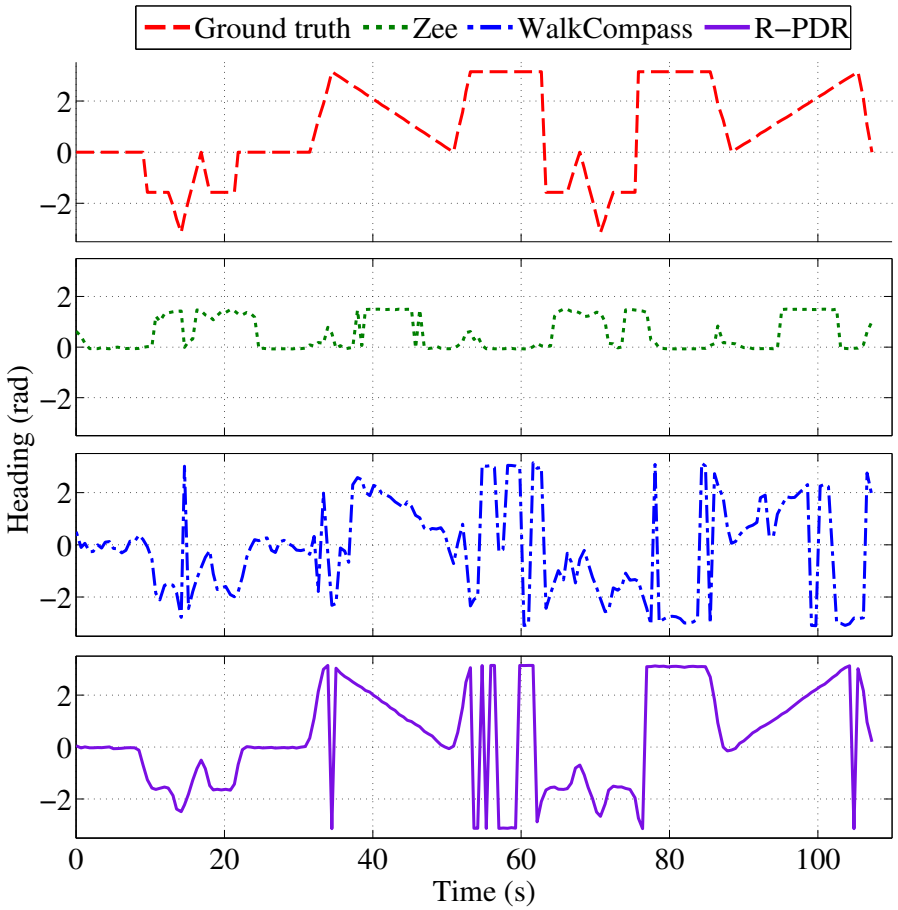


Figure 4.10: Comparison of heading estimation techniques. Zee cannot distinguish between two directions 180° apart so the estimated heading ranges from 0 to π. WalkCompass can estimate the heading without ambiguity. However, the headings estimated from WalkCompass are not as accurate as from R-PDR.

pedestrian is always consistent with the heading of the mobile devices [81], which is not true in real world settings. To address this limitation, Zee [80] estimates the heading from the typical frequency spectrum of walking behavior while WalkCompass gets the heading of the user from the average acceleration after filtering.

Fig. 4.10 compares the estimated heading with R-PDR and competing techniques for tracking a user in a sports centre for 110 seconds. Since Zee can only provide two direction estimates that are 180° apart, it is not able to solve the direction ambiguity, which explains why we can only observe headings between 0° to 180°. WalkCompass can do a good job when the device is attached tightly to the lower part of the body and the user walks straight. But the performance of WalkCompass is degraded when the device is mounted in the upper part of the body or the user takes a turn or slightly shakes the device. In comparison, the proposed R-PDR approach does not

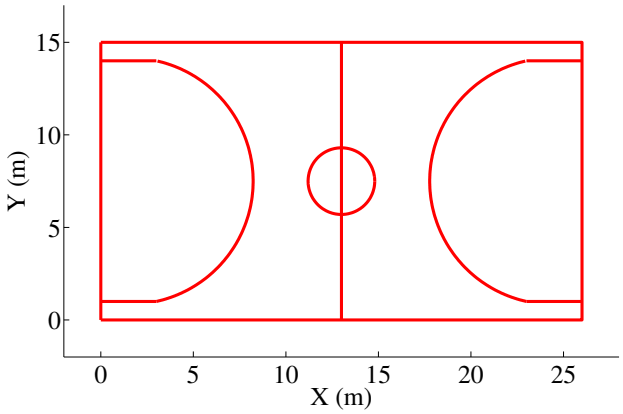
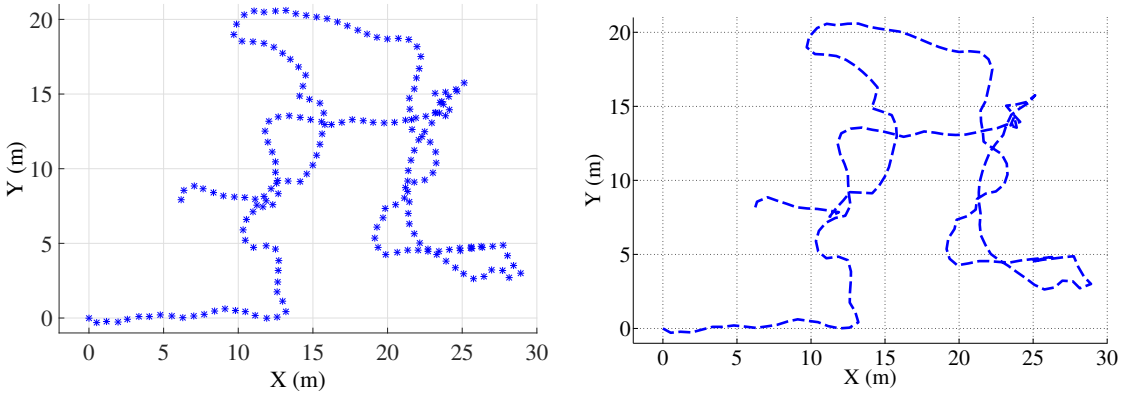


Figure 4.11: Ground truth of test trajectory in sports center.



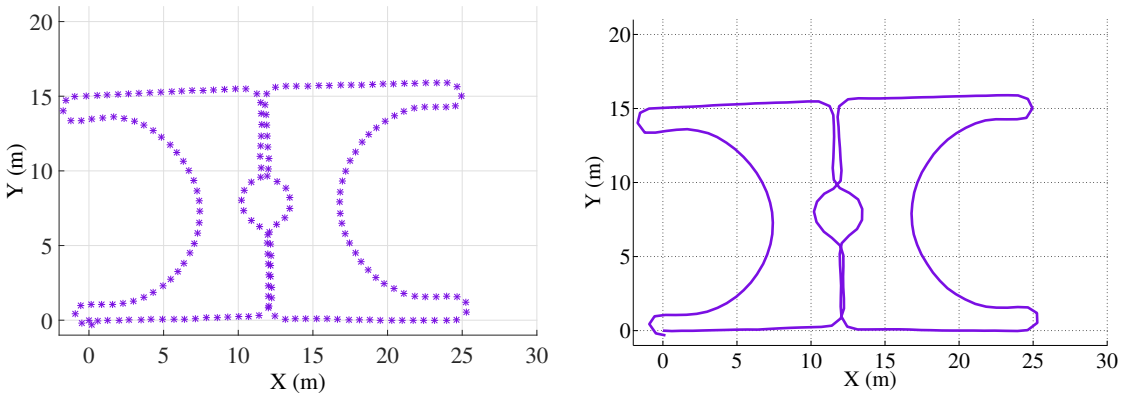
(a) Discrete estimated locations

(b) Continuous trajectory

Figure 4.12: Reconstructed trajectory by WalkCompass.

suffer from noisy heading estimates, thus yielding higher accuracy than the competing approaches.

Experiments have been conducted in the sports centre to compare the estimated trajectories with R-PDR and competing approaches for the tracking. The ground truth of the test trajectory is shown in Fig. 4.11. Since it is an open space and no map constraints are available, Zee cannot address the issue of heading ambiguities in two opposite directions and thus no trajectories can be generated. The reconstructed trajectory by WalkCompass is shown in Fig. 4.12. Since the headings estimated by WalkCompass suffer from noisy acceleration signals, the tracking accuracy is significantly degraded. In contrast, the proposed R-PDR reconstructs a trajectory that is very close to the ground truth.



(a) Discrete estimated locations

(b) Continuous trajectory

Figure 4.13: Reconstructed trajectory by R-PDR.

4.5 Discussion

Pedestrian dead reckoning is a technique that can act as the foundation of an accurate indoor positioning system by exploiting low cost, infrastructure free device sensors. Work to date has made great strides in demonstrating the potential of PDR, but subject to constraints on device placement. These constraints are limiting the widespread adoption of PDR. Rather than seeking a placement specific approach, or attempting to derive the placement of the device, we have taken an alternative approach in this work and presented a suite of techniques that together provide *robust* PDR (R-PDR). Through extensive experiments using R-PDR, we have demonstrated that step detection and heading estimation are accurately performed even when the position of the device is changed dynamically through the test. In addition, we also showed how R-PDR is able to reject common behaviours like typing that appear like steps to competing PDR techniques.

We believe that the techniques presented here will have wide applicability not only in consumer devices like smartphones but also in wearable devices like smartwatches and glasses. In summary, R-PDR is a comprehensive technique that will enable the adoption of accurate, reliable and robust indoor positioning, without requiring any calibration or tuning.

Given that IMU and WiFi can both provide useful information on indoor positioning, we can expect that an efficient fusion of these two, potentially with the floor plan of the site and other possible sensor modalities like barometer, will provide much higher positioning accuracy than each of them individually. Given the resource limitation of mobile devices in terms of both computation and energy consumption,

an lightweight and flexible fusion approach is needed to achieve this goal. The next chapter proposes an extremely lightweight and flexible framework that can fuse various location-related information from IMU, WiFi, floor plan, and information from other sensors available using conditional random fields.

Chapter 5

Lightweight Map Matching

The ultimate objective of an indoor tracking system is to provide continuous, reliable and accurate positioning on smartphone class devices. We identify maps as the key to providing accurate indoor location; this is a reasonable assumption given that indoor mapping is a high priority for companies, such as Google, Microsoft, Apple, Qualcomm, and so on. A map can be viewed in the broadest sense as a spatial graph which provides constraints. At the simplest level this takes the form of a floor plan of a building. This constrains the allowable motion of a user - people cannot walk through walls and can only enter a room through a door. Other maps (meta-maps essentially) provide additional constraints or features, such as the positions of access points, radio fingerprints, signal strength peaks or distorted geomagnetic fields. Based on a time-series of observations, such as inertial trajectories or RF scans, the goal is to reconcile the observations with the constraints provided by the maps in order to estimate the most feasible trajectory of the user, i.e. the sequence that violates the fewest constraints. Although indoor maps are becoming increasingly available, there is no practical and reliable indoor map matching solution available at present.

This chapter presents MapCraft, a novel, robust and responsive technique that is extremely computationally efficient (running in under 10 ms on an Android smartphone), does not require training in different sites, and tracks well even when presented with very noisy sensor data. Key to our approach is expressing the tracking problem as a conditional random field (CRF), a technique which has had great success in areas such as natural language processing, but has yet to be considered for indoor tracking. Unlike directed graphical models like Hidden Markov Models, CRFs capture arbitrary constraints that express how well observations support state transitions, given map constraints. Extensive experiments in multiple sites show how MapCraft outperforms state-of-the art approaches, demonstrating excellent tracking error and accurate reconstruction of tortuous trajectories with zero training effort.

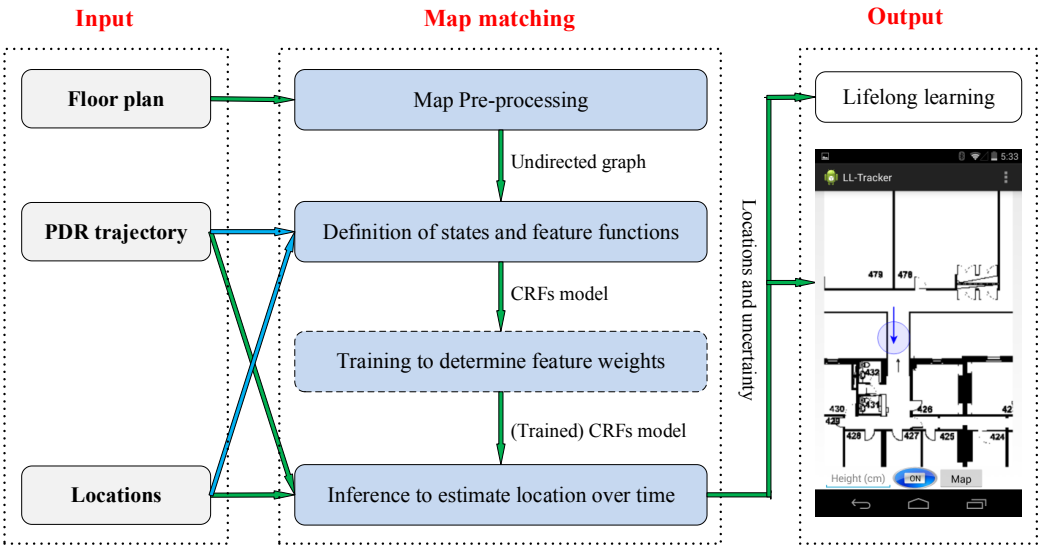


Figure 5.1: System architecture.

As proof of its robustness, we also demonstrate how it is able to accurately track the position of a user from accelerometer and magnetometer measurements only (i.e. gyro- and WiFi-free). We believe that such an energy-efficient approach will enable always-on background localization, enabling a new era of location-aware applications to be developed.

This chapter is organized as follows. Sec. 5.1 outlines the system architecture. Sec. 5.2 introduces the CRF model and Sec. 5.3 proposes MapCraft, a map matching solution that uses CRFs for indoor tracking. Sec. 5.4 extensively evaluates MapCraft in three indoor settings, and compares it with competing techniques. Sec. 5.5 summarizes this chapter and discusses ideas for future work.

5.1 System Architecture

The system architecture is shown graphically in Fig. 5.1, and is described through the use of an example. When a user enters a building and launches the tracking application, the application requests a floor plan (along with other meta-data as generated by other systems, which could include fingerprint maps) from the server, if not already within the cache. Note that this is the only time that a user needs to reveal any data about their coarse position to a third party. The floor plan provides constraints over the set of possible positions a user can take, as well as allowed transitions between locations (i.e. a user cannot directly travel from one end of the building to the other without visiting intermediate locations). As such, the floor plan forms

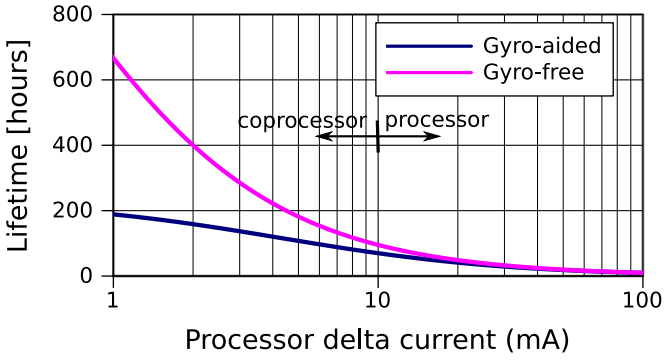


Figure 5.2: The power consumption gap between gyro-free and gyro-aided systems will increase with the emergence of digital motion processors (assuming 1000mAh battery, gyro-free current of 0.5mA and gyro-aided 4.3mA).

a sparse graph and can thus be efficiently stored in memory. Sensors on the user's phone collect data about the motion and (radio) environment. Motion sensors can include accelerometers, magnetometers and gyroscopes. Radio sensors can include WiFi, Bluetooth (low energy), FM radio and so forth. Other sensors can also be used such as acoustic or vision. Raw sensor data is typically not immediately usable and needs to be processed. In the case of motion data, this could include dead reckoning trajectories based on counting steps and estimating heading, or using full IMU tracking in the case of foot mounted sensors. For RF data, a channel/propagation model can be used to relate received signal strengths to physical distances. Alternatively, raw signal strengths may be directly forwarded to the CRF model, to be later combined with RF fingerprint map data if available. These data sources form a set of observations which indicate the likely position of a user.

Maps and observations are combined using conditional random fields, an undirected graphical model described in Sec. 5.2. The CRF model is particularly well suited to this sequential problem because it allows us to flexibly define *feature functions* that capture the extent to which observations support states and state transitions, given map constraints. As a user moves through the building, certain paths become unlikely, as they violate map constraints. The Viterbi algorithm is used to efficiently find the most likely sequence of states through the transition graph, culminating in an estimate of the user's location and quality thereof.

5.2 Conditional Random Fields

CRFs are undirected probabilistic graphical models introduced by Lafferty et al. [210]. They have been successfully applied to a number of tasks in computer vision (e.g. classifying regions of an image), bioinformatics (e.g. segmenting genes in a strand of DNA), and natural language processing (e.g. extracting syntax from natural-language text) [211, 212]. In all of these applications, the input is a vector of observations $Z = \{Z_0, \ldots, Z_T\}$, and the task is to predict a vector of latent variables $S = \{S_0, \ldots, S_T\}$ given input Z. In this work, we model sensor observations as input variables and locations as output variables, and show how CRFs can be used in the context of indoor localisation.

5.2.1 Maximum Entropy Model

In order to introduce CRFs, we must first introduce the *Maximum Entropy Model (MEM)*. A chain CRF is an extension of MEM for state sequences, in the same way that a HMM extends a naive Bayes model, as illustrated in Fig. 5.3. The Maximum Entropy Model assumes that given incomplete knowledge of the probability distribution $p(S_0|Z_0)$, the only unbiased estimate is a distribution that is as uniform as possible given training data (consisting of several (Z_0, S_0) values). This implies finding the model that has the largest possible conditional entropy:

$$p^*(S_0|Z_0) = \underset{p(S_0|Z_0)\in P}{\operatorname{argmax}} H(S_0|Z_0) \tag{5.1}$$

where P is the set of all models consistent with the training material. To explain the meaning of consistency, let's consider a set of m features $f_1, \ldots, f_m$, each one of which is a function of observation and state variables. A model is consistent with the training material when the expected value of each feature in the empirical distribution (training dataset) is equal to its expected value in the model's distribution. Each feature thus introduces a constraint, and finding $p^*(S_0|Z_0)$ becomes a constrained optimisation problem. For each constraint a Lagrange multiplier λ_i is introduced; the general form that the optimal solution in the maximum entropy sense is log linear [210, 213]:

$$p^*_\lambda(S_0|Z_0) \propto \exp(\sum_{i=1}^{m} \lambda_i * f_i(S_0, Z_0)) \tag{5.2}$$

The conditional probability distribution of states given observations is thus proportional to the exponentiated sum of weighted features.

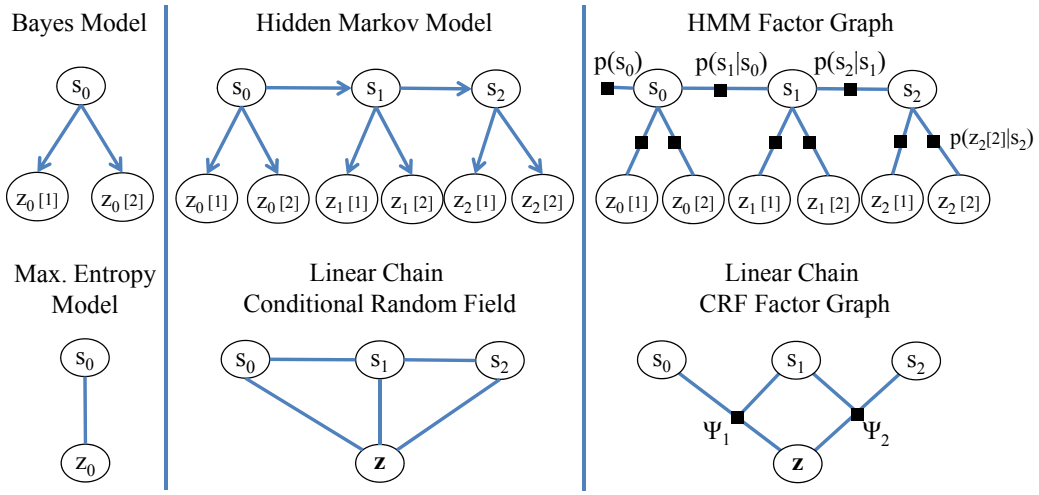


Figure 5.3: Directed generative graphical models (top) vs. undirected discriminative models (bottom).

5.2.2 Linear Chain Conditional Random Fields

Linear Chain Conditional Random Fields can be viewed as the sequence version of Maximum Entropy Models, in the same way that HMMs is an extension of the naive Bayes classifier model. In linear chain CRFs, the conditional probability of states given observations is proportional to the product of potential functions that link observations to consecutive states, as expressed in the equation below and shown in Fig. 5.3 (bottom right).

$$p^*_\lambda(S|Z) \propto \prod_{j=1}^{T} \Psi(S_{j-1}, S_j, Z, j) \tag{5.3}$$

where j denotes the position in the observation sequence, and Ψ are potential functions. A potential function is composed of multiple feature functions f_i, each of which reflects in a different way how well the two states S_{j-1} and S_j are supported by the observations Z. Depending on how we select features, they might refer to a different subset of observations (from a single step or from multiple steps), and may depend on one of the two S_{j-1}, S_j states or on both:

$$\Psi_j(S_{j-1}, S_j, Z, j) = \exp(\sum_{i=1}^{m} \lambda_i * f_i(S_{j-1}, S_j, Z, j)) \tag{5.4}$$

5.2.3 HMMs and CRFs

HMMs are directed generative graphical models: they are trained to maximize the joint probability distribution of observation and state variables, which they compute

as a product of state priors and conditional probabilities of observations given states (top right of Fig. 5.3). In contrast to HMMs, CRFs are undirected discriminative models: they are trained to directly maximise the conditional probability of state variables given observation variables, which they compute as a product of potential functions (bottom right of Fig. 5.3). Thus, unlike HMMs, in CRFs there is no need to model the exact conditional probability distributions of observations given states (or state transitions). Instead one only has to define feature functions, as discussed in Sec. 5.3.2.

Furthermore, as shown in the CRF factor graph (bottom right of Fig. 5.3), the power of CRFs over HMMs lies in how they link observations to each other and to states. CRFs are able to model both: 1) how observations relate to individual states, as well as 2) how they relate to transitions between states. This is very convenient for tracking systems that make use of inertial sensor data, which naturally depend on the transition between two locations rather than on a single location. Using inertial observations in a HMM would complicate things by either having to use an input-output first order HMM, where transition probabilities depend on inertial data, or having to model the problem as a second-order HMM, where a state represents a transition between locations.

In HMMs observations at a given timestamp are typically considered independent of each other given the state (naive Bayes assumption), whereas in CRFs, it is possible to define features that capture these dependencies. In addition, in HMMs, observations generated at a given step only depend on that step's state. In CRFs, we are flexible to define features that link an entire chain of observations (of arbitrary length) with a state or a state transition. This is useful when nearby observations are perturbated with correlated errors, e.g. when a local distortion of the magnetic field affects several consecutive heading observations. Such dependencies can be flexibly captured by functions that link a whole subchain of observations to a state transition. It is further useful when using RSS landmarks for tracking; for instance, several RSS observations must be received after time t before deciding if the observation at time t is a peak value.

Finally, in CRFs, it is possible to define more than one feature function that captures the dependency between a sub-chain of observations and a state (or state transition). New feature functions can be used to accommodate new sensing modalities in a natural way.

In summary, CRFs are a powerful model that allows us to flexibly introduce new types of sensors into our tracking system, and to define their dependencies not only

with single states, but also with state transitions, and with each other over a time window of arbitrary size.

5.3 Map Matching Using CRFs

Having introduced Conditional Random Fields, and their advantages over directed graphical models, we are now in a position to describe our algorithm, called MapCraft, which makes use of CRFs to track people in indoor environments[1]. MapCraft involves four distinct steps: 1) Map pre-processing; 2) Definition of states and feature functions; 3) Training to determine feature weights; and 4) Inference to estimate location over time. The first three steps are performed once for each building by either a mobile phone or a service in the cloud. The fourth step is performed online on the user's smartphone to track themselves.

5.3.1 Map Pre-processing

This step takes a floor plan as input, and produces a graph that 1) encodes a set of discrete states (locations), and 2) represents physical constraints between discrete states imposed by the map. This information will then be fed to the second step, to help us define the CRF's states and feature functions. In our implementation, such information is obtained from maps in various image formats. The main task is to extract edges from the image needed to perform map structure recognition or reconstruction, using standard edge detection algorithms. Note that indoor maps are typically cleaner than big Google maps, which makes the extraction quite simple. Standard edge detection algorithms, including simple ones that use pixel grey scale detection, could accomplish this goal. We can then use a connectivity test to find out the reachable regions in the map. A graph is built on the reachable region of the map. We first divide the graph into identical squares with edge length e. The size of e impacts both the position accuracy and the computational cost of the map matching algorithm. On the one hand, the larger the edge length the coarser the achievable position accuracy. In addition, to have connected vertices in narrow corridors, the edge length should not exceed the width of corridors connecting different parts of the building. On the other hand, as we decrease the edge length, we increase the computational cost of the map matching algorithm, which is quadratic on the number

[1]Since we assume that a floor plan (and optionally a radio map) is readily available, we interchangeably refer to the tracking problem as the map matching problem. Note however that CRFs as such do not require the presence of a map; they could be used to infer location without a map, using as input location sensor data from various modalities.

of vertices and bi-quadratic on the number of edges. On balance, a suitable choice of e in our system is the width of the narrowest corridor in most buildings, e.g. 0.8 m. The accuracy benefits of more fine-grained maps were observed to be negligible compared to the additional computational cost that they incurred.

The neighbours of a target vertex are vertices which have a common geographical border with the target vertex and pass the connectivity test as well. The removal of unreachable vertices is important to the system performance, because there is typically a large number of vertices in the map that cannot be reached from the legal region. The graph derived from the map is a finite, irreducible (consisting of a single strongly connected component), and aperiodic (the number of steps to return to a certain state could be both even and odd) that guarantees a unique stationary distribution for the corresponding Markov chain. An example of the graph construction process is shown in Fig. 5.4. If a central server is available in the building, the graph could be stored, generated and explicitly used by other users. The process of map generation and graph construction only happens once when a new map is used in the system.

