# TOWARD SECURE AND USABLE INDOOR POSITIONING

by

Yunzhi Li

A Dissertation submitted to the Faculty of the University of Delaware in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science

Summer 2022

**TOWARD SECURE AND USABLE INDOOR POSITIONING**

by

Yunzhi Li

Approved: _____
Rudolf Eigenmann, Ph.D.
Interim Chair of the Department of Computer and Information Sciences

Approved: _____
Levi T. Thompson, Ph.D.
Dean of the College of Engineering

Approved: _____
Louis F. Rossi, Ph.D.
Vice Provost for Graduate and Professional Education and
Dean of the Graduate College

I certify that I have read this Dissertation and that in my opinion it meets the academic and professional standard required by the University as a Dissertation for the degree of Doctor of Philosophy.

Signed: _____

Rui Zhang, Ph.D.
Professor in charge of Dissertation

I certify that I have read this Dissertation and that in my opinion it meets the academic and professional standard required by the University as a Dissertation for the degree of Doctor of Philosophy.

Signed: _____

Chien-Chung Shen, Ph.D.
Member of Dissertation committee

I certify that I have read this Dissertation and that in my opinion it meets the academic and professional standard required by the University as a Dissertation for the degree of Doctor of Philosophy.

Signed: _____

Matthew Louis Mauriello, Ph.D.
Member of Dissertation committee

I certify that I have read this Dissertation and that in my opinion it meets the academic and professional standard required by the University as a Dissertation for the degree of Doctor of Philosophy.

Signed: _____

Jie Yang, Ph.D.
Member of Dissertation committee

## ACKNOWLEDGEMENTS

First, I would like to express my sincere gratitude to my advisor Prof. Rui Zhang for his continuous support and guidance to my study and research during my Ph.D. study at the University of Delaware. He is a brilliant mentor who guided me through this journey. Through his guidance, I have learned how to do research. He always encouraged me to explore varied research interests and think deeply about research problems. I learned how to read papers, identify practical and meaningful research problems, conceive novel ideas from shallower to deeper, and turn the ideas into complete technical solutions. He also helped me with academic writing, experiment design, and data collection and analysis. I could not have imagined having a better advisor for my Ph.D. study.

Second, I would like to thank my other committee members: Prof. Chien-Chung Shen, Prof. Matthew Louis Mauriello, and Prof Jie Yang from Florida State University. Their immense knowledge and plentiful experience enriched my project. My sincere gratitude to Prof. Chien-Chung Shen for all the courses he taught me, for the report he shared with me and for his guidance to my research. I greatly appreciate Prof. Matthew Louis Mauriello's for his precious suggestions and the inspiration to my project. A sincere thank you to Prof. Jie Yang for his constructive and brilliant comments to my research and experiments.

Third, I have received the help and support from a great number of people, including, but not limited to, my colleagues Dr. Yidan Hu, Dr. Aishah Aseeri, Yukun Dong, Zheyuan Liu, and Tianye Ma. They provided me with a friendly and constructive environment. They also shared valuable ideas on my research projects. I would like to express great appreciation to them.

Last but not least, my deepest gratitude goes to my family. To my wife Chang whose love, strength and advice accompanied me. To my parents for their love, understanding and support throughout my Ph.D. studies.

## TABLE OF CONTENTS

## LIST OF TABLES

## LIST OF FIGURES

## ABSTRACT

Recent years have witnessed growing interests from both academia and industry in developing effective Indoor Positioning Systems (IPSes). An IPS allows users to learn their locations and navigate in large unfamiliar indoor venues such as shopping malls, hospitals, and airports, where GPS signals are absent or unreliable. Despite significant efforts made in the past, existing IPSes suffer from two major limitations. First, most existing IPSes are designed for improving positioning accuracy without considering potential malicious attacks. Second, many IPSes require costly infrastructure upgrades, which may hinder their wide adoption and deployment. This dissertation tackles three challenges in developing secure and usable IPSes to pave the way for their wide adoption.

First, we introduce MV-IPS, a novel WiFi-based IPS that can achieve high positioning accuracy regardless of the presence of malicious attacks. It is well known that a malicious attacker can severely degrade the positioning accuracy of WiFi-based IPSes by deploying fake WiFi Access Points (APs) to impersonate legitimate ones. We observe that existing WiFi-based IPSes suffers from an inherent trade-off between positioning accuracy and attack resilience, i.e., the strong attack resilience comes at the cost of reduced positioning accuracy in the absence of attacks. To tackle this challenge, we develop a novel multi-vote WiFi-based IPS without suffering from such trade-off. In the proposed IPS, we view APs as voters and determine a user's location by jointly considering APs' weighted votes. Doing so can effectively limit the impact of potential malicious APs without sacrificing the positioning accuracy in the absence of attacks.

Second, we introduce ImPos, a novel image-based IPS for indoor venues where WiFi infrastructure is unavailable, which allows a user to learn her location by taking

a small number of photos of the surrounding using smartphones. While several image-based IPSes have been proposed in the literature, they all suffer from low positioning accuracy due to inaccurate angle measurements from smartphone sensor readings. To address their limitation, our system greatly improves the angle measurement through image analysis and effectively copes with varying number of landmarks. Detailed experimental studies confirm the significant advantages of our IPS over prior solutions.

Last but not the least, we develop CITS, a novel RSS-based continuous indoor tracking system. Existing RSS-based IPSes suffer from the ambiguity of RSS fingerprints and device homogeneity which could greatly limit its positioning accuracy in practice. To tackle these challenges, CITS estimates a user's location via differential RSS fingerprint matching and path tracking. Our experiment results confirm that CITS can achieve high positioning accuracy in the presence of fingerprint ambiguity and device diversity and significantly outperform prior solutions.

# Chapter 1

# INTRODUCTION

Indoor positioning systems(IPSes) have attracted interest from both academia and industry in recent years. In large indoor vennues covered with ceilings such as airports, hospitals and shopping malls, GPS signals are unavailable or unreliable. Therefore, the user's location cannot be accurately determined. Due to the necessary of indoor positioning services, researchers are trying to propose efficient and convenient IPSes. According to some existing investigations such as [1], the market of IPSes is predicted to reach 256.39 billion in 2028.

To provide an efficient and convenient Indoor positioning service to the user, existing works mainly utilize user's smartphones as well as some indoor environment information. For example, [2, 3, 4] use WiFi signals, [5, 6, 7] uses blue-tooth signals,[8, 9, 10] use ambient sounds to locate the users. This introduces the first limitation that obstacles the development of IPSes. The service provider is required to install extra infrastructure. This is because some indoor environment information is not general in every indoor venue. For example, WiFi-based IPSes and blue-tooth based IPSes require the service provider to install multiple access points(APs) to cover the whole indoor area with WiFi or blue-tooth signals. The sound-based IPSes require microphones to receive the sound signals. Apart from the infrastructure at the server end, there is also an heterogeneous device problem at the user end. Different model of smartphones have different capabilities to collect the indoor environment information,i.e., different smartphone models measure different received signal strengths(RSSes) at the same location and same time. This also influences the accuracy of IPSes. Motivated by the above problems, to design an usable IPS that satisfies different environments and smartphone models is important.

With the rapid development of IPSes, the security issues also attract people. Some recent works analyze the security of existing IPSes. Taking the WiFi-based IPSes as an example, [11] mentioned that for WiFi-based IPS, by placing some absorbing materials between the user and the WiFi access point, the localization accuracy can be highly influenced. [12] proposed a novel WiFi signal attack which uses some commercial WiFI routers to impersonate benign WiFi access points then further degrade the locating accuracy. More recent work [13] mentioned that the IPS can even be misled into a desired location. Therefore, considering the attack resilience when designing a new IPS is important.

Observing the two important problems about usable and secure IPS, this dissertation aims to tackle three challenges by developing usable and secure IPSes.

### 1.1 Multi-vote WiFi-based Indoor Localization

The WiFi received signal strength(RSS)-based IPS is vulnerable to RSS attacks at the physical layer. Previous work[12, 14, 15] mentioned that by placing some absorbing materials between the WiFi access point and the user, the performance of WiFi-based IPS is obviously degraded. [11, 13] mentioned that by using some commercial WiFi routers to impersonate the benign WiFi access points, the IPS can be misled to return a wrong location to the user. Even though some existing works[12, 14, 15, 16, 13] proposed defense to this problem, they sacrifice the locating accuracy for attack resilience. To design a secure WiFi-based IPS without suffering such trade-off between localization accuracy and attack resilience, we propose a multi-vote WiFi-based Indoor localization system. Inspired by bounda count voting scheme, we view each AP as a voter to determine the user's location. Since different AP plays different significance in voting, we assign each AP a weight. By doing so, the impact of malicious AP to the returned location is limited and the overall locating accuracy is guaranteed.

### 1.2 Image-based Indoor Localization

It is impossible to guarantee that the necessary indoor information is available in all indoor venues. Even WiFi signal, which is the most popular signal utilized by IPSes, cannot be guaranteed to cover every corner of the building. Therefore, previous works[17, 18] proposed image-based IPSes which utilize smartphone cameras to collect the images for surrounding landmarks. The server locates the user by recognizing these landmarks through images, then locate the user in a triangulation way with the angles obtained from the smartphone sensor readings. The key advantage of image-based IPSes is they do not rely on the extra infrastructures in the indoor environment. However, previous image-based IPSes suffers from low positioning accuracy due to the inaccurate angle measurement. To address their limitation, we propose a vision-based IPS. By analyzing the orientation location of the landmark on the image, we correct the angle which obtained from the sensor readings. Besides, we observe that the number of landmarks is strictly limited to three in triangulation technique, which also impacts the system performance. We formulate and solve an optimization problem which considers a varying number of landmarks to further improve system accuracy. We validate the effectiveness of the new technique through detailed experiments.

### 1.3 WiFi-based Continuous Indoor Tracking

The ambiguity of RSS fingerprints and the heterogeneity of user's smartphone impact the locating accuracy of IPS significantly. First, prior studies such as [13] have shown that multiple locations could have the same RSS fingerprint. As a result, it is difficult for the IPS server to distinguish these locations solely based on the user's RSS measurement. Moreover, it has been well known that in WiFi-based IPS, the RSSes measured by different models of smartphones could be different even at the same location and same time [19, 20, 21, 22]. In particular, the device used by the IPS operator to collect the RSS fingerprints may be different from the mobile devices used by the users during the online navigation phase, which means that the RSS measurement collected by the user may not match the RSS fingerprint at the location. To tackle these

two challenges, we introduce the design and evaluation of CITS, a novel continuous RSS-based indoor tracking system that can achieve high positioning accuracy in the presence of fingerprint ambiguity and device heterogeneity. First, instead of locating a user by matching his RSS measurement with the RSS fingerprint, we match the difference between adjacent RSS measurements with the difference between RSS fingerprints at adjacent locations, which offers great resilience to device heterogeneity. Second, we study continuous indoor tracking and match a sequence of RSS measurements from user to a movement path for much improved positioning accuracy in the presence of RSS fingerprints ambiguity.

## 1.4 Organization

The remainder of this dissertation is structured as follows. Chapter 2 introduces MV-IPS, a novel secure RSS-IPS that can achieve higher positioning accuracy than prior solutions in spite of the presence or absence of RSS attacks. Chapter 3 presents ImPos, a novel image-based IPS that achieves high positioning accuracy by accurately estimating the direction from a user to recognized landmarks through image analysis and fully utilizing all the recognized landmarks. Chapter 4 further introduces CITS, a novel RSS-based continuous indoor tracking system that can achieve high positioning accuracy in the presence of RSS fingerprint ambiguity and device diversity via differential RSS fingerprint matching and path matching. Chapter 5 finally concludes this disseration.

# Chapter 2

# SECURE INDOOR POSITIONING AGAINST SIGNAL STRENGTH ATTACKS VIA OPTIMIZED MULTI-VOTING

## 2.1 Introduction

Indoor Positioning Systems (IPSes) have attracted tremendous interest from the academia and industry. An IPS not only help mobile users obtain real-time locations in large indoor venues where GPS signals are unavailable or unreliable, but also provide relevant location contexts to enable a wide range of exciting applications. Exemplary IPS applications include location-based proximity advertising in shopping malls, patient and visitor guidance in hospitals, personnel and asset tracking in factories, and so on.

WiFi-based RSS-IPSes [23, 2] are among the most promising types of IPSes and expected to foster a market of 2.5 billion US dollars by 2020 [24]. Relying on ubiquitous smartphones and existing indoor WiFi infrastructures, RSS-IPSes explore distinguishable received signal strength (RSS) measurements at different indoor locations as their location fingerprints. RSS-IPSes are very attractive to indoor venue owners because there is no need to perform costly infrastructure updates. A typical RSS-IPS works in two phases. In the offline training phase, the IPS operator collects RSS fingerprints at indoor positions to build a fingerprint database of sufficient spatial granularity. In the online positioning phase, on receiving a location query with an RSS measurement from the user, the IPS searches its fingerprint database for the most matching RSS fingerprint and returns the corresponding reference position to the querier.

RSS-IPS is unfortunately vulnerable to received signal strength (RSS) attacks at the physical layer that cannot be thwarted by conventional cryptographic techniques. Early studies [11] demonstrate that RSS measurements can be easily manipulated by

placing absorbing materials such as book, water, and foil between transmitting and receiving devices. Most recently, Li *et al.* [13] showed that, by impersonating a few WiFi APs with off-the-shelf wireless routers and fine-tuning their transmission power, the attacker can control the RSS values at a target location to either maximize the distance error or mislead the IPS server into returning an arbitrary wrong location. While several defenses against RSS attacks have proposed in the literature [12, 14, 15, 16, 13], they nevertheless all exhibit a tradeoff between attack resilience and positioning accuracy in the absence of RSS attacks. In other words, the high resilience to RSS attacks is usually achieved at the sacrifice of positioning accuracy when there is no RSS attack. This inherent tradeoff makes existing defenses less appealing to IPS operators and thus less likely to be adopted in reality.

**Is it possible to design an RSS-IPS that can achieve high positioning accuracy in spite of the presence or absence of RSS attacks?**

In this thesis, we provide an affirmative answer to the above question by introducing MV-IPS, a novel RSS-IPS with higher positioning accuracy than prior defenses in both cases. We find that the key to achieve high positioning accuracy in both situations is to fully utilize every AP's information while limiting the impact of any individual AP on the final decision. Specifically, making full use of all available APs can improve positioning accuracy in the absence of RSS attacks, whereas limiting each individual AP's role in final decision making can provide resilience against RSS attacks.

Based on this observation, we design MV-IPS based on the Borda count [25], a family of single-winner preferential voting methods widely used in practice. In MV-IPS, the IPS server uses every AP as a voter to give a weighted vote for every reference (or candidate) location in its RSS fingerprint database. A user still submits a location query as usual, which contains the RSS measurement from each available AP. Then the IPS server produces a ranked list of reference locations per AP according to the similarity between the received RSS and the RSS fingerprint for the AP at each reference location. In addition, each reference location is assigned a point value that corresponds to the AP's weighted vote and depends on both its rank and parameters pre-trained

from RSS measurements. Finally, the IPS server outputs the reference location that receives the highest total point values from all APs as the querier's most likely location.

Our contributions can be summarized as follows.

- We identify a key limitation of existing defenses against physical-layer signal strength attacks on RSS-IPSes, which forces IPS operators to choose between attack resilience and positioning accuracy.
- We propose MV-IPS, a novel RSS-IPS that explores Borda count voting mechanisms for indoor localization, in which WiFI APs cast weighted votes to jointly determine user locations.
- We formulate the location-weight assignment as an optimization problem to accommodate different APs' capabilities in differentiating user locations and present the optimal solution based on the projected gradient descent.
- We conduct trace-driven simulation studies based on prototype implementations and real RSS data to confirm the effectiveness of MV-IPS in the presence and absence of RSS attacks. Specifically, our evaluation results show that MV-IPS can achieve an average distance error of 1.68 m in the absence of RSS attacks in contrast to 1.96 m achieved by the state-of-art defense [13] while being highly resilient to RSS attacks.

The rest of the chapter is structured as follows. Section 2.2 briefs the related work. Section 2.3 presents the MV-IPS design. Section 2.4 evaluates the performance of MV-IPS via trace-driven simulations. Section 2.5 concludes this chapter.

## 2.2 Related Work

In this section, we review some most related work.

RSS-IPSes have been studied extensively in the past two decades, and existing solutions differ in how a user's RSS measurement is matched with RSS fingerprints. In the deterministic RSS-IPS, on receiving a location query from the user, the IPS operator evaluates the similarity between the user's RSS measurement and the stored RSS fingerprints using a proper distance metric and returns the reference location

whose RSS fingerprint is most similar to the user's RSS measurement. As a representative deterministic-matching RSS-IPS, Radar[23, 2] uses Euclidean distance as the distance metric. In addition, cosine similarity [26] and Tanimato similarity [27] have been shown to yield satisfactory positioning accuracy. Moreover, Wu *et al.* [28] applied support vector machine, and Nuno *et al.* [29] adopted linear discriminant analysis for RSS fingerprint matching. Probabilistic matching has also been used in RSS-IPS. For example, Horus [3] represents the RSS fingerprint at each reference location as the probability distribution of the RSS value and determines the user's location using maximum-likelihood estimation. Other probabilistic matching algorithms have been proposed, including Bayesian network [30], expectation-maximization [31], and Gaussian process [32]. None of these works are resilient to RSS attacks.

There have been some efforts to design RSS-IPS resilient to RSS attacks. Li *et al.* [12] introduced a median-distance based defense in which the distance between the user's RSS measurement and the RSS fingerprint is calculated with respect to every AP, and the reference location with the smallest median distance is chosen as the user's location. The work [15] explored K-means cluster to distinguish good APs and attacked APs according to their geometric relationship. Kushki *et al.* [14] proposed to select a subset of reliable APs according to their confidence scores based on the covariance matrix. Fang *et al.* [16] introduced an attack-resistant localization scheme based on a probabilistic inclusive disjunction model. Yang *et al.* [33] explored Trained Mean Matching (TMM) to detect the evil twin attack in RSS-IPS. Most recently, Yuan *et al.* [13] introduced a defense against RSS attacks. However, all these solutions would force an IPS operator to choose between attack resiliency and positioning accuracy, which makes them less likely to be adopted in practice.

There are also some works loosely related to our work. For example, Li *et al.* [34] introduced several mechanisms to filter out fake RSS data in crowdsourced IPS systems. As another example, PriWFL [35] protects user's location privacy by encrypting a user's location query using Pallier cryptosystem, which is subsequently improved by Yang *et al.* [36] to further protect the fingerprint database at the IPS operator. There

are also several IPSes that do not solely rely on RSS. For example, BSurroundSend [37] explores ambient information such as sound and light information to enrich the fingerprint and improve positioning accuracy. As another example, PinLoc [38] explores detailed physical layer information such as channel frequency responses to improve the position accuracy of WiFi-based IPS. Wu *et al.* [4] showed that signal fingerprints based on Channel State Information (CSI) can improve the indoor localization performance. Similarly, DeepFi [39] adopts deep learning to perform indoor localization using fine-grained CSI-based fingerprints.

## 2.3 MV-IPS Design

In this section, we first give an overview of MV-IPS and then detail its design.

### 2.3.1 Overview

The design of MV-IPS is inspired by the Borda count, a family of single-winner preferential voting methods widely used in both political and non-political elections. In a typical Borda count voting, every voter ranks candidates in order of preference and assigns a point value, i.e., weight vote, to every candidate based on the candidate's ranking such that higher-ranked candidates receive more point values. When all votes are cast, the candidate who receives the maximum total points is chosen as the winner. Different Borda count methods vary in how point values are assigned in accordance with rankings.

In MV-IPS, we view APs as voters and reference locations as candidates. A user still submits a location query as usual, which contains the RSS from each AP. Each AP is associated with a set of RSS fingerprints and corresponding reference locations in the IPS server's fingerprint database. So the IPS server can easily generate a ranked list of reference locations for each AP according to the difference between the received RSS and corresponding fingerprint in the database: smaller difference leads to higher ranking. The IPS server also assigns a point value to each reference location for each

AP, which corresponds to the AP's weighted vote. Finally, the reference location that receives the maximum total point values is considered the user's most likely location.

A key difference between MV-IPS and the standard Borda count voting lies in how APs assign point values to their rankings. In particular, we observe that different APs could have diverse capabilities in determining user locations. For example, an AP of which the RSS exhibits large variation across different locations can provide more reliable evidence about a user's location than the one with extremely low or very similar RSS values at many reference locations. As a result, unlike traditional Borda count methods in which all voters share the same point assignment rule, the APs follow different point assignment rules under MV-IPS. In MV-IPS, we formulate the point assignment as an optimization problem based on RSS training data and then find an optimal point assignment rule via the projected gradient decent.

### 2.3.2 Detail Design

As many other RSS-IPSes, MV-IPS consists of two phases. In the offline training phase, we collect RSS fingerprints at both reference locations and training locations in the indoor venue of interest and train the system parameters. In the online positioning phase, the IPS server answers location queries from users based on the received RSS measurements and its RSS fingerprint database. In what follows, we first introduce how system parameters are trained based on RSS measurements and then explain how the IPS operator determines a user's location on receiving a location query.

#### 2.3.2.1 Data Collection

We first choose $n$ reference locations $x_1, \cdots, x_n$ and $m$ training locations $y_1, \ldots, y_m$ in the indoor venue. We then pre-compute $d[i, j]$ as the Euclidian distance between reference location $x_i$ and training location $y_j$ for all $1 \leq i \leq n$ and $1 \leq j \leq m$.

We then collect one RSS fingerprint at each of the $n$ reference locations. The RSS measurement collected at reference location $x_i$ is denoted by $\mathsf{rss}_i = (\mathsf{rss}_{i,1}, \ldots, \mathsf{rss}_{i,p})$, where $\mathsf{rss}_{i,z}$ is the $z$th AP's RSS at reference location $x_i$ for all $1 \leq z \leq p$ and $1 \leq i \leq n$,

and $p$ is the number of APs in the indoor venue. These RSS measurements serve as the RSS fingerprint of the reference locations. We also collect the RSS measurements at $m$ training locations. We denote the RSS measurement collected at training location $y_i$ by $\mathsf{rss}'_i = (\mathsf{rss}'_{i,1}, \ldots, \mathsf{rss}'_{i,p})$, where $\mathsf{rss}'_{i,z}$ is the $j$th AP's RSS at training location $y_i$ for all $1 \leq z \leq p$ and $1 \leq i \leq m$.

#### 2.3.2.2 Parameter Training

Let $W = [w]_{p \times n}$ be the weight matrix, where $w[z, i]$ is weight assigned by AP $z$ to the reference location ranked $i$th for all $1 \leq z \leq p$ and $1 \leq i \leq n$. We use the following method to train $W$ using the collected RSS measurements $\{\mathsf{rss}_i | 1 \leq i \leq n\} \bigcup \{\mathsf{rss}'_i | 1 \leq i \leq m\}$.

First, for every training location $y_j, 1 \leq j \leq m$, with RSS measurement $\mathsf{rss}'_j = (\mathsf{rss}'_{j,1}, \cdots, \mathsf{rss}'_{j,p})$, we first find its ranking under each AP. Specifically, each AP $z, 1 \leq z \leq p$, calculates the difference between $\mathsf{rss}'_{j,z}$ and the fingerprint of reference location $x_i$ as

$$\triangle_z(i, j) = |\mathsf{rss}_{i,z} - \mathsf{rss}'_{j,z}|,$$

for all $1 \leq i \leq n$. Each AP $z$, $1 \leq z \leq p$, then ranks the $n$ reference locations based on $\triangle_z(i, j)$. Let $(\phi_{z,j}(1), \phi_{z,j}(2), \ldots, \phi_{z,j}(n))$ be a permutation of $(1, 2, \ldots, n)$, such that $\triangle_z(\phi_{z,j}(1), j) < \triangle_z(\phi_{z,j}(2), j) < \cdots < \triangle_z(\phi_{z,j}(n), j)$. Reference position $x_{\phi_{z,j}(i)}$ is then ranked $i$th by AP $z$ for all $1 \leq z \leq p$ and $1 \leq i \leq n$. We also define $\phi^{-1}_{z,j}(\cdot)$ as the inverse of permutation $\phi_{z,j}(\cdot)$, i.e., reference location $x_i$ is ranked $\phi^{-1}_{z,j}(i)$th by AP $z$.

We repeat the above procedure for all $m$ training locations to obtain $m$ rank matrixes $\Phi^1, \ldots, \Phi^m$, where

$$\Phi^j = \begin{bmatrix} \phi_{1,j}(1) & \phi_{1,j}(2) & \ldots & \phi_{1,j}(n) \\ \phi_{2,j}(1) & \phi_{2,j}(2) & \ldots & \phi_{2,j}(n) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_{p,j}(1) & \phi_{p,j}(2) & \ldots & \phi_{p,j}(n) \end{bmatrix}$$

for all $1 \leq j \leq m$.

Second, for every training location $y_j, 1 \leq j \leq m$, we find the returned reference location according rank matrix $\Phi^j$ under weight matrix $W$. Specifically, under a given weight matrix $W$, each AP $z$ gives reference location $i$ a weighted vote $w[z, \phi_{z,j}^{-1}(i)]$ for all $1 \leq z \leq p$ and $1 \leq i \leq n$. The total weight that reference location $x_i$ receives from all $n$ APs is then given by

$$w[i|j] = \sum_{z=1}^{p} w[z, \phi_{z,j}^{-1}(i)], \tag{2.1}$$

for all $1 \leq i \leq n$. Given $w[1|j], \ldots, w[n|j]$, the reference location $x_{j^*}$ with the highest total weight is estimated as the location for RSS measurement $\mathsf{rss}'_j$, where

$$j^* = \arg\max_{i \in \{1,\ldots,n\}} w[i|j],$$

which results in a distance error $d[j^*, j]$.

We now formulate the training of weight matrix $W$ as an optimization problem where we seek to find a weight matrix that minimizes the average error distance across $m$ training locations. Our cost function is inspired by softargmax function widely used in multiclass classification. Consider training location $y_j$ as an example. There are $n$ possible reference locations $x_1, \ldots, x_n$ that $\mathsf{rss}'_j$ may be estimated into under MV-IPS. Since the distance between $y_j$ and reference location $x_i$ is $d[i, j]$, we define the loss function with respect to training location $y_j$ as

$$\mathcal{L}_j(W) = \frac{\sum_{i=1}^{n} \exp(\gamma \cdot w[i|j]) \cdot d[i, j]}{\sum_{i=1}^{n} \exp(\gamma \cdot w[i|j])} \tag{2.2}$$

where $w[i|j]$ is given in Eq. (2.1) and $\gamma > 0$ is a system parameter. It is easy to see that as $\gamma \to \infty$, the term $\exp(\gamma \cdot w[j^*|j])$ dominates other terms and $\mathcal{L}(j)$ converges to $d[j^*, j]$. We further define the cost function as

$$\mathcal{J}(W) = \frac{1}{m} \sum_{j=1}^{m} \mathcal{L}_j(W). \tag{2.3}$$

To find the optimal weight assignment, we seek to solve the following optimization problem

$$\begin{aligned} &\textbf{Minimize} \quad \mathcal{J}(W) \\ &\textbf{Subject to} \quad \sum_{i=1}^{n} w[z,i] = 1, \quad \forall 1 \leq z \leq p, \end{aligned} \tag{2.4}$$

where the constraint indicates that every AP has a total weight of one.

We use the projected gradient method [40] to find a local minimum for $\mathcal{J}(W)$. Specifically, let us first consider loss function $\mathcal{L}_j(W)$ by rewriting it in terms of $\{w[z,i]|1 \leq z \leq p, 1 \leq i \leq n\}$ as

$$\mathcal{L}_j(W) = \frac{\sum_{i=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z, \phi_{z,j}^{-1}(i)]) \cdot d[i,j]}{\sum_{i=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z, \phi_{z,j}^{-1}(i)]))}. \tag{2.5}$$

Let $i' = \phi_{z,j}^{-1}(i)$. It follows that $i = \phi_{z,j}(i')$. Substituting $\phi_{z,j}^{-1}(i)$ and $i$ by $i'$ and $\phi_{z,j}(i')$, respectively, we can rewrite $\mathcal{L}_j(W)$ as

$$\mathcal{L}_j(W) = \frac{\sum_{i'=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z,i']) \cdot d[\phi_{z,j}(i'), j]}{\sum_{i'=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z,i']))}. \tag{2.6}$$

We observe that every $w[z,i'], 1 \leq z \leq p, 1 \leq i' \leq n$ appears in both the numerator and denominator of $\mathcal{L}_j(W)$.

We now derive partial derivative of $\mathcal{L}_j(W)$ with respect to $\mathcal{L}_j(W)$. Specifically, let us define two additional functions as

$$f = \sum_{i'=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z,i']) \cdot d[\phi_{z,j}(i'), j],$$

and

$$g = \sum_{i'=1}^{n} \exp(\gamma \cdot \sum_{z=1}^{p} w[z,i']).$$

The partial derivatives of $f$ and $g$ with respect to $w[u, v]$ are given by

$$\frac{\partial f}{\partial w[u,v]} = \gamma \cdot \exp(\gamma \cdot \sum_{z=1}^{p} w[z,v]) \cdot d[\phi_{z,j}(v), j] \tag{2.7}$$

and

$$\frac{\partial g}{\partial w[u,v]} = \gamma \cdot \exp(\gamma \cdot \sum_{z=1}^{p} w[z,v]), \tag{2.8}$$

respectively, for all $1 \leq u \leq p$ and $1 \leq v \leq n$. We can then compute the partial derivative $\mathcal{L}_j(W)$ with respect to each $w[u, v]$ as

$$\frac{\partial \mathcal{L}_j}{\partial w[u,v]} = \frac{\frac{\partial f}{\partial w[u,v]} \cdot g - f \cdot \frac{\partial g}{\partial w[u,v]}}{g^2}, \tag{2.9}$$

for all $1 \leq u \leq p, 1 \leq v \leq n$, where $\frac{\partial f}{\partial w_{u,v}}$ and $\frac{\partial g}{\partial w_{u,v}}$ are given in Eqs. (2.7) and (2.8), respectively. Finally, we can derive the partial derivative of cost function $\mathcal{J}(W)$ with respect to $w[u, v]$ as

$$\frac{\partial \mathcal{J}}{\partial w[u,v]} = \frac{1}{m} \sum_{j=1}^{m} \frac{\partial \mathcal{L}_j}{\partial w[u,v]}. \tag{2.10}$$

To apply the projected gradient method [40], we rewrite the constraint in the optimization problem as

$$AW = b,$$

where

$$A = \begin{bmatrix} \mathbf{1} & \mathbf{0} & \dots & \mathbf{0} \\ \mathbf{0} & \mathbf{1} & \dots & \mathbf{0} \\ \vdots & \vdots & \ddots & \vdots \\ \mathbf{0} & \mathbf{0} & \dots & \mathbf{1} \end{bmatrix},$$

$\mathbf{0} = (0, \dots, 0)$, $\mathbf{1} = (1, \dots, 1)$, $W = [W_1, W_2, \dots, W_p]^T$, $W_z = (w[z, 1], \dots, w[z, n])$ for all $1 \leq z \leq p$, and $b_{1\times p} = (1, 1, \dots, 1)^T$. We can then compute the orthogonal projector matrix as

$$P = I_{np} - A^T(AA^T)^{-1}A, \tag{2.11}$$

**Algorithm 1:** Weight Matrix Training

**input :** Initial weight matrix $W^{(0)}$, error distances $\{d[i,j]|1 \le i \le n, 1 \le j \le m\}$, rank matrices $\Phi^1, \ldots, \Phi^m$, learning rate $\eta$, and terminal parameter $\epsilon$

**output:** Weight matrix $W^{(t)}$

```
t ← 1;
while True do
    foreach j ∈ {1, ..., m} do
        foreach i ∈ {1, ..., n} do
            Compute w[i|j] according to Eq. (2.1);
        end
        Compute L_j(W^(t−1)) according to Eq. (2.6);
    end
    J(W^(t−1)) ← (1/m) Σ_{j=1}^{m} L_j(W^(t−1));
    Compute ∇J(W^(t−1)) according to Eq. (2.10);
    W^(t) ← W^(t−1) − ηP∇J(W^(t−1));
    if |J(W^(t−1)) − J(W^(t))| < ε then
        break;
    else
        t ← t + 1;
    end
end
return W^(t);
```

where $I_{np}$ is the $np \times np$ identity matrix and $T$ on superscript denotes matrix transposition.