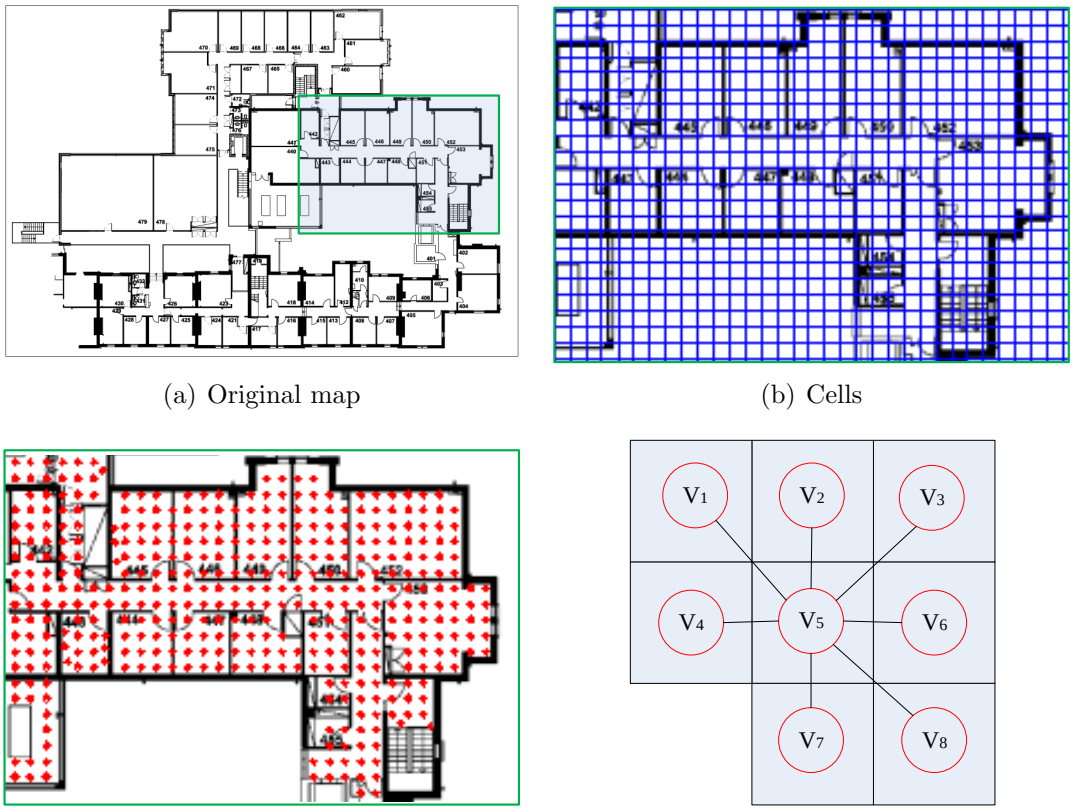
5.3.2 Definition of States and Feature Functions

The output of the previous step directly allows us to define the state space of the CRF, as the set of discrete locations encoded in the vertices of the generated graph. We are now in a position to introduce the set of feature functions used by MapCraft. Recall that a feature function f_i defines the degree to which observations Z support our belief about two consecutive states (S_{t-1} and S_t); the stronger the support, the higher the value of the feature function $f_i(S_{t-1}, S_t, Z)$. Note that in CRFs, we are free to use any subset of observations, generated at a single or multiple time steps, though in most cases, the observations that matter are those temporally close to time t. In what follows, we specify for each feature the subset of observations that it uses, and how it relates them to state transitions or, in some cases, to individual states.

The first feature in our system expresses the extent to which an inertial measurement Z_t^{in} supports the transition between states S_{t-1} and S_t:

$$f_1(S_{t-1}, S_t, Z_t^{in}) = I(S_{t-1}, S_t) f_1^{\theta}(S_{t-1}, S_t, Z_t^{\theta} + f_1^{l}(S_{t-1}, S_t, Z_t^{l})) \tag{5.5}$$

where $I(S_{t-1}, S_t)$ is an indicator function equal to 1 when states S_{t-1} and S_t are connected and 0 otherwise. The inertial observation Z_t^{in} has two components: the measured length Z_t^{l} and angle (heading) Z_t^{θ} of displacement, which are assumed to



(a) Original map (b) Cells

(c) Vertices remained after connectivity test (d) Underlying graph example

Figure 5.4: Example showing how a floorplan is automatically converted into a connected graph.

be independent. f_1^θ and f_1^l are the functions to relate the angle and length to the underlying graph, respectively. The function f_1^θ is given as

$$f_1^\theta(S_{t-1}, S_t, Z_t^\theta) = \ln \frac{1}{\sigma_\theta \sqrt{2\pi}} - \frac{(Z_t^\theta - \theta(S_{t-1}, S_t))^2}{2\sigma_\theta^2} \tag{5.6}$$

where $\theta(S_{t-1}, S_t)$ is the orientation of the edge between states S_{t-1} and S_t, and σ_θ^2 is the heading variance of the observation Z_t^θ. The feature function f_1^l is defined likewise.

The purpose of the second feature is to handle correlations in heading errors in a recent time window. It does so by measuring how well a *corrected* inertial measurement $Z_t^{in}(\hat{\theta})$, derived by rotating Z_t^{in} by angle $\hat{\theta}$, supports the transition between states S_{t-1} and S_t:

$$f_2(S_{t-1}, S_t, Z_{0:t}) = f_1(S_{t-1}, S_t, Z_t^{\text{in}}(\hat{\theta})), \tag{5.7}$$

The rotation angle $\hat{\theta}$ is estimated as the average heading difference between the estimated and measured headings from time stamp $t - w$ to t, where w is a window

size parameter. The estimated headings are based on MLE state estimates generated by MapCraft's inference step (Sec. 5.3.4). More specifically:

$$\hat{\theta} = \sum_{i=1}^{w} \Theta\left(S_{t-i}^{\mathrm{MLE}} - S_{t-i-1}^{\mathrm{MLE}}, Z_{t-i}^{\theta}\right)/w, \tag{5.8}$$

where Θ represents the angle between two vectors, and S_t^{MLE} is the maximum likelihood estimate (MLE) of the state at time t taking into account all measurement from 0 to t. MLE state estimates are computed efficiently using the Viterbi algorithm as explained in Sec 5.3.4.

The third feature function is optional, and it takes into account the signal strength observations in conjunction with a radio fingerprint map, if it is available in a building. Unlike the previous two feature functions, that constrain state transitions, this feature constraints individual states.

$$f_3(S_t, Z_t^{\mathrm{RSS}}) = -(S_t - \boldsymbol{\mu_t})^T(\Sigma_t)^{-1}(S_t - \boldsymbol{\mu_t}), \tag{5.9}$$

in which the observation Z_t^{RSS} is the estimated mean $\boldsymbol{\mu}_t$ and covariance Σ_t of current position given RSS fingerprint data. This feature measures the negative squared Mahalanobis distance between the state and the RSS-based position estimate.

When the fingerprinting database is not available, this feature function can make use of the transmitter-receiver distances estimated from Chapter 3. Then $\boldsymbol{\mu}_t$ and Σ_t are the mean and covariance of the estimated distance set.

The CRF model used by MapCraft combines the three features above into a potential function $\Psi_j(S_{j-1}, S_j, Z, j)$, which is computed as the exponentiated function of their weighted sum as shown in Eqn. 5.4. The way that weights λ_i are determined is explained in the next subsection (Training step). However, as we will show in Section 5.4, in typical indoor environments, the training step is not strictly required, as using equal weights typically yields comparable location accuracy to that obtained after careful weight training. Hence, in practice the following training step can be skipped and equal weights can be assigned to the three features above.

The power of the CRF model is that it does not constrain us to only use the features above. Depending on the sensor data available and the maps available, it might be useful to extend the list of features. For example, suppose that we are in possession of a radio map, denoted with PeakPointInMap(S_t), that contains the locations where the RSSI from an access point takes a local peak value (e.g. provided

by [91]). We could then define a fourth feature as follows:

$$f_4(S_t, Z_{t-w:t+w}) = \begin{cases} 1 & \text{if } Z_t = \max(Z_{t-w:t+w}) \\ & \text{and } \text{PeakPointInMap}(S_t) \\ 0 & \text{otherwise} \end{cases} \tag{5.10}$$

In case the building is equipped with cameras or sensors alike, additional features could be added to easily incorporate visual and other sensor data. In general, CRFs provide a flexible model where a number of different observations can be fused into the model. Recall that since we use a CRF model, we do not need know the exact observation probabilities given states, and need not restrict ourselves to assuming conditional independence of observations given state. Finally we are free to associate a state at a particular time step (or a state transition) with observations that go beyond this time step.

5.3.3 Training to Determine Feature Weights

In many scenarios where CRFs are applied, the freedom of being able to define a number of different features comes at the cost of needing to estimate their weights. This step requires training material T, which consists of one or more true trajectories, paired with respective sequences of sensor observations. Training the CRFs to estimate weights is performed by maximising the log-likelihood on the training material T, i.e. the log of the conditional probability of states given observations:

$$L_\lambda(T) = \sum_{((S,Z)\in T)} \log p^*_\lambda(S|Z) \tag{5.11}$$

By taking the partial derivative of the log likelihood function with respect to each feature λ_j and setting it to 0, we get the maximum entropy model constraint:

$$E_T(f_i) - E(f_i) = 0 \tag{5.12}$$

that is, the expected value of the i-th feature under the empirical distribution $E_T(f_i)$ is equal to its expected value under the model distribution $E(f_i)$, where the two expected values are:

$$E_T(f_i) = \sum_{(S,Z)\in T} \sum_{j=1}^{T} f_i(S_{j-1}, S_j, Z, j) \tag{5.13}$$

$$E(f_i) = \sum_{(S,Z)\in T} \sum_{S'\in StateSeq} \sum_{j=1}^{T} f_i(S'_{j-1}, S'_j, Z, j) \tag{5.14}$$

Setting the gradient to zero does not always give us an analytical solution for the weights λ_i. This requires resorting to iterative methods, such as iterative scaling or gradient-based methods. Independent of the method used, one needs to be able to compute $E(f_i)$ and $E_T(f_i)$ efficiently. This is easily done for $E_T(f_i)$, since all we have to do is go over each training sequence, sum up the weighted sum of feature functions over all time steps, and sum up the result for all training sequences.

Computing $E(f_i)$, however, is slightly more complicated and requires the use of a dynamic programming approach, known as the Forward-Backward algorithm, similar to the one typically used for Hidden Markov Models. The details of how this algorithm is applied to CRFs can be found in [213]. The time complexity of the Forward-Backward algorithm is $O(|S|^2T)$, where T is the length of the sequence and $|S|$ is the number of discrete states.

The goal of the training is to tune the feature weights λ_i in Eqn. 5.4 in order to make the features best support the training data. The weight actually reflects how much we trust the corresponding feature. For instance, if we set a large weight, e.g. 5, for feature $f_1(S_{t-1}, S_t, Z_t^{in})$, we can see from Eqn. 5.6 that it is equivalent to decreasing the length variance σ_l^2 and angle variance σ_θ^2 much smaller, which means we consider the length and angle measurements to be more accurate than indicated by these variances defined in Eqn. 5.6. Therefore, it is not necessary to tune the weights from training data if the variances of measurements are well defined.

It is also worth noting that large feature weights should be avoided, because they amplify slight differences in the feature values of two tracking solutions into significant differences in the potential function ψ, due to the exponential function in Eqn. 5.4. We suggest that all feature weights should be chosen from $[0.5, 2]$, which works well in all our experiments in different environments. It is also demonstrated in Sec. 5.4 with our empirical results from three different indoor environments that the training step only has slight impact ($< 10\%$) on localisation accuracy. The default setting of weights equal to 1 for all features yields similar performance to weights derived from training. Hence, in practice we do not need training material (ground truth trajectories); we can directly proceed to the location inference step described below.

5.3.4 Inference to Estimate Locations

The final step is finding the most likely sequence of hidden states, i.e. the most likely trajectory $\boldsymbol{S}^*$. This requires solving the following optimisation problem:

$$\boldsymbol{S}^* = \underset{\boldsymbol{S}}{\operatorname{argmax}}\, p(\boldsymbol{S}|\boldsymbol{Z}), \tag{5.15}$$

The Viterbi algorithm, a dynamic programming algorithm, offers an iterative solution. The Viterbi algorithm is often used for error correction in convolutional coding and is computationally efficient with a worst case time complexity of $O(|S|^2T)$, where T is the length of the trajectory in steps and $|S|$ the number of states. It is similar to the Forward-Backward algorithm used in each learning iteration, with the subtle difference that it applies a maximisation instead of a summing operation in each induction step. More specifically, in each step, it evaluates the highest score $\delta_j(s)$ along a path at position j that ends in each possible value s for state S_j, as follows, and gradually fills a lattice with these values:

$$\delta_j(s|Z) = \max_{s' \in S} \delta_{j-1}(s')\Psi_j(s', s, Z, j) \tag{5.16}$$

In the case of on line *real-time tracking*, the most recently filled column of the lattice, which represents a discrete distribution $p(S_i|Z_{1:i})$ is normalised and converted into a 2D Gaussian distribution and displayed on the user's map. If MapCraft is extended to employ features that use a few observations after the current step (e.g. feature f_4), a slight delay will be introduced in displaying a user's location. In the case of delay tolerant off line tracking, it is possible to wait until the location accuracy is high before performing the path backtracking step of the Viterbi algorithm, and computing the optimal path form the lattice. As an alternative, we can combine the online and offline approaches above and refresh the users map to show at each step the most up-to-date previous and current location estimates.

Specifically, the mean $\hat{\mu}$ and variance $\hat{\Sigma}$ of the approximated Gaussian distribution are determined by the MLE estimates.

$$\hat{\mu} = \sum_{i=1}^{N} S_i p(S_i|Z_{1:i}), \tag{5.17}$$

$$\hat{\Sigma} = \sum_{i=1}^{N} (S_i - \hat{\mu}) p(S_i|Z_{1:i}). \tag{5.18}$$

Then the accuracy of the current position estimate is determined by the 2σ or 3σ rule. The pseudo code of MapCraft is shown in Algorithm 1.

The input of the algorithm consists of three structures: graph G, feature function handles F, and sensor observations Z, as shown in Line 1. Graph G is a structure with the following elements: $G.N$ is the number of states in the graph, and $G.S_j$ $(j = 1, 2, \cdots, G.N)$ is the ith state including: $S_j.location$ – location of this state, $S_j.neighbourList$ – the neighbour list of S_j. The features input F has the following

Algorithm 1 MapCraft algorithm

```
    procedure MAPCRAFT(G, F, Z)
        if G.N < 1 then return false
 3:     if F.N < 1 then return false
        if Z is empty then return false
        for S_j in G do
 6:         for s_n in F do
               v(j,n) ← s_n(S_j, Z)                          ▷ Eqns. (5.9), (5.10)
           end for
 9:         for t_n in F do
               for S_j' in S_j.neighbourList do
                   e(j,n) ← t_n(S_j', S_j, Z)                ▷ Eqns. (5.5), (5.7)
12:            end for
           end for
           Ψ_j(S_j', S_j) ← exp(∑(λ_v v(j,n) + λ_e e(j,n)))  ▷ Eqn. (5.4)
15:    end for
       Ψ ← Ψ / ∑Ψ                                            ▷ Normalization
       μ̂ ← ∑_{j=1}^{N} Ψ_j S_j.location                      ▷ Eqn. (5.17))
18:    Σ̂ ← ∑_{j=1}^{N} Ψ_j (S_j.location − μ̂)
       [index, likelihood] ← max(Ψ)
       decodeSeq(t) ← index
21:    for τ = t − 1 : 1 do                                  ▷ Viterbi decoding
           [index, likelihood] ← max_{s'∈S} δ_{τ−1}(s') Ψ_τ(s', s, Z, τ)   ▷ Eqn. (5.16)
           decodeSeq(τ) ← index
24:    end for
       return μ̂, Σ̂, decodeSeq
```

elements: $F.N_S$ is the number of feature functions applied to states, $F.N_T$ is the number of feature functions applied to transitions between states, $F.s_i$ ($i = 1, 2, \cdots, F.N_S$) is the ith feature function handle applied to states, and $F.t_i$ ($i = 1, 2, \cdots, F.N_T$) is the ith feature function handle applied to state transitions. In this implementation, features functions are shown in Eqns. 5.5, 5.7, 5.9, and 5.10. The sensor observation includes displacement vector (l, θ), WiFi RSS vectors $(rss_1, rss_2, \cdots, rss_N)$, etc.

In Algorithm 1, lines 2–4 check the sanity of the input including the graph, the feature function handles, and the sensor observations. Lines 5–15 iterate over all states of the graph and calculate the potential of each state give the sensor observations. Specifically, lines 6–8 compute the feature functions applied to states. Lines 9–13 calculate the feature functions applied to transitions between states through iterations over all neighbours of the current state. Line 14 obtains the potential of the current

state by evaluating weighted average of all feature functions. Line 16 normalizes the potential values for the whole graph to prepare for calculation of location mean and variance. Lines 17 and 18 calculate the mean and variance of the location estimate. Lines 19–20 get the maximum potential value at the current time stamp to start the viterbi decoding. Lines 21–24 obtain the decoded state sequence by backtracking the Viterbi trellis. Finally, line 25 returns the estimated location including the mean and variance, and the decoded state sequence.

5.4 Evaluation

This section demonstrates and evaluates proposed and competing map matching algorithms through extensive data collection at different times and places. Extensive experiments are conducted by different users with different devices in three experimental sites with different floor plans to account for context variability. Note that in this chapter an existing strawman PDR algorithm (referred to as S-PDR) instead of the proposed R-PDR algorithm is implemented. The performance of the combination of R-PDR and MapCraft, together with other components of the system, is discussed in Chapter 7.

S-PDR is based on two assumptions. The first assumption is that the user hold the mobile device at hand in texting mode. The second assumption is that the heading of the mobile device is always the same as the heading of the user. These two assumptions make S-PDR simple and efficient to implement and widely used. However, it has problems in terms of practicality because users are not likely to hold the mobile device all the time and keep the heading of the device the same as the heading of the walking. We combine S-PDR with MapCraft to form the first competing approach in the fusion category: S-Tracker. To avoid penalising S-Tracker for its assumptions, we compare the performance of the proposed algorithms using a variety of mobile device attachments, to that of S-Tracker using a single device attachment (texting mode).

5.4.1 Experimental Setup

5.4.1.1 Sites

To demonstrate the real world applicability of the tracking system, our map matching algorithm is evaluated and compared against competing approaches in three real-world settings, namely an office building, a market, and a museum. All of these have different floor plans as shown in Fig. 5.10 and methods of construction which affect

the obtained sensor data. The office environment ($65 \times 35m^2$, where the majority of the tests have been conducted) is a multi-storey office building with a stone and brick construction, reinforced with metal rebars - testing was conducted on the fourth floor. The market ($108 \times 53m^2$) consists of a number of small shops, laid out over a single floor. Construction is mainly brick and mortar, with a metal roof. The museum ($109 \times 89m^2$) is a multi-storey stone building with large, open spaces. Testing was conducted on the ground floor. Overall, 500 trajectories of average length 200 m were collected over 15 days. Error is expressed in [m] RMS.

5.4.1.2 Participants

The variations between different people are taken into account by acquiring data from 20 people of different genders, heights, and ages. They may not appear in all three sites, but each one of them has participated in the experiments in at least two different sites. During the experiments, the subjects held the mobile phones in their hands and then walked anywhere in the building without planned routes, to realistically capture real pedestrian motion, rather than artificial, constant speed trajectories.

5.4.1.3 Devices and Implementation

Different types of mobile phones and pads are involved in experiments, including LG Nexus 4, Asus Nexus 7, Samsung Nexus S, Samsung Galaxy S IV, Samsung Galaxy S III, Samsung I9100G Galaxy S II, HTC Hero S and Huawei U8160. These mobile phones differ greatly in terms of sensors, functionality and price. But one thing in common is that they all run the Android operating system no earlier than version 2.3.6. A snapshot of our application prototype has been shown in Fig. 5.1.

5.4.1.4 Ground truth

To provide accurate ground truth, numbered labels were placed along corridors and within rooms on a 3 m grid. Using the device's camera, these were filmed at the same time experiments were conducted. The time-synchronized video streams were then mapped to locations on the floorplan, and intermediate locations interpolated using footstep timing, also obtained from the video.

5.4.1.5 Training

Fig. 5.11 shows the impact of training on the performance of MapCraft. Training alters weights for the various input features, away from the nominal case of weights

1 for all features (when the training iteration is 0). Several iterations of training were run on a set of training trajectories obtained from the office environment. The weights from each training iteration were then applied to each of the three data sets (trajectories from the office, museum, and market) for cross validation. Note that the RMS error of the trajectories in the office environment, where training was performed, is decreased with the number of training iterations, but only slightly (up to 9%). The RMS error of market and museum trajectories also do not vary much; it can even slightly increase because the training is performed in a different environment than testing. This implies that training is not critical to good performance and in practice, it can be skipped. The remainder of experiments, which are conducted with weights 1 for all features, show that MapCraft is accurate without requiring careful training to a particular environment.

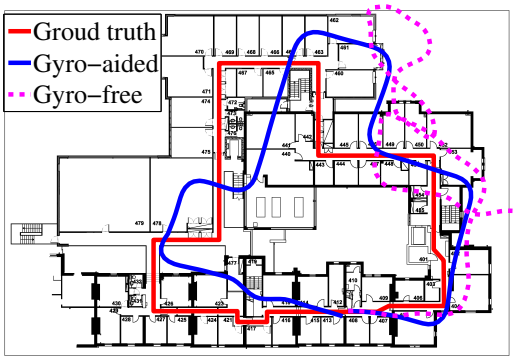
5.4.1.6 Competing Approaches

We compare MapCraft with several state-of-the-art map matching algorithms. For readability, their names reveal the underpinning Bayesian estimation technique with references that point to source papers with implementation details.

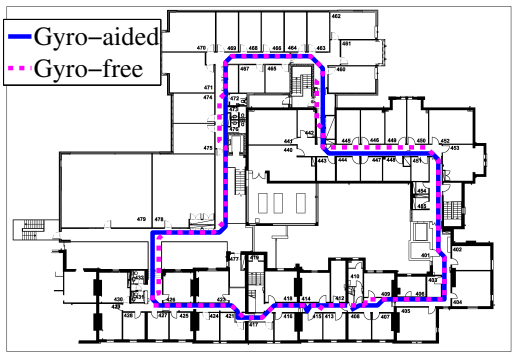
- **HMM** [154]: This algorithm uses a first order HMM, where states represent discrete positions, with equal transition probabilities between neighboring states (regardless of the vertex degree). A typical example is VTrack [154] which works very well using ground truth position estimates from GPS as observations. Seitz et al. [214] extends VTrack and obtains the underlying graph of HMM from RSS rather than GPS, by also exploiting position estimates from the inertial trajectory as observations;

- **HMM-IO** [164]: This is an Input-Output first order HMM; the difference from the first order HMM is that inertial data is not used to generate observations at each step, but to calculate the transition probability between consecutive states;

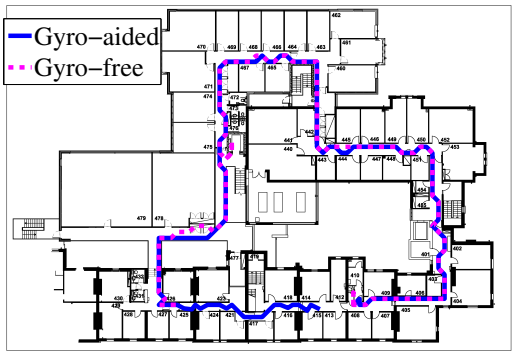
- **HMM2** [156]: In the second-order HMM, states are path segments connecting two locations in the map. Observations encompass both inertial displacement vectors and RSS-based position estimates. Transition and observation models are defined as in AutoWitness [156];



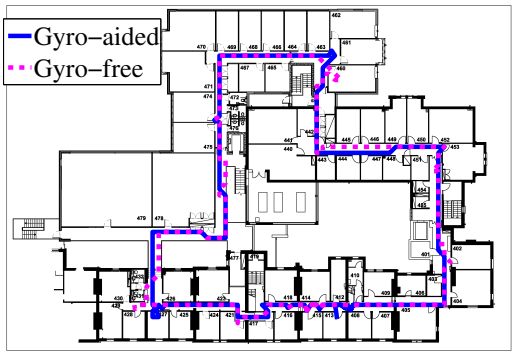
(a) Ground truth and raw trajectory



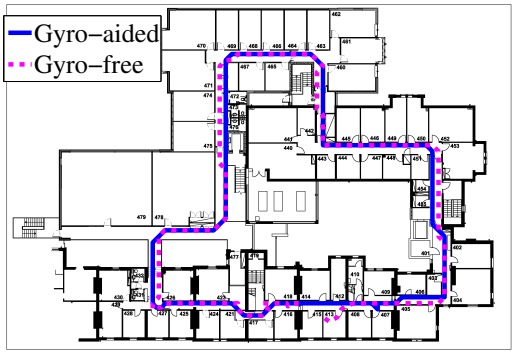
(b) MapCraft



(c) HMM



(d) HMM-IO



(e) HMM2

(f) PF

Figure 5.5: Matched trajectories using MapCraft and competing approaches with gyro-free and gyro-aided simple raw trajectories.

- **PF** [80, 166]: This algorithm uses a particle filter implementation similar to Zee [80, 166]. This implementation maintains a four-dimensional joint probability distribution as a particle filter, with a set of particles (samples), $X = (X_1, X_2, \cdots, X_N)$ representing the probability distribution where $X_t = (x_t, y_t, \theta_t, \Delta l_t)$, with (x_t, y_t) being the 2D location, θ_t being the heading, and Δl_t being the step

length offset. Taking the heading θ_t and step length offset Δl_t into states can enable the particle filter to automatically learn the heading drift and step length offset of the trajectory. With the observations including step length l_t and step heading change α_t from inertial sensors, the particles are updated as

$$\begin{aligned} x_t &= x_{t-1} + (l_t + \Delta l_t)\cos(\theta_t), \\ y_t &= y_{t-1} + (l_t + \Delta l_t)\sin(\theta_t), \\ \theta_t &= \theta_{t-1} + \alpha_t + n_\theta, \\ \Delta l_t &= \Delta l_{t-1} + n_l, \end{aligned} \tag{5.19}$$

where n_l is the step length variation uniformly distributed within $\pm 30\%$ of the estimated step length, and n_θ is zero-mean Gaussian distributed perturbation $n_\theta \sim \mathcal{N}(0, 20^\circ)$. This particle filter starts with 4,000,000 particles to estimate the start position of the trajectory and continues with 20,000 particles after convergence.

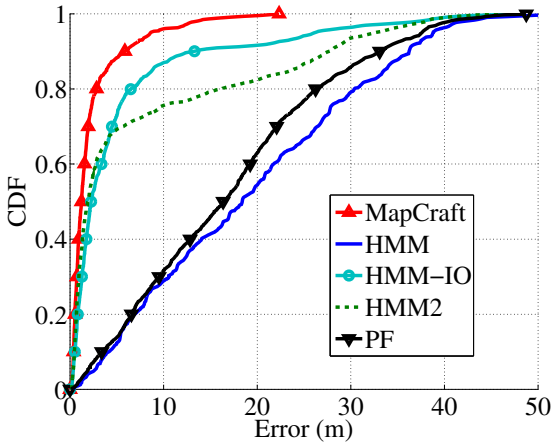
Generally speaking, these competing approaches work well with simple structure trajectories, e.g. those generated in major corridors in a building, as shown in Fig. 5.5. It is demonstrated below that the accuracy of these approaches decreases dramatically with tortuous trajectories shown in Fig. 5.10.

5.4.2 Performance Comparison

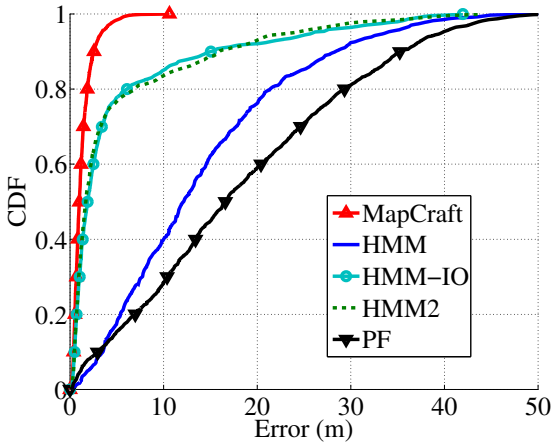
This section compares the accuracy and robustness of MapCraft with competing approaches.