Let $W^{(0)}$ be the initial weight matrix, where we set $w[z,i] = \frac{2(n-i+1)}{n(n+1)}$ for all $1 \le i \le n$ and $1 \le z \le p$ as in the standard Borda count voting. We repeatedly compute

$$W^{(t)} = W^{(t-1)} - \eta P \nabla \mathcal{J}(W^{(t-1)}) \tag{2.12}$$

for $t = 1, 2, 3, \ldots$, where $\eta$ is the learning rate and $\nabla \mathcal{J}(W^{(t-1)})$ is the gradient of $\mathcal{J}$ with respect to $W^{(t-1)}$ given by Eq. (2.10). The learning rate $\eta$ is usually set dynamically

via backtracking line search [41]. The process terminates if

$$|\mathcal{J}(W^{(t)}) - \mathcal{J}(W^{(t+1)})| < \epsilon,$$

where $\epsilon$ is a small constant.

We summarize the training process in Algorithm 1. Line 1 initializes the iteration index $t = 1$. In Lines 2-17, we iteratively update the weight matrix until the terminal condition is met. Specifically, in Line 4-6, for every training location $y_j$, we compute the total weight each reference location $x_i$ receives according to Eq. (2.1). Next, based on $\{w[i|j]|1 \leq i \leq n\}$, the loss function of each training location $y_j$ is calculated according to Eq. (2.6) in Line 7. We then calculate the cost function as the average loss across all $m$ training locations in Line 9. In Lines 10-11, we compute the gradient of the cost function using Eq. (2.10) and then update the weight matrix updating according to Eq. (2.12). If the difference between the cost function of the new weight matrix and the cost function of the previous weight matrix is less than $\epsilon$, we terminate the process and output the current weight matrix. Otherwise, we repeat the same procedure in the next iteration.

#### 2.3.2.3 Online Positioning

In the online positioning phase, the IPS operator processes location queries from the user. Assume that the user issues a location query with RSS measurement $\mathsf{rss}_u = (\mathsf{rss}_{u,1}, \ldots, \mathsf{rss}_{u,p})$. The IPS operator first computes

$$\triangle_z(i, u) = |\mathsf{rss}_{i,z} - \mathsf{rss}_{u,z}|,$$

for all $1 \leq z \leq p$ and $1 \leq i \leq n$. Each AP $z$ $(1 \leq z \leq p)$ then ranks the $n$ reference locations based on $\triangle_z(i, u)$. The IPS operator then computes

$$w[i|u] = \sum_{z=1}^{p} w[z, \phi_{z,u}^{-1}(i)], \tag{2.13}$$

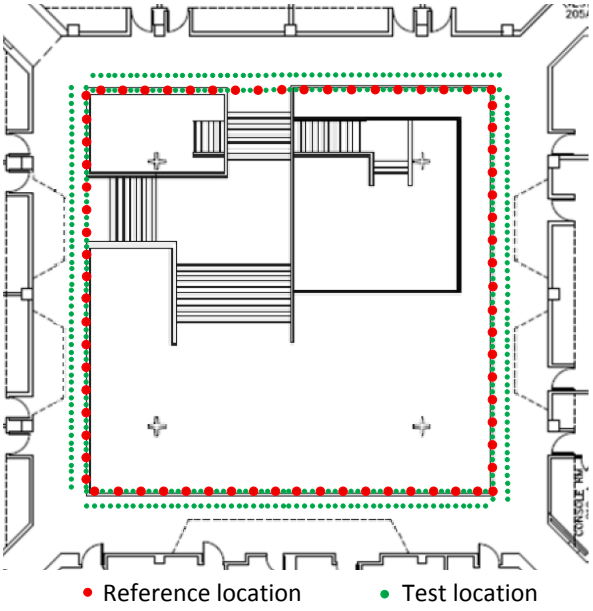



**Figure 2.1:** The floor plan of the indoor venue.

for all $1 \leq i \leq n$, where $\phi_{z,u}^{-1}(i)$ is the rank of reference location $x_i$ under AP $z$ given $\mathsf{rss}_u$. Given $w[1|u], \ldots, w[n|u]$, the reference location $x_{u^*}$ with the highest total weight is estimated as the user's location, where

$$u^* = \arg\max_{i \in \{1,\ldots,n\}} \; w[i|u]$$

## 2.4 Performance Evaluation

In this section, we report the simulation results for MV-IPS.

### 2.4.1 Simulation Settings

We have implemented a prototype of MV-IPS. The prototype system is based on Android studio/Java on a Huawei Honor8 smartphone, which has a 2.3 GHz octa-core CPU and 4 GB RAM. The sampling frequency of the WiFi module is 0.67 Hz. We deploy the prototype on a square zone of 17.8×17.8 $\mathrm{m}^2$ inside an office building with $m = 35$ WiFi APs. Fig. 2.1 shows the floor plan of the indoor venue. We collect RSS

measurements at $n = 72$ reference locations and $m = 360$ training locations as shown in Fig. 2.1, where every reference location is surrounded by five reference locations. We use the RSS measurements collected at the 72 reference locations as the RSS-fingerprint database and the ones collected at the 360 training locations to train the weight matrix and evaluate the performance of the MV-IPS. In particular, we divide the 360 training measurements into five groups of equal size, in which every group contains one training measurement close to each reference location.

We use 5-fold cross-validation to evaluate the performance of MV-IPS. Specifically, we select 4 groups of training measurements to train the weight matrix and use the remaining group as the testing RSS measurements. We repeat this process for five times such that every group is used as the testing set once. The results we report below are the average across the 5 runs.

We mainly use mean distance error (MDE) to evaluate the performance of MV-IPS. For every $\mathsf{rss}'_j$ in a testing group $G$, let $y_j$ be the training location at which it was was taken and $x_{j^*}$ the reference location returned by MV-IPS. We define the MDE as

$$\mathsf{MDE} = \frac{\sum_{\mathsf{rss}'_j \in G} d[j^*, j]}{|G|}.$$

We consider an attacker model similar to the one in [13]. In particular, we assume that the attacker is able to impersonate a subset of $k$ APs of his choice with fake ones under his control. He can also fine-tune the transmission powers of the fake APs to manipulate the RSS values experienced by a target user. We consider the following two attack strategies.

- *Distance error maximization (DEM) attack* [13]: The attacker aims to maximize the distance error experience by a target user. In this attack, the attackers first learns the RSS fingerprints stored at the IPS operator by acting as a normal user repeatedly issuing location queries and then finds the furthest reference location from the target user's location that he can mislead the IPS operator into returning through impersonating and manipulating the transmission powers of the $k$ fake

APs. With respect to MV-IPS, we assume that the attacker knows the weight matrix used by the IPS operator and is able to perfectly control the fake APs' RSS values experienced by the target user.

- *Random RSS attack*: In this attack, the attacker controls the RSSes of $k$ APs randomly chosen from all the APs. The user's RSS measurement under this attack is assumed to be uniform at random in the range from $-30$ dB to $-95$ dB.

We compare the MV-IPS with the following RSS-IPSes.

- Radar [23, 2]: As the most representative RSS-IPS, Radar returns the reference location whose RSS fingerprint is the closest to the user's RSS measurement under the Euclidean distance.
- Median [12]: As a defense against RSS attack, the median-based defense uses the median among the $p$ element-wise distances as the metric to measure the similarity between the user's RSS measurement and the stored RSS fingerprint. The reference location with the smallest median element-wise distance is selected as the user's location.
- TDFM [13]: As the state-of-art defense against RSS attacks, TDFM generalizes Median [12] and Radar. Specifically, the IPS operator calculates $p$ element-wise distances between the user's RSS measurement and each RSS fingerprint with each corresponding to one AP. The similarity between the user's RSS measurement and an RSS fingerprint is measured by the $\lambda$-truncated distance [13], which is the sum of $p-\lambda$ element-wise distances after dropping the $\lambda/2$ largest and $\lambda/2$ smallest element-wise distances. When $\lambda = 0$, TDFM is equivalent to Radar. When $\lambda = (p-1)/2$, TDFM is equivalent to the median-based defense.

Table 2.1 summarizes the default parameters in our simulation unless stated otherwise.

### 2.4.2 Simulation Results

We now report our simulation results.

**Table 2.1:** Default Settings

| Para. | Value | Description |
|---|---|---|
| $n$ | 72 | # of reference locations |
| $m$ | 360 | # of training locations |
| $p$ | 35 | # of APs |
| $\gamma$ | 200 | The exponential parameter in Eq. (2.2) |
| $\eta$ | 0.1 | Learning rate |
| $\epsilon$ | 0.9 | The terminating condition |

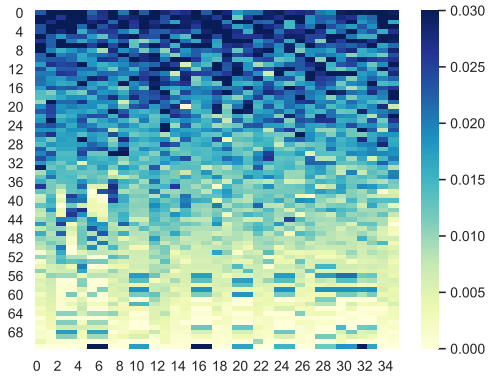



**Figure 2.2:** An example of weighted matrix under MV-IPS.

#### 2.4.2.1 An Example Of Weight Matrix In MV-IPS

Fig. 2.2 shows an example of the weighted matrix trained in MV-IPS, where the $x$-axis represents the IDs of APs and $y$-axis represents the indexes of reference locations. As we can see, different APs have very different weight assignments over their rankings. This clears highlights the key difference between MV-IPS and the standard Boada count voting where all the voters use the same weight assignment.

#### 2.4.2.2 Performance In The Absence Of RSS Attack

Fig. 2.3 compares the CDFs of the error distance under Radar, Median, TDFM, and MV-IPS in the absence of RSS attack. As we can see, MV-IPS not only outperforms

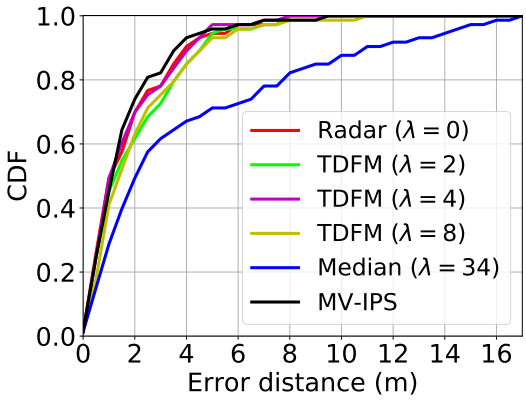



**Figure 2.3:** Comparison of the CDFs of distance error under Radar, TDFM, Median, and MV-IPS.

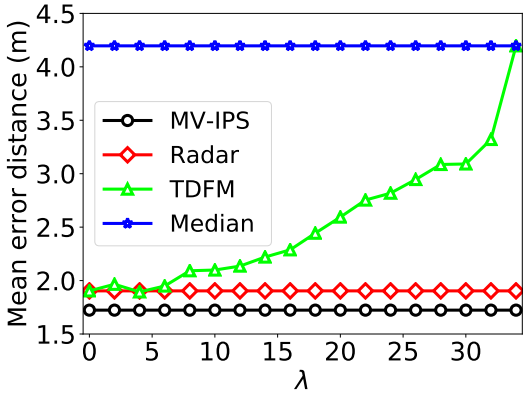



**Figure 2.4:** MDE vs. $\lambda$ in the absence of RSS attack.

Median and TDFM, but also achieves smaller MAE than Radar that is designed for benign environment. For example, 83% of distance errors are smaller than 2.5m under MV-IPS, whereas 78 % and 71 % of distance errors are smaller than 2.5m under TDFM and Median, respectively. These results demonstrate that MV-IPS achieves higher positioning accuracy than prior defenses and Radar in the absence of RSS attack and is thus more appealing to IPS operators in reality.

Fig. 2.4 compares the MDEs under Radar, Median, TDFM, and MV-IPS as

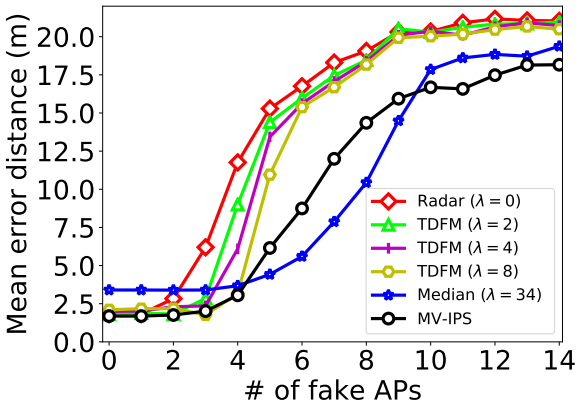



**Figure 2.5:** MDE vs. # of fake APs under DEM attack.

$\lambda$ varying from 0 to 34, where the MDEs under Radar, Median, and MV-IPS are not affected by the change in $\lambda$ and are plotted for reference only. We can see that MV-IPS has the smallest MDE among the four. In addition, the MDE under TDFM increases as $\lambda$ increases. This is expected, because the more element-wise distances dropped, the lower the positioning accuracy for TDFM in the absence of RSS attack, and vice versa.