5.4.2.1 Accuracy/Sensor Tradeoff

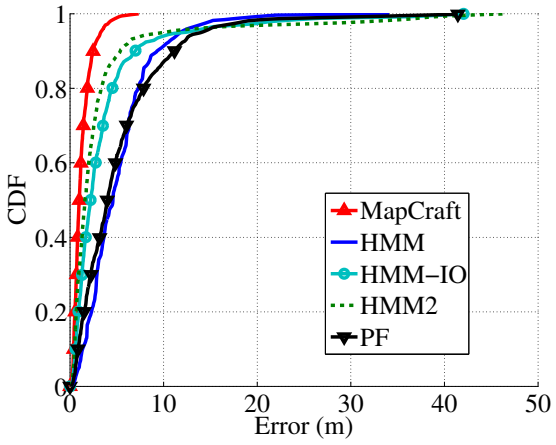
The goal of the first experiment, which was conducted in the office site, is to explore the tradeoff between sensor usage and position accuracy. Different sensors provide different accuracy in location or heading estimations. For instance, gyro sensors help improve heading estimates, esp. in the presence of magnetic field distortions, as are typically encountered in indoor settings. WiFi scans provide helpful information about absolute location. Fig. 5.6 shows three different regimes of map matching with increasing levels of sensor usage: 1) accelerometer and magnetometer only; 2) full inertial sensing: accelerometer, magnetometer, and gyroscope (no WiFi; 3) full inertial sensing with periodic WiFi RSS measurements. Notice that MapCraft significantly



(a) Accelerometer + magnetometer



(b) Accelerometer + magnetometer + gyroscope



(c) Accelerometer + magnetometer + gyroscope + WiFi

Figure 5.6: Error CDFs for various map matching approaches with increasing number of sensors a) gyro-free and b) gyro-aided and c) wifi- and gyro-aided.

outperforms competing algorithms under all three regimes, typically resulting in errors two to three-fold lower than the next best approach (2000 particles are used in PF-based approach).

Specifically, Fig. 5.6(a) shows the error CDF with only accelerometer and magnetometer measurements, such as would be used in a system aiming to consume minimal energy. The lack of gyro readings makes the heading estimation very inaccurate, especially in areas with high concentrations of metal. As a consequence, the whole raw trajectory is very noisy. These distorted trajectories greatly deteriorate the performance of existing methods whilst our approach remains robust to noisy sensors. One of the key reasons is the use of feature $f2$ that handles heading error correlations. Fig. 5.6(b) shows the error distributions with full inertial measurements. The gyroscope provides more accurate heading estimation over a short period of time. However, over time, the accumulated error becomes excessively large. However, due to the features in our system, the accumulated error is gradually reduced in each step, which guarantees the accuracy of the heading estimation and yields accurate matching results. Fig. 5.6(c) shows the error CDF with periodic WiFi measurements, taken every 16 seconds. These are used, in conjunction with a radio fingerprint map, to provide absolute position estimates. With these measurements, the performance of HMM and PF improves significantly. However, the performance of MapCraft increases further still with the combination of relative and absolute measurements. This is because it captures more features of the measurements across both time and space.

5.4.2.2 Convergence distance

The proposed and competing algorithms are designed for online tracking. However, in the absence of WiFi measurements and without knowledge of the initial pose, they initially incur a convergence cost, which we measure as the average minimum distance needed for the algorithm to find the correct location within an error of 3 m. Fig. 5.7 shows that for 97% of cases, MapCraft converges within 50 m and the next best, HMM2, within 60 m. The HMM (Seitz) and PF approach show considerably worse performance, in some cases never converging. Even with a very large number of particles (100k), the PF approach fails to converge in many cases due to impoverishment. HMM is unable to estimate the correct location, as it requires a good initial starting point estimation. Note that when WiFi-aiding is used, all algorithms provide a good location estimate after 2-4 m.

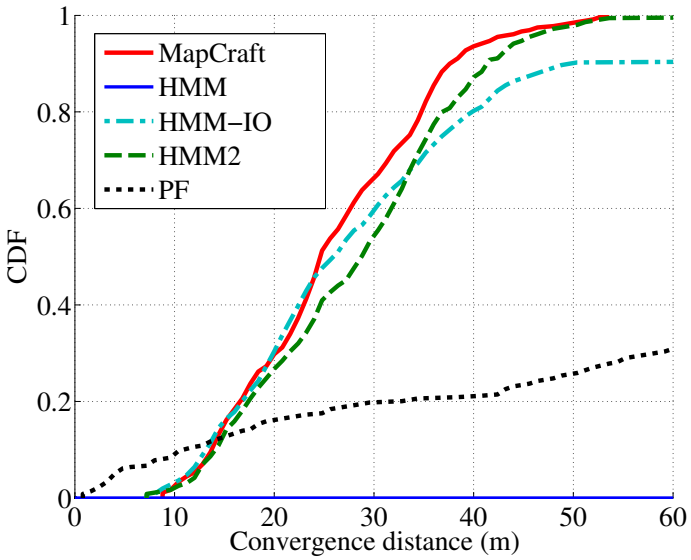


Figure 5.7: Convergence distances of MapCraft and competing approaches (WiFi-free).

5.4.2.3 Discussion

The underlying reason for the superiority of MapCraft in tracking accuracy is the ability to model displacements of inertial trajectories without prior assumptions, as we have discussed in Section 5.2. The first-order HMM matches the locations rather than the displacements of the raw trajectory to possible location sequences in the map, and thus its performance can be easily degraded with a very noisy raw trajectory due to the increasingly accumulated location error from the inertial measurements. We have to make prior assumptions on the transition and observation probability distributions for HMM and HMM2, which are only approximations of the real world scenarios. In addition, the HMM category (HMM, HMM2, and HMM-IO) all assume the independence of observations given the state but unfortunately different observations are likely to be correlated, both spatially and temporally. As explained in Section 5.2, CRF makes no similar assumptions, which makes it easy to capture these dependence and accurately model the tracking process as it is. As a result, CRF has better accuracy in general, especially with tortuous trajectories in long-term tracking.

The PF approach is sensitive to motion sensing errors, especially the heading bias caused by magnetic disturbance of metallic objects, which is ubiquitous in indoor environments. In our experiments, the magnetic disturbance from one big metallic object usually lasts for 5 to 15 meters and can be as large as 30 degrees, which is very likely to mislead all particles into the wrong room, especially with very tortuous trajectories generated in our experiments. Since the PF approach only has local

information about the region covered by particles, if all particles enter a wrong room it is hard to recover to the correct position. Furthermore, if the heading bias comes at the beginning of the tracking process, it is difficult for the particle filter to converge without knowledge of the initial pose of the pedestrian.

The MapCraft is lightweight in both running time and memory usage because it neither stores transition or emission matrices (only several feature functions) nor performs expensive matrix operations. The HMM category, especially HMM2 whose state space is much bigger, take more running time and memory for processing transitions and emissions. The running time and memory usage of HMM-IO are comparable to MapCraft when the number of features is small, e.g. 2 or 3 in our experiments because the transition and emission probabilities are computed in a real-time manner.

The running time and memory usage of particle filter largely depends on the number of particles. The major computation cost comes from checking whether the position update of each particle violates the map constraints. The computation complexity of 4,000,000 particles at the beginning and 20,000 particles after convergence would way beyond the computation power of mobile devices.A graph-based particle filter [215] has been proposed to reduce the computational complexity of particle filters. This approach first discretizes the floor plan into a connected graph. Then a particle filter is applied on top of this graph. Each particle maintains a state vector of the vertex in graph and orientation instead of physical coordinates as in normal particle filters. The particles are updated by the absolute orientation estimates from the compass, relative displacement estimates from the pedometer, and absolute location estimates from WiFi-fingerprinting. This discrete particle filter approach is reported to achieve the same positioning accuracy with only 10% computation cost of normal continuous particle filters.In essence, the graph-based particle filter is a particle filter running on top of a discrete graph space with many fewer possible transitions than a continuous space, which explains the significant reduction in computational complexity. However, just like a normal particle filter, it only has local information of vertices covered by particles instead of global information of the whole graph like MapCraft. As a result, it is hard to recover if all particle enter a wrong room (a set of vertices), especially with complex and tortuous trajectories. Given this disadvantage, this graph-based particle filter is expected to have similar accuracy to a normal particle filter and is thus not evaluated in this experiment.

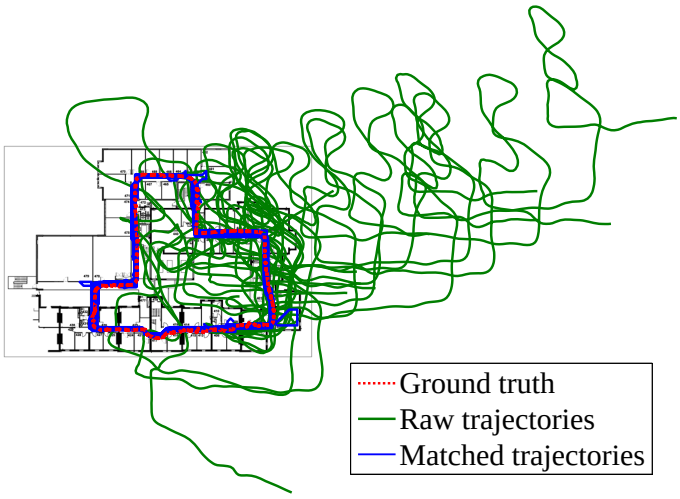


Figure 5.8: Experiment in the Office test site in which the same route was walked 50 times. The uncorrected and resulting trajectories are shown.

5.4.3 Robustness

The next set of experiments are performed to examine the robustness of MapCraft and its applicability to real world scenarios. It must be emphasized that all these results are for the WiFi-free case, i.e. the only input to the system is a floor plan and IMU data. The results that we had with WiFi were even better, but we do not show them for space reasons.

5.4.3.1 Long-term tracking

First, we studied the repeatability of accuracy results as we run MapCraft for long time periods, resulting in inertial trajectories that are increasingly distorted with respect to the true trajectory. Fig. 5.8 shows the results of an experiment where the same route was followed 50 times in the office environment by different people with different mobile phones and the resulting trajectories calculated. Note that although the raw inertial measurements are significantly different each time, the map-matched trajectories are very similar. This shows that the map matching process is able to accurately reconstruct the correct trajectory, in spite of excessive and varying metal-induced distortions to the heading estimation.

5.4.3.2 Multi-site performance

We then studied the robustness of MapCraft in a variety of environments, namely an office building, a museum and a supermarket.

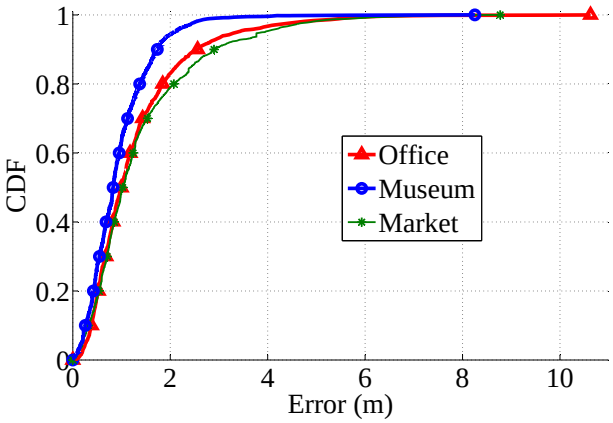


Figure 5.9: Location errors in office, museum, and market sites.

Very complex and tortuous trajectories typically are the weakness of inertial tracking systems, due to drift and the absence of absolute anchor measurements. However, by using the map matching approach, very accurate reconstruction can be provided even when the user executes a complex trajectory. This is shown in Figure 5.10. The cumulative distribution function of location errors in the three environments are shown in Fig. 5.9.

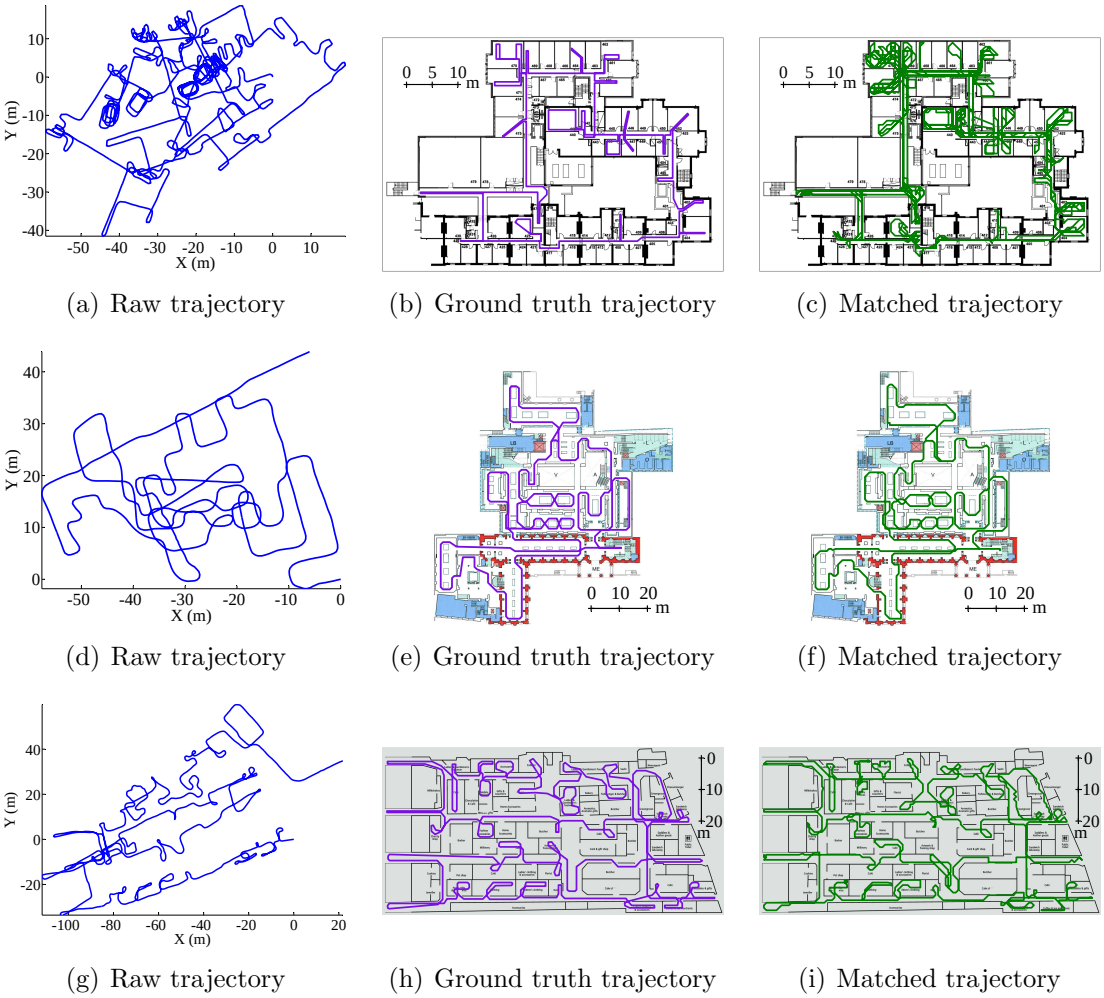
The room-level accuracy, defined as the percentage of matched trajectories entering the correct room, and overall 97 percentile accuracy is shown in Table 5.1. The room level accuracy is above 90% in all environments, which demonstrates the applicability of our approach to tackling real-world navigation problems. This compares well with other approaches based on extensive and multimodal radio fingerprinting [89].

Table 5.1: RMS error, 97 percentile accuracy and room-level accuracy of MapCraft in three sites.

Site	office	museum	market
RMS error (m)	1.69	1.14	1.83
97 percentile (m)	4.10	2.37	4.53
Number of rooms	20	15	29
Room entry-events	357	360	82
Room identification (%)	93.0	100	96.3

5.4.3.3 Impact of training

Finally we studied the impact of training on the tracking accuracy of MapCraft. In Fig. 5.11, we train the model with ground truth trajectories from the office environment and apply the same feature weights to all three experiment sites. First notice



(a) Raw trajectory (b) Ground truth trajectory (c) Matched trajectory

(d) Raw trajectory (e) Ground truth trajectory (f) Matched trajectory

(g) Raw trajectory (h) Ground truth trajectory (i) Matched trajectory

Figure 5.10: Experiments in office (top), museum (middle), and market (bottom) site, showing raw, ground-truth and matched trajectories.

that the training of feature weights does not improve position accuracy significantly in the case of the office environment. Second, we used the various trained feature weights from the office environment in the museum and market environments, but again did not observe a significant change in position accuracy as we move from one iteration's weights to another.

5.4.3.4 Discussion of MapCraft Performance

Intuitively, map matching algorithms are only useful when there are maps available, e.g. floor plans, radio maps, etc. It is these maps that provide additional information that indicates the current location, which is then taken into account in the location estimation process by map matching algorithms. As a result, map matching algorithms, including MapCraft, cannot contribute more to either the positioning

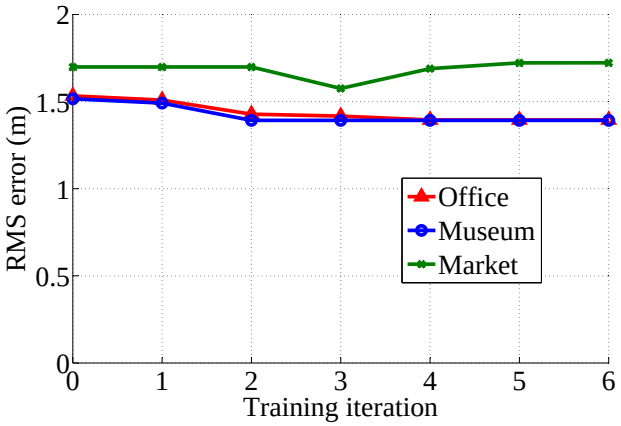


Figure 5.11: Impact of training on the performance of MapCraft. All parameters are trained from data collected in the office site and tested in all three sites.

accuracy or the robustness of the system in an environment without constraints to bound the trajectories, e.g. open space with no radio maps available. Experiments have been conducted in a basketball court without any physical constraints or signal maps to test the performance of MapCraft. In these experiments, the accuracy of MapCraft is almost the same as that of the PDR algorithm used. Detailed results are reported in the evaluation (Chapter 7).

The performance of MapCraft also hinges on the selection of appropriate grid size of the underlying graphs.As stated in Sec. 5.3.1, the maximum allowable grid size depends on the width of corridors to ensure that the underlying graph is connected. As a result, the selection of grid size should follow the instructions in Sec. 5.3.1.

5.5 Discussion

This chapter has demonstrated the merit of a novel map matching technique, based on the application of conditional random fields. It has shown how it is robust, being able to operate with very noisy sensor data; lightweight, running in under 10 ms on a smartphone; and accurate, achieving the lowest RMS errors compared with other state-of-the-art approaches. It does not require per-site training, which will allow for easy and widespread adoption, as the only information that is required to use our approach is a floor plan.

In the future, the proposed system has the potential to make crowd-sourcing of WiFi fingerprints practical, without requiring time-consuming manual scans. This is because MapCraft is able to establish a user's position using only dead-reckoned

trajectories and a floorplan, without any external information such as a starting location or knowledge of WiFi access point locations. We believe that MapCraft has widespread application to a number of domains, as this single approach can be used with a wide variety of sensors and map information. One particularly relevant area is estimating location online and in real-time in resource-constrained body-worn sensors.

Although MapCraft has been demonstrated to work quite well across different people and environments, the positioning accuracy can be further improved. The positioning error of MapCraft mainly comes from two facts that 1) the pedestrian dead reckoning algorithm usually requires context-specific tuning, like the bias of various sensors, the step length/heading estimation parameters, etc., and 2) the location-specific information can be further explored from the environment which can only be learned automatically rather than constructed manually in practice. To address these two issues and further improve the positioning accuracy, the next chapter proposes a lifelong learning component which can automatically learn both the context-specific parameters for PDR component such as the sensor bias and heading bias, and the radio maps like WiFi and magnetic distortion maps, with zero user effort.

Chapter 6

Lifelong Learning

The fact that PDR parameters often require context-specific tuning has posed a general challenge to the real world application of PDR algorithms. However, it is argued in this chapter that this information could be automatically learnt, rather than manually tuned or assumed as input. In addition, location specific observations, such as radio fingerprints, magnetic distortions, etc. can also be learned automatically every time we visit a new environment to improve later visits. This is not dissimilar to the way that people improve their navigation skills when they repeatedly visit the same environment.

This chapter exploits the interplay between the pedestrian dead reckoning (PDR) component and the map matching component based on the principle of lifelong learning. Existing work typically feeds PDR output into a map matching technique (e.g. particle filter/HMM/CRF), which exploits map information to correct the inertial trajectory. The proposed system introduces a reverse link between the two layers. It augments the functionality of the map matching component to learn context-specific parameters, which are then fed back to the underlying PDR layer. Although in this chapter we demonstrate the efficacy of lifelong learning in tuning parameters of the proposed robust pedestrian dead reckoning algorithm (R-PDR), the idea of lifelong learning is broadly applicable to improve the performance of any PDR implementation that requires parameter tuning.

This chapter is structured as follows. Sec. 6.1 proposed the system architecture of the proposed lifelong learning approach. Sec. 6.2 discusses the general framework for learning PDR parameters and environment features. Sec. 6.3 presents several illustrative examples of parameter learning, including sensor bias, heading bias, step length, and environment fingerprints. Sec. 6.4 evaluates the performance of the proposed lifelong learning approach through extensive real world experiments. Sec. 6.5 summarizes this chapter and illustrates the benefits of lifelong learning.

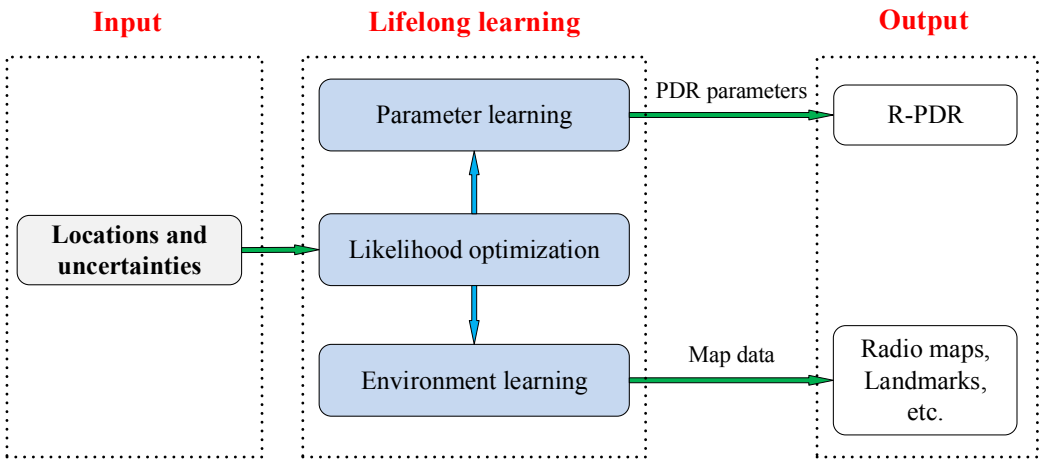


Figure 6.1: System architecture of the proposed lifelong learning approach.

6.1 System Architecture

The inertial tracking system generally needs careful tuning of parameters such as sensor bias and step length estimation parameters, to name a few. The fact that certain parameters are context-specific, i.e. dependent on the environment, device or individual, poses a great challenge to the widespread application of inertial tracking systems.

In addition, the signal variations in indoor environments, e.g. WiFi signal, magnetic distortion, etc. are strong indications of locations. However, the construction of the signal maps is key to leveraging these signal variations for positioning. Therefore, the automatic learning of the signal maps rather than labor intensive manual collection is both significant and challenging.

In this chapter, we propose a lifelong learning system, as shown in Fig. 6.1. Generally speaking, the lifelong learning component is a feedback loop, wherein the output of map matching is used to fine tune the parameters of the proposed R-PDR algorithm and simultaneously build the signal/landmark maps. Specifically, the learning takes place in three steps. To start with, the lifelong learning component takes the locations and uncertainties estimated from the map matching component as the input and estimates the likelihood of the matched trajectory given the observations (PDR trajectory) through optimization in an iterative manner. Secondly, parameters and maps can be derived when the likelihood has been optimized. The last step for the lifelong learning component is to feed the learned PDR parameters to the R-PDR component and store learned map data to the database. The learning process is discussed in detail in Sec. 6.2. Illustrative examples on how lifelong learning benefits

the whole positioning system is discussed in Sec. 6.3.

6.2 Proposed Learning Strategy

The accuracy of R-PDR depends on a number of parameters employed in its four steps. These are hard to tune manually because they may depend on the device, attachment, user and environment. To address this issue, we propose to use an unsupervised approach to learning R-PDR parameters. This approach is triggered every time the map constraints are such that there is one and only one matched trajectory that can best support the raw trajectory with high probability. This guarantees the existence and uniqueness of the matched trajectory solution. When this condition is met, R-PDR parameter learning becomes the following optimization problem:

$$\boldsymbol{x}^* = \underset{\boldsymbol{x}}{\operatorname{argmax}} \ln p(\boldsymbol{S}(\boldsymbol{x})|\boldsymbol{Z}(\boldsymbol{x})), \tag{6.1}$$

where $\boldsymbol{x}$ is the parameter (or vector of parameters) of R-PDR (or indeed another PDR implementation) that requires tuning, $\boldsymbol{S}(\boldsymbol{x})$ is the matched trajectory and $\boldsymbol{Z}(\boldsymbol{x})$ is the raw trajectory fed from R-PDR to map matching. The key idea is that only when the estimated parameter values are the same as (or very close to) the correct ones, can we maximize the conditional probability of the matched trajectory given the raw trajectory.

Generally, the solution to this optimization can be obtained with the expectation maximization (EM) approach by iterating the forward-backward algorithm [216]. However, the soft EM approach does not work for the learning of step length parameters because the optimization of step length parameters $\boldsymbol{x}$ in the M-step actually changes the state space of the model, which makes the E-step unable to evaluate the expectation in the next iteration [216]. Therefore, the hard EM approach, also known as Viterbi training is employed to solve the optimization problem [217] for the learning of step length parameters, details in Sec. 6.3.

6.3 Illustrative Learning Examples

We now show a couple of examples that demonstrate the effectiveness of the feedback loop between R-PDR and map matching for parameter tuning.

6.3.1 Learning of Sensor Bias

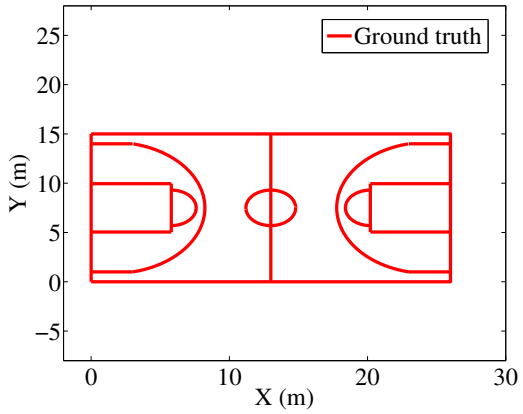
The learning of sensor bias, especially for low-cost sensors, is crucial to the performance of PDR algorithms. The motion sensor bias is the bottleneck that stops PDR algorithms from being widely used because the sensor bias is accumulated, leading to significant errors in the trajectory, especially when used in open space where no additional constraints can be applied. To make things worse, motion sensors also suffer from both time drift and thermal drift, which makes the bias vary with time and temperature. Therefore, the *lifelong* learning of sensor bias is essential to accurate PDR tracking. In practice all three motion sensors available, including accelerometer, magnetometer, and gyroscope, have drift or error. However, of all three motion sensors, the gyroscope plays the key role in terms of determining the accurate orientation of the device, which builds the foundation for long-term tracking. In addition, the calibration of gyroscope sensors is especially difficult compared to the calibration of accelerometers and magnetometers. Therefore, the learning of the gyroscope bias is an essential capability.