#### 2.4.2.3 Performance Under DEM Attack

Figure. 2.5 compares the MDE under TDFM and MV-IPS under the DEM attack with the number of fake APs varying from 0 to 14. We can see that the MDE increases as the number of fake APs increases under both TDFM and MV-IPS. In particular, when the number of fake APs is small, e.g., 2, MV-IPS outperforms TDFM with smaller MDE. As the number of fake APs increases to 3, MV-IPS still outperforms other mechanisms except for TDFM with $\lambda = 8$. This is anticipated as when $\lambda$ is set to be slightly larger than twice of the number of fake APs, all the element-wise distances involving fake APs are likely dropped and the remaining good RSS values can ensure sufficiently high positioning accuracy. However, properly setting $\lambda$ would require the IPS operator to know the number of fake APs in advance, which is usually unavailable

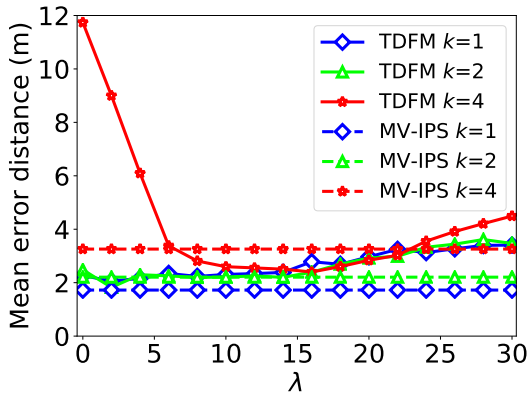
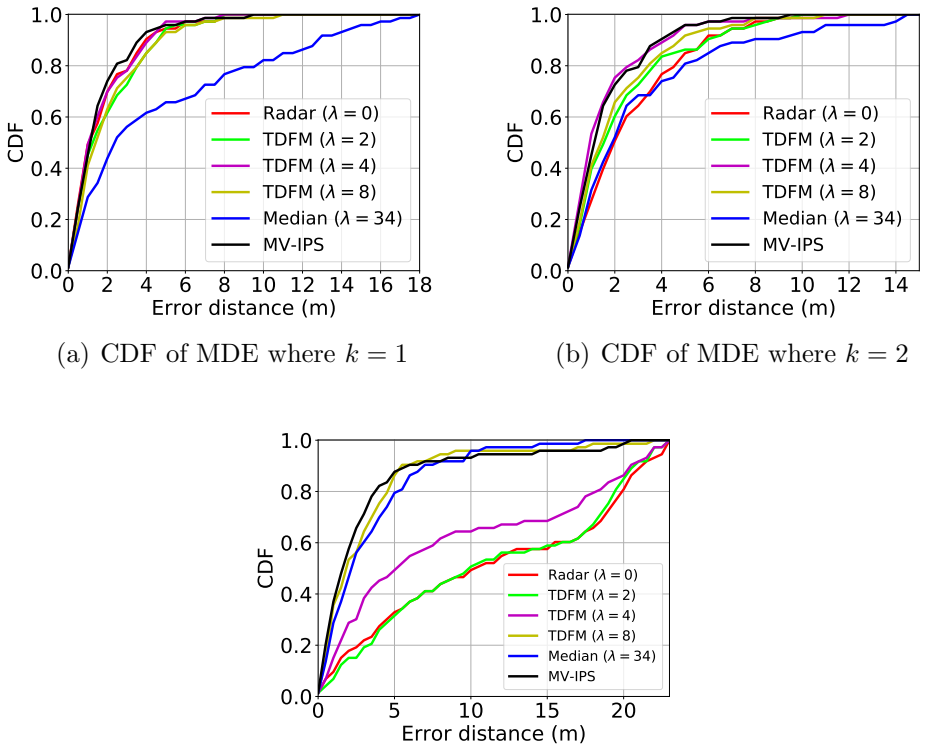



**Figure 2.6:** MDE vs. $\lambda$ under DEM attack.

in practice. On the other hand, when $\lambda$ is set too small or too large, either some fake RSS values will be used for determining user's location or too many good RSS values are dropped, leading to the increase in MDE and thus lower positioning accuracy. In contrast, MV-IPS does not require the IPS operator to tune any parameter and can always maintain high positioning accuracy.

Fig. 2.6 compares the MDEs under TDFM and MV-IPS with $\lambda$ varying from 0 to 30, where the MDE under MV-IPS is not affected by the change in $\lambda$ and plotted for reference only. As we can see that for any given number of fake APs, the MDE under TDFM first decreases and then increases as $\lambda$ increases. The reason is that when $\lambda$ is set too small, some fake RSS values are included for location determination, which results in lower positioning accuracy. When the $\lambda$ is large enough, all fake RSS values are likely dropped, leading to higher positioning accuracy of TDFM. Even in this scenario, we can see that the MDE under MV-IPS is still very close to that under TDFM. Furthermore, as $\lambda$ further increases, MV-IPS outperforms TDFM again. Since $\lambda$ is normally difficult to set without knowing the number of fake APs in advance, these results highlight the significant advantage of MV-IPS over the state-of-art defense TDFM.

Figs. 2.7(a) to 2.7(c) show the CDFs of distance error under Radar, TDFM, Median, and MV-IPS where the number of fake APs is 1, 2, and 4, respectively. Generally




(a) CDF of MDE where $k = 1$

(b) CDF of MDE where $k = 2$

(c) CDF of MDE where $k = 4$

**Figure 2.7:** CDF of MDEs under DEM attack.

speaking, MV-IPS outperforms other three mechanisms in most cases. In particular, as shown in Figs. 2.7(a) to 2.7(c), the MDE under MV-IPS is always smaller than those under Radar, Median, and TDFM except for TDFM with $\lambda = 2$. We can also see from Fig. 2.7(c) that when $k = 4$, the MDE under Median is acceptable and is close to that under MV-IPS, but the MDE under Radar increases drastically. The MDE under TDFM is always between that under Radar and that under Median but still larger than that under MV-IPS. These results show that MV-IPS is highly resilient to DEM attack when the number of fake APs is small.

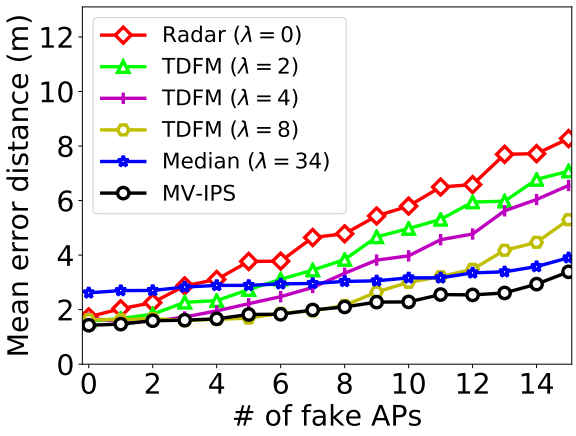



**Figure 2.8:** MDE vs. # of fake APs under random RSS attack.

#### 2.4.2.4 Performance Under Random RSS Attack

Fig. 2.8 shows the MDEs under Radar, TDFM, Median, and MV-IPS under random RSS attack with the number of fake APs varying from 0 to 15. We can see that the MDEs under Radar, TDFM, and MV-IPS all increase as the number of fake APs increases. This anticipated as the more fake APs, the more fake RSS values being used for determining the user's location. We can also see that the MDE under MV-IPS grows much slower than under TDFM and Radar. The MDE under MV-IPS is also lower that under Median when there are more than 15 fake APs.

Fig. 2.9 compares the MDEs under TDFM and MV-IPS under random RSS attack with $\lambda$ varying from 0 to 32, where the MDEs of MV-IPS are not affected by the change in $\lambda$ and are plotted for reference only. We can see that the MDE under TDFM first declines and then increases as $\lambda$ increases, which once gain highlights the importance of properly setting $\lambda$ for TDFM. Furthermore, while the MDE under TDFM is acceptable when $\lambda$ is in the range of $(5, 15)$, the MDE under MV-IPS is either very close to or smaller than that under TDFM.

Figs. 2.10(a) to 2.10(c) show the CDFs of distance error under Radar, TDFM, Median, and MV-IPS under random RSS attack. Once again, we can see that MV-IPS outperforms the other three mechanisms in most cases. While we can see from

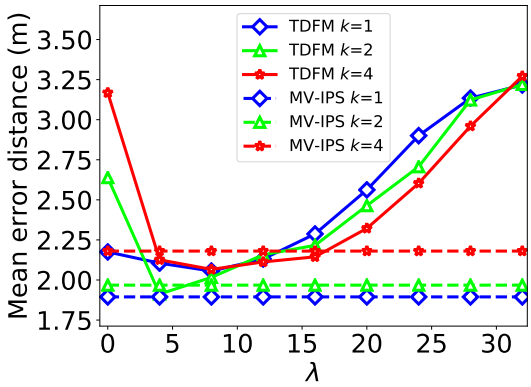
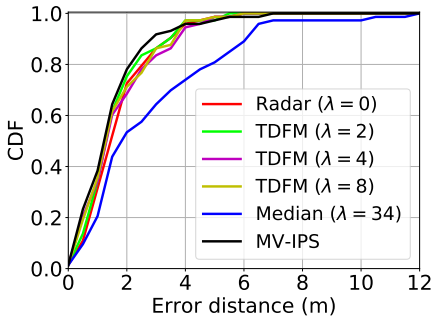
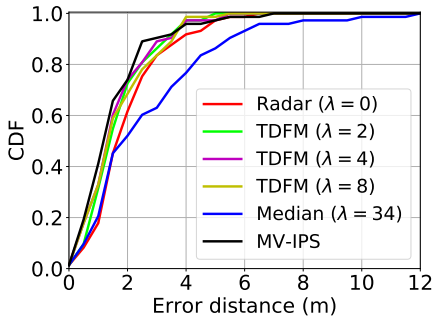


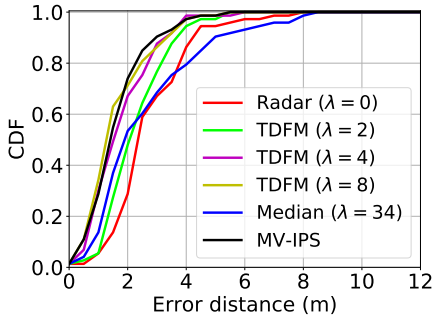

**Figure 2.9:** MDE vs. $\lambda$ under random RSS attack.




(a) CDF of distance error where $k = 1$




(b) CDF of distance error where $k = 2$




(c) CDF of distance error where $k = 4$

**Figure 2.10:** CDFs of distance error under random RSS attack.

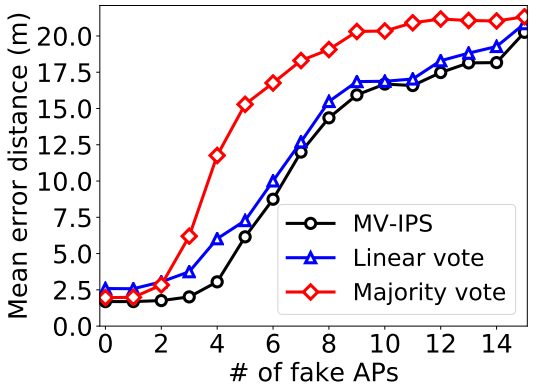



**Figure 2.11:** Impact of weight matrix on MDE where the number of fake APs varying from 0 to 14.

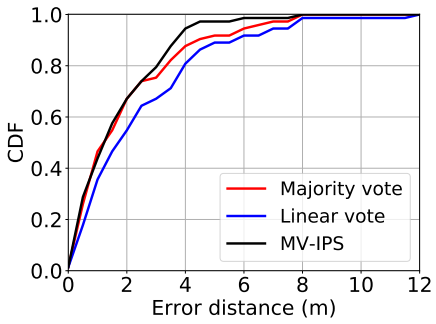
Fig. 2.10(b) that TDFM achieves a MDE comparable to MV-IPS when $\lambda = 2$, properly setting $\lambda$ is difficult without knowing the number of fake APs in advance. Finally, the MDEs under all four mechanisms are smaller than the corresponding cases under the DEM attack, which is anticipated as random RSS attack is less effective than DEM attack. It is thus not surprising that Median always has the highest MDE even when $k = 4$.

#### 2.4.2.5 Impact of Different Weight Matrices

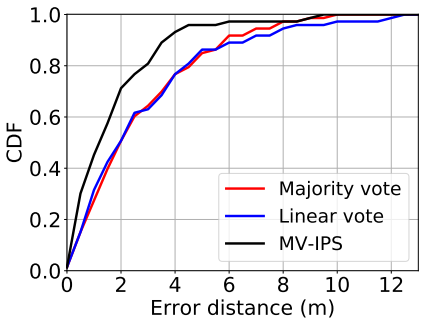
While the weight matrix is trained from RSS data under MV-IPS, we also evaluate the impact of different weight matrices. Specifically, we compare MV-IPS with the mechanisms based on the following two weight matrices.

- *Linear vote*: the weight matrix is the same as the initial weight matrix $W^{(0)}$, where $w[z, i] = \frac{2(n-i+1)}{n(n+1)}$ for all $1 \leq i \leq n$ and $1 \leq z \leq p$.
- *Majority vote*: the weight matrix is defined by $w[z, 1] = 1$ and $[z, j] = 0$ for all $1 \leq z \leq p$ and $2 \leq j \leq n$.

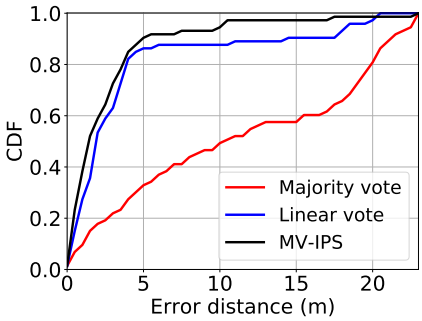
Fig. 2.11 compares the MDEs under MV-IPS, Linear note, and Majority vote. As we can see, as the number of fake APs increases, the MDEs under all three mechanisms increase, which is expected. Moreover, MV-IPS that uses the trained weight




(a) CDF of distance error where $k = 1$




(b) CDF of distance error where $k = 2$




(c) CDF of distance error where $k = 4$

**Figure 2.12:** CDF of distance error under DEM attack under different weight matrices.

matrix always has the lowest MDE among the three, which demonstrates the effectiveness of training in finding a good weight assignment.

Figs. 2.12(a) to 2.12(c) show the CDFs of MDEs under MV-IPS, Linear vote, and Majority vote where the number of fake APs is 1,2, and 4, respectively. Not surprisingly, MV-IPS outperforms both Linear Vote and Majority Vote by large margins, especially for the case where $k = 4$. These results confirm the advantages of MV-IPS based on the weight matrix trained by RSS data.

#### 2.4.2.6 Impact of Different Cost Functions

We also evaluate the impact of different cost functions on the positioning accuracy of MV-IPS. In particular, we evaluate MV-IPS under the following three cost

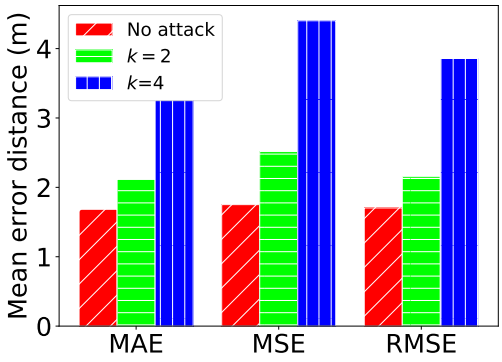



**Figure 2.13:** Comparison of different cost functions.

functions.

- Mean Absolute Error (MAE): the cost function used by MV-IPS and given in Eq. (2.3).
- Mean Square Error (MSE): the cost function is given by

$$\mathcal{J}(W) = \frac{1}{m}\sum_{j=1}^{m}(\mathcal{L}_j(W))^2.$$

- Root Mean Square Error (RMSE): the cost function is given by

$$\mathcal{J}(W) = \sqrt{\frac{1}{m}\sum_{j=1}^{m}(\mathcal{L}_j(W))^2}.$$

Fig. 2.13 compares the distance errors under three cost functions where the number of fake APs is 0, 2, and 4. We can see that the MDEs under MAE and RMSE cost functions are approximately 3.2m when the number of fake APs is 4, whereas that that under MSE is above 4.2m. In addition, the MDE under MSE is always the lowest among the three cost functions. These results indicate that the cost function chosen by MV-IPS outperforms the other two options and leads to high positioning accuracy.