The key idea of learning the gyro bias is that only when the estimated heading is the same as (or very close to) the real heading can we maximize the conditional probability $p(\boldsymbol{S}(\boldsymbol{x})|\boldsymbol{Z}(\boldsymbol{x}))$ in Eqn. (6.1). Since the orientation of the device $\boldsymbol{q}(\boldsymbol{\omega}, \boldsymbol{b})$ can be easily derived from the gyro bias $\boldsymbol{b}$ and the angular velocity $\boldsymbol{\omega}$ [86], the learning of gyro bias can be achieved by optimizing

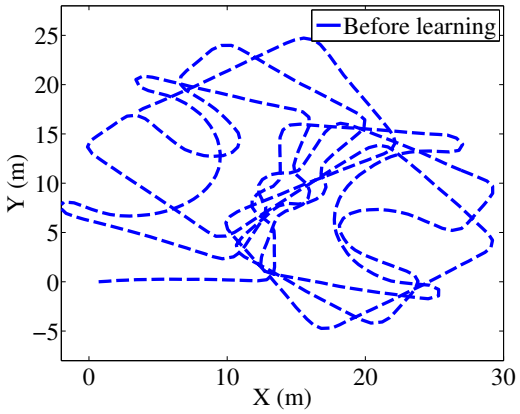
$$\boldsymbol{b}^* = \underset{\boldsymbol{b}}{\operatorname{argmax}} \ln p(\boldsymbol{S}|\boldsymbol{Z}(\boldsymbol{q}(\boldsymbol{\omega}, \boldsymbol{b}))), \tag{6.2}$$

in which the formulation of $p(\boldsymbol{S}|\boldsymbol{Z}(\boldsymbol{q}(\boldsymbol{\omega}, \boldsymbol{b})))$ has been discussed in detail in Chapter 5. Since MapCraft uses conditional random fields to perform map matching, this optimization problem can be solved by iterating the forward-backward algorithm [216].

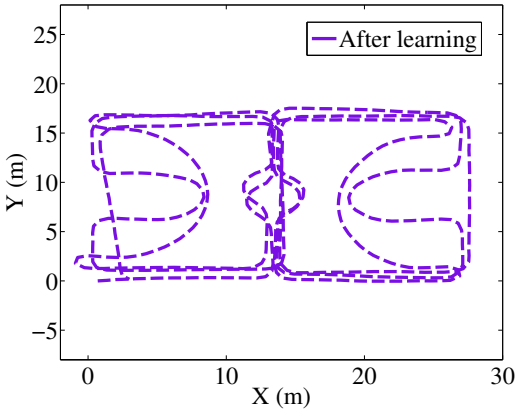
We have conducted experiments to test the effectiveness of the sensor bias learning. Fig. 6.2 shows the trajectory with one Nexus 5 device in an indoor basketball court without any floorplan or other map constraints. It is observed that the trajectory largely deviates from the ground truth without sensor bias learning. It is greatly improved in terms of heading estimation after the sensor bias of this device has been learned in a completely different (office) environment shown in Fig. 6.9. This example serves to show how lifelong learning can exploit one environment's structure to benefit navigation in more challenging open space environments. The implications of cross-environment learning can extend to other users: the high tracking accuracy of a user with "lifelong learning experience" can be exploited to create an accurate radio map



(a) Trajectory ground truth



(b) Trajectory before bias learning



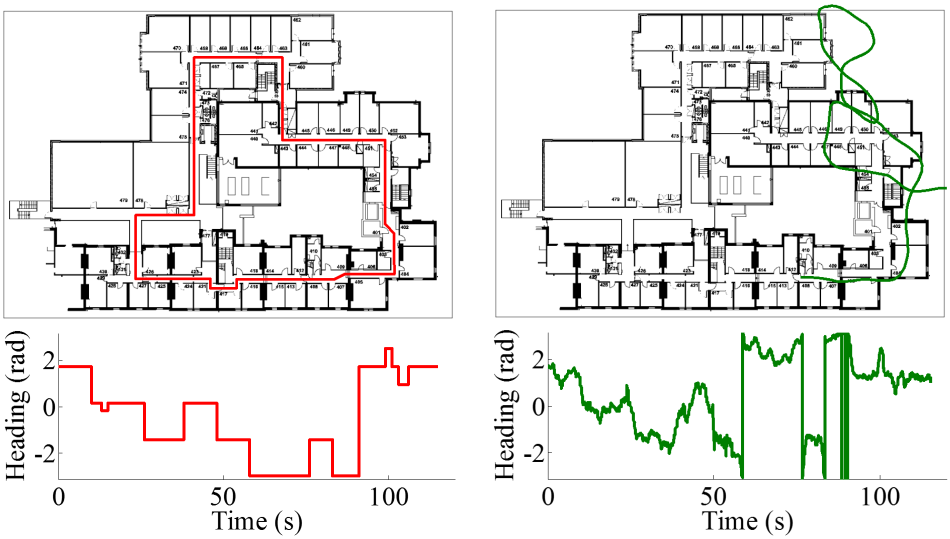
(c) Trajectory after bias learning

Figure 6.2: Inertial trajectories of 0.5 km in a basketball court without any map constraints before and after sensor bias learning, showing an improvement in loop closing loop error from 17.3m to 2.3m, and demonstrating the importance and effectiveness of the learning algorithm. Note that the bias parameters were learned in a different experiment site with floor plan constraints.

of the environment, which can in turn benefit other algorithms like WiFiSLAM [179] and users tracking with only radio maps.

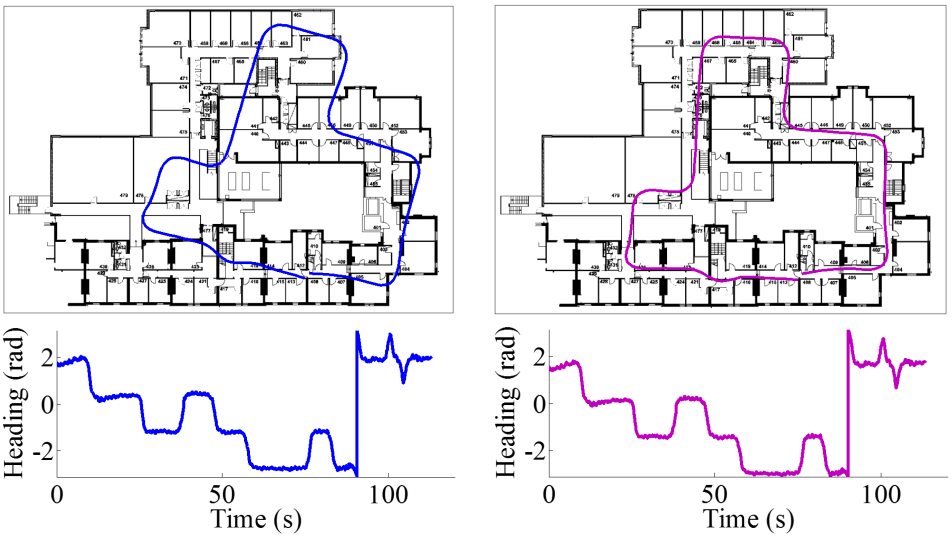
6.3.2 Learning of Heading Bias

The heading parameter learning, or specifically the initial heading bias (denoted with Δh) learning is much simpler than the learning of step length parameters, by fusing gyroscope and magnetometer data. Gyro sensors have high accuracy within a short time period while suffer significantly from long-term bias because of thermo-mechanical events. Magnetometers are the converse without long-term drift but lack of short-term accuracy due to the soft or hard iron effect. Therefore, the two types



(a) Ground truth

(b) Magnetometer



(c) Gyroscope

(d) Initial heading bias compensation

Figure 6.3: Estimated trajectories and headings from a) ground truth, b) gyroscope, c) magnetometer, and d) initial heading bias compensation.

of sensors can compensate each other to offer highly accurate heading estimations.

However, the initial heading can only be estimated from the magnetometer and thus a large initial bias might be introduced. To make things worse, we have no other information from the map constraints before map matching algorithm converges, e.g. at the beginning, to correct the initial heading estimation. The initial heading is crucial because it impacts on all heading estimations in the future. Therefore, it is necessary to learn the initial heading bias as soon as possible and use this learned

value to improve the performance of the CRFs-based tracking algorithm proposed in Chapter 5.

The proposed approach to learning the initial heading bias is to minimize the difference between headings estimated from gyroscope and magnetometers. After the system has worked for a period of time, we have both magnetometer and gyroscope readings. Then the initial heading bias can be easily learned by solving the following optimization problem.

$$\Delta h^* = \underset{\Delta h}{\operatorname{argmax}} \sum_{t=1}^{t'} (\theta_t^{\text{gyro}} - \theta_t^{\text{mag}} - \Delta h)^2, \tag{6.3}$$

where Δh is the heading bias at time $t = 0$, t' is the time the pedestrian has walked before the bias estimation[1], θ_t^{gyro} and θ_t^{mag} are the heading estimated from gyro sensors and magnetometers at time t, respectively.

The estimated initial heading bias is fed into the unscented Kalman filter mentioned in Chapter 4 to make accurate heading estimations. We have also conducted experiments in the office test site to show the effectiveness of the initial heading bias compensation mechanism. Starting from an arbitrary location in the office test site, a pedestrian walked a trajectory shown in Fig. 6.3(a). Figs. 6.3(b) and 6.3(c) show the resulting heading estimations and trajectories with heading derived from magnetometer only and gyroscope only, respectively. It is observed that the trajectory is extremely noisy but immune from long-term drift with only magnetometer data. In comparison, the trajectory with gyroscope data is very clean during the experiment period but suffers significantly from the initial heading bias. Besides, the accuracy of this trajectory is also weakened by long-term drift near the end of the trajectory. Fig. 6.3(d) shows the headings and trajectories after the initial heading bias has been learned. It is observed that after the initial heading bias learning and correction, the accuracy of the trajectory is greatly improved. It is later shown in Chapter 7 that the initial heading bias learning mechanism can not only improve the tracking accuracy, but also greatly speed up the convergence of the map matching algorithm even in the absence of WiFi data.

6.3.3 Learning of Step Length

This example shows how we can use the feedback loop between R-PDR and map matching to improve the performance of R-Step Length Estimation, the third step of

[1]Empirically 30 $\sim$ 90 seconds in our experiments.

R-PDR. In particular, we learn the step constant γ (Eqn. (4.5)) for different individuals because 1) the average step length plays a crucial role in the tracking accuracy due to the high consistency of step length in human walking patterns; and 2) the parameters α and β are very similar for different individuals in our experiments.

The step length constant learning is achieved in the same way as the learning of the gyro bias by replacing learning variable from $\boldsymbol{b}$ in Eqn. (6.2) to the step length constant γ.

The general solution to this optimization problem can be obtained with the expectation maximization (EM) approach. However, the feature functions shown in Sec. 5.3.2 cannot be used in learning the step length constant. We take the first feature function (Eqn. (5.5)) as an example to explain this in detail. In essence, this feature function is the product of the probability of the estimated step length given the graph edge length $p(Z_t^l|S_{t-1}, S_t)$ and the probability of the estimated heading given the heading of the map graph edges $p(Z_t^\theta|S_{t-1}, S_t)$. The log-likelihood in Eqn. (6.1) is maximized when the estimated step length is the same as the graph edge length because the item $p(Z_t^l|S_{t-1}, S_t)$ keeps the maximum for each state transition during the whole state sequence while the item $p(Z_t^\theta|S_{t-1}, S_t)$ is not significantly affected.

As a result, it is necessary to change the feature functions in Eqn. 5.5 to learn the step length constant. The essence of the step length parameter estimation is to find a γ that best supports the heading observations given the underlying graph of the map. To find this optimum γ, we start with an initial γ and equally divide the whole training trajectory into segments with the same length as the edge of the graph. Then a new feature function taking only the heading observation into account is defined as

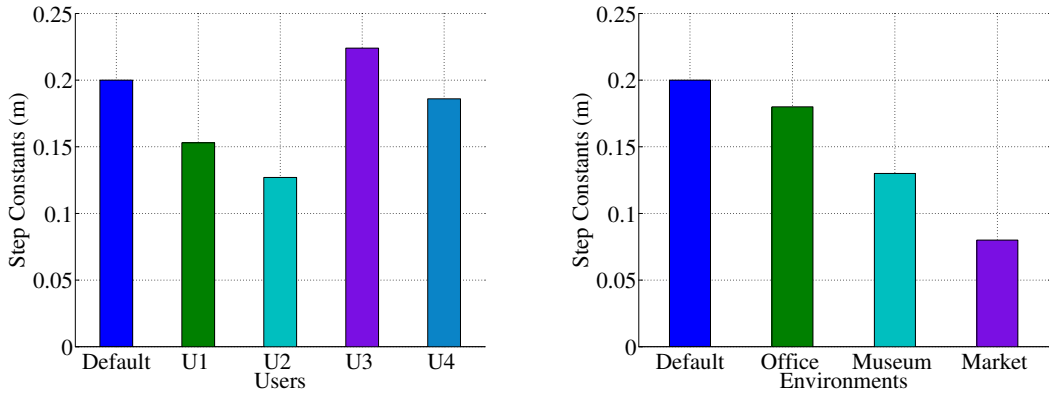
$$f_t(S_{t-1}, S_t, Z_t^\theta) = \mathcal{N}(Z_t^\theta, \theta(S_{t-1}, S_t), \sigma_\theta^2), \tag{6.4}$$

where Z_t^θ is the heading observation at time t, $\theta(S_{t-1}, S_t)$ is the orientation of the edge from state S_{t-1} to state S_t, and σ_θ^2 is the heading observation variance.

To solve the optimization problem in Eqn. (6.1), we first consider the (soft) conditional EM approaches which optimize the model parameters in two steps:

$$\begin{aligned} \text{E-step:} \quad & p(\boldsymbol{S})^i = \underset{\boldsymbol{S}}{\operatorname{argmax}}\, p(\boldsymbol{S}|\boldsymbol{Z}, \gamma^{i-1}), \\ \text{M-step:} \quad & \gamma^i = \underset{\gamma}{\operatorname{argmax}}\, E_{p(\boldsymbol{S})^i}\left[\ln\, p(\boldsymbol{S}|\boldsymbol{Z}), \gamma\right], \end{aligned} \tag{6.5}$$

where λ is the parameter to be estimated and the superscript i indicates the current iteration step.



(a) Step constants of different pedestrians in an office environment.

(b) Step constants of the same pedestrian in different environments.

Figure 6.4: Experiments results showing the importance and effectiveness of the proposed lifelong learning approach on step length estimation.

Unfortunately the (soft) EM approach in Eqn. (6.5) does not fit our model formulation. It is apparent that the expectation in the M-step can only be evaluated when the state space from the E-step $p(\boldsymbol{S})^i$ remains the same during the optimization process. However, the goal of the M-step is to find a γ^i that maximizes the log-likelihood. Therefore, γ^i experiences many different values during the optimization process in the M-step. From our training model formulation described above, different γ lead to training sequences of different lengths and thereby different state spaces for the state sequence $\boldsymbol{S}$. This contradicts the condition that the state space must remain the same for the expectation maximization and thus the algorithm fails.

Fortunately, it is justified in [217] that the most likely value of the state sequences given the model and the observed data, which is the result of the inference step, also maximizes the conditional likelihood. Therefore, by replacing the probability distribution in Eqn. (6.5) with maximization, we transform the (soft) conditional EM to hard conditional EM, also known as Viterbi training. Then the hard EM is the process of iterating over the following two steps:

$$\begin{aligned} \text{E-step:} \quad & \boldsymbol{S}^i = \underset{\boldsymbol{S}}{\operatorname{argmax}}\, p(\boldsymbol{S}|\boldsymbol{Z}, \boldsymbol{\lambda}^{i-1}), \\ \text{M-step:} \quad & \boldsymbol{\lambda}^i = \underset{\boldsymbol{\lambda}}{\operatorname{argmax}} \ln\, p(\boldsymbol{S}|\boldsymbol{Z}), \end{aligned} \tag{6.6}$$

We have conducted experiments in an office environment, a museum environment, and a market to show the effectiveness of the feedback loop on step length estimation. Notice in Fig. 6.4(a) that the step constants γ are very different for different pedestrians (U1 to U4), and very different even for the same pedestrian in different

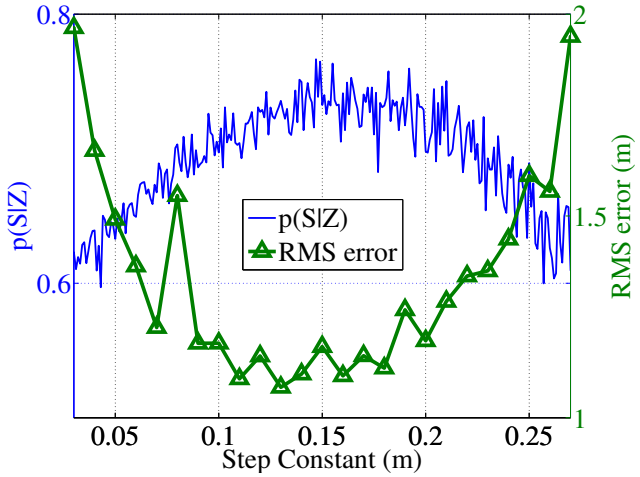


Figure 6.5: The step constant that maximizes $p(\boldsymbol{S}|\boldsymbol{Z})$ in Eqn. (6.1) minimizes the tracking error.

environments (office, musuem, and market) as well, as shown in Fig. 6.4(b). Fig. 6.5 shows that the step constant which maximizes the conditional probability $p(\boldsymbol{S}|\boldsymbol{Z})$ in Eqn. (6.1) also minimizes the RMS error of tracking, which demonstrates the effectiveness of the proposed algorithm in estimating the step constants.

6.3.4 Learning of Environment Maps

In addition to the device or individual-specific parameters, there are also environment specific parameters including radio and magnetic distortion maps. However, building these maps is a very time consuming and labour intensive operation, typically requiring manual surveying and sampling. In addition, RSS maps dynamically vary over time, and thus require constant updating. Traditionally, these maps usually can be obtained via either manual mapping [111] or SLAM [179]. However, the manual mapping is extremely labor intensive, which makes it impractical to be widely adopted. Meanwhile, the SLAM approaches are either unable to provide global sensible maps for localization or require the manual setting of some landmarks [175] and fingerprinting points [179].

Compared with existing mapping techniques, the proposed lifelong learning algorithm can globally map the environment-specific parameters with zero-effort from users. Based on the estimated trajectories from the map matching algorithm, we can then build the radio map or magnetic distortion map easily. These radio maps, in return, can be used by either R-PDR or map matching layer for later users to speed up the convergence of positioning.

It is well-known that the magnetic distortions can be used as the unique signatures of locations in indoor environments because of the metallic objects ubiquitous in indoor environments greatly distort the magnetic field of the earth and make it over locations. However, the construction of the distortion map is very labor intensive, which obstructs its widespread application. Therefore, the automatic learning of the magnetic distortion is also crucial.

The learning of the magnetic distortions is similar to the learning of radio maps. The magnetic distortion map can be used as unique signature for localization, similar to radio maps. In addition, together with the radio map, the magnetic distortion map can also improve the heading estimation accuracy of PDR, especially the initial heading.

6.4 Evaluation

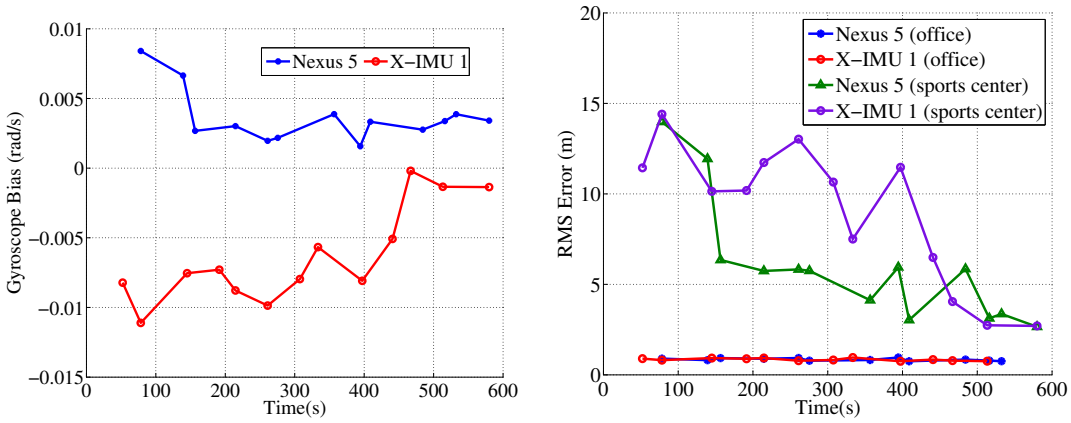
This section evaluates the performance of the proposed lifelong learning component through the learning of typical parameters including sensor bias, heading bias, step length, and the environment.

6.4.1 Experimental Setup

The lifelong learning component is evaluated in three real-world settings with known floor plans: an office building ($65 \times 35m^2$), a museum ($109 \times 89m^2$), and a sports center hall ($30 \times 20m^2$). To test the effectiveness of learning, 15 people of different genders, heights, and ages, and different types of mobile devices are taken into account. To provide accurate ground truth, numbered labels were placed along corridors and within rooms on a 2 m grid. During the experiments, the test subject always held one camera at hand. Labels were filmed at the same time that experiments were conducted. The time-synchronized video streams were then mapped to locations on the floorplan, and intermediate locations interpolated using footstep timing, also obtained from the video.

6.4.2 Performance of Sensor Bias Learning

The first set of experiments was designed to evaluate the performance of the sensor bias learning algorithm. Five different devices with different sensor biases, including Nexus 5, Nexus 4, Samsung S4, X-IMU 1 and X-IMU2 are involved in the experiments. To minimize the impact from other factors, we fixed other variables like the



(a) Bias learning results

(b) Impact of bias learning on error

Figure 6.6: Tracking accuracy with different devices and attachments, showing the effectiveness of the lifelong bias learning algorithms. The bias was learned in the office environment for a period of time, and then tested both in the same office and a completely different sports centre environment.

experiment site (office environment), users (User 1), and attachment (handheld). It is shown in Fig. 6.6(b) that the tracking accuracy for different devices is remarkably stable after each device has been used for $5 \sim 10$ minutes. The trend is especially apparent for the devices used in open spaces, like the sports centre, where no map constraints are available to improve the tracking accuracy. It is observed that after walking for 0.5km in the basketball court, the RMS tracking error can be as high as 15m without the bias learning algorithms while the error can decrease to around only 2m if the gyro bias has been learned in the office environments before it is used in the basketball court. In addition, in the office environment where the map matching can compensate for the sensor bias, the sensor bias learning still improves the tracking accuracy by over 20% (from 0.96m to 0.75m). Note that though the sensor bias can only be learned in environments where maps are available, the learned sensor bias can be used in environments without map constraints afterwards.

Another way of evaluating the performance of the proposed bias learning algorithm is to compare the estimated gyroscope bias with the measured bias. There are two approaches of measuring gyroscope bias. One is rotating tables which provides the ground truth for the angular velocity. Then the angular velocity from gyroscope can be compared with the ground truth to obtain the real bias involved. This real bias can then be compared with the bias estimated using the proposed algorithm. A rotating table is expensive and not widely accessible. The other much simpler approach is to place the gyroscope on a static table and obtain the raw angular velocity directly

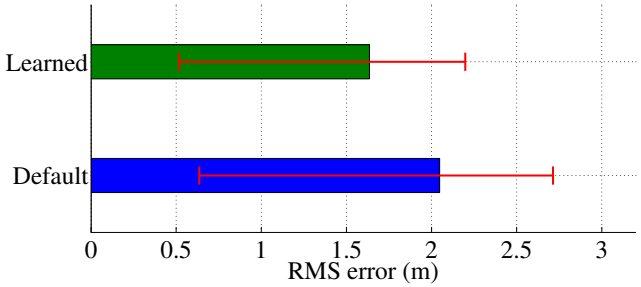


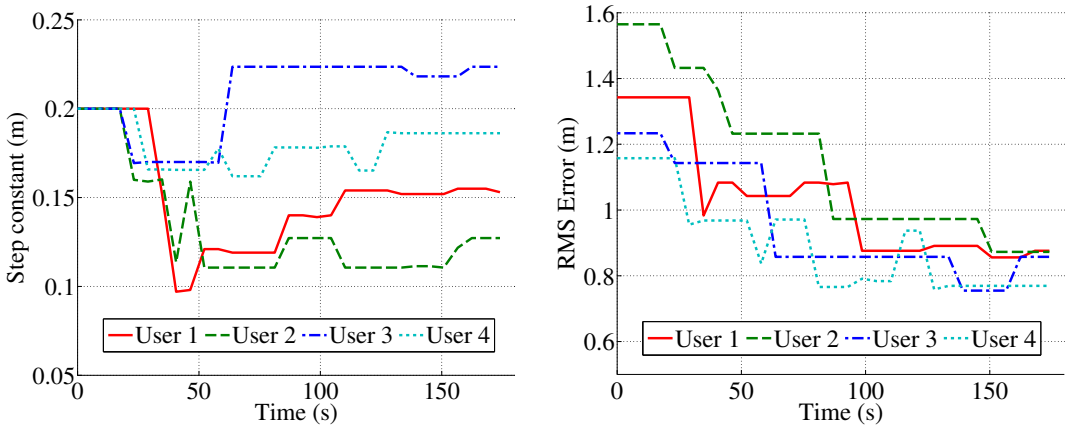
Figure 6.7: The RMS error (WiFi-free) with and without initial heading bias learning.

from the gyroscope. The angular velocity should be zero in this case. Therefore, the angular velocity obtained in this approach is the bias of the gyroscope. This approach can only measure the bias of gyroscope in static conditions which is different from the bias in dynamic conditions. In addition, it is hard to eliminate the impact of the environment which was observed during the experiments. As an alternative, this thesis directly compares the reconstructed trajectories with the ground truth trajectories to see how the learning of gyroscope can gradually approach the real bias value and improve the tracking accuracy.

It should be noted that this thesis only provides an intelligent way of learning gyroscope bias. However, the gyroscope bias is dependent on many environmental factors like temperature, humidity, etc. Therefore, the time stability of the bias needs to be taken into account with extra care because the bias of the same gyroscope can be slightly or even significantly different in different environments at different times. The experiments discussed in this thesis were all conducted on the same day. Further experiments in different weeks, months, or even years need to be conducted to test the time stability of the gyroscope bias but this is out of the scope of this research.

6.4.3 Performance of Heading Bias Learning

WiFi-free experiments have also been conducted in the office environment to test the performance of the proposed initial heading bias learning algorithm. We have taken trajectories with an average length of 100 meters starting from 1000 different locations randomly selected from the office environment. Then the RMS errors of these trajectories were calculated with and without the initial heading bias learning. The RMS errors along with 15 and 85 percentiles are shown in Fig. 6.7. It is observed that the initial heading bias learning algorithm is proved to improve the RMS error of the tracking by over 20 percent.



(a) Step learning results

(b) Step learning performance

Figure 6.8: The improvement of tracking accuracy over time with the learning of step length.

6.4.4 Performance of Step Length Learning

This set of experiments was designed to test the performance of the step length learning. To minimize the influence of other factors, we fixed the experiment site to be the office environment, the device to be Nexus 5 phone and attachment to be handheld only. Different step constant parameters are learned for different pedestrians. Fig. 6.8(a) shows some typical step constants for different people in the office environment and the time taken to learn these parameters. Fig. 6.8(b) shows how the tracking performance improves with the convergence of the parameter learning.

6.4.5 Performance of Environment Learning

The learning of environment maps, including the radio and magnetic distortion maps is straightforward. Experiments in the office environments have been conducted to test the accuracy of the learned radio map. We first manually collected the radio signal strength from one access point in the corridor of the office environment. Then the radio map along the same corridor was constructed automatically by the proposed learning algorithm. Fig. 6.9 compares the radio map learned and the one manually built to test the accuracy of the proposed environment learning technique. It is observed that the learned map is highly consistent with the one manually built.

The magnetic distortions, in terms of both magnitude and heading, can also be automatically learned from the environments. Figs. 6.10(a) and 6.10(b) show the magnitude and heading distortion of the magnetic field of the earth in an office building.