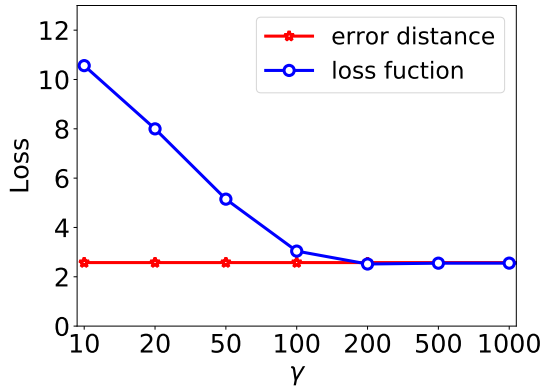



**Figure 2.14:** Impact of $\gamma$ on loss function $\mathcal{L}_j(W)$.

#### 2.4.2.7 Impact of Parameter $\gamma$

The loss function given in Eq. (2.2) involves the parameter $\gamma$. Intuitively, as $\gamma$ approaches $\infty$, loss function $\mathcal{L}_j(W)$ approaches to $d[j^*, j]$. We also evaluate the impact of $\gamma$. Fig. 2.14 plots the values of $\mathcal{L}_j(W)$ and $d[j^*, j]$ as $\gamma$ increases from 10 to 1000. We can see that as $\gamma$ increases, the difference between $\mathcal{L}_j(W)$ and $d[j^*, j]$ decreases. When $\gamma$ exceeds 200, the difference between $\mathcal{L}_j(W)$ and $d[j^*, j]$ becomes negligible. By choosing $\mathcal{L}_j(W)$ as the loss function, we are able to derive the close form of the gradient of the cost function $\mathcal{J}(w)$.

### 2.4.3 Summary

We summarize the simulation result as follows.

- MV-IPS achieves higher positioning accuracy than Radar, Median, and TDFM in the absence of RSS attacks.
- In the presence of RSS attacks, MV-IPS achieves higher positioning accuracy than Radar and Median. It also either outperforms TDFM or achieves a positioning accuracy closer to TDFM when the parameter $\lambda$ is set to be approximately twice of the number of fake APs.

- Unlike TDFM whose performance is highly dependent on properly setting of parameter $\lambda$ that requires the knowledge of the number of APs, MV-IPS is oblivious to the number of fake APs and can always achieve satisfactory positioning accuracy no matter whether RSS attacks are present.
- MV-IPS relies on a weight matrix properly trained from the RSS data and significantly outperforms other weight matrices used in the standard Borda count and majority vote.

## 2.5 Summary Of MV-IPS

In this thesis, we have introduced the design and evaluation of MV-IPS, a novel RSS-IPS based on weighted multi-voting. Inspired by the Borda count voting, MV-IPS treats every AP as a voter to cast a weighted vote for every reference location, and the reference location that receives the highest accumulative vote is considered as the user's location. Unlike existing RSS-IPSes that suffer from the inherent tradeoff between indoor positioning accuracy and attack resilience, MV-IPS can achieve high indoor positioning accuracy no matter whether RSS attacks are present. Trace-driven simulation studies based on real RSS measurements have confirmed the significant advantages of MV-IPS over prior RSS-IPSes.

# Chapter 3

# IMPOS: AN IMAGE-BASED INDOOR POSITIONING SYSTEM

## 3.1 Introduction

Indoor positioning systems (IPSes) are indispensable for users to navigate in large and unfamiliar indoor venues such as airports, hospital complexes, and shopping malls. The lack of reliable GPS signals in indoor venues makes accurate indoor positioning a non-trivial task and has prompt the design and development of various types of IPSes based on different techniques over the past two decades. Existing IPSes include WiFi fingerprint-based IPS [23, 3], WiFi Channel State Information (CSI)-based IPS [4], Bluetooth-based IPS [5], image-based IPS [18, 42], Radio Frequency Identification (RFID)-based IPS [43], sound-based IPS [9, 44], visible light-based IPS [45], and so on. The global IPS market is expected to reach 256.59 billions in 2028 [1].

As a promising type of IPS, image-based IPSes are particularly suitable for venues without WiFi infrastructure. In a typical image-based IPS, the IPS server identifies a number of landmarks in the indoor venue. A user only needs to take a few pictures of his surrounding using his smartphone and submits the images to the IPS server. The landmarks in the query images are recognized and matched with the ones stored at the server. The IPS server can then estimate the user's position based on the locations of the recognized landmarks via 3D model reconstruction [46, 47, 48, 49], triangulation [18, 42], or trilateration [50].

We observe that existing image-based IPSes suffer from two main limitations. First, prior solutions based on 3D model reconstruction [46, 47, 48, 49] incur expensive setup costs and long processing delays for users' location queries. Second, existing solutions based on triangulation [18, 42] and trilateration [50] suffer from low positioning

accuracy. For example, the state-of-art image-based IPS Sextant [18] reports a median positioning error of 3.6 m largely due to inaccurate angle measurements smartphone compass readings. Third, since triangulation and trilateration only require three recognized landmarks but query images may capture more than three landmarks, different combinations of three landmarks may result in different positioning accuracy. How to choose the best three landmarks remains unclear despite some heuristic guidelines. These situations call for efficient image-based IPSes with high positioning accuracy.

To tackle this challenge, we introduce the design and evaluation of ImPos, a novel image-based IPS based on triangulation with much improved positioning accuracy. The key idea behind the ImPos is a novel method for accurate angle estimation that jointly considers smartphone compass readings and the positions of the landmarks in the query image. Moreover, instead of trying to select the best three landmarks, we formulate the position estimation as an optimization problem to fully utilize all the recognized landmarks to improve positioning accuarcy. By doing so, ImPos achieves a much higher positioning accuracy than the state-of-art image-based IPS. Our contributions in this chapter can be summarized as follows.

- We introduce ImPos, a novel image-based IPS with much improved positioning accuracy.
- We propose a novel method for angle estimation that jointly considers smartphone compass readings and the positions of the landmarks in the query image.
- Experiment studies based on a prototype confirm the advantages of ImPos over prior image-based IPSes. For example, our experiment results show that ImPos achieves a median distance error at 1.09 m in contrast to the 3.23 m reported in [18].

The rest of this chapter is structured as follows. Section 3.2 discusses the related work. Section 3.3 presents the design of ImPos. Section 3.4 reports our experiment results. This chapter is finally concluded in Section 3.5.

## 3.2 Related Work

In this section, we review some works that are most germane to our work.

Image-based IPSes have attracted significant attentions in the past two decades because they do not require any costly infrastructure upgrade but explore the ubiquitous presence of smartphones in people's daily life. Existing image-based IPSes can be broadly divided into two categories. The first category utilizes 3D modeling and locates a user by constructing the 3D model of the landmark and projecting the model into query images through machine learning techniques. Arnold *et al.* [46] proposed a compressed 3D representation for landmarks then project the 3D model to the query image to locate the user. Kawaji *et al.* [51] used omnidirectional panoramic images and KNN methods for comparison to recognize the landmarks. Ke *et al.* [52] achieved high positioning accuracy by improving the accuracy of image matching through an advanced RANdom SAmple Consensus (RANSAC) algorithm. In [47], Tommaso *et al.* trained a regression forest model for 2D image to 3D model correspondences whereby to predict the camera pose. InLoc [48] explores dense matching rather than local features to calculate the similarities between the query image and landmark images and then infer the user's location according to landmark orientation depth on images. Lu *et al.* [49] coped WiFi with the 3D model of the landmark to recognize the landmark quickly. However, these solutions suffer from costly 3D model construction and large processing latency for location queries because the server must search the matching feature points among million of feature descriptors on the landmark model to construct the 3D model of the landmark then project the 3D model to the query image.

The second category estimates a user's location from recognized landmarks using geometric approaches. MoVips [17] calculates the ratio of the distance between the same pair of feature points shown on the query and landmark image and infers the user's distance to the landmark based on the focal length. Hamed *et al.* [53] used a weighted KNN with Epipolar Geometry to infer the user's location from recognized the landmarks. Sextant [18] obtains the angles between each capturing gesture from the smartphone interior sensors then calculates the user's location by triangulation.

Guan *et al.* [54] considered both user's height and sight region to infer the user's relative location to the landmark. Yuanqing *et al.* [55] obtained the user's trace from the pedestrian dead reckoning(PDR) system and then paired the trace with the stored guider's trace HAIL[56] infers a possible region with a predefined compass error and then searches the user's accurate position in the region with the landmarks cope with compass and gyroscope readings. Jiang *et al.* [57] uses images to identifies users' rough position and facing direction, then locate the user with the help of dead reckoning systems. However, these solutions suffer low positioning accuracy due to inaccurate angle or distance estimation.

Besides image-based IPSes, there are also several other types of IPSes based on different technologies. For example, WiFi received signal strength (RSS)-based IPSes [23, 3] explore the distinguishable WiFi received signal strengths to serve as the location fingerprints to estimate a user's location. FILA [4] utilizes WiFi CSI signals as location fingerprints. In [58], Bose *et al.* used the path loss model to calculate the distance between the user and WiFi access point then uses trilateration to locate the user. As another example, Fischer *et al.* [5] utilize the angle of arrival (AOA) and time of arrival (TOA) of Bluetooth signal to infer the distance between the Bluetooth access point and a user whereby to locate the user through trilateration. Similar to RADAR, Wang *et al.* used Bluetooth RSS signals as fingerprints in [6]. There are also IPSes that explore acoustic signals [9, 44], visible light [45], or RFID signals [43] to locate a user. However, these solutions typically require extra infrastructures which may not be readily available in many indoor venues.

## 3.3 Design of ImPos

In this section, we first give an overview of ImPos and then detail its design.

### 3.3.1 Overview

ImPos works in two phases, the offline phase and online localization phase. In the offline phase, the server constructs a landmark database by taking images of

all landmarks and recording their physical coordinates on the floor. In the online localization phase, the server answers location queries from users based on the received query images and the constructed landmark database. Specifically, different from prior image-based IPSes [18, 56], which estimate angles only considering camera headings and identify user's location using angle-based localization techniques, ImPos jointly considers the camera headings and the positions of the landmarks in the query images to estimate angles with high accuracy. Moreover, we formulate localization as an optimization problem instead of directly applying the naive triangulation techniques.

In what follows, we first introduce how to construct the landmark database offline. Next, we introduce how the server determines a user's location which mainly includes an algorithm for angle estimation, and an localization mechanism by solving the formulated optimization problem.

### 3.3.2 Offline Landmark Database Construction

We first select $N$ landmarks $\mathcal{L} = \{L_1, L_2, \cdots, L_N\}$ in the indoor venue. Next, we take one image for each landmark $L_i \in \mathcal{L}$ and record its physical coordinate, $(x_i, y_i) \in \mathcal{D}$, where $\mathcal{D}$ is the domain of the physical coordinate. For each landmark image, we extract a feature vector, $F_i = \{\langle f_{i,1}, (p_{i,1}, q_{i,1})\rangle, \cdots, \langle f_{i,m_i}, (p_{i,m_i}, q_{i,m_i})\rangle\}$ using the classical Speeded Up Robust Features (SURF) algorithm [59], where $f_{i,j}, (1 \leq j \leq m_i)$, is a feature point, $(p_{i,j}, q_{i,j})$ is the pixel position of the feature point, and $m_i$ is the number of feature points in $F_i$. The landmark database can be represented by $\{\langle (x_i, y_i), F_i\rangle | 1 \leq i \leq N\}$.

### 3.3.3 Online Positioning

In the online positioning phase, on receiving a location query from a user, the server estimates the user's location via image processing and triangulation. Assume that a user issues a location query $Q = \{(I_1, c_1), \cdots, (I_n, c_n)\}$, where each $I_i$, $1 \leq i \leq n$, is a query image taken by the users, and $c_i$ is the corresponding compass reading

recorded when taking $I_i$. We assume that every query image $I_i$ includes at least one landmark.

Upon receiving the location query $Q$, the server processes the query in three steps. First, the server identifies the landmarks captured by each query image and obtains their physical coordinates according to the landmark database. Next, the server estimates the *intersecting angle* between the line from the user to a landmark and the line from the user to another landmark. Finally, the server uses the recognized landmark coordinate and the estimated intersection angles to estimate the user's location. In what follows, we detail these steps.

#### 3.3.3.1 Landmark Recognition

Given a query $Q = \{(I_1, c_1), \cdots, (I_n, c_n)\}$, the server first extracts the feature vector from each image. For every image $I_i, 1 \leq j \leq n$, the server converts it into $I_i = \{\langle f_{i,1}, (p_{i,1}, q_{i,1})\rangle, \cdots, \langle f_{i,k_i}, (p_{i,k_i}, q_{i,k_i})\rangle\}$, where $f_{i,j}$ is a feature point and $(p_{i,j}, q_{i,j})$ is the pixel position of the feature point, and $k_i$ is the number of feature points in image $I_i$ for all $1 \leq j \leq k_i$.

For each query image $I_i$, the server identifies the landmarks it captures as follows. First, we compare the feature vector of the query image $I_i$ with the feature vector of every landmark $L_j \in \mathcal{L}$. Specifically, for each landmark $L_j$ with feature vector $F_j$, we calculate its matching degree with the query image $I_i$ as follows. Assume that $F_j$ has $m_j$ feature points, we then have $k_i m_j$ possible feature point pairs. We adopt the standard feature matching techniques [59] to determine whether every pair of feature points are a match. The total number of the matched pairs $n_{i,j}$ is deemed as the matching degree with respect to query image $I_i$ and landmark $L_j$. If the matching degree is larger than a threshold $\theta$, we consider that the landmark $L_j$ matches the query image $I_i$, and the query image $I_i$ captures landmark $L_j$ with high probability.

Second, we refine the matched landmarks by filtering out possible duplicate matches. Since some landmarks may share similar shapes and features, it is possible that the same set of features in the query image are matched to multiple landmarks. Let

$\mathcal{L}_i \subseteq \mathcal{L}$ be the set of matched landmarks in image $I_i$ identified the first step. For each landmark $l_j \in \mathcal{L}_i$ with a set of matched feature points $\{\langle f_{j,1}, (p_{j,1}, q_{j,1})\rangle, \cdots, \langle f_{j,n_{i,j}}, (p_{j,n_{i,j}}, q_{j,n_{i,j}})\}$, we compute the pixel position of the landmark $l_j$ in image $I_i$ as

$$\begin{cases} p_{i,j} &= \frac{1}{n_{i,j}} \sum_{x=1}^{n_{i,j}} p_{j,x} \\ q_{i,j} &= \frac{1}{n_{i,j}} \sum_{x=1}^{n_{i,j}} q_{j,x} \ . \end{cases} \tag{3.1}$$

For every pair of landmarks $L_j, L_k \in \mathcal{L}_i$, we compute their pixel distance as

$$d(j,k) = \sqrt{(p_{i,j} - p_{i,k})^2 + (q_{i,j} - q_{i,k})^2} \ .$$

If $d(j,k)$ is below a predetermined threshold, we consider $L_j$ and $L_k$ are matched to the same set of feature points and remove the one with lower matching degree from $\mathcal{L}_i$.

#### 3.3.3.2 Intersecting Angle Estimation

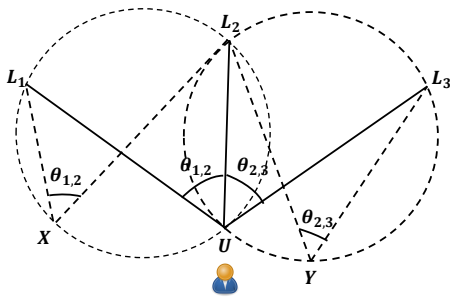



**Figure 3.1:** Illustration of localization via triangulation.

In this subsection, we first introduce the classical triangulation technique and then present a novel method for estimating the intersecting angles.

Triangulation is a classical method to estimate a user's position from the positions of three landmarks and two intersecting angles. Consider Fig. 3.1 as an example, in which three landmarks $L_1, L_2$ and $L_3$ are placed in a clockwise order. Suppose that

we want to estimate a user's position $U$. Assume that we know the two intersecting angles $\angle L_1UL_2 = \theta_{1,2}$ and $\angle L_2UL_3 = \theta_{2,3}$. We can find a point $X$ such that $\angle L_1XL_2 = \theta_{1,2}$ and draw a circumscribed circle that passes through $L_1, L_2$ and $X$. It is easy to see that any point on the arc $\widehat{L_1XL_2}$ has a circle angle of $\theta_{1,2}$. Similarly, we can find another point $Y$ such that $\angle L_2YL_3 = \theta_{2,3}$ and draw a circumscribed circle that passes through $L_2, L_3$, and $Y$ so that any point on the arc $\widehat{L_2YL_3}$ has a circle angle of $\theta_{1,2}$. Therefore, the user's position $U$ must be the intersection point of the two circles.

Accurate measurement of the intersecting angles from users to the recognized landmarks is key for accurate location estimation via triangulation. Prior works [18, 56] all directly use the compass readings recorded by the smartphone when taking the picture as the direction from the user to the landmarks. We find that such angle measurements are inaccurate because the camera may not be perfectly facing the landmark when picture is taken. The difference between the true direction towards the landmark and the reported direction from the compass reading is commonly referred to as *angle drift*. Large angle drifts would introduce large errors to the estimated intersecting angles and lead to large positioning error. In what follows, we introduce an effective method to estimate such angle drifts.

Consider Fig. 3.2 as an example in which two reference lines pass $X$ and $Y$, respectively. Suppose that the user is at position $U$ when taking a picture of landmark $L$ and that the camera's position is at point $F$. The true direction from the user to the landmark can be represented by $\angle LUY$. Now suppose that the compass reading is $c$ which is the angle $\angle HFX = \angle HUY$. The angle drift is thus $\angle HUL = \alpha$, which is difficult to directly estimate.

Our key idea is that the angle drift $\alpha$ can be approximated by angle $\angle HFL = \gamma$. The reason is that the length of the line segment $FU$ is approximately the user's arm length, which is about tens of centimeter and much smaller than the distance between the user and the landmark $LU$, which is usually several meters or more in practice. Therefore, the angle $\angle FLU$ is very small in practice. Since $\gamma = \alpha + \beta$, we have $\gamma \approx \alpha$.

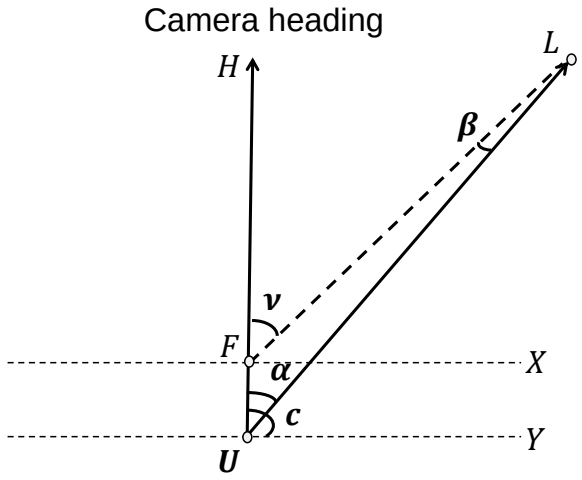



**Figure 3.2:** Illustration of angle drift.

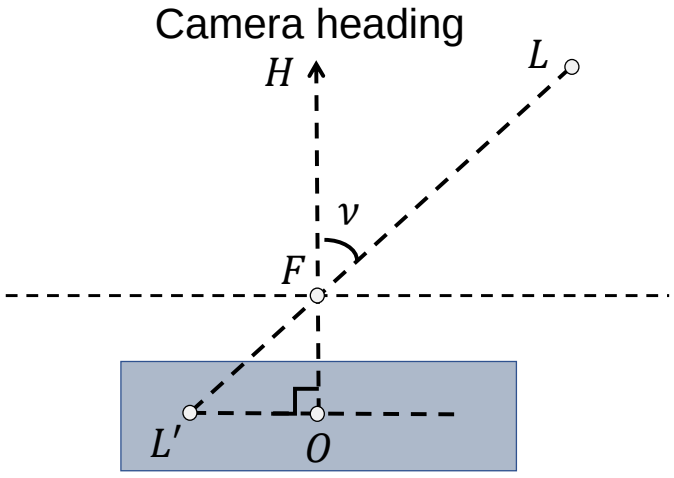



**Figure 3.3:** Angle drift estimation

We now discuss how to estimate angle $\gamma$ from the landmark $L$'s pixel position in the image. Consider Fig. 3.3 as an example, in which $F$ is the camera's focal point, $O$ is the the optical center of the camera, and $L'$ is the position of the landmark $L$ in the image. We can see that $\angle L'FO = \gamma$ and that triangle $\triangle L'OF$ is a right triangle.

It follows that

$$\gamma = \arctan \frac{\overline{OL'}}{\overline{FO}} \ , \tag{3.2}$$

where $\overline{FO}$ is the focal length of the camera and $\overline{OL'}$ can be computed from the pixel size and the pixel distance between $L'$ and $O$.

For every image $I_i$ and every landmark $L_j \in \mathcal{L}_i$, we apply the above method to find angle drift $\gamma_{i,j}$. Denote by $\theta_{i,j}$ the direction from the user to landmark $L_j$ according to image $I_i$ for all $1 \leq i \leq n$ and $j \in \mathcal{L}_i$. We compute each $\theta_{i,j}$ as

$$\theta_{i,j} = \begin{cases} c_i - \gamma_{i,j} & \text{if landmark } L_j \text{ is on the left,} \\ c_i + \gamma_{i,j} & \text{if landmark } L_j \text{ is on the right,} \end{cases} \tag{3.3}$$

where $c_i$ is the compass reading recorded when taking image $I_i$.

Since the same landmark may be captured in multiple images, we further estimate the direction to each recognized landmarks by taking the average of multiple estimations. Let $\mathcal{L}_q = \bigcup_{i=1}^{n} \mathcal{L}_i$ be the set of all the recognized landmarks. For each landmark $L_j \in \mathcal{L}_q$, we define $\mathcal{I}_j = \{i | L_j \in \mathcal{L}_i, 1 \leq i \leq n\}$ as the set of images that contain $L_j$ and denote by $\theta_j$ as the direction from user towards $L_j$. We compute each $\theta_j$ as

$$\theta_j = \frac{\sum_{j \in \mathcal{I}_j} \theta_{i,j}}{|\mathcal{I}_j|} \tag{3.4}$$

#### 3.3.3.3 User Positioning

Given a set of recognized landmarks $\mathcal{L}_q$, where each landmark $L_j \in \mathcal{L}_q$ is at position $(x_j, y_j)$ and estimation $\theta_j$, we formulate the location estimation as an optimization problem. Specifically, for every pair of landmarks $L_j, L_k \in \mathcal{L}_q$, we define their estimated intersecting angle as

$$\phi_{j,k} = |\theta_j - \theta_k| \ , \tag{3.5}$$

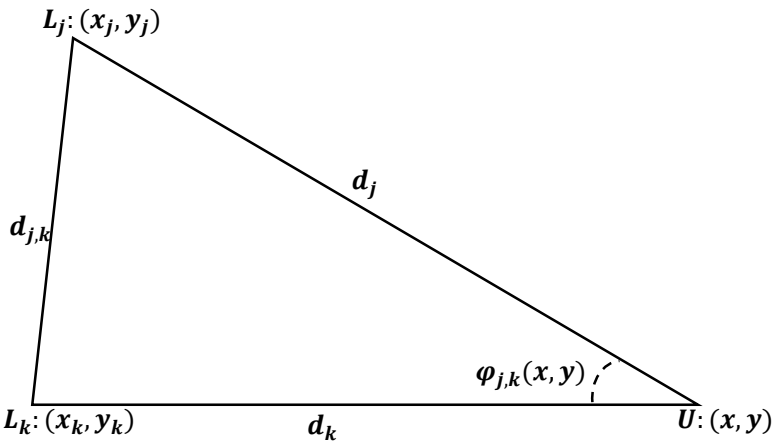



**Figure 3.4:** An example of the triangle formed by the user's location and two landmarks

where $\theta_j$ and $\theta_k$ are the estimated directions from the user to $L_j$ and $L_k$, respectively. Assume the user's location is at $(x, y)$. We can draw a triangle formed by the user's location and two landmarks. Consider Fig. 3.4 as an example. The intersecting angle can be computed as

$$\phi_{j,k}(x, y) = \arccos \frac{d_j^2 + d_k^2 - d_{j,k}^2}{2 d_j d_k} \ , \tag{3.6}$$

where

$$\begin{aligned} d_j &= \sqrt{(x_j - x)^2 + (y_j - y)^2}, \\ d_k &= \sqrt{(x_k - x)^2 + (y_k - y)^2}, \\ d_{j,k} &= \sqrt{(x_j - x_k)^2 + (y_j - y_k)^2} \ . \end{aligned}$$

Ideally, the user's true location $(x, y)$ should minimize the difference between the measured intersecting angle $\phi_{j,k}$ and the computed intersecting angle $\phi_{j,k}(x, y)$.

We now introduce two optimization problem formulation for estimating the user's location based on the above observation. Our first optimization problem formulation seeks to minimize the total square difference between the measured and computed

intersecting angles for all pairs of recognized landmarks, which is given by

$$\begin{aligned} \text{Minimize} \quad & \sum_{j \in \mathcal{L}_q} \sum_{k \in \mathcal{L}_q} (\phi_{j,k}(x,y) - \phi_{j,k})^2 \ , \\ \text{Subject to} \quad & (x,y) \in \mathcal{D} \ , \end{aligned} \tag{3.7}$$

where $\mathcal{D}$ denotes the set of possible locations in the indoor venue after proper discretization.

While the first formulation is intuitive, it suffers from one limitation. In particular, we notice that the change in the user's location has different impact on the computed intersecting angles of different pairs of recognized landmarks. Specifically, if the user is far away from both landmarks, then the change in the user's location has a relatively small impact on the computed intersecting angle, so minimizing the square difference between the measured and computed intersecting angles may require a large change in the user's location and thus should be given a smaller weight. On the other hand, if the user is very close to one of the two landmarks, then a small change in user's location would have a very large impact on the computed intersecting angle, which should be given a higher weight. While we don't know the user's location in advance, we find that the smaller the measured intersecting angle and the larger the distance between the two landmarks, the more likely that the user is far away from both landmarks, and vice versa. Based on this observation, we define a weight for every pair of landmarks $L_j$ and $L_k$ as

$$w_{j,k} = \frac{\phi_{j,k}}{\sqrt{(x_j - x_k)^2 + (y_j - y_k)^2}} \tag{3.8}$$

and define the cost function as

$$f(x,y) = \sum_{j \in \mathcal{L}_q} \sum_{k \in \mathcal{L}_q} w_{j,k} (\phi_{j,k}(x,y) - \phi_{j,k})^2 \ . \tag{3.9}$$

Our second optimization problem formulation seeks to minimize the total weighted

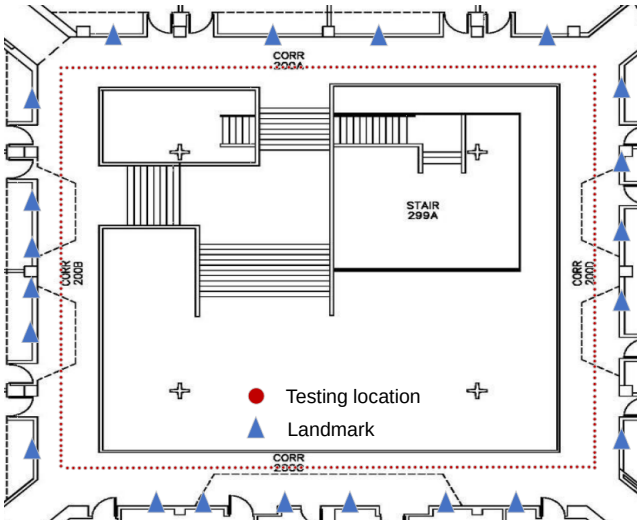



**Figure 3.5:** The floor plan of an office building

square difference between the measured and computed intersecting angles for all pairs of recognized landmarks, which is given by

$$\begin{aligned} \text{Minimize} \quad & f(x, y) \\ \text{Subject to} \quad & (x, y) \in \mathcal{D} \, . \end{aligned} \tag{3.10}$$

Both optimization problems can be solve via exhaustive search.

## 3.4 Performance Evaluation

In this section, we evaluate the performance of ImPos through experimental studies.

### 3.4.1 Data Collection and Experiment Settings

We implement a prototype of the ImPos in Android Studio/Java and test it on a Huawei P30 smartphone, which is equipped with a 2.8 mm physical focal length camera and SONY IMX600 camera sensor. The resolution of image is $2736 \times 3648$, and the pixel size for SONY IMX600 $1\mu m$.

We deploy the prototype system in an office building of 400 $m^2$, and Fig. 3.5 shows the floor plan. As we can see, we take $N = 21$ posters on the wall as landmarks. For every landmark, we record its coordinate, take a picture of it, and extract the SURF features from the image. We then choose 216 testing locations and take five images and record the corresponding compass readings at each testing location.

We compare the performance of ImPos with the Sextant system [18], which is the state of art image-based IPS that uses triangulation. In Sextant, the IPS server recognizes the landmarks from images and obtains the intersecting angles directly from the compass readings. If more than three landmarks are recognized, Sextant first estimates a rough location of the user and then chooses the three landmarks that are closest to the rough location to estimate the user's location via triangulation.

We use the following two performance metrics for our evaluation.

- **Angle error**: For every image $I_i$ taken at testing location $l = (x, y)$ and every recognized landmark $L_j$ at location $(x_j, y_j)$, the true direction from the testing location $l$ to landmark $L_j$ is given by

$$\bar{\theta}_{i,j} = \arctan \frac{y_j - y}{x_j - x} .$$

  The corresponding angle error is defined as

$$\Delta\theta_{i,j} = |\theta_{i,j} - \bar{\theta}_{i,j}| ,$$

  where $\theta_{i,j}$ is given by Eq. (3.3).