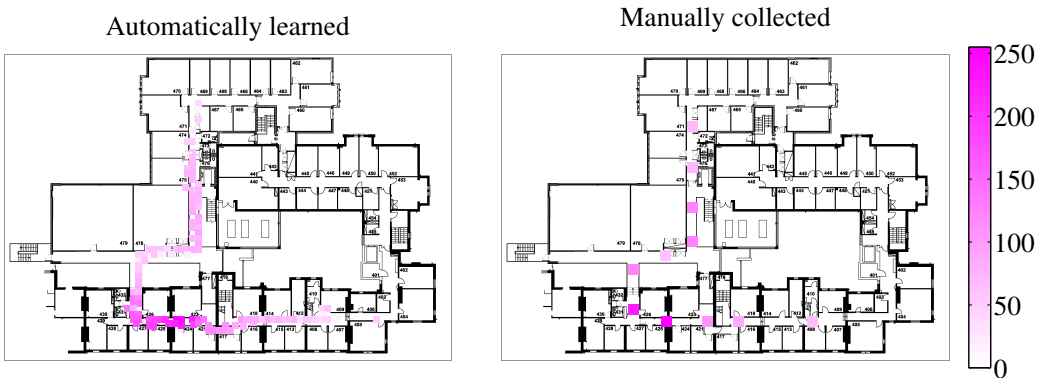
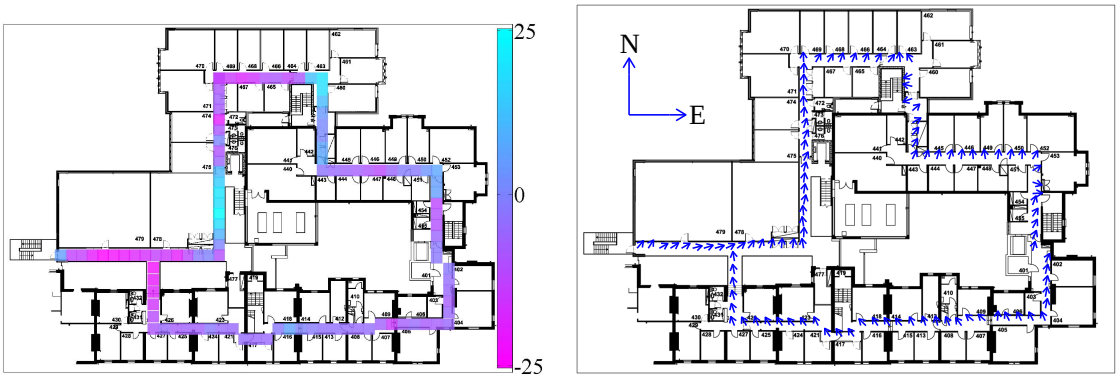


Figure 6.9: Radio maps automatically learned (left) and manually collected (right).



(a) Magnetic magnitude distortion learned.

(b) Magnetic heading distortion learned.

Figure 6.10: Experiments results showing the effectiveness of the lifelong learning component in learning environment maps.

It is observed that the magnetic field of the earth suffers from significant distortion in indoor environments, which can then be easily used as the fingerprinting of locations.

6.5 Discussion

This chapter has demonstrated the merits of the lifelong learning algorithm.The lifelong learning component is a novel approach to learn various parameters in indoor positioning, which can optimize the performance of the positioning system: it is a simple yet general approach that can be applied to any positioning system implementation. Unlike existing work, where lifelong learning has only been used to optimize the map matching layer, in this chapter we generalise this idea to optimize the entire system, reaping the fruits of cross-layer optimisation. This component learns PDR parameters in an unsupervised manner and thus requires zero user effort. Notably, we have illustrated cases in which its operation in one environment can boost its perfor-

mance in a new and more challenging environment. The idea of lifelong learning has widespread application, as it can be used with a wide variety of sensors and different types of maps. The future plan is to learn a wider variety of parameters, and to study in depth the sensitivity of each parameter to context (user, device, attachment, or environment) variability. The next step is to incorporate algorithms that detect context changes, and select appropriate parameter settings.

Online SLAM algorithms, e.g. DP-SLAM [184] can also learn parameters including the map of the environments. DP-SLAM implements what is essentially a simple particle filter over maps and robot poses. However, with the notion of particle ancestry, it uses a technique called distributed particle mapping (DP-Mapping), which enables it to maintain a large number of maps very efficiently. Compared with other SLAM approaches, the biggest advantage of DP-SLAM lies in the fact that the computational complexity is only $O(ADP\lg P)$ where A is the number of grid squares, D is the ancestry depth, and P is the number of particles, which is significantly lower than normal particle filters. However, DP-SLAM still has two disadvantages. Firstly, DP-SLAM requires loop closing in certain scenarios. Moreover, the computational complexity is still too high for mobile applications.

Having introduced all four components of the system, it is time to evaluate the performance of the whole indoor positioning system. The next chapter will discuss the implementation of the whole system and show how the proposed system meets the four key requirements of a practical system, low-cost, accurate, robust, and scalable, by means of extensive real world experiments.

Chapter 7

Evaluation

The proposed indoor positioning system has been implemented on the Android system, and tested in various experimental settings. Recall that the **key objectives** of a practical indoor positioning system are low-cost, accurate, robust, and scalable. The proposed system has demonstrated itself to be **low-cost** for using only existing infrastructure and mobile devices. In addition, since the proposed system relies only on the floor plan and does not require any time-consuming and labor-intensive site survey, it is extremely **scalable** so that it can be quickly and easily deployed in a new environment. In order to demonstrate that the proposed system is a practical indoor positioning system, this chapter shows that how it is also **accurate** and **robust** through extensive real-world experiments.

This chapter is organized as follows. Sec. 7.1 introduces the experimental setup of the extensive real world experiments to evaluate the performance of the whole indoor positioning system. Sec. 7.2 evaluates the accuracy of the proposed system and compares it with state-of-the-art competing approaches. Sec. 7.3 evaluates the robustness of the system. Sec. 7.4 summarizes this chapter with discussions on the performance of the whole indoor positioning system.

7.1 Experimental Setup

This section describes the setup of the experiments, including experimental sites, participants, devices, etc. The state-of-the-art competing approaches of the proposed algorithms are also introduced.

7.1.1 Sites

To demonstrate the real world applicability of the proposed indoor positioning system, LL-Tracker is evaluated and compared against competing approaches in three real-world settings, namely an office building, a museum, and a sports center. All of these have different floor plans as shown in Figs. 7.6, 7.7, and 7.8, and methods of construction which affect the obtained sensor data. The office environment ($65 \times 35m^2$, where the majority of the tests have been conducted) is a multi-storey office building with a stone and brick construction, reinforced with metal rebars - testing was conducted on the fourth floor. The museum ($109 \times 89m^2$) is a multi-storey stone building with large, open spaces. Testing was conducted on the ground floor. The experiments in the sports center ($30 \times 20m^2$) were mainly conducted in a basketball court where no map constraints are available. Construction is mainly brick and mortar, with a metal roof. 224 trajectories of average length over 400 m were collected over 30 days. Specifically, 146 trajectories were taken from the office building, 46 from the museum, and 32 from the sports center. Positioning error is expressed in [m] RMS, as justified in Sec. 1.2.

7.1.2 Participants

The variations between different people are taken into account by acquiring data from 15 people of different genders, heights, and ages. During the experiments, the subjects are mounted with several devices simultaneously in different parts of the body, typically hand, watch, glasses, shirt pocket, and trousers pocket. Then they walk anywhere in the building without planned routes, to realistically capture real pedestrian motion, rather than artificial, constant speed trajectories. They are told to move freely and may have different motion modes including walking, standing still, sitting down, bending to pick up something on the floor, etc.

7.1.3 Devices and Implementation

Since the proposed indoor positioning system does not rely on extra infrastructure or devices, the only hardware that was used to build the system is the mobile devices people use, e.g. smartphones, pads, laptops, etc. Generally speaking, the hardware variations only have slight impact on the tracking accuracy. One key difference is the sampling rate of the IMU sensors (accelerometer, magnetometer, and gyroscope). Normally, the sampling rate of the gyroscope cannot be under 50 Hz if we use the angular velocity provided by gyro sensors to get the relative rotation. Meanwhile,

to obtain the offset of the mobile device by means of double integration over acceleration data, the sampling rate of accelerometer cannot be under 50 Hz. Therefore, gyroscopes and accelerometers of devices on which we have deployed the whole system have a minimum sampling rate of 50 Hz. Device models on which we have implemented the proposed system includes LG Nexus 4, LG Nexus 5, Asus Nexus 7, Samsung Nexus S, Samsung Galaxy S II, Samsung Galaxy S III, and Samsung Galaxy S IV. We have also implemented the system in different devices of the same model, e.g. four different Asus Nexus 7, two different Samgsung S IV, and two different LG Nexus 4. These devices differ greatly in terms of sensors, functionality and price. But one thing in common is that they all run the Android operating system no earlier than version 2.3.6. The experiment results indicate that different phone models only have very slight impact on the tracking accuracy.

7.1.4 Ground Truth

To provide accurate ground truth, a vision-aided system, along with both automatic and manual image recognition have been utilized in the following steps.

Firstly, numbered labels were placed along corridors and within rooms on a 2 m grid. The coordinates of these labels were manually measured using laser range finder (mm accuracy). These labels are then accurately pinned on the floor plan.

Secondly, during the experiments, the test subject always held one camera in their hand which points to the ground. These labels were filmed at the same time experiments were conducted.

The time-synchronized video streams were then mapped to locations on the floorplan. Specifically, those moments when the test subjects have their feet on or above the labels were first identified – the location of the label serves as the ground truth location of the test subject at this moment. These moments were identified both automatically and manually to ensure the accuracy of the ground truth to be at centimeter level.

7.1.5 Proposed Algorithms

In this section we evaluate the performance of two proposed and two competing end-to-end positioning solutions. The proposed ones are LL-Tracker and R-Tracker. In particular, *LL-Tracker* combines R-PDR and MapCraft with lifelong learning, i.e. with the feedback loop from map matching to fine tune the parameters of R-PDR.

On the other hand, *R-Tracker* consists of R-PDR and MapCraft without lifelong learning.

7.1.6 Competing Algorithms

Practical indoor positioning algorithms are required to be infrastructure-free or use existing infrastructures like WiFi and Bluetooth low energy (BLE). Existing practical algorithms fall into two categories: RF category where only WiFi/BLE are used and fusion category where inertial data and WiFi/BLE data are fused to perform positioning. Typical examples in the RF category are HORUS [113], RADAR [111], and EZ [218]. The state-of-the-art algorithms in the fusion category are S-Tracker [216], N-Tracker [80, 85], UnLoc [175], WifiSLAM [179], and the algorithm in [81]. To evaluate the performance of R-Tracker and LL-Tracker, This thesis selects the algorithm that reports the best accuracy from each category, HORUS from RF category, and S-Tracker and N-Tracker from the fusion category, as competing approaches. The details of these competing approaches are discussed below.

- **HORUS**: HORUS [113] is a fingerprinting-based localization approach which uses an existing radio map, e.g. RSS-location pairs denoted with $< \boldsymbol{R}_i, \boldsymbol{L}_i >$, $i = 1 \cdots N$. To localize mobile devices, HORUS 1) takes a set of RSS $\boldsymbol{r}$, 2) estimates the likelihood of this set given RSS-location pairs in the radio map as

$$p(\boldsymbol{r}|\boldsymbol{R}_i) = \prod_{j=1}^{M} p(r_j = R_{i,j}|L_i), \tag{7.1}$$

  where M is the dimension of r and $p(r_j = R_{i,j}|L_i)$ is the distribution of the RSS at location L_i (assumed to be Gaussian) and then 3) selects the location from the pair with the highest likelihood or a weighted mean of all locations based on the likelihood as the estimated location.

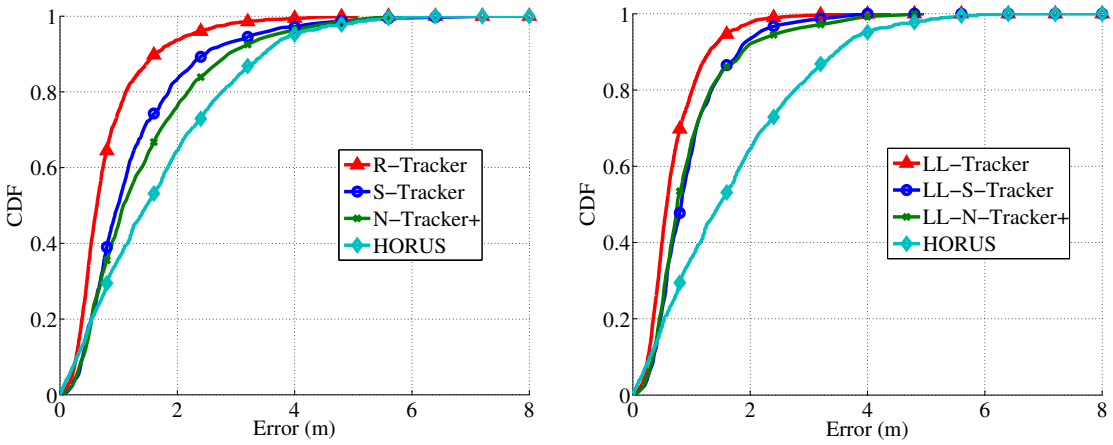
- **S-Tracker**: S-Tracker combines a strawman PDR algorithm referred to as S-PDR with MapCraft. S-PDR is based on two assumptions. The first assumption is that the user hold the mobile device at hand in texting mode. The second assumption is that the heading of the mobile device is always the same as the heading of the user. These two assumptions make S-PDR simple and efficient to implement and widely used. However, it has problems in terms of practicality because users are not likely to hold the mobile device all the time and keep the heading of the device the same as the heading of the walking. We

combine S-PDR with MapCraft to form the first competing approach in the fusion category: S-Tracker. To avoid penalising S-Tracker for its assumptions, we compare the performance of the proposed algorithms using a variety of mobile device attachments, to that of S-Tracker using a single device attachment (texting mode).

- **N-Tracker+**: N-Tracker+ combines a state-of-the-art PDR which we refer to as N-PDR, with the proposed MapCraft algorithm. N-PDR is a combination of Normalized Autocorrelation [80] and WalkCompass [85]. The normalized autocorrelation has reported the best step detection results compared with other existing approaches, as discussed in Sec. 4.3. However, since normalized autocorrelation cannot distinguish between real steps and other repetitive motions like nodding and tapping, we only evaluate the accuracy of N-PDR using trajectories without such repetitive motions. WalkCompass is a state-of-the-art heading estimation approach, but it has two limitations. The first limitation lies in the fact that WalkCompass has no mechanism to correct the long term drift of inertial sensors, which indicates that the performance of WalkCompass can be significantly degraded in long-term tracking. Since we have some fairly long trajectories (>2 km) which WalkCompass cannot manage, we enhance WalkCompass with the proposed robust orientation estimation to make it capable of working for long term tracking (called N-PDR+). The other limitation is that WalkCompass does not work well with devices attached to the shirt pocket or head, as the authors also suggested. Therefore, we only evaluate the performance of this approach with devices at hand or in the pocket. The combination of N-PDR+ and MapCraft makes N-Tracker+.

7.2 Accuracy

To evaluate the accuracy of the proposed tracking system, we have conducted experiments with 15 users, 5 different devices and 5 different attachments in the office and museum environments. The same IMU and WiFi data were fed into the proposed R-Tracker, LL-Tracker, and the competing approaches including S-Tracker, N-Tracker+, and HORUS. Fig. 7.1 compares the error cumulative distribution function of the four approaches without lifelong learning ability. It is observed from Fig. 7.1(a) that R-Tracker outperforms all other approaches, followed by S-Tracker, N-Tracker+, and lastly HORUS which is used as a baseline for tracking accuracy. Note that all PDR algorithms compared in this section are fused with the map matching algorithm used



(a) Lifelong learning disabled

(b) Lifelong learning enabled

Figure 7.1: Accuracy comparison of various tracking algorithms.

Table 7.1: Comparison of tracking systems in terms of accuracy (RMS [m]).

Lifelong Learning	PDR algorithms			HORUS
	S-PDR*	N-PDR+	R-PDR	
Disabled	1.61	1.78	1.07	4.62
Enabled	1.16	1.23	0.86	

* S-PDR is tested only with devices held by hand in texting mode.

in MapCraft to estimate the locations while HORUS only exploits knowledge of the radio map, but does not perform map matching.

The tracking accuracy of these algorithms is further improved when the lifelong learning capability is enabled, as shown in Fig. 7.1(b). The tracking errors of these combinations of approaches are compared in Table. 7.1. It can be observed that the lifelong learning ability can improve the tracking accuracy of not only the proposed approach from 1.07m (R-Tracker) to 0.86m (LL-Tracker), but also other PDR-based approaches from 1.61m (S-Tracker) to 1.16m (LL-S-Tracker), and from 1.78m (N-Tracker+) to 1.23m (LL-N-Tracker+). The reason lies in the fact that the proposed lifelong learning algorithm optimizes generic PDR parameters, e.g. sensor bias, step constant, etc. instead of algorithm-specific parameters.

LL-Tracker outperforms competing approaches because it combines robust pedestrian dead reckoning (i.e. the ability to distinguish between real and fake steps) with lifelong learning of R-PDR parameters. Unlike LL-Tracker, the performance of both S-Tracker and N-Tracker+ can be easily degraded under non-ideal experiment conditions. For example, motions of the test subjects like occasionally shaking the

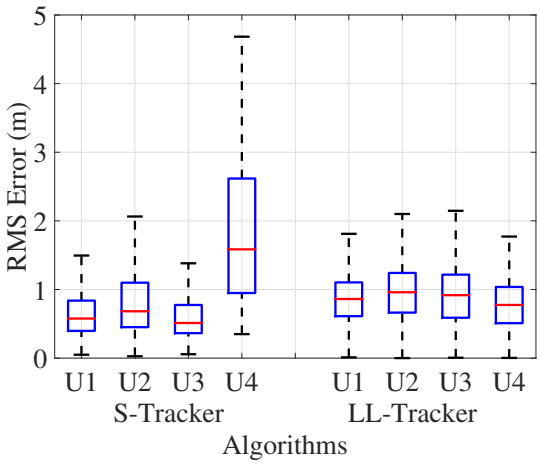


Figure 7.2: The robustness of the tracking system over user variability.

device, opening/closing the door, etc. generate signals similar to steps and thus lead to over counting of steps using these algorithms, which thereby degrades the tracking accuracy.

7.3 Robustness

The next set of experiments is designed to validate the robustness of the proposed LL-Tracker as we change the experimental conditions by varying users, devices, attachments, and environments.

7.3.1 User Variability

To evaluate the robustness to user variability, 15 different people with different genders, aged from 20 to 55, with height from 150 cm to 190 cm are involved in the experiments. To minimize the impact of other factors, we fixed the experiment site to be the office environment, the device to be Nexus 5 phone and attachment to be handheld only. Different step constant parameters are learned for different pedestrians. It is shown in Fig. 7.2 that the proposed LL-Tracker has almost the same RMS tracking error regardless of different users while the counterpart, MapCraft, has individual specific tracking errors.

To further demonstrate the robustness of LL-Tracker, we have performed experiments taking into account various motion behaviors and time-varying walking speed. A set of behaviors including walking, sitting down, standing up, standing still, accelerating, decelerating, bending down to pick up things on the floor were performed by the pedestrian. We can observe some of these behaviors from the acceleration shown

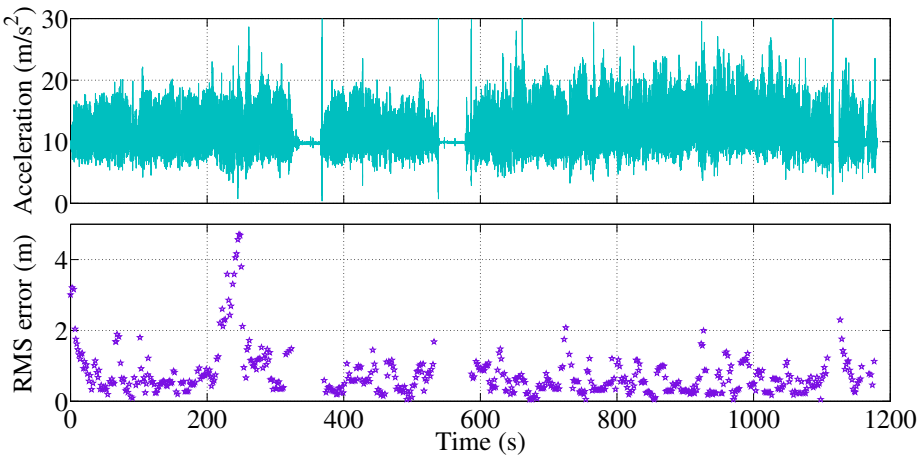


Figure 7.3: Tracking accuracy with various behaviors and varying walking speed.

at the top of Fig. 7.3 while the bottom of Fig. 7.3 shows the tracking accuracy over the whole trajectory. Note that the big tracking error shortly after 200 seconds is due to the deliberate sudden significant change of walking speed and random shaking of the device. In this case the learning algorithm takes around 20 seconds to converge again.

7.3.2 Device Variability

To evaluate the robustness over the device variability, five different devices, including Nexus 5, Nexus 4, Samsung S4, X-IMU 1 and X-IMU2 are involved in the experiments. Again, we fixed other variables like the experiment site (office environment), users (User 1), and attachment (handheld). It is observed that in the office environment where the map matching can compensate for the sensor bias, the sensor bias learning still improves the tracking accuracy by over 20% (from 0.96m to 0.75m).

7.3.3 Attachment Variability

Five typical attachments are tested in this experiment to evaluate the robustness of LL-Tracker, including handheld, watch, glasses, shirt pocket, and trousers pocket. This experiment is also conducted by User 1 with Nexus 5 phone in the office environment. The traveled distance in this set of experiments is over 12 km in total. The test subject had different motion modes during the experiments including walking, standing still, sitting down, bending to pick up something on the floor, etc. It is observed from Fig. 7.5 that the RMS errors of LL-Tracker are extremely similar for different attachments. Please note that MapCraft only works for handheld attachment.

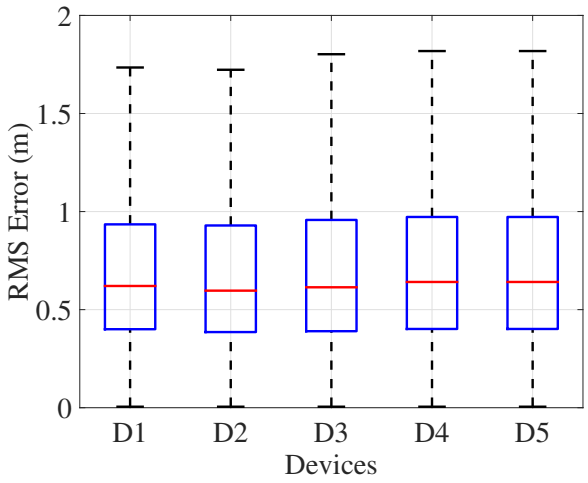


Figure 7.4: The robustness of the tracking system over device variability.

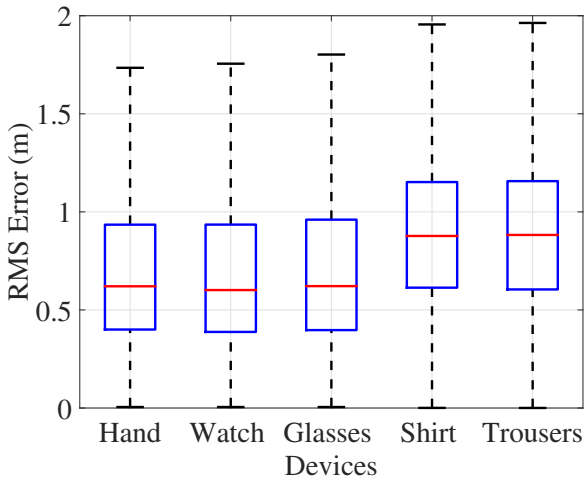
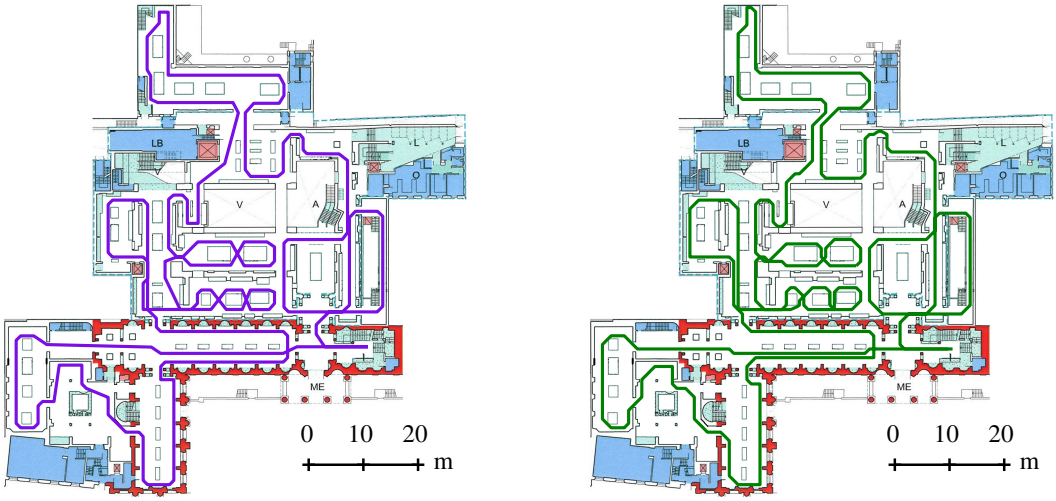


Figure 7.5: The robustness of the tracking system over attachment variability.

7.3.4 Environment Variability

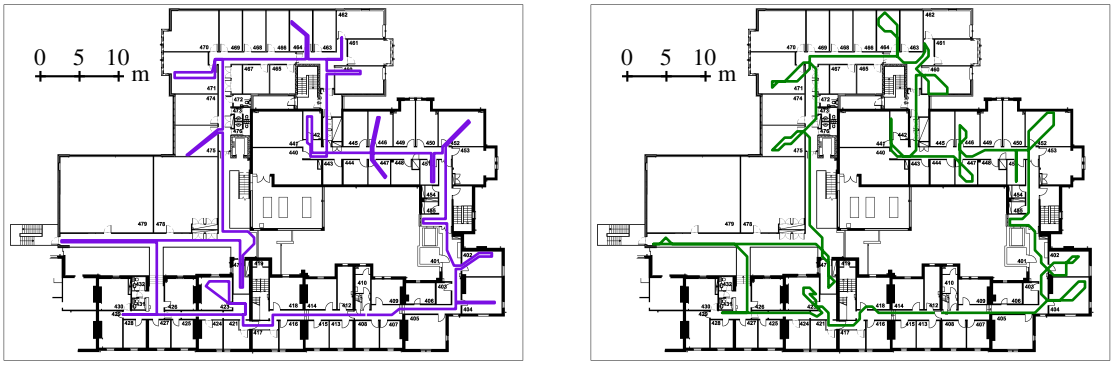
We then evaluate the performance of LL-Tracker in a variety of environments, namely an office environment, a museum, and a sports center. The experiments in the sports center were conducted in the basketball court for the convenience of ground truth collection. In this set of experiments, data from all 15 experiment subjects, 5 different devices, and 5 different attachments are taken into account.

We have taken very complex and tortuous trajectories which are particularly challenging for inertial tracking systems, due to drift and the absence of absolute anchor measurements. Figs. 7.6, 7.7, and 7.8, show a couple of illustrative examples of how LL-Tracker succeeds in accurately tracking a pedestrian through the museum, office and sports center. The cumulative distribution functions of these environments are



(a) Museum ground truth (b) Reconstructed trajectory

Figure 7.6: Experiments in the museum, showing the ground truth and reconstructed trajectories.