- **Error distance**: For each testing location $l = (x, y)$ and corresponding estimated location $\hat{l} = (\hat{x}, \hat{y})$, the error distance is defined as

$$\Delta d = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2} .$$

### 3.4.2 Experiment Results

#### 3.4.2.1 Angle Estimation Accuracy

Fig. 3.6 shows the CDFs of angle error under the Sextant and the ImPos. We can see that the ImPos achieves a median angle error of 3.5 degrees, which is much smaller

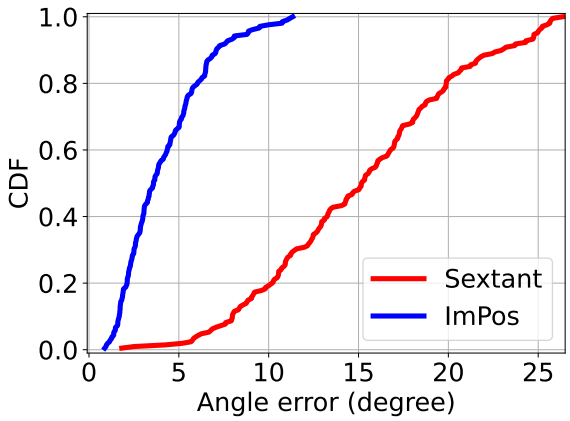



**Figure 3.6:** Comparison of Sextant and ImPos in terms of angle error

than the median angle error of 15.1 degrees achieved by the Sextant. In addition, the angle error is below 6.2 degrees for 80% of the cases under the ImPos which is significantly smaller than 19.8 degree under Sextant. These results clearly show that the ImPos can measure the angle much more accurately than the Sextant that directly uses compass readings as the directions to landmarks. We can also see that the maximum angle error under ImPos is 11.91 degree. This is because there is always some difference between the true angle drift $\alpha$ and the approximated angle $\gamma$ in Fig. 3.2. In addition, the position of a landmark shown on an image is not always the center of all the matched features. We can also see that the maximum angle error under the Sextant is 25.9 degree. This is also reasonable, because the angle of view of HUAWEI P30 is about 60 degrees. If a landmark appears on the edge of a query image, it would result in an angle error of up to 30 degree.

#### 3.4.2.2 Positioning Accuracy

We now compare the ImPos and the Sextant in terms of error distance. Since the ImPos makes use of all recognized landmarks to estimate the user's location and the number of recognized landmarks affects its localization accuracy, we consider different numbers of landmarks for the ImPos. In particular, since we took five images at each

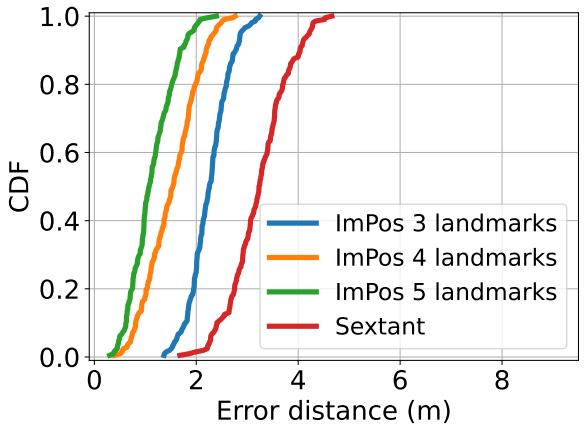



**Figure 3.7:** The impact of the number of recognized landmarks on localization accuracy

testing location, the number of recognized landmarks is at least five. For each testing location, we enumerate all the combinations of 3, 4, and 5 recognized landmarks. For every combination of landmarks, we estimate the user's location and compute the corresponding error distance. As we can see from Fig. 3.7, ImPos achieves a median error distance of 2.33 m, 1.56 m, and 1.09 m with 3, 4, and 5 recognized landmarks, respectively, which is much lower than the 3.23 m under the Sextant system. These results indicate that the ImPos achieves a much higher positioning accuracy than the Sextant because it can effectively make use of all recognized landmarks to improve its positioning accuracy.

Fig. 3.8 shows the average error distance for at every testing location with the number of recognized landmarks varying from 3 to 5. We can see that different testing locations have different error distances, which is expected. In addition, the average error distance decreases as the number of recognized landmarks increases for every testing location. Fig. 3.9 shows the box-plot of error distance with the number of recognized landmarks varying from 3 to 5. Once again, we can see that the ImPos achieves an median error distance of 2.33 m, 1.56 m, and 1.09 m with 3, 4, and 5 recognized landmarks respectively. These results further confirms the advantage of ImPos in fully utilizing all recognized landmarks to improve its localization accuracy.

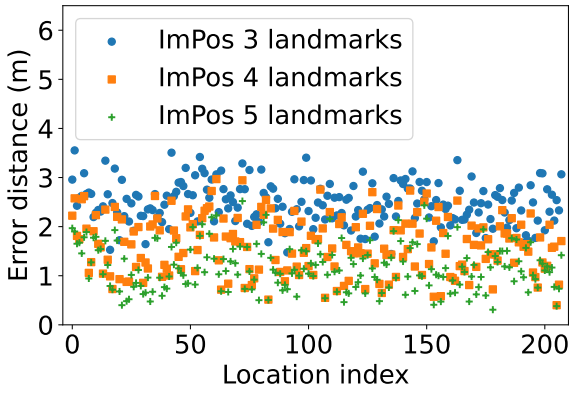



**Figure 3.8:** Error distance at difference testing locations

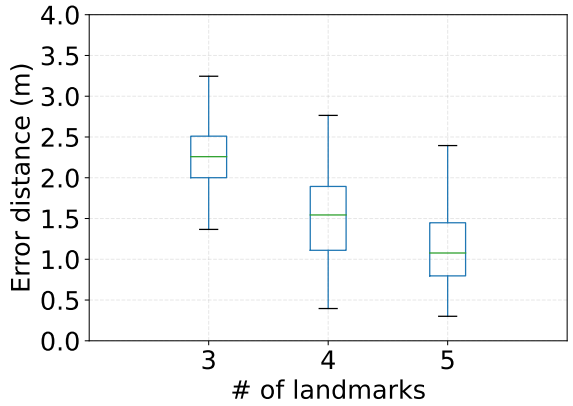



**Figure 3.9:** Box-plot of error distance with different number of recognized landmarks

#### 3.4.2.3 Comparison of Two Optimization Problem Formulations

We also compare the error distances produced by the two optimization problem formulations introduced in Section 3.3.3.3. Fig. 3.10 shows CDFs of error distance produced by the two optimization problem formulations where the number of recognized landmarks varying from 4 to 5. We ignore the case of 3 recognized landmarks here because both formulations produce the same estimated location. We can see from Fig. 3.10 that the error distance produced by the first problem formulation with

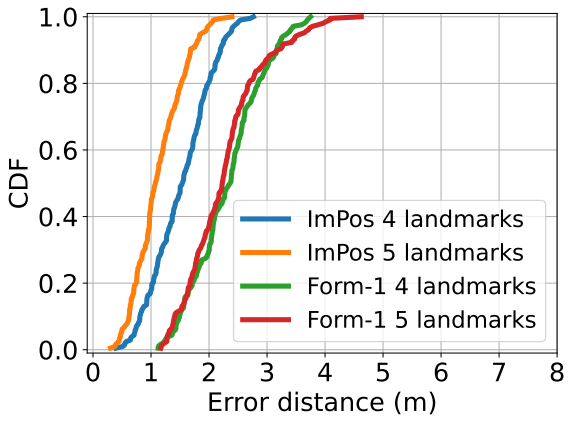



**Figure 3.10:** The impact of different cost functions on localization accuracy

unweighted cost function has a larger error distance than ImPos (i.e., the second formulation) in both cases. In addition, as the number of recognized landmarks increases from 4 to 5, the median error distance produced by the first problem formulation only reduces from 2.4 m to 2.3 m. In contrast, the median error distance under the ImPos reduces from 1.6 m to 1.2 m. These results show that the first formulation is unable to make full use of additional recognized landmarks to improve the positioning accuracy.

## 3.5 Summary

In this chapter, we have introduced the design and evaluation of ImPos, a novel image-based IPS that achieves high positioning accuracy by accurately estimating the direction from a user to recognized landmarks through image analysis and fully utilizing all the recognized landmarks. Detailed experiment results confirm the significant advantages of ImPos over prior image-based solutions based on triangulation.

# Chapter 4

# CONTINUOUS INDOOR TRACKING VIA DIFFERENTIAL RSS FINGERPRINTING

## 4.1 Introduction

Indoor navigation is necessary for users to locate themselves and reach their destinations in large unfamiliar venues such as airports, shopping malls, and hospital complex, where GPS signals are either absent or unavailable. According to a recent report [1], the global market of indoor positioning and navigation service is expected to reach 256.59 billion in 2028.

The capability of continuously tracking a user's locations is key to any indoor navigation system, which is required for computing and updating the optimal route from the user's current location to intended destinations and providing the user with visual or audio turn-by-turn instructions. As a result, a typical indoor navigation system involves an Indoor Positioning System (IPS) module and requires the user to periodically submit location queries to learn their whereabouts. Existing indoor navigation systems mainly differ in the technology behind their IPS modules, which include Computer Vision[18, 42, 60], RFID[61, 43, 62], Wi-Fi[23][3], visible light[63, 64, 45], Bluetooth[5, 6, 65], acoustic sound [9][66] and so on.

WiFi Received Signal Strength-based IPS (RSS-IPS) [23][3] is considered as one of the most classical IPSes, which exploit distinguishable RSSes at different locations as their fingerprints and locates a user by comparing the user's RSS measurement with the RSS fingerprints collected at different locations in advance. In comparison with other types of IPSes, RSS-based IPSes explore ubiquitous smartphones and WiFi infrastructure widely available in target venues and do not require costly infrastructure updates. A typical RSS-IPS works in two phases. In the offline training phase, the

IPS operator collects RSS fingerprints at different indoor reference locations. In the online positioning phase, on receiving an RSS measurement from a user, the IPS server returns the reference location of which the RSS fingerprint is the closest to the user's measurement.

RSS-IPSes face two critical challenges that limit their positioning accuracy in practice, the ambiguity of RSS fingerprints and device diversity. First, prior studies such as [67][13] have shown that multiple locations could have the same RSS fingerprint. As a result, it is difficult for the IPS server to distinguish these locations solely based on the user's RSS measurement. Second, it is also well known that different mobile devices may detect different Received Signal Strengths at the same location and time [19, 20, 21, 22]. In particular, the device used by the IPS operator to collect the RSS fingerprints may be different from the mobile devices used by the users during the online navigation phase, which means that the RSS measurement collected by the user may be different from the RSS fingerprint at the location. Both factors could result in larger differences between the RSS fingerprint and the user's RSS measurement at the same location and reduce the positioning accuracy achieved by RSS-IPSes.

Fortunately, we find that periodic location queries from the user during indoor navigation offers new opportunities to tackle WiFi fingerprint ambiguity and device heterogeneity. Specifically, instead of matching a user's RSS measurement with the fingerprint, we find that matching the difference between adjacent RSS measurements with the difference between RSS fingerprints at adjacent locations offers great resilience to device diversity. Moreover, the ambiguity of WiFi fingerprints can be effectively tackled by matching a sequence of RSS measurements from user to a movement path. Based on these observations, we introduce the design and evaluation of CITS, a novel continuous RSS-based indoor tracking system that can achieve high positioning accuracy in the presence of WiFi fingerprint ambiguity and device heterogeneity. Our contributions in this chapter are summarized as follows.

- We are the first to study continous indoor tracking in the presence of fingerprint ambiguity and device diversity.

- We introduce CITS, a novel RSS-based IPS based on differential fingerprinting and path matching that can achieve much improved positioning accuracy.
- Experiment studies based on a prototype confirm the advantages of CITS over prior RSS-based IPSes. For example, our experiment results show that CITS achieves a mean distance error at 0.8 m in contrast to the 1.68 m reported in [23] when the user travels more than 25 seconds in the indoor environment.

The rest of this chapter is structured as follows. Section 4.2 discusses the related work. Section 4.3 presents the design of CITS. Section 4.4 reports our experiment results. This chapter is finally concluded in Section 4.5.

## 4.2 Related Work

In this section, we review some of the prior works that are most germane to our work.

A conventional technique for indoor tracking is the pedestrian dead reckoning (PDR) [68]. A PDR system records the Inertial Measurement Unit sensor readings when the user is moving and calculates the user's displacements to determine his current location. Since the displacement error accumulates with the user's moving, several types of indoor signals have been explored to augment the PDR. Fang *et al.* [69] enhanced the PDR system with wireless telemetry. Carrera *et al.* [8] fused RSS, PDR, and building information to track the user's movement. In [70], Samuel *et al.* corrected the error of PDR systems in a smart building environment. They equipped the user with a RFID tag reader to read the RFID tags which are placed throughout the smart building then correct the cumulative error by reading the tags. Recently, hybrid algorithms have been proposed in [71, 72, 73] to highly improve the positioning accuracy via combining PDR approach with WiFi fingerprinting approach. Shen *et al.* [74] pointed out that user's heading error is the key factor of the error in PDR, and used WiFi-RSS to minimize the heading error resulting in a higher position accuracy in PDR. In [75], Ho *et al.* removed the interference signals for interior sensor reading with a fast Fourier transform-based smoother on the collected data then further improved

the PDR accuracy. In addition, exploring the magnetic field information is another promising way to minimize the accumulating error in PDR [76, 77, 78]. Apart from the above papers, there still exist plenty of works that involve PDR. However, none of them consider the heterogeneous device problem. In other words, the heterogeneous device problem impacts the process of correcting the accumulating error.

Apart from PDR, researchers also try to track the user with WiFi signals. Yang *et al.* [79] analyzed the localization errors of the RSS-based fingerprinting localization methods. Hoang *et al.* [80] inputted a series of WiFI RSS into hidden Markov chain (HMM) to predict the user's movement. In [81], Anthea *et al.* proposed a novel RSS indoor tracking system by using a Compressive Sensing-based positioning scheme. Channel station information has been used to infer the user's movement and activity in [82, 83]. Kleisouris *et al.* [84] utlized multiple antennas to improve localization results, while Zheng *et al.* [85] combined multiple frequencies and powers to reduce the localization errors. A RSS gradient fingerprint database was constructed in [86] to track user's movement with improved accuracy via comparing the RSS gradient between each two locations. Liu *et al.* [87] combined acoustic ranging and RSS-based localization to reduce the large errors of smartphone tracking. In [88], Yang *et al.* proposed a improved lateration based method for mitigating multipath effects. Hernández *et al.*[89] used a topological RSS radio-map then tracked the user's location through Bayes filter. Chandrasekaran *et al.*[90] compared the performance of several widely used RSS based localization algorithms under a laboratory setting. Recently, Mai *et al.*[91] utilized convolutional neural network (CNN) and Tiwary *et al.*[92] trained a deep neural network based on RSS difference to track the users in the indoor environment. There are some research efforts dedicated to robust indoor localization, such as robust localization in the presence of signal strength and access point attacks in [93, 94, 95]. However, the above works do not consider the heterogeneous problem or solve this problem under a very strict condition with a strong assumption.

Besides WiFi signals, there are also several other types of IPSes based on other different technologies. For example, image-based IPSes have been proposed in [18,

42, 60] to locate users with high positioning accuracy through recognizing landmarks in the photos. Apart from image-based IPSes, acoustic signal-based IPSes [9, 66] utilized acoustic signals as fingerprints for localization. In addition, visible light [63, 45, 64], Radio Frequency Identification (RFID) [61, 43, 62], and Bluetooth signals [5, 6, 65] could also been used for user positioning. However, the above works can only repetitively infer the user's location in a tracking system instead of considering user's prior locations.

## 4.3 CITS Design

In this section, we first give an overview and then detail CITS's design.

### 4.3.1 Overview

We design CITS based on two key ideas.