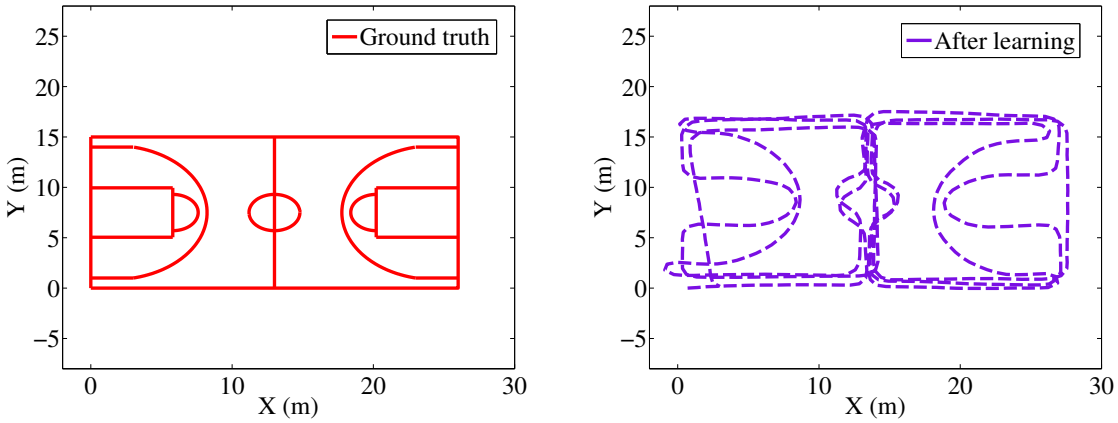


(a) Office ground truth (b) Reconstructed trajectory

Figure 7.7: Experiments in the office, showing the ground truth and reconstructed trajectories.

shown in Fig. 7.9(a). The RMS errors are 0.86m in the office and 0.90m in the museum. Note that LL-Tracker starts tuning R-PDR parameters after it converges to a single trajectory on the map with high probability; the CDF of the convergence distance in the office and museum environments is presented in Fig. 7.9(b).

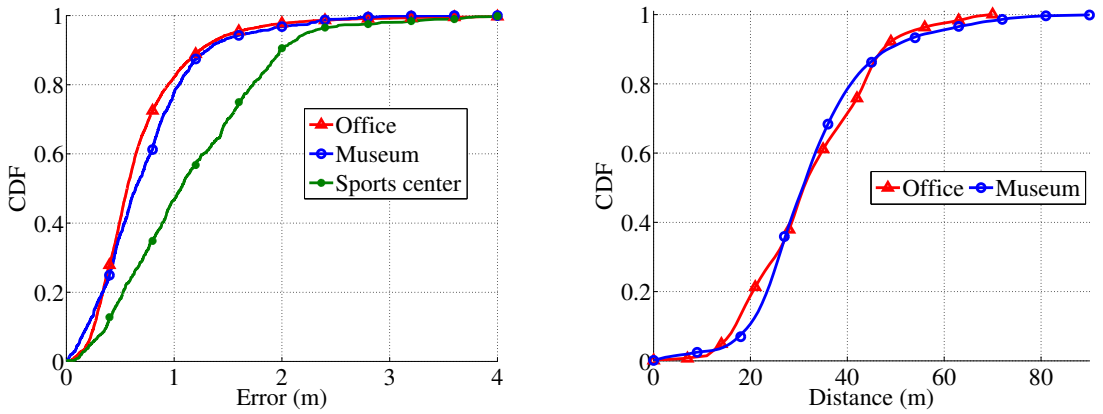
The performance in open space is always the limitation of PDR algorithms because the errors can be accumulated quickly within a short time when no external constraints, e.g. the floor plan, can be utilized to correct the drift, especially with low-cost IMU sensors. In order to understand the limitation of the proposed R-PDR



(a) Sports center ground truth

(b) Reconstructed trajectory

Figure 7.8: Experiments in the sports center, showing the ground truth and reconstructed trajectories.



(a) Multi-site performance to validate the robustness of LL-Tracker against environmental variations

(b) Convergence distance of the proposed learning algorithms

Figure 7.9: The performance of proposed LL-Tracker in multi-sites.

algorithm, we have tested the performance of various devices in a sports center with results reported in Fig. 7.9(a). The devices used in this experiment have been calibrated with the feedback loop from map matching layer in the office environment shown in Fig. 7.7(b). It is observed that the proposed LL-Tracker only has a RMS error of around 1.27 meters even in open space (basketball court) with trajectories as long as 0.5 km. An example trajectory in this environment can be found in Fig. 7.8.

The automatic learning of parameters is crucial to the performance of PDR-based tracking systems. We have observed from the extensive experiments that PDR parameters especially the gyro bias have much greater impact on the tracking performance

in environments with less constraints like the sports center than in more constrained environments like the office building and the museum where the floor plan can provide sufficient constraints. The gyro bias plays a crucial role in the tracking performance because the proposed system heavily relies on the accurate orientation estimation of the device. In comparison, the impact of learning and tuning the step constant offset is less significant when the map matching algorithm is applied in environments with rich floorplan constraints.

7.4 Discussion

This chapter has demonstrated the merits of a novel indoor positioning system LL-Tracker. LL-Tracker achieves sub meter accuracy more than 80% of the time. It is robust to device, attachment, user, and environment variability. It learns PDR parameters in an unsupervised manner and thus requires zero user effort. Notably, we have illustrated cases in which its operation in one environment can boost its performance in a new and more challenging environment. LL-Tracker has widespread application, as it can be used with a wide variety of sensors and different types of maps.

Recall that the NIMIT component is only used when the fingerprinting database is not available. Since the proposed LL-Tracker has the learning ability to automatically build the fingerprinting database, the NIMIT component is evaluated separately in Chapter 3 but not included in the evaluation in this Chapter.

Chapter 8

Conclusion and Future Work

This work was set out to explore how to make a practical indoor positioning system. The need of a practical indoor positioning system in various scenarios, e.g. exhibition centers, shopping malls, etc. strongly motivates this study. To facilitate the widespread application in these real world scenarios, a practical system should be designed to utilize widely available devices such as smartphones, pads, etc. and existing infrastructure like WiFi signals. The key requirements of the practical system are low-cost, accurate, robust, and scalable. To achieve this goal, we have proposed a system consisting of four components including robust pedestrian dead reckoning (R-PDR), non-line-of-sight identification and mitigation (NIMIT), lightweight map matching (MapCraft), and lifelong learning. Extensive real world experiments have demonstrated not only the merits of the proposed system, but the superiority of each individual component over the state-of-the-art approaches as well. This chapter first summarizes each individual component and the whole system, then discusses the limitations of the proposed system and ideas for future work.

8.1 Conclusion

This section concludes the functionality and performance of each individual component of the proposed system and explains how the whole system meets all key requirements we have defined for a practical indoor positioning system.

1. Non-line-of-sight Identification and Mitigation

Given the RSS measurements from the WiFi/BLE scanner, this component infers whether the direct path between the transmitter (e.g. AP) and receiver (e.g. mobile devices) is obstructed by objects such as walls and doors, called NLOS identification,

and estimate the transmitter-receiver distance, called NLOS mitigation. The goal of NLOS identification and mitigation is to achieve better positioning accuracy with the collected RSS.

We propose and compare distinct NLOS identification and mitigation algorithms based on machine learning techniques and hypothesis testing, namely least square support vector machine (LS-SVM), Gaussian processes (GP), and hypothesis testing (HT). LS-SVM has a low computational cost compared with other machine learning techniques while GPC has slightly higher classification/regression accuracy. To reduce the training phase, which is expensive for most applications in terms of training data collection, we also propose HT which only requires least square estimation of the mean and variance of all RSS measurements.

Extensive experiments in various indoor environments show that our techniques can distinguish between LOS/NLOS conditions with an accuracy of around 95%. Furthermore, the presented techniques improve distance estimation accuracy by 60% as compared to state-of-the-art NLOS mitigation techniques. Finally, improvements in distance estimation accuracy of 50% are achieved even without environment-specific training, demonstrating the practicality of our approach to real world implementations.

2. Robust Pedestrian Dead Reckoning

The R-PDR component converts the IMU measurements into PDR trajectories – a sequence of <distance, heading> pairs. It addresses three key challenges of existing PDR algorithms, including step detection with unconstrained mobile devices, distinguishing between real steps and other periodic motions, and achieving robust long term tracking even with significant sensor drift, especially with low-cost IMU sensors. To address these issues, three novel capabilities have been added to the PDR system, including 1) novel motion recognition algorithm which only includes two classes, but captures a wide spectrum of human motion and device placement combinations, 2) rotationally invariant orientation tracking algorithm, which corrects long-term gyroscope drift and tracks the orientation of the device robustly, and 3) accurate acceleration estimation algorithm which further removes the short-term bias of acceleration and extracts accurate acceleration signals for human motion.

R-PDR was extensively evaluated with real world experiments. It is demonstrated that the overall step detection accuracy can be up to 99.4% regardless of the device attachment. Specifically, it can reject over 99% of false positives. R-PDR thus addresses the three main limitations facing existing PDR techniques.

3. Lightweight Map Matching

This component takes the PDR trajectory from the R-PDR component and the coarse location estimations if available from either radio maps or NLOS identification and mitigation component as the input and estimates the locations of the user. Floor plans, various maps, and sensor observations are combined using conditional random fields, an undirected graphical model. The CRF model is particularly well suited to this sequential problem because it allows us to flexibly define *feature functions* that capture the extent to which observations support states and state transitions, given map constraints.

Four distinct steps are involved in the map matching process, including *map pre-processing* which converts floor plan into a graph with a set of discrete states and edges, *feature definition* which defines various feature functions from different sensor modalities, *optional training* which optimizes the model parameters for individual sensor observations and buildings, and *inference* which estimates the location over time using efficient Viterbi decoding. The first three steps are performed once for each floor plan. The fourth step is performed online on the user's mobile devices to track themselves.

It is validated in extensive real world experiments in three different environments showing how MapCraft outperforms state-of-the art approaches, achieving low tracking error and accurate reconstruction of tortuous trajectories with zero training effort. As proof of its robustness, we also demonstrate how it is able to accurately track the position of a user from accelerometer and magnetometer measurements only (i.e. gyro- and WiFi-free).

4. Lifelong Learning

This component proposes a reverse link between the map matching component and the PDR component. It augments the functionality of the map matching component to learn context-specific parameters including maps (radio maps, landmarks, etc.) to build additional maps for positioning, and PDR parameters fed back to the R-PDR component to improve its robustness and accuracy.

The learning processes of PDR parameters and maps are different. The learning of PDR parameters, such as sensor bias and step-related parameters, is achieved by maximizing the likelihood of the trajectory after the map matching given the PDR trajectories provided by R-PDR component. The learning of maps like WiFi/BLE radio map, magnetic distortion map, map of landmarks, etc. is achieved by embedding

the radio/magnetic/landmarks records along the trajectory obtained from the map matching component.

Extensive real world experiments have demonstrated how the lifelong learning can improve the accuracy and robustness of the proposed indoor positioning system. Although in this thesis we only demonstrate the efficacy of lifelong learning in learning parameters for R-PDR component, the idea of lifelong learning is broadly applicable to improve the performance of any PDR implementation that requires parameter tuning.

5. Key Objectives

Recall that the key objectives of a practical indoor positioning system are low-cost, accurate, robust, and scalable. We can now check how the proposed indoor positioning system achieves all four objectives.

Low-cost: The cost of existing indoor positioning solutions comes from two aspects: infrastructure deployment which deploys new expensive infrastructure such as UWB transceivers which serve as the landmarks for positioning service, and site survey which takes measurements in known locations before the positioning actually takes place. The proposed system only makes use of existing devices (smartphones, pads, etc.) and infrastructure (i.e. WiFi APs), and does not require any time-consuming site survey or expensive manual maintenance. Therefore, the proposed system is a low-cost system.

Accurate: It has been demonstrated in Chapter 7 that the proposed system can offer sub-meter accuracy for over 80% of the time across different experiment sites. The accuracy can even improve over time as the lifelong learning component learns more device- and site- specific parameters. Therefore, the proposed system is an accurate system.

Robust: The robustness measures whether the system can work equally or similarly well in different conditions, which is another fundamental requirement for positioning systems. Extensive experiments in Chapter 7 have demonstrated that the proposed system is robust against different context variations, including device variation, attachment variation, user variation, and environment variation.

Scalable: The scalability of a positioning solution is its ability to be easily deployed in new environments or to handle a growing number of positioning objects in a capable manner. Since the proposed system requires neither extra infrastructure nor labor intensive site survey, it is quite scalable and can be easily applied to a new building.

8.2 Limitations and Future Work

In spite of the good positioning performance of the proposed system, there are a number of limitations in the current system which also motivates directions for future work.

1. Power consumption

The major limitation of the proposed system is its power consumption. Since tracking requires continuous sensor data, especially from IMU sensors, the energy consumption can be significant with the processor running all the time to read sensor data. In addition, the energy consumption of gyro sensors also notably contributes to the total energy burning, up to ten times more than accelerometer and magnetometer.

To avoid draining the battery of mobile devices, the on-demand localization request can be answered with the position estimated from only existing radio maps automatically constructed by previous visitors of the same environment instead of always-on background service at the cost of lower accuracy at the start of tracking. Note that the accuracy can be quickly improved/recovered as the user walks around for a short while. Alternatively, sensor data can also be read and pre-processed with the aid of motion processors which are currently available only in latest smartphones with a much lower energy consumption. It can be expected that motion processors will be the trend of future smartphones given the dramatic increase in people's demand of ubiquitous health and daily life applications.

2. Destructive motions

Another limitation is the ability of the robust pedestrian dead reckoning component (R-PDR) to manage the combination of motions. A typical example is that if the user keeps shaking the device violently and randomly while walking, it is then beyond the ability of R-PDR to detect and validate steps because the sensor signal from the walking behavior is totally overwhelmed by the noise (signals from shaking behavior). In other words, the PDR trajectories are less reliable with malicious destructive motions.

To address this issue, a straightforward solution is to relax the constraints of PDR trajectories in the map matching component and put relatively higher weights to other feature functions discussed in Chapter 5. This would make the map matching component trust more other features and less the PDR features. Another method to approach this problem is to filter the sensor data to clean the signal before the

step detection according to the unique features of walking behavior. In addition, building a classifier to distinguish between malicious destructive motions and normal walking patterns and thereby skipping the PDR observations during this period is also another future research direction.

3. Expensive learning

The lifelong learning component requires the optimization of the likelihood of the output sequence given the observations, which, in essence, is the iterative computation of the whole map matching process until it reaches the optimal solution. This optimization process is probably the most computationally intensive part of the whole system. For resource constrained mobile devices, this computationally expensive learning process also poses challenges to the online optimization of the PDR parameters and radio maps.

This issue can be tackled in three ways. Firstly, not all learning requires such intensive computation therefore the less intensive learning can be done online in real time manner while others can be done when the device is idle. For instance, the learning of the environment can be easily achieved by the mobile devices. Furthermore, the computation can be offloaded to the cloud. Since the personalization of the algorithm for each individual happens only once, it is quite reasonable to send the data back to the server and do the calculation in the cloud. Finally and probably more attractively, we can develop a more computationally efficient algorithm to learn all the parameters involved. This might involve redesigning some part of the system but it is a very interesting future research direction.

4. Costly training

A minor limitation is the costly training phase of the NLOS identification and mitigation component. Generally speaking, the algorithms used in this components are supervised and require training phase before it can work well in a new environment. The training requires time consuming and labor intensive manual data collection, which is an obstacle for the widespread application of this algorithm.

However, as demonstrated in Chapter 3, the set of parameters trained from one environment still yield reasonably good positioning accuracy in a new environment. Therefore, future models could also be initialized with our existing data sets and then adapted to new environments using the radio map automatically constructed from the lifelong learning component. Alternatively, the position can also be estimated

by fingerprinting instead of distance estimation, which could also skip the training phase.

8.3 Closing Remarks

As the next frontier for the location services market, indoor positioning is likely to change the way people live and think. Since it provides people their locations in indoor environments where GPS cannot, numerous opportunities and experiences can be brought to people in their daily life. It is highly likely that people no longer waste time searching and waiting for each other in public areas like shopping malls and the shop owners have clear understanding of what commodities are popular to what types of customers and thus measures can be taken to optimize customer experience in their shops and maximize their profit as well.

The positioning system designed in this study is practical and efficient, which enables a new era of location-aware applications to be developed in various scenarios. In addition, some algorithms proposed in this system should not be limited to indoor positioning because they are sufficiently general to transcend the boundaries of indoor positioning and be used in other applications. However, further work is needed to examine the suitability of these algorithms for applications in other fields.

Appendix A

UKF-based Orientation Tracking

In the unscented Kalman filter used for orientation tracking of the device, the acceleration and magnetic field are used as the observation variables and angular velocity given by gyro sensors as the input variables of the process model. To avoid the singularity, we use the quaternion to represent the angles. Let the variable $f_g = 1$ denote the situation when gyro sensors are available and $f_g = 0$ otherwise. Let $\boldsymbol{q}_t = [\epsilon_0\ \epsilon_1\ \epsilon_2\ \epsilon_3]^T$ be the system state (angle of the device in quaternion) at time t. The process model is defined as follows.

$$\begin{aligned} \boldsymbol{q}_t &= \boldsymbol{q}_{t-1} + \frac{1}{2} f_g \boldsymbol{C}(\boldsymbol{q}_{t-1}) \boldsymbol{\omega}_{t-1} \Delta t + \tau(1 - f_g)(\boldsymbol{q}_{t-1} - \boldsymbol{q}_{t-2}) + \boldsymbol{v}, \\ &= F(\boldsymbol{q}_{t-1,t-2}, f_g, \boldsymbol{\omega}_{t-1}, \tau, \boldsymbol{v}). \end{aligned} \tag{A.1}$$

where the well-known $\boldsymbol{C}(\boldsymbol{q}_{t-1})$ is

$$\boldsymbol{C}(\boldsymbol{q}_{t-1}) = \frac{1}{||\boldsymbol{q}_{t-1}||} \begin{bmatrix} -\epsilon_1 & -\epsilon_2 & -\epsilon_3 \\ \epsilon_0 & -\epsilon_3 & \epsilon_2 \\ \epsilon_3 & \epsilon_0 & -\epsilon_1 \\ -\epsilon_2 & \epsilon_1 & \epsilon_0 \end{bmatrix}, \tag{A.2}$$

$\boldsymbol{\omega}_{t-1} = [\omega_{1,t-1}\ \omega_{2,t-1}\ \omega_{3,t-1}]^T$ is the angular velocity output of the gyroscope at time instant $t-1$, $\boldsymbol{v}$ is the process noise, Δt is the interval from time instant $t-1$ to t, and τ is the tuning parameter which controls the correlation between two consecutive steps.

Denote the observation variables with $\boldsymbol{u}_t = [g_1\ g_2\ g_3\ m_1\ m_2\ m_3]^T$ where g_i and $m_i, i = 1, 2, 3$ are the reported gravity and magnetic field, respectively. Then the observation model is as follows.

$$\begin{aligned} \boldsymbol{z}_t &= \begin{bmatrix} \boldsymbol{R}(\boldsymbol{q}_t) & \boldsymbol{0}_{3\times3} \\ \boldsymbol{0}_{3\times3} & \boldsymbol{R}(\boldsymbol{q}_t) \end{bmatrix} \boldsymbol{u}_t + \boldsymbol{n}. \\ &= H(\boldsymbol{q}_t, \boldsymbol{u}_t, \boldsymbol{n}). \end{aligned} \tag{A.3}$$

where $\boldsymbol{n}$ is the observation noise, and the rotation matrix $\boldsymbol{R}(\boldsymbol{q}_t)$ is

$$\boldsymbol{R}(\boldsymbol{q}_t) = \begin{bmatrix} \epsilon_0^2 + \epsilon_1^2 + \epsilon_2^2 + \epsilon_3^2 & 2(\epsilon_1\epsilon_2 - \epsilon_0\epsilon_3) & 2(\epsilon_1\epsilon_3 + \epsilon_0\epsilon_2) \\ 2(\epsilon_1\epsilon_2 + \epsilon_0\epsilon_3) & \epsilon_0^2 - \epsilon_1^2 + \epsilon_2^2 - \epsilon_3^2 & 2(\epsilon_2\epsilon_3 - \epsilon_0\epsilon_1) \\ 2(\epsilon_1\epsilon_3 - \epsilon_0\epsilon_2) & 2(\epsilon_2\epsilon_3 + \epsilon_0\epsilon_1) & \epsilon_0^2 - \epsilon_1^2 - \epsilon_2^2 + \epsilon_3^2 \end{bmatrix}. \tag{A.4}$$

A.1 Process and Observation Noise

Typical MEMS gyroscopes exhibit very high frequency noise caused by thermo-mechanical events, the process noise $\boldsymbol{v}$ is modeled as white gaussian noise with a long term average to zero. We assume no correlation between the four components in quaternion. The initial covariance matrix of the process noise is

$$\boldsymbol{P}_0^v = \sigma_g^2 \boldsymbol{I}_{4\times 4}. \tag{A.5}$$

The covariance of the system state is

$$\boldsymbol{P}_0^q = E[(\boldsymbol{q}_0 - \hat{q}_0)(\boldsymbol{q}_0 - \hat{q}_0)^T]. \tag{A.6}$$

The noise of the observed acceleration and magnetic field $\boldsymbol{n}$ could also be modeled as white Gaussian noise. Therefore,the initial observation covariance matrix is

$$\boldsymbol{P}_0^n = \begin{bmatrix} \sigma_a^2 \boldsymbol{I}_{3\times 3} & \boldsymbol{0}_{3\times 3} \\ \boldsymbol{0}_{3\times 3} & \sigma_m^2 \boldsymbol{I}_{3\times 3} \end{bmatrix} \tag{A.7}$$

A.2 Tracking Algorithm

At time instant t, we define

$$\boldsymbol{x}_t = \begin{bmatrix} \boldsymbol{q}_t \\ \boldsymbol{v}_t \\ \boldsymbol{n}_t \end{bmatrix} \tag{A.8}$$

and the augmented state dimension is the sum of the dimensions of the three vectors $\boldsymbol{q}_t$, $\boldsymbol{v}_t$, and $\boldsymbol{n}_t$.

$$d = d_q + d_v + d_n. \tag{A.9}$$

Similarly, the covariance matrix is constructed from the covariance matrix of the three vectors,

$$\boldsymbol{P}_t = \begin{bmatrix} \boldsymbol{P}_t^q & \boldsymbol{0}_{4\times 3} & \boldsymbol{0}_{4\times 6} \\ \boldsymbol{0}_{3\times 4} & \boldsymbol{P}_t^v & \boldsymbol{0}_{3\times 6} \\ \boldsymbol{0}_{6\times 4} & \boldsymbol{0}_{6\times 3} & \boldsymbol{P}_t^n \end{bmatrix} \tag{A.10}$$

A.2.1 Initialization at $t = 0$

$$
\begin{aligned}
f_g &= 0/1, && \text{(A.11)}\\
\hat{\boldsymbol{q}}_0 &= E[\boldsymbol{q}_0], && \text{(A.12)}\\
\hat{\boldsymbol{x}}_0 &= [\hat{\boldsymbol{q}}_0\ 0\ 0], && \text{(A.13)}\\
\boldsymbol{P}_0 &= \boldsymbol{P}_t|_{t=0}. && \text{(A.14)}
\end{aligned}
$$

A.2.2 Sigma-points

In total, $2d+1$ sigma-points are generated from the current state mean and covariance as follows.

$$
\boldsymbol{X}_{t-1} = \begin{cases} \hat{\boldsymbol{x}}_{t-1} & j = 0, \\ \hat{\boldsymbol{x}}_{t-1} + \gamma \boldsymbol{S}_j & j = 1, \cdots, d, \\ \hat{\boldsymbol{x}}_{t-1} - \gamma \boldsymbol{S}_j & j = d+1, \cdots, dN, \end{cases} \tag{A.15}
$$

where $\boldsymbol{S}_j$ is the jth column of the matrix $\boldsymbol{S}$ from the square-root decomposition of covariance matrix at time $t-1$

$$
\boldsymbol{S} = \sqrt{\boldsymbol{P}_{t-1}}, \tag{A.16}
$$

and γ is a scaling parameter

$$
\gamma = \sqrt{d + \lambda}, \quad \lambda = \alpha^2(d + \kappa) - d, \tag{A.17}
$$

in which $\alpha (0 \leq \alpha \leq 1)$ and $\kappa \geq 0$ are tuning parameters controlling the sigma-point distribution.

The sigma-points matrix could be decomposed to

$$
\boldsymbol{X}_{t-1} = \begin{bmatrix} \boldsymbol{Q}_{t-1} \\ \boldsymbol{V}_{t-1} \\ \boldsymbol{N}_{t-1} \end{bmatrix} \tag{A.18}
$$

where $\boldsymbol{Q}_{t-1},\ \boldsymbol{V}_{t-1},\ \boldsymbol{N}_{t-1}$ are the sigma-points matrix of the system state, process noise, and observation noise, respectively.

A.2.3 Time Update

Calculate the system states at time t by transforming the sigma-points through the system model

$$
\boldsymbol{Q}_{t|t-1} = F(\boldsymbol{Q}_{t-1}, f_g, \boldsymbol{\omega}_{t-1}, \tau, \boldsymbol{v}), \tag{A.19}
$$

where F is defined in Equation A.1. The *a priori* state mean and covariance estimate

$$\hat{\boldsymbol{q}}_t^k = \sum_{j=0}^{2d} (w_j^m \boldsymbol{Q}_{j,t|t-1}), \tag{A.20}$$

$$\boldsymbol{P}_t^k = \sum_{j=0}^{2d} (w_j^c (\boldsymbol{Q}_{j,t|t-1} - \hat{\boldsymbol{q}}_t^k)(\boldsymbol{Q}_{j,t|t-1} - \hat{\boldsymbol{q}}_t^k)^T), \tag{A.21}$$

where $\boldsymbol{Q}_{j,t|t-1}$ is the jth column of matrix $\boldsymbol{Q}_{t|t-1}$, w_j^c and w_j^m are the weights of the jth column of the mean and covariance matrix, defined as,

$$\begin{aligned} w_0^m &= \frac{\lambda}{d+\lambda}, & j = 0, \\ w_0^c &= \frac{\lambda}{d+\lambda} + (1 - \alpha^2 + \beta), & j = 0, \\ w_j^m &= w_j^c = \frac{1}{2(d+\lambda)}, & j = 1, \cdots, 2d, \end{aligned} \tag{A.22}$$

in which for Gaussian prior we set $\beta = 2$ to affect the calculation of the covariance for the zeroth sigma-points as suggested in [219].

A.2.4 Observation Update

According to the observation model, the new observation sigma-points are

$$\boldsymbol{Z}_{t|t-1} = H(\boldsymbol{Q}_{t|t-1}, \boldsymbol{u}_t, \boldsymbol{n}). \tag{A.23}$$

where H is defined in Equation A.3. The mean and the covariance of the observation are calculated according to

$$\hat{\boldsymbol{z}}_t = \sum_{j=0}^{2d} (w_j^m \boldsymbol{Z}_{j,t|t-1}), \tag{A.24}$$

$$\boldsymbol{P}_t^z = \sum_{j=0}^{2d} (w_j^c (\boldsymbol{Z}_{j,t|t-1} - \hat{\boldsymbol{z}}_t)(\boldsymbol{Z}_{j,t|t-1} - \hat{\boldsymbol{z}}_t)^T). \tag{A.25}$$

The cross variance is

$$\boldsymbol{P}_t^{qz} = \sum_{j=0}^{2d} (w_j^c (\boldsymbol{Q}_{j,t|t-1} - \hat{\boldsymbol{q}}_t)(\boldsymbol{Z}_{j,t|t-1} - \hat{\boldsymbol{z}}_t)^T). \tag{A.26}$$

The Kalman gain is

$$\boldsymbol{K}_t = P_t^{qz} (P_t^z)^{-1}. \tag{A.27}$$

At last we get the state estimate and covariance from the unscented Kalman filter as follows.

$$\hat{\boldsymbol{q}}_t = \hat{\boldsymbol{q}}_t^k + \boldsymbol{K}_t (\boldsymbol{z}_t - \hat{\boldsymbol{z}}_t). \tag{A.28}$$

$$\boldsymbol{P}_t = \boldsymbol{P}_t^k - \boldsymbol{K}_t \boldsymbol{P}_t^z \boldsymbol{K}_t^T. \tag{A.29}$$

A.2.5 Parameters

The implementation of the UKF-based orientation tracking algorithm in this thesis has the parameter settings shown in Table A.1.

Table A.1: Parameter settings

Parameter	Value	Description of the parameter
d_q	4	Dimension of Kalman states
d_v	4	Dimension of state noise
d_n	3	Dimension of observation noise
σ_g	$2e^{-3}$	Standard deviation of angular velocity from gyroscope
σ_a	0.3	Standard deviation of gravity observation from accelerometer
σ_m	20	Standard deviation of magnetic field from magnetometer
α	1	Tuning parameter for sigma-points distribution
κ	0	Tuning parameter for sigma-points distribution
β	2	Gaussian prior covariance parameter
τ	0.8	Correlation tuning parameters between two consecutive steps

All other parameters involved in this implementation were derived from parameters in Table A.1 using the corresponding equations given in this appendix.

Bibliography

[1] Indoor Air Division Office, "Report to Congress on indoor air quality volume II: Assessment and control of indoor air pollution," tech. rep., U.S. Environmental Protection Agency, Washington, DC, USA, 1999.

[2] A. Malik, *RTLS For Dummies*. John Wiley & Sons, 1st ed., 2009.

[3] A. Hopper, R. Want, and A. Harter, "The Active Badge System."

[4] D. Boontrai, T. Jingwangsa, and P. Cherntanomwong, "Indoor localization technique using passive RFID tags," in *Proc. 9th Int. Symp. Commun. Info. Technol. (ISCIT'09)*, (Icheon, South Korea), pp. 922–926, Sept. 2009.

[5] S. Saad and Z. Nakad, "A standalone RFID indoor positioning system using passive tags," *IEEE Trans. Ind. Electron.*, vol. 58, no. 5, pp. 1961–1970, 2011.

[6] P. Vorst and J. Sommer, "Indoor positioning via three different RF technologies," in *Proc. 4th European Workshop RFID Syst., Technol. (RFID SysTech'08)*, (Freiburg, Germany), pp. 1–10, 2008.

[7] M. Bouet and A. L. dos Santos, "RFID tags: Positioning principles and localization techniques," in *Proc. 1st IFIP Wirel. Days (WD'08)*, (Dubai), pp. 1–5, Nov. 2008.

[8] S. Ting, S. Kwok, A. Tsang, and G. Ho, "The study on using passive RFID tags for indoor positioning," *Int. J. Engneer. Business Manage.*, vol. 3, no. 1, pp. 9–15, 2011.

[9] F. Seco and C. Plagemann, "Improving RFID-based indoor positioning accuracy using Gaussian processes," in *Proc. Int. Conf. Indoor Pos. Indoor Nav. (IPIN'10)*, (Zurich, Switzerland), pp. 1 – 8, 2010.

[10] H. Zou, H. Wang, L. Xie, and Q.-S. Jia, "An RFID indoor positioning system by using weighted path loss and extreme learning machine," in *Pos. Loc. Nav. Symp. (PLANS'04)*, (Taipei, Taiwan, China), pp. 66 – 71, 2013.

[11] J. Wu, *Three-dimensional indoor RFID localization system.* PhD thesis, University of Nebraska, 2012.

[12] A. Athalye, V. Savic, M. Bolic, and P. Djuric, "Novel semi-passive rfid system for indoor localization," *IEEE Sensors J.*, vol. 2, no. 13, pp. 528–537, 2013.

[13] Apple Inc., "Getting Started with iBeacon," tech. rep., Apple Inc., 2014.

[14] X. Zhao, Z. Xiao, A. Markham, N. Trigoni, and Y. Ren, "Does BTLE measure up against WiFi? A comparison of indoor location performance," in *Proc. 20th European Wirel. Conf. (EW'14)*, (Barcelona, Spain), pp. 1–6, 2014.

[15] S. Hay and R. Harle, "Bluetooth tracking without discoverability," in *Proc. 4th Int. Symp. Location Context Awareness (LoCA '09)*, (Tokyo, Japan), pp. 120–137, 2009.

[16] M. J. Kuhn, J. Turnmire, M. R. Mahfouz, and A. E. Fathy, "Adaptive leading-edge detection in UWB indoor localization," in *Proc. IEEE Radio, Wirel. Symp. (RWS'10)*, (New Orleans, LA, USA), pp. 268–271, Jan. 2010.

[17] Y. Zhou and C. L. Law, "Indoor elliptical localization based on asynchronous UWB range measurement," *IEEE Trans. Instrument. Measure.*, vol. 60, no. 1, pp. 248–257, 2011.

[18] N. Arrue, M. Losada, L. Zamora-Cadenas, A. Jimenez-Irastorza, and I. Velez, "Design of an IR-UWB Indoor Localization System Based on a Novel RTT Ranging Estimator," in *Proc. 1st Int. Conf. Sensor Device Technol., App. (SENSORDEVICES '10)*, (Venice, Italy), pp. 52–57, July 2010.

[19] M. Segura, V. Mut, and C. Sisterna, "Ultra wideband indoor navigation system," *IET Radar Sonar, Nav.*, vol. 6, no. 5, p. 402, 2012.

[20] Z. Irahhauten, H. Nikookar, and M. Klepper, "2D UWB localization in indoor multipath environment using a joint ToA/DoA technique," in *Proc. IEEE Wirel. Commun. Netw. Conf. (WCNC'12)*, (Shanghai, China), pp. 2253–2257, Apr. 2012.

[21] Z. Irahhauten, H. Nikookar, and M. Klepper, "A joint ToA/DoA technique for 2D/3D UWB localization in indoor multipath environment," in *Proc. IEEE Int. Conf. Commun. (ICC'12)*, (Ottawa, ON, Canada), pp. 4499–4503, June 2012.

[22] A. D. Angelis, S. Dwivedi, and P. Handel, "Characterization of a flexible UWB sensor for indoor localization," *IEEE Trans. Instrument. Measure.*, vol. 62, no. 5, pp. 905–913, 2013.

[23] P. Meissner, M. Gan, and F. Mani, "On the use of ray tracing for performance prediction of UWB indoor localization systems," in *Proc. IEEE Int. Conf. Commun. (ICC'13)*, (Budapest, Hungary), pp. 68–73, 2013.

[24] W. Vinicchayakul and S. Promwong, "Improvement of fingerprinting technique for UWB indoor localization," in *Proc. 4th Joint Int. Conf. Info. Commun. Technol., Electro. Electri. Engineer (JICTEE'14)*, (Chiang Rai, Thailand), pp. 1–5, Mar. 2014.

[25] T. Gigl, G. Janssen, V. Dizdarevic, K. Witrisal, and Z. Irahhauten, "Analysis of a UWB indoor positioning system based on received signal strength," in *Proc. 4th Workshop Pos. Nav., Commun. (WPNC'07)*, (Hannover, Germany), pp. 1–5, 2007.

[26] G. Bellusci, *Ultra-Wideband Ranging for Low-Complexity Indoor Positioning Applications*. PhD thesis, Delft University of Technology, 2010.

[27] P. Meissner, E. Leitinger, M. Fröhle, and K. Witrisal, "Accurate and robust indoor localization systems using ultra-wideband signals," in *Proc. European Conf. Nav. (ENC'13)*, (Vienna, Austria), pp. 1–9, Apr. 2013.

[28] S. Ingram, D. Harmer, and M. Quinlan, "Ultrawideband indoor positioning systems and their use in emergencies," in *Pos. Loc., Nav. Symp. (PLANS'04)*, (Monterey, CA, USA), pp. 706–715, 2004.

[29] Z. Li, W. Dehaene, and G. Gielen, "A 3-tier UWB-based indoor localization system for ultra-low-power sensor networks," *IEEE Trans. Wirel. Commun.*, vol. 8, pp. 2813–2818, June 2009.

[30] H. Wymeersch and S. Maranò, "A machine learning approach to ranging error mitigation for UWB localization," *IEEE Trans. Commun.*, vol. 60, no. 6, pp. 1719–1728, 2012.

[31] L. Yang and G. B. Giannakis, "Ultra-wideband communications: an idea whose time has come," *IEEE Signal Process. Mag.*, vol. 21, no. 6, pp. 26–54, 2004.

[32] A. F. Molisch, "Ultra-wide-band propagation channels," *Proc. IEEE*, vol. 97, pp. 353–371, Feb. 2009.

[33] M. Z. Win and R. A. Scholtz, "Ultra-wide bandwidth time-hopping spread-spectrum impulse radio for wireless multiple-access communications," *IEEE Trans. Commun.*, vol. 48, no. 4, pp. 679–691, 2000.

[34] M. Z. Win and R. A. Scholtz, "Characterization of ultra-wide bandwidth wireless indoor channels : a communication-theoretic view," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 9, pp. 1613–1627, 2002.

[35] M. Z. Win and R. A. Scholtz, "Impulse radio: how it works," *IEEE Commun. Lett.*, vol. 2, pp. 36–38, Feb. 1998.

[36] J. A. Dabin, A. M. Haimovich, and H. Grebel, "A statistical ultra-wideband indoor channel model and the effects of antenna directivity on path loss and multipath propagation," *IEEE J. Sel. Areas Commun.*, vol. 24, no. 4, pp. 752–758, 2006.

[37] J. Blankenbach, A. Norrdine, and H. Hellmers, "A robust and precise 3D indoor positioning system for harsh environments," in *Proc. Int. Conf. Indoor Pos. Indoor Nav. (IPIN'12)*, (Sydney, NSW), pp. 1–8, 2012.

[38] V. Schlageter, P.-A. Besse, R. Popovic, and P. Kucera, "Tracking system with five degrees of freedom using a 2D-array of Hall sensors and a permanent magnet," *Sens. and Actuators A*, vol. 92, no. 1-3, pp. 37–42, 2001.

[39] D. Roetenberg, P. Slycke, A. Ventevogel, and P. Veltink, "A portable magnetic position and orientation tracker," *Sens. and Actuators A*, vol. 135, pp. 426–432, Apr. 2007.

[40] A. Sheinker, B. Ginzburg, N. Salomonski, L. Frumkis, and B.-Z. Kaplan, "Localization in 2D Using Beacons of Low Frequency Magnetic Field," *IEEE J. Sel. Topics Applied Earth Observ. Remote Sens.*, vol. 6, pp. 1020–1030, Apr. 2013.

[41] J. Blankenbach and A. Norrdine, "Position estimation using artificial generated magnetic fields," in *Proc. Int. Conf. Indoor Pos. Indoor Nav. (IPIN'10)*, (Zurich, Switzerland), pp. 1–5, 2010.

[42] E. Paperno, I. Sasada, and E. Leonovich, "A new method for magnetic position and orientation tracking," *IEEE Trans. Magn.*, vol. 37, no. 4, pp. 1938–1940, 2001.

[43] A. Sheinker, B. Ginzburg, N. Salomonski, L. Frumkis, and B.-Z. Kaplan, "Localization in 3-D Using Beacons of Low Frequency Magnetic Field," *IEEE Trans. Instrument. Measure.*, vol. 62, pp. 3194–3201, Dec. 2013.

[44] D. D. Arumugam, M. Sibley, J. D. Griffin, D. D. Stancil, and D. S. Ricketts, "An active position sensing tag for sports visualization in American football," in *Proc. IEEE Int. Conf. RFID (RFID'13)*, (Penang), pp. 96–103, Apr. 2013.

[45] E. Paperno and P. Keisar, "Three-Dimensional Magnetic Tracking of Biaxial Sensors," *IEEE Trans. Magnetics*, vol. 40, pp. 1530–1536, May 2004.

[46] H. Hellmers, A. Norrdine, J. Blankenbach, and A. Eichhorn, "An IMU/magnetometer-based indoor positioning system using Kalman filtering," in *Proc. Int. Conf. Indoor Pos. Indoor Nav. (IPIN'13)*, (Montbeliard-Belfort), pp. 1–9, 2013.

[47] G. Pirkl and P. Lukowicz, "Robust, low cost indoor positioning using magnetic resonant coupling," in *Proc. ACM Conf. Ubi. Comput. (UbiComp'12)*, (New York, NY, USA), pp. 431–440, 2012.

[48] B. Alavi and K. Pahlavan, "Analysis of undetected direct path in time of arrival based UWB indoor geolocation," in *Proc. IEEE 62nd Veh. Technol. Conf. (VTC'05)*, vol. 4, pp. 2627–2631, 2005.

[49] V. Dyo, K. Yousef, S. a. Ellwood, D. W. Macdonald, A. Markham, N. Trigoni, R. Wohlers, C. Mascolo, B. Pásztor, and S. Scellato, "Wildsensing: Design and deployment of a sustainable sensor network for wildlife monitoring," *ACM Trans. Sens. Netw.*, vol. 8, pp. 1–33, Sept. 2012.

[50] A. Markham, N. Trigoni, S. a. Ellwood, and D. W. Macdonald, "Revealing the hidden lives of underground animals using magneto-inductive tracking," in *Proc. 8th ACM Conf. Embedded Netw. Sensor Syst. (SenSys'10)*, (Zurich, Switzerland), pp. 281–285, 2010.

[51] A. Markham, N. Trigoni, D. W. Macdonald, and S. a. Ellwood, "Underground Localization in 3-D Using Magneto-Inductive Tracking," *IEEE Sens. J.*, vol. 12, pp. 1809–1816, June 2012.

[52] R. Feynman, *The Feynman lectures on physics*. Pasadena, CA, USA: Addison Wesley Longman, 1970.

[53] Health Canada, "Limits of human exposure to raidofrequency electromagnetic energy in the frequency range from 3 kHz to 300 GHz," tech. rep., Health Canada, Ottawa, ON, Canada, 2009.

[54] H. Kim and J. Choi, "Advanced indoor localization using ultrasonic sensor and digital compass," in *Proc. Int. Conf. Control, Automat. Syst. (ICCAS'08)*, (Seoul, Korea), pp. 223–226, Oct. 2008.

[55] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Commun. Surveys Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[56] Sana, "A Survey of Indoor Localization Techniques," *IOSR J. Electrical and Electronics Engineer.*, vol. 6, no. 3, pp. 69–76, 2013.

[57] Y.-k. Ku, C.-s. Nam, and D.-r. Shin, "Efficient indoor localization and error correction algorithm," in *Proc. 12th Int. Conf. Adv. Commun. Technol. (ICACT'10)*, (Dublin, Ireland), pp. 453–457, 2010.

[58] S. K. Park and Y. S. Suh, "A zero velocity detection algorithm using inertial sensors for pedestrian navigation systems.," *Sensors*, vol. 10, pp. 9163–9178, Jan. 2010.

[59] R. Harle, "A Survey of Indoor Inertial Positioning Systems for Pedestrians," *IEEE Commun. Surveys Tutorials*, vol. 15, pp. 1281–1293, Jan. 2013.

[60] J. Kim, H. Jang, D. Hwang, and C. Park, "A step, stride and heading determination for the pedestrian navigation system," *J. Global Pos. Syst.*, vol. 3, no. 1, pp. 273–279, 2004.

[61] L. Fang, P. Antsaklis, L. Montestruque, M. B. McMickell, M. Lemmon, Y. Sun, and H. Fang, "Design of a wireless assisted pedestrian dead reckoning system-the NavMote experience," *IEEE Trans. Instrument. Measure.*, vol. 54, no. 6, pp. 2342–2358, 2005.

[62] I. Bylemans, M. Weyn, and M. Klepal, "Mobile Phone-Based Displacement Estimation for Opportunistic Localisation Systems," in *Proc. 3rd Int. Conf. Mob. Ubi. Comput. Syst. Services Technol. (UbiComp'09)*, (Sliema), pp. 113–118, Oct. 2009.

[63] S. Beauregard and H. Haas, "Pedestrian dead reckoning: A basis for personal positioning," in *Proc. 3rd Workshop Pos. Nav. Commun. (WPNC'06)*, (Hannover, Germany), pp. 27–36, 2006.

[64] S. H. Shin, M. S. Lee, and P. C. G., "Pedestrian dead reckoning system with phone location awareness algorithm," in *Proc. IEEE/ION Position Location Nav. Symp. (PLANS'10)*, (Indian Wells, CA, USA), pp. 97–101, 2010.

[65] P. Goyal, V. J. Ribeiro, H. Saran, and A. Kumar, "Strap-down Pedestrian Dead-Reckoning system," in *Proc. Int. Conf. Indoor Pos. Indoor Nav. (IPIN'11)*, (Guimarães, Portugal), pp. 1–7, Sept. 2011.

[66] N. Ravi, N. Dandekar, P. Mysore, and M. Littman, "Activity recognition from accelerometer data," in *Proc. 7th Conf. Innov. App. Artificial Intell. (AAAI'05)*, (Pittsburgh, PA, USA), pp. 1541–1546, 2005.

[67] J.-g. Park, A. Patel, D. Curtis, S. Teller, and J. Ledlie, "Online pose classification and walking speed estimation using handheld devices," in *Proc. ACM Conf. Ubi. Comput. (UbiComp'12)*, (New York, NY, USA), pp. 1–10, 2012.

[68] A. Brajdic and R. Harle, "Walk detection and step counting on unconstrained smartphones," in *Proc. ACM Conf. Ubi. Comput. (UbiComp'13)*, (Zurich, Switzerland), pp. 225–234, ACM Press, 2013.

[69] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, pp. 8507–8525, Jan. 2012.

[70] J. Rose and J. G. Gamble, *Human Walking*. Baltimore, PA, USA: Lippincott, Williams and Wilkins, 3rd ed., 2006.

[71] H. Ying, C. Silex, A. Schnitzer, S. Leonhardt, and M. Schiek, "Automatic step detection in the accelerometer signal," in *Proc. 4th Int. Workshop Wearable and Implantable Body Sens. Netw. (BSN'07)*, (Aachen University, Germany), pp. 80–85, 2007.

[72] M. Susi, V. Renaudin, and G. Lachapelle, "Motion Mode Recognition and Step Detection Algorithms for Mobile Phone Users," *Sensors*, vol. 13, no. 2, pp. 1539–1562, 2013.

[73] J. Yang, "Toward physical activity diary: motion recognition using simple acceleration features with mobile phones," in *Proc.1st Int. workshop Interactive multimedia for consumer electronics (IMCE'09)*, (Beijing, China), pp. 1–9, 2009.

[74] J. S. Wang, C. W. Lin, Y. T. Yang, and Y. J. Ho, "Walking pattern classification and walking distance estimation algorithms using gait phase information," *IEEE Trans. Biomedical Engineer.*, vol. 59, no. 10, pp. 2884–2892, 2012.

[75] M. Alzantot and M. Youssef, "UPTIME: Ubiquitous pedestrian tracking using mobile phones," in *Proc. IEEE Wirel. Commun. Netw. Conf. (WCNC'12)*, (Paris, France), pp. 3204–3209, Apr. 2012.

[76] E. Foxlin, "Pedestrian tracking with shoe-mounted inertial sensors," *IEEE Comput. Graph. App.*, vol. 25, no. 6, pp. 38–46, 2005.

[77] R. Faragher and R. Harle, "SmartSLAMan efficient smartphone indoor positioning system exploiting machine learning and opportunistic sensing," in *Proc. 26th Int. Tech. Meet. Satel. Div. Institute of Nav. (ION GNSS+ '13)*, (Nashville, TN, USA), pp. 1–14, 2013.

[78] S. H. Shin, C. G. Park, and J. W. Kim, "Adaptive step length estimation algorithm using low-cost MEMS inertial sensors," in *Proc. IEEE Sens. App. Symp. (SAS'07)*, (San Diege, CA, USA), pp. 1–5, 2007.

[79] W. Chen, R. Chen, Y. Chen, H. Kuusniemi, and J. Wang, "An effective Pedestrian Dead Reckoning algorithm using a unified heading error model," in *Proc. IEEE/ION Position, Location and Nav. Symp*, (Indian Wells, CA, USA), pp. 340–347, May 2010.

[80] A. Rai and K. Chintalapudi, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proc. 18th Ann. Int. Conf. Mob. Comput. Netw. (MobiCom'12)*, (Istanbul, Turkey), pp. 1–12, 2012.

[81] F. Li, C. Zhao, G. Ding, J. Gong, C. Liu, and F. Zhao, "A reliable and accurate indoor localization method using phone inertial sensors," in *Proc. ACM Conf. Ubi. Comput. (UbiComp'12)*, (New York, NY, USA), pp. 421–430, 2012.

[82] Y. S. Suh, "Orientation estimation using a quaternion-based indirect Kalman filter with adaptive estimation of external acceleration," *IEEE Trans. Instrument. Measure.*, vol. 59, no. 12, pp. 3296–3305, 2010.

[83] X. Yun and E. R. Bachmann, "Design, Implementation, and Experimental Results of a Quaternion-Based Kalman Filter for Human Body Motion Tracking," *IEEE Trans. Robotics*, vol. 22, pp. 1216–1227, Dec. 2006.

[84] B. Huyghe, J. Doutreloigne, and J. Vanfleteren, "3D orientation tracking based on unscented Kalman filtering of accelerometer and magnetometer data," in *Proc. IEEE Sens. App. Symp. (SAS'09)*, (New Orleans, LA, USA), pp. 148 – 152, 2009.

[85] N. Roy, H. Wang, and R. Roy Choudhury, "I am a smartphone and i can tell my user's walking direction," in *Proc. 12th Ann. Int. Conf. Mob. Syst., App., Serv. (MobiSys'14)*, (New York, NY, USA), pp. 329–342, 2014.

[86] T. Harada, T. Mori, and T. Sato, "Development of a Tiny Orientation Estimation Device to Operate under Motion and Magnetic Disturbance," *The Int. J. Robotics Res.*, vol. 26, pp. 547–559, June 2007.

[87] M. H. Afzal, V. Renaudin, and G. Lachapelle, "Assessment of indoor magnetic field anomalies using multiple magnetometers," in *Proc. 23th Int. Tech. Meet. Satel. Div. Institute of Nav. (ION GNSS+ '10)*, (Portland, Oregon), pp. 21–24, 2010.

[88] M. B. Kjæ rgaard, "Indoor location fingerprinting with heterogeneous clients," *Pervasive Mob. Comput.*, vol. 7, pp. 31–43, Feb. 2011.

[89] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "FM-based indoor localization," in *Proc. 10th Int. Conf. Mob. Syst. App. Serv. (MobiSys'12)*, (Low Wood Bay, Lake District, UK), pp. 169–181, 2012.

[90] M. Azizyan, I. Constandache, and R. Roy Choudhury, "SurroundSense: mobile phone localization via ambience fingerprinting," in *Proc. 15th Ann. Int. Conf. Mob. Comput. Netw. (MobiCom'09)*, (New York, NY, USA), pp. 261–272, 2009.

[91] G. Shen, Z. Chen, P. Zhang, T. Moscibroda, and Y. Zhang, "Walkie-Markie: indoor pathway mapping made easy," in *Proc. 10th USENIX Conf. Netw. Syst. Design, Implement. (NSDI'13)*, (Lombard, IL, USA), pp. 85–93, 2013.

[92] E. Elnahrawy and R. Martin, "The limits of localization using signal strength: a comparative study," in *Proc.1st Ann. IEEE Commun. Soc. Conf. Sens. Ad Hoc Commun., Netw. (SECON'04)*, (Piscataway, NJ, USA), pp. 406–414, 2004.

[93] Y. Chen and H. Kobayashi, "Signal Strength Based Indoor Geolocation," in *Proc. IEEE Int. Conf. Commun., (ICC'02)*, (New York, USA), pp. 436–439, 2002.

[94] F. Reichenbach and D. Timmermann, "Indoor localization with Low complexity in wireless sensor networks," in *Proc. IEEE Int. Conf. Ind. Info. (INDIN'06)*, (Singapore), pp. 1018–1023, 2006.

[95] X. Li, "RSS-Based Location Estimation with Unknown Pathloss Model," *IEEE Trans. Wirel. Commun.*, vol. 5, pp. 3626–3633, Dec. 2006.

[96] L. Xie, Y. Wang, and X. Xue, "A new indoor localization method based on inversion propagation model," in *Proc. 6th Int. Conf. Wirel. Commun. Netw. Mob. Comput. (WiCom'10)*, (Chengdu, China), pp. 1–4, Sept. 2010.

[97] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, "A novel approach to indoor RSSI localization by automatic calibration of the wireless propagation model," in *Proc. IEEE 69th Veh. Technol. Conf. (VTC'09-Spring)*, (Barcelona, Spain), pp. 1–5, Apr. 2009.

[98] K. Li, M. A. Ingram, and A. V. Nguyen, "Impact of clustering in statistical indoor propagation models on link capacity," *IEEE Trans. Commun.*, vol. 50, no. 4, pp. 521–523, 2002.

[99] C. Lim, J. L. Volakis, K. Sertel, R. W. Kindt, and A. Anastasopoulos, "Indoor propagation models based on rigorous methods for site-specific multipath environments," *IEEE Trans. Antennas Propag.*, vol. 54, no. 6, pp. 1718–1725, 2006.

[100] I. Guvenc, C. Chong, and F. Watanabe, "NLOS identification and mitigation for UWB localization systems," in *Proc. IEEE Wirel. Commun. Netw. Conf. (WCNC'07)*, (Kowloon, China), pp. 1571–1576, 2007.

[101] A. Maali, A. Ouldali, H. Mimoun, and G. Baudoin, "Joint TOA estimation and NLOS identification for UWB localization systems," in *Proc. 3rd Int. Conf. Sen. Technol. App. (SENSORCOMM'09)*, (Athens, Glyfada), pp. 212–216, June 2009.

[102] Y. Qi, H. Kobayashi, and H. Suda, "Analysis of wireless geolocation in a non-line-of-sight environment," *IEEE Trans. Wirel. Commun.*, vol. 5, no. 3, pp. 3–5, 2006.

[103] S. Venkatesh and R. M. Buehrer, "Non-line-of-sight identification in ultra-wideband systems based on received signal statistics," *IET Microw. Antennas Propag.*, vol. 1, no. 6, pp. 1120–1130, 2007.

[104] Y. Jo, J. Lee, D. Ha, and S. Kang, "Accuracy enhancement for UWB indoor positioning using ray tracing," in *IEEE/ION Pos. Loc. Nav. Symp.*, pp. 565–568, 2006.

[105] J. Khodjaev, Y. Park, and A. Saeed Malik, "Survey of NLOS identification and error mitigation problems in UWB-based positioning algorithms for dense environments," *Ann. of Telecommun.*, vol. 65, pp. 301–311, Aug. 2010.

[106] S. Maran, W. M. Gifford, and H. Wymeersch, "NLOS identification and mitigation for localization based on UWB experimental data," *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, 2010.

[107] K. Yu, Y. J. Guo, and S. Member, "Statistical NLOS identification based on AOA, TOA, and signal strength," *IEEE Trans. Veh. Technol.*, vol. 58, no. 1, pp. 274–286, 2009.

[108] P. Chen, "A non-line-of-sight error mitigation algorithm in location estimation," in *Proc. IEEE Wirel. Commun. Netw. Conf. (WCNC'99)*, (New Orleans, LA, USA), pp. 316–320, 1999.

[109] X. Li, "An iterative NLOS mitigation algorithm for location estimation in sensor networks," in *Proc. 15th IST Mob. Wirel. Commun. Summit*, (Miconos, Greece), pp. 1–5, 2006.

[110] S. Nawaz and N. Trigoni, "Convex programming based robust localization in NLOS prone cluttered environments," in *Proc. 10th Int. Conf. Informat. Process. Sens. Netw. (IPSN'11)*, (Chicago, IL, USA), pp. 318–329, 2011.

[111] P. Bahl and V. N. Padmanabhan, "RADAR : An in-building RF-based user location and tracking system," in *Proc. 19th Annu. IEEE Int. Conf. Comput. Commun. (INFOCOM'00)*, (Tel-Aviv, Israel), pp. 775–784, 2000.

[112] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, J. Tabert, P. Powledge, G. Borriello, and B. Schilit, "Place lab: device positioning using radio beacons in the wild," in *Proc. 3rd Int. Conf. Pervasive Comput. (PERVASIVE'05)*, (Berlin, Germany), pp. 116–133, 2005.

[113] M. Youssef and A. Agrawala, "The Horus WLAN Location Determination System," in *Proc. 3rd Int. Conf. Mob. Syst. (MobiSys'05)*, (New York, NY, USA), pp. 205–218, 2005.

[114] G.-Y. Park, M.-H. Jeon, and C.-H. Oh, "Indoor wireless localization using Kalman filtering in fingerprinting-based location estimation system," *Int. J. Softw. Engineer App.*, vol. 8, pp. 235–246, Jan. 2014.

[115] F. Subhan, H. Hasbullah, and K. Ashraf, "Kalman filter-based hybrid indoor position estimation technique in bluetooth networks," *Int. J. Nav. Observ.*, vol. 2013, no. TT, pp. 1–13, 2013.

[116] D. Liu, Y. Xiong, and J. Ma, "Exploit Kalman filter to improve fingerprint-based indoor localization," in *Proc. Int. Conf. Comput. Sci. Netw. Technol. (ICCSNT'11)*, (Harbin, China), pp. 2290–2293, Ieee, Dec. 2011.

[117] M. Abdullah, K. Hatta, and E. Jeganathan, "Adaptive WiFi fingerprinting for location approximation," *Int. J. Comput. Info. Syst. Control Engineer*, vol. 7, no. 2, pp. 181–188, 2013.

[118] S. Ali-Loytty, T. Perala, V. Honkavirta, and R. Piche, "Fingerprint kalman filter in indoor positioning applications," in *Proc. 3rd IEEE Multi-conf. Syst. Control*, (Saint Petersburg, Russia), pp. 1678–1683, 2009.

[119] V. Honkavirta, T. Perala, S. Aliloytty, and R. Piche, "A comparative survey of WLAN location fingerprinting methods," in *Proc. 6th Workshop Pos., Nav., Commun. (WPNC'09)*, (Hannover, Germany), pp. 243–251, 2009.

[120] C. Figuera, I. Mora, and A. Guerrero, "Nonparametric model comparison and uncertainty evaluation for signal strength indoor location," *IEEE Trans. Mob. Comp.*, vol. 8, no. 9, pp. 1250–1264, 2009.

[121] J. Gu, S. Chen, and T. Sun, "Localization with incompletely paired fata in complex wireless sensor network," *IEEE Trans. Wirel. Commun.*, vol. 10, pp. 2841–2849, Sept. 2011.

[122] I. Vallivaara, J. Haverinen, A. Kemppainen, and J. Roning, "Simultaneous localization and mapping using ambient magnetic field," in *Proc. IEEE Conf. Multisensor Fusion, Integration (MFI'10)*, (Salt Lake City, UT, USA), pp. 14–19, Sept. 2010.

[123] W. H. K. de Vries, H. E. J. Veeger, C. T. M. Baten, and F. C. T. van der Helm, "Magnetic distortion in motion labs, implications for validating inertial magnetic sensors.," *Gait and posture*, vol. 29, pp. 535–41, June 2009.

[124] E. Vildjiounaite, E.-J. Malm, J. Kaartinen, and P. Alahuhta, "Location estimation indoors by means of small computing power devices, accelerometers, magnetic sensors, and map knowledge," in *Proc. 1st Int. Conf. Pervasive Comput. (PerCom'02)*, (Zürich, Switzerland,), pp. 211–224, 2002.

[125] K. P. Subbu, B. Gozick, and R. Dantu, "LocateMe: Magnetic-fields-based indoor localization using smartphones," *ACM Trans. Intell. Syst. Tchnol.*, vol. 4, no. 4, pp. 1–27, 2013.

[126] P. Subbu, *Indoor localization using magnetic fields.* PhD thesis, University of North Texus, 2011.

[127] W. Storms, *Magnetic field aided indoor navigation.* PhD thesis, Air University, 2009.

[128] H. Lu, J. Zhao, X. Li, and J. Li, "A new method of double electric compass for localization," in *Proc. 6th World Congress Intell. Control, Autom. (WCICA'06)*, (Dlian, China), pp. 5277–5281, 2006.

[129] B. Li, T. Gallagher, A. Dempster, and C. Rizos, "How feasible is the use of magnetic field alone for indoor positioning?," in *Proc. Int. Conf. Indoor Pos. Indoor Nav. (IPIN'12)*, (Sydney, NSW), pp. 1–11, 2012.

[130] B. Gozick, K. P. Subbu, R. Dantu, and T. Maeshiro, "Magnetic maps for indoor navigation," *IEEE Trans. Instrument. Measure.*, vol. 60, pp. 3883–3891, Dec. 2011.

[131] M. Frassl, M. Angermann, M. Lichtenstern, P. Robertson, B. J. Julian, M. Doniec, and A. Motivation, "Magnetic Maps of Indoor Environments for Precise Localization of Legged and Non-legged Locomotion," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS'13)*, (Tokyo, Japan), pp. 913–920, 2013.

[132] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman, "Indoor location sensing using geo-magnetism," in *Proc. 9th Int. Conf. Mob. Syst. App. Serv. (MobiSys'11)*, (Bethesda, Maryland, USA), p. 141, 2011.

[133] M. Angermann, M. Frassl, M. Doniec, B. Julian, and P. Robertson, "Characterization of the indoor magnetic field for applications in localization and mapping," in *Proc. Int. Conf. Indoor Pos. Indoor Nav. (IPIN'12)*, (Sydney, NSW), pp. 1–9, 2012.

[134] P. Robertson, M. Frassl, M. Angermann, M. Doniec, B. J. Julian, M. Garcia Puyol, M. Khider, M. Lichtenstern, and L. Bruno, "Simultaneous localization and mapping for pedestrians using distortions of the local magnetic field intensity in large indoor environments," in *Proc. Int. Conf. Indoor Pos. Indoor Nav. (IPIN'13)*, (Montbeliard-Belfort), pp. 1–10, Oct. 2013.

[135] E. L. Grand and S. Thrun, "3 - Axis Magnetic Field Mapping and Fusion for Indoor Localization," in *Proc. IEEE Int. Conf. Multisensor Fusion, Integration for Intell. Syst. (MFI'12)*, (Hamburg, Germany), pp. 358–364, 2012.

[136] T. H. Riehle, S. M. Anderson, P. a. Lichter, J. P. Condon, S. I. Sheikh, and D. S. Hedin, "Indoor waypoint navigation via magnetic anomalies.," in *Proc. 33rd Ann. Int. Conf. IEEE Engineer. Med., Biol. Society. (EMBS'11)*, (Boston, MA, USA), pp. 5315–5318, Jan. 2011.

[137] T. H. Riehle, S. M. Anderson, P. a. Lichter, N. a. Giudice, S. I. Sheikh, R. J. Knuesel, D. T. Kollmann, and D. S. Hedin, "Indoor magnetic navigation for the blind.," in *Proc. 34th Ann. Int. Conf. Engineer. Med., Biol. Society. (EMBS'12)*, (San Diege, CA, USA), pp. 1972–1975, Jan. 2012.

[138] W. Storms, J. Shockley, and J. Raquet, "Magnetic field navigation in an indoor environment," in *Proc. Ubiq. Pos. Indoor Nav. Loc. Based Serv. (UPINLBS'10)*, (Kirkkonummi, Finland), pp. 1–10, 2010.

[139] G. Glanzer and U. Walder, "Self-contained indoor pedestrian navigation by means of human motion analysis and magnetic field mapping," in *Proc. Workshop on Pos. Nav. Commun. (WPNC'10)*, (Dresden, Germany), pp. 303–307, Mar. 2010.

[140] J. Burnett and P. D. Yaping, "Mitigation of extremely low frequency magnetic ÿelds from electrical installations in high-rise buildings," *Building and Environments*, vol. 37, no. 8-9, pp. 769–775, 2002.

[141] G. Casinovi, A. Geri, and G. Veca, "Magnetic field near a concrete wall during a lightning stroke," *IEEE Trans. Magnetics*, vol. 25, no. 5, pp. 4006–4008, 1989.

[142] K. Yamazaki, K. Kato, K. Ono, H. Saegusa, K. Tokunaga, Y. Iida, S. Yamamoto, K. Ashiho, K. Fujiwara, and N. Takahashi, "Analysis of magnetic disturbance due to buildings," *IEEE Trans. Magnetics*, vol. 39, pp. 3226–3228, Sept. 2003.

[143] J. Haverinen and A. Kemppainen, "Global indoor self-localization based on the ambient magnetic field," *Robot. Autonom. Syst.*, vol. 57, pp. 1028–1035, Oct. 2009.

[144] C. E. White, D. Bernstein, and A. L. Kornhauser, "Some map matching algorithms for personal navigation assistants," *Transport. Research Part C: Emerg. Technol.*, vol. 8, pp. 91–108, Feb. 2000.

[145] S. Brakatsoulas, D. Pfoser, S. Randall, and C. Wenk, "On map-matching vehicle tracking data," in *Proc. 31st Int. Conf. Very Large Databases (VLDB'05)*, (Trondheim, Norway), pp. 853–864, 2005.

[146] C. Wenk, R. Salas, and D. Pfoser, "Addressing the need for map-matching speed: Localizing global curve-matching algorithms," in *Proc. 18th Int. Conf. Sci. Stat. Database Manage. (SSDBM'06)*, (Vienna, Austria), pp. 379 – 388, 2006.

[147] S. S. Chawathe, "Segment-Based Map Matching," in *Proc. IEEE Intell. Veh. Symp. (IV'07)*, (Istanbul, Turkey), pp. 1190–1197, June 2007.

[148] L. Xi, Q. Liu, M. Li, and Z. Liu, "Map Matching Algorithm and Its Application," in *Proc. Intell. Syst. Knowl. Engineer. (ISKE'07)*, (Paris, France), pp. 1–7, 2007.

[149] S. H. Shin, "New Map-Matching Algorithm Using Virtual Track for Pedestrian Dead Reckoning," *ETRI J.*, vol. 32, pp. 891–900, Dec. 2010.

[150] M. Weber, L. Liu, and K. Jones, "On map matching of wireless positioning data: a selective look-ahead approach," in *Proc. ACM 18th SIGSPATIAL Int. Conf. Adv. Geograph. Info. Syst. (SIGSPATIAL'14)*, (San Jose, CA, USA), pp. 290–299, 2010.

[151] A. Javanmard, M. Haridasan, and L. Zhang, "Multi-track map matching," in *Proc. 20th Int. Conf Adv. Geographic Informat. Syst. (SIGSPATIAL'12)*, (Redondo Beach, CA, USA), pp. 394–397, 2012.

[152] J. Krumm, J. Letchner, and E. Horvitz, "Map matching with travel time constraints," in *Proc. Soc. Automotive Engineers World Congress (SAE'07)*, (Detroit, USA), pp. 1–11, 2007.

[153] P. Newson and J. Krumm, "Hidden Markov map matching through noise and sparseness," in *Proc. 17th ACM SIGSPATIAL Int. Conf. Adv. Geographic Infomat. Syst. (GIS'09)*, (Seattle, WA, USA), pp. 336–343, 2009.

[154] A. Thiagarajan and L. Ravindranath, "VTrack: accurate, energy-aware road traffic delay estimation using mobile phones," in *Proc. 7th ACM Conf. Embed. Netw. Sens. Syst. (SenSys'11)*, (Berkeley, CA, USA), pp. 1–14, 2009.

[155] A. Thiagarajan, L. Ravindranath, H. Balakrishnan, S. Madden, and L. Girod, "Accurate, low-energy trajectory mapping for mobile devices," in *Proc. 8th USENIX Symp. Netw. Syst. Design Implement. (NSDI'11)*, (Boston, MA, USA), pp. 1–14, 2011.

[156] S. Guha, K. Plarre, D. Lissner, S. Mitra, B. Krishna, P. Dutta, and S. Kumar, "AutoWitness: locating and tracking stolen property while tolerating GPS and radio outages," in *Proc. 8th ACM Conf. Embed. Netw. Sens. Syst. (SenSys'10)*, (Zurich, Switzerland), pp. 29–42, 2010.

[157] S. Syed and M. E. Cannon, "Fuzzy logic-based map matching algorithm for vehicle navigation system in urban canyons," in *Proc. ION Nation. Tech. Meet. (NTM'07)*, (San Diego, CA, USA), pp. 1–12, 2004.

[158] A. Dewandaru, A. M. Said, and A. N. Matori, "A novel map-matching algorithm to improve Vehicle Tracking System accuracy," in *Int. Conf. Intell. Adv. Syst. (ICIAS'07)*, (Kuala Lumpur, Malaysia), pp. 177–181, Ieee, Nov. 2007.

[159] M. Jabbour, P. Bonnifait, and V. Cherfaoui, "Map-Matching Integrity Using Multi-Hypothesis Road Tracking," *J. Intellig. Transport. Syst. Technol., Plan. Operat.*, vol. 12, no. 4, pp. 189–201, 2008.

[160] J. Yuan, Y. Zheng, and C. Zhang, "An interactive-voting based map matching algorithm," in *Proc. 11th Int. Conf. Mob. Data Manage. (MDM'10)*, (Washington, USA), pp. 1–10, 2010.

[161] C. M. Bishop, *Pattern recognition and machine learning*. Springer, 1st ed., 2006.

[162] J. Seitz, T. Vaupel, S. Meyer, J. Gutirrez Boronat, and J. Thielecke, "A Hidden Markov Model for pedestrian navigation," in *Proc. 7th Workshop Position. Nav. Commun. (WPNC'10)*, (Dresden, Germany), pp. 120–127, 2010.

[163] H. Wen, Z. Xiao, A. Markham, and N. Trigoni, "On Assessing the Accuracy of Positioning Systems in Indoor Environments," in *Proc. 10th European Conf. Wirel. Sens. Netw. (EWSN'14)*, (Ghent, Belgium), pp. 1–17, 2013.

[164] W. Burgard, D. Fox, and S. Thrun, "Markov localization for mobile robots in dynamic environments," *J. Artificial Intell. Research*, vol. 11, pp. 391–427, 2011.

[165] S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp, "A tutorial on particle filters for online nonlinear / non-Gaussian Bayesian tracking," *IEEE T. Signal Process.*, vol. 50, no. 2, pp. 174–188, 2002.

[166] O. Woodman and R. Harle, "Pedestrian localisation for indoor environments," in *Proc. 10th Int. Conf. Ubi. Comput. (UbiComp'08)*, (Seoul, Korea), pp. 114–123, 2008.

[167] S. Beauregard, Widyawan, and M. Klepal, "Indoor PDR performance enhancement using minimal map information and particle filters," in *Proc. IEEE/ION Pos. Location Nav. Symp. (PLANS'08)*, (Monterey, CA, USA), pp. 141–147, 2008.

[168] M. M. Atia, M. J. Korenberg, and A. Noureldin, "Particle-Filter-Based WiFi-Aided Reduced Inertial Sensors Navigation System for Indoor and GPS-Denied Environments," *Int. J. Nav. Observ.*, vol. 2012, pp. 1–12, 2012.

[169] W. Chai, C. Chen, E. Edwan, J. Zhang, and O. Loffeld, "INS/Wi-Fi based indoor navigation using adaptive Kalman filtering and vehicle constraints," in *Proc. 9th Workshop Pos., Nav. Commun. (WPNC'12)*, (Dresden, Germany), pp. 36–41, 2012.

[170] D. Obradovic, H. Lenz, and M. Schupfner, "Fusion of map and sensor data in a modern car navigation system," *J. VLSI Signal Process.*, vol. 45, pp. 111–122, Dec. 2006.

[171] A. S. Paul and E. A. Wan, "RSSI-based indoor localization and tracking using sigma-point Kalman smoothers," *IEEE J. Sel. Topics Signal Process.*, vol. 3, no. 5, pp. 860–873, 2009.

[172] B. Ferris, D. Fox, and N. Lawrence, "WiFi-SLAM using Gaussian process latent variable models," in *Proc. 20th Int. Joint Conf. Artificial Intell. (IJCAI'07)*, (Hyderabad, India), pp. 2480–2485, 2007.

[173] P. Robertson, M. Angermann, and M. Khider, "Improving Simultaneous Localization and Mapping for pedestrian navigation and automatic mapping of buildings by using online human-based feature labeling," in *Proc. IEEE/ION Pos. Location Nav. Symp. (PLANS'10)*, (Indian Wells, CA, USA), pp. 365–374, 2010.

[174] P. Robertson, M. Angermann, B. Krach, and M. Khider, "SLAM Dance: Inertial-Based Joint Mapping and Positioning for Pedestrian Navigation," in *Proc. ION GNSS+*, (Portland, Oregon), pp. 1–12, 2010.

[175] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. R. Choudhury, "No need to war-drive: unsupervised indoor localization," in *Proc.10th Int. Conf. Mob. Syst. App. Serv. (MobiSys'12)*, (Low Wood Bay, Lake District, UK), pp. 197–210, 2012.

[176] H. Shin, Y. Chon, and H. Cha, "Unsupervised Construction of an Indoor Floor Plan Using a Smartphone," *IEEE Trans. Syst. Man Cyber., Part C (App. and Rev.)*, vol. 42, pp. 889–898, Nov. 2012.

[177] Y. Kim, Y. Chon, and H. Cha, "Smartphone-Based Collaborative and Autonomous Radio Fingerprinting," *IEEE Trans. Syst. Man Cyber., Part C (App. and Rev.)*, vol. 42, pp. 112–122, Jan. 2012.

[178] Y. Kim, H. Shin, and H. Cha, "Smartphone-based Wi-Fi Pedestrian-Tracking System Tolerating the RSS Variance Problem," in *Proc. 18th IEEE Int. Conf. Pervasive Comp. Commun. (PerCom'12)*, (Lugano, Switzerland), pp. 11–19, 2012.

[179] J. Huang, D. Millman, M. Quigley, D. Stavens, S. Thrun, and A. Aggarwal, "Efficient, generalized indoor WiFi GraphSLAM," in *Proc. IEEE Int. Conf. Robot. Automat.*, (Shanghai, China), pp. 1038–1043, May 2011.

[180] Z. Yang, C. Wu, and Y. Liu, "Locating in fingerprint space: wireless indoor localization with little human intervention," in *Proc. 18th Ann. Int. Conf. Mob. Comput. Netw. (MobiCom'12)*, (Istanbul, Turkey), pp. 1–12, 2012.

[181] J. Koo and H. Cha, "Autonomous construction of a WiFi access point map using multidimensional scaling," in *Proc. 9th IEEE Int. Conf. Pervasive Comp. Commun. (PerCom'12)*, (San Francisco, CA, USA), pp. 1–18, 2011.

[182] G. Grisetti, R. Kummerle, C. Stachniss, and W. Burgard, "A Tutorial on Graph-Based SLAM," *IEEE Intell. Transport. Syst. Mag.*, vol. 2, no. 4, pp. 31–43, 2010.

[183] D. Wolf and G. Sukhatme, "Online simultaneous localization and mapping in dynamic environments," in *Proc. IEEE Int. Conf. Robot. Automat. (ICRA '04)*, (New Orleans, LA, USA), pp. 1301–1307, Ieee, 2004.

[184] A. Eliazar and R. Parr, "DP-SLAM: fast, robust simultaneous localization and mapping without predetermined landmarks," in *Proc. 18th Int. Joint Conf. Artificial Intell. (IJCAI'08)*, (Acapulco, Mexico), pp. 1135–1142, 2003.

[185] M. Angermann and P. Robertson, "FootSLAM: Pedestrian Simultaneous Localization and Mapping Without Exteroceptive SensorsHitchhiking on Human Perception and Cognition," *Proc. IEEE*, vol. 100, no. Special Centennial Issue, pp. 1840–1848, 2012.

[186] C. G. Koay and P. J. Basser, "Analytically exact correction scheme for signal extraction from noisy magnitude MR signals.," *J. Magn. Resonance*, vol. 179, pp. 317–22, Apr. 2006.

[187] C. Tepedelenlioglu, A. Abdi, and G. B. Giannakis, "The Ricean K factor: estimation and performance analysis," *IEEE Trans. Wirel. Commun.*, vol. 2, no. 4, pp. 799–810, 2003.

[188] R. Caruana and A. Niculescu-Mizil, "An empirical comparison of supervised learning algorithms," in *Proc. 23rd Int. Conf. Machine learning (ICML'06)*, (Pittsburgh, PA, USA), pp. 161–168, 2006.

[189] M. Buhmann, *Radial basis functions: theory and implementations*. Cambridge, UK: Cambridge University Press, 1st ed., 2003.

[190] J. Suykens and J. Vandewalle, "Least squares support vector machine classifiers," *Neural Process. Lett.*, vol. 9, no. 3, pp. 293–300, 1999.

[191] H. W. Kuhn and A. W. Tucker, "Nonlinear Programming," in *Proc. 2nd Berkeley Symp. Math. Statistics, Prob.*, (Berkeley, CA, USA), pp. 481–492, 1951.

[192] C. E. Rasmussen and C. K. I. Williams, *Gaussian processes for machine learning*. Cambridge, MA: MIT Press, 2nd ed., Apr. 2006.

[193] E. L. Snelson, *Flexible and efficient Gaussian process models for machine learning*. Ph.d, University College London, 2007.

[194] C. M. Bishop, *Neural networks for pattern recognition*. Oxford, UK: Oxford University Press, Dec. 1997.

[195] C. Williams and D. Barber, "Bayesian classification with Gaussian processes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 20, no. 12, pp. 1342–1351, 1998.

[196] H. Nickisch and C. Rasmussen, "Approximations for binary Gaussian process classification," *J. Mach. Learn. Research.*, vol. 9, no. 2008, pp. 2035–2078, 2008.

[197] K. Cheung and R. D. Murch, "A new empirical model for indoor propagation prediction," *IEEE Trans. Veh. Technol.*, vol. 47, no. 3, pp. 996–1001, 1998.

[198] W. Honcharenko, H. L. Bertoni, and J. Dailing, "Mechanisms governing propagation between different floors in buildings," *IEEE Trans. Antennas Propag.*, vol. 41, no. 6, pp. 787–790, 1993.

[199] W. Honcharenko, H. Bertoni, J. Dailing, J. Qian, and H. Yee, "Mechanisms governing UHF propagation on single floors in modern office buildings," *IEEE Trans. Veh. Technol.*, vol. 41, no. 4, pp. 496–504, 1992.

[200] X. Zhou, Z. Zhang, G. Wang, and X. Yu, "Practical conflict graphs for dynamic spectrum distribution," in *Proc. ACM Int. Conf. Measure. Model. Computer Syst. (SIGMETRICS'13)*, (Pittsburgh, PA, USA), pp. 5–16, 2013.

[201] A. Dempster, N. Laird, and D. Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *J. Royal Statistical Society*, vol. 39, no. 1, pp. 1–38, 1977.

[202] N. K. Sharma, "A weighted center of mass based trilateration approach for locating wireless devices in indoor environment," in *Proc. Int. workshop Mob. Manage. Wirel. Access (MobiWac'06)*, (New York, NY, USA), pp. 112–115, ACM Press, 2006.

[203] H. Shi, X. Li, Y. Shang, and D. Ma, "Cramer-Rao bound analysis of quantized RSSI based localization in wireless sensor networks," in *Proc. 11th Int. Conf. Parallel Distrib. Syst. (ICPADS'05)*, (Fukuoka, Japan), pp. 32–36, 2005.

[204] D. Mizell, "Using gravity to estimate accelerometer orientation," in *Proc. 7th IEEE Int. Symp. Wearable Comput. (ISWC'03)*, (New York, NY, USA), pp. 252–253, 2003.

[205] H. Lu, J. Yang, Z. Liu, N. D. Lane, T. Choudhury, and A. T. Campbell, "The Jigsaw continuous sensing engine for mobile phone applications," in *Proc. 8th ACM Conf. Embedded Netw. Sensor Syst. (SenSys'10)*, (New York, NY, USA), pp. 71–84, ACM Press, 2010.

[206] S. Hemminki, P. Nurmi, and S. Tarkoma, "Accelerometer-based transportation mode detection on smartphones," in *Proc. 11th ACM Conf. Embedded Netw. Sensor Syst. (Sensys'13)*, (New York, NY, USA), pp. 1–14, 2013.

[207] P. Zhou, M. Li, and G. Shen, "Use it free: Instantly knowing your phone attitude," in *Proc. 20th Ann. Int. Conf. Mob. Comput. Netw. (MobiCom'14)*, (New York, NY, USA), pp. 605–616, 2014.

[208] V. Renaudin, M. Susi, and G. Lachapelle, "Step length estimation using handheld inertial sensors," *Sensors*, vol. 12, no. 7, pp. 8507–8525, 2012.

[209] L. Gupta, D. L. Molfese, R. Tammana, and P. G. Simos, "Nonlinear alignment and averaging for estimating the evoked potential," *IEEE Trans. Biomedical Engineer.*, vol. 43, no. 4, pp. 348–356, 1996.

[210] J. D. Lafferty, A. McCallum, and F. C. N. Pereira, "Conditional Random Fields: Probabilistic Models for Segmenting and Labeling Sequence Data," in *Proc. 18th Int. Conf. Machine Learning (ICML'01)*, (Bellevue. WA, USA), pp. 282–289, 2001.

[211] S. Baloch, E. Cheng, Y. Zhu, A. Mohamed, H. Ling, and T. Fang, "Shape based conditional random fields for segmenting intracranial aneurysms," *Lecture Notes in Comput., Vision, Biomechanics*, vol. 3, no. TT, pp. 55–67, 2013.

[212] F. Ramos, D. Fox, and D.-W. Hugh, "CRF-Matching: Conditional Random Fields for Feature-Based Scan Matching.," in *Proc. Robotics Sci. Syst. (RSS'07)*, (Atlanta, GA, USA), pp. 1–8, 2007.

[213] R. Klinger and K. Tomanek, "Classical Probabilistic Models and Conditional Random Fields," (Dortmund, Germany), pp. 1–32, Technische Universitat Dortmund, 2007.

[214] J. Seitz, T. Vaupel, J. Jahn, S. Meyer, J. G. Boronat, and J. Thielecke, "A Hidden Markov Model for urban navigation based on fingerprinting and pedestrian dead reckoning," in *Proc. 13th Conf. Informat. Fusion (FUSION'10)*, (Edinburgh, UK), pp. 1–8, 2010.

[215] S. Hilsenbeck, D. Bobkov, G. Schroth, R. Huitl, and E. Steinbach, "Graph-based data fusion of pedometer and WiFi measurements for mobile indoor positioning," in *Proc. ACM Int. Joint Conf. Pervasive, Ubiquitous Comput. (UbiComp'14)*, (Acapulco, Mexico), pp. 147–158, 2014.

[216] Z. Xiao, H. Wen, A. Markham, and N. Trigoni, "Lightweight map matching for indoor localization using conditional random fields," in *Proc. Int. Conf. Info. Process. Sens. Netw. (IPSN'14)*, (Berlin, Germany), pp. 131–142, 2014.

[217] H. Trinh, *A machine learning approach to recovery of scene geometry from images*. Ph.d thesis, Toyota Technological Institute at Chicago, 2010.

[218] K. Chintalapudi, A. Padmanabha, and V. Padmanabhan, "Indoor localization without the pain," in *Proc. 16th Ann. Int. Conf. Mob. Comput. Netw. (MobiCom'10)*, (Chicago, IL, USA), pp. 173–184, 2010.

[219] R. V. D. Merwe, *Sigma-point Kalman filters for probabilistic inference in dynamic state-space models*. Ph.d thesis, Oriegon Health & Science University, 2004.