First, we observe that matching the difference between two adjacent RSS measurements from the user to the difference between two RSS fingerprints can effectively mitigate the impact of device diversity. Specifically, several prior studies [19, 20, 21, 22] have shown that for any two different devices, say $A$ and $B$, there is a linear relationship between their RSS measurements from the same AP at the same location and time. In particular, let $\mathsf{rss}_A$ and $\mathsf{rss}_B$ be the RSS measurements of devices $A$ and $B$, respectively. We have

$$\mathsf{rss}_A = \theta \cdot \mathsf{rss}_B + \delta_{AB} \ , \tag{4.1}$$

where $\theta$ and $\delta_{AB}$ are two device-dependent constants. Moreover, it has also been shown in [22] that the parameter $\theta$ is close to one in most cases, which is also used in recent work [96]. This allows us to ignore parameter $\theta$ hereafter and rewrite Eq. (4.1) as

$$\mathsf{rss}_A = \mathsf{rss}_B + \delta_{AB} \ , \tag{4.2}$$

Further let $\mathsf{rss}'_A$ and $\mathsf{rss}'_B$ denote the RSS measurements of the two devices at a different location. We get

$$\mathsf{rss}'_A = \mathsf{rss}'_B + \delta_{AB} \ . \tag{4.3}$$

Subtracting Eq. (4.3) from Eq. (4.2), we have

$$\mathsf{rss}_A - \mathsf{rss}'_A = \mathsf{rss}_B - \mathsf{rss}'_B \ , \tag{4.4}$$

which indicates that the difference of device $A$ measurements at the two locations is the same as that of device $B$ even if they are of different device models.

Second, we find that the ambiguity of RSS fingerprints can be addressed by path matching that takes a user's prior estimated locations into accounts. In particular, assume that the time is divided into intervals of same length and that a user submits one RSS measurement collected by a mobile device of an unknown type at each time interval $t = 1, 2, \ldots$. If we know that the user is at reference location $x$ at time $t$ and that the maximum distance a user can travel within one time interval is $d$, then the user must be at one of the reference locations within a distance of $d$ from reference location $x$, including reference location $x$ itself if the user has not moved. While it is possible for multiple reference locations share the same RSS fingerprint, by limiting the search space for user's location at time $t+1$ can greatly reduce the chance of such ambiguity.

In what follows, we detail the design of CITS, which consists of *differential RSS fingerprint database construction* and *continuous tracking via path matching*.

### 4.3.2 Differential RSS Fingerprint Database Construction

We first divide the whole indoor venue into $n$ cells of equal size, e.g., 0.5×0.5 $\mathrm{m}^2$, and select the center of each cell as one reference location. We then construct a directed graph $G = (V, E)$, where $V$ is the set of $n$ vertexes with each corresponding to one reference location, and $E$ is the set of edges. Two vertexes $x$ and $y$ are connected by a pair of antiparallel edges edge $e(x, y)$ and $e(y, x)$ if the minimal distance between

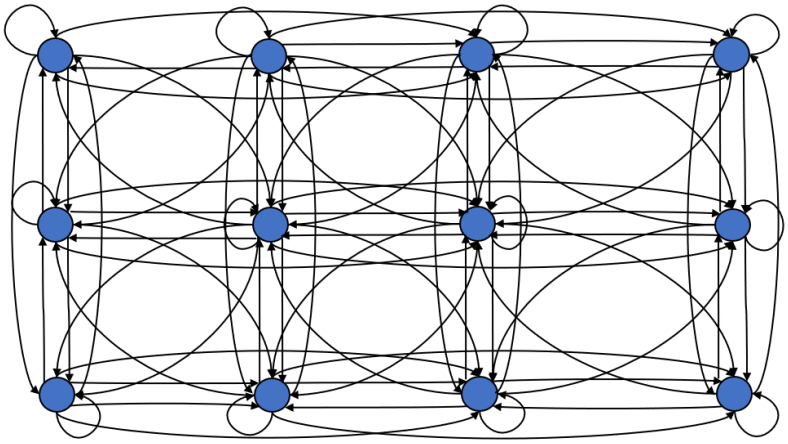


**Figure 4.1:** An example of the graph with 12 vertexes, i.e., reference locations.

cell $x$ to cell $y$ is no more than $d$, where $d$ is a system parameter denoting the maximum distance a user can travel between two consecutive RSS measurements. In addition, every vertex $x \in V$ has a self-loop, i.e., an edge that originates from and terminates at $x$, to account for the cases that the user remain in the same cell between two consecutive RSS measurements. Fig. 4.1 shows an example of the graph with 12 vertexes, where two vertexes are neighboring vertexes if there is at most one cell between them.

Assume that there are $m$ WiFi APs in the indoor venue. The CITS operator collects one RSS fingerprint $\mathsf{rss}_x = (\mathsf{rss}_x^1, \ldots, \mathsf{rss}_x^m)$ at each reference location $x \in V$, where $\mathsf{rss}_x^j$ is the RSS of AP $j$ for all $1 \leq j \leq m$.

Next, for every edge $e(x, y) \in E$, we calculate the RSS difference between vertexes $x$ and $y$ as

$$\triangle \mathsf{rss}_{x,y} = (\triangle \mathsf{rss}_{x,y}^1, \ldots, \triangle \mathsf{rss}_{x,y}^m) \ . \tag{4.5}$$

where

$$\triangle \mathsf{rss}_{x,y}^j = \mathsf{rss}_y^j - \mathsf{rss}_x^j \ ,$$

is the difference between the AP $j$'s RSS fingerprints at reference locations $x$ and $y$ for all $1 \leq j \leq m$.

Finally, we store the constructed RSS differential fingerprint database as $\{\mathsf{rss}_x | x \in V\} \bigcup \{\triangle\mathsf{rss}_{x,y} | e(x,y) \in E\}$.

### 4.3.3 Continuous Tracking via Path Matching

With the differential WiFi fingerprint database in place, we now illustrate how to continuously track a user via path matching. Denote by $\mathsf{rss}_t = (\mathsf{rss}_t^1, \ldots, \mathsf{rss}_t^m)$ the user's RSS measurement at time $t = 1, 2, \ldots$, where $\mathsf{rss}_t^j$ is the RSS measurement for AP $j$ for all $1 \leq j \leq m$. Consider two adjacent RSS measurements $\mathsf{rss}_{t-1}$ and $\mathsf{rss}_t$. We define the RSS difference for AP $j$ from time $t-1$ to time $t$ as

$$\triangle\mathsf{rss}_t = (\triangle\mathsf{rss}_t^1, \ldots, \triangle\mathsf{rss}_t^m) \ ,$$

where $\triangle\mathsf{rss}_t^j = \mathsf{rss}_t^j - \mathsf{rss}_{t-1}^j$ for all $1 \leq j \leq m$.

Now consider an edge $e(x,y) \in E$ with RSS difference $\triangle\mathsf{rss}(x,y)$. Intuitively, if $\triangle\mathsf{rss}_t \approx \triangle\mathsf{rss}(x,y)$, then it is likely that the user has moved from reference location $x$ to $y$ between time $t-1$ and $t$. Based on this observation, we further define the overall RSS difference with respect to time $t$ and edge $e(x,y)$ as

$$\phi(t, e(x,y)) = \sum_{j=1}^{m} |\triangle\mathsf{rss}_t^j - \triangle\mathsf{rss}_{x,y}^j| \ .$$

Similarly, for any path $p = \{x_1 \rightarrow x_2 \rightarrow \cdots \rightarrow x_t\}$ consisting of a sequence of $t$ vertexes, we define the overall path RSS difference as

$$\phi_p = \sum_{i=2}^{t} \phi(i, e(x_{i-1}, x_i)) \ .$$

The smaller $\phi_p$, the more likely the user traverses path $p$ from time 1 to $t$.

The CITS server always maintains the $s$ most likely paths the user traverses that have the smallest overall path RSS difference, where $s \geq 1$ is a system parameter. Upon receiving a new RSS measurement $\mathsf{rss}_t$ from the user at time $t$, the CITS server

recomputes the $s$ most likely paths. Let $p_\varphi = \langle x_1^\varphi \to \cdots \to x_{t-1}^\varphi \rangle$ denote the $\varphi$th most likely path at time $t-1$ for all $1 \le \varphi \le s$. Also denote by $N(x)$ the set of neighboring cells of cell $x$. Then the set of all possible new paths are $P = \{\langle x_1^\varphi \to \cdots \to x_{t-1}^\varphi \to x \rangle | x \in N(x_l^\varphi) \bigcup \{x_l^\varphi\}, 1 \le \varphi \le s\}$. The CITS server then finds the $s$ paths with the smallest overall path RSS difference and returns the last reference location of the most likely path as the user's current location.

In what follows, we detail the procedure of estimating the user's location at each time $t = 1, 2, \ldots$.

#### 4.3.3.1 At Time $t = 1$

Since the user has submitted only one RSS measurement $\mathsf{rss}_1$ at time $t = 1$ which is insufficient for path matching, we estimate the user's location $l_1$ according to the classical RSS-IPS Radar [23] and record the $s$ most likely reference locations for later path matching.

Specifically, on receiving the user's RSS measurement $\mathsf{rss}_1 = (\mathsf{rss}_1^1, \ldots, \mathsf{rss}_1^m)$, we estimate the user's location at time 1 as

$$l_1 = \arg\min_{x \in V} \sum_{j=1}^{m} (\mathsf{rss}_x^j - \mathsf{rss}_1^j)^2 \ , \tag{4.6}$$

which is the reference location with the closest RSS fingerprint to $\mathsf{rss}_1$ measured by the Euclidean distance. In addition, we also record the set of $s$ reference locations whose RSS fingerprints are closest to $\mathsf{rss}_1$, denoted by $P_1$.

#### 4.3.3.2 At Time $t = 2$

On receiving the user's RSS measurement $\mathsf{rss}_2$ at time $t = 2$, we estimate the user's location $l_2$ via path matching.

First, we generate a set of candidate paths from the set of most likely reference locations $S_1$ and graph $G$. Specifically, for each reference location $x \in P_1$, we generate

a set of candidate path as $C_x = \{\langle x \to y\rangle | y \in N(x) \bigcup \{x\}\}$. The set of all candidate paths is then $C = \bigcup_{x \in P_1} C_x$.

Second, we find the candidate path with the smallest overall path difference whereby to determine the user's current location $l_t$. Specifically, for every candidate path $p = \langle x \to y\rangle \in P$, we computes its overall path difference as

$$\begin{aligned} \phi_{\langle x \to y\rangle} &= \phi(2, e(x, y)) \\ &= \sum_{j=1}^{m} |\triangle \mathsf{rss}_t^j - \triangle \mathsf{rss}_{x,y}^j| \ . \end{aligned} \tag{4.7}$$

The most likely path is then given by

$$\langle x^*, y^*\rangle = \arg\min_{\langle x \to y\rangle \in C} \phi_{\langle x \to y\rangle} \ ,$$

and the user's current location is estimated as $l_2 = y^*$. Moreover, we also record the set of $s$ most likely paths denoted by $P_2$.

#### 4.3.3.3 At Time $t > 2$

On receiving the user's RSS measurement $\mathsf{rss}_t$ at time $t > 2$, we estimate the user's location $l_t$ in a similar fashion.

First, we generate a set of candidate paths from the set of most likely paths $P_{t-1}$ and graph $G$. Specifically, for each path $p = \langle x_1 \to \cdots \to x_{t-1}\rangle \in P_{t-1}$, we generate a set of candidate path as $C_p = \{\langle x_1 \to \cdots \to x_{t-1} \to y) | y \in N(x_{t-1}) \bigcup \{x_{t-1}\}\}$. The set of all candidate paths is then $C = \bigcup_{p \in P_{t-1}} C_p$.

Second, we find the candidate path with the smallest overall path difference whereby to determine the user's current location $l_t$. Specifically, for every candidate path $p = \langle x_1 \to \cdots \to x_{t-1} \to y\rangle \in C$, we computes its overall path difference as

$$\begin{aligned} \phi_p &= \phi_{\langle x_1 \to \cdots \to x_{t-1} \to y\rangle} \\ &= \sum_{i=1}^{t-1} \phi(i, e(x_{i-1}, x_i)) + \phi(t, e(x_{t-1}, y)) \ . \end{aligned}$$

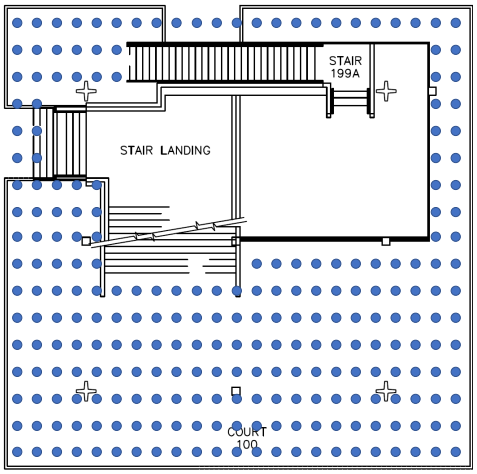



**Figure 4.2:** The floor plan of the office space

The most likely path is then given by

$$\langle x_1^* \to \cdots \to x_t^* \rangle = \arg\min_{p \in C} \phi_p \ ,$$

and the user's current location is estimated as $l_t = x_t^*$. Moreover, we also record the set of $s$ most likely paths denoted by $P_t$.

We summarize the whole procedure in Algorithm 1.

## 4.4 Performance Evaluation

In this section, we evaluate the performance of CITS via detailed experimental studies based on real prototype.

### 4.4.1 Prototype Implementation and Data Collection

We implement a prototype of CITS in Android studio/Java and deploy it on a square zone of 23.8 × 23.8 $m^2$ inside an office building with the floor plan shown in Fig. 4.2. We chose $n = 368$ reference locations in the indoor venue and detected $m = 41$ WiFi APs based on unique SSIDs.

**Algorithm 2:** Location Estimation

**input :** Graph $G = (V, E)$, RSS measurement $\mathsf{rss}_t$ at time $t = 1, 2, \ldots,$ parameter $s$

**output:** User's location $l_t$ at each time $t$

**if** $t = 1$ **then**
  $l_1 \leftarrow \arg\min_{x \in V} \sum_{j=1}^{m} (\mathsf{rss}_x^j - \mathsf{rss}_1^j)^2$;
  Record the set of $s$ most likely reference locations as $P_1$;
**end**
**if** $t = 2$ **then**
  **foreach** $x \in P_1$ **do**
    $C_x \leftarrow \{\langle x \rightarrow y \rangle | y \in N(x) \bigcup \{x\}\}$;
  **end**
  $C \leftarrow \bigcup_{x \in P_1} C_x$;
  **foreach** $\langle x \rightarrow y \rangle \in C$ **do**
    $\phi_{\langle x \rightarrow y \rangle} \leftarrow \sum_{j=1}^{m} |\triangle \mathsf{rss}_t^j - \triangle \mathsf{rss}_{x,y}^j|$;
  **end**
  $\langle x^*, y^* \rangle \leftarrow \arg\min_{\langle x \rightarrow y \rangle \in C} \phi_{\langle x \rightarrow y \rangle}$;
  $l_2 \leftarrow y^*$;
  Record the set of $s$ most likely candidate paths as $P_2$;
**end**
**if** $t > 2$ **then**
  **foreach** $p = \langle x_1 \rightarrow \cdots \rightarrow x_{t-1} \rangle \in P_{t-1}$ **do**
    $C_p \leftarrow \{\langle x_1 \rightarrow \cdots \rightarrow x_{t-1} \rightarrow y) | y \in N(x_{t-1}) \bigcup \{x_{t-1}\}\}$;
  **end**
  $C \leftarrow \bigcup_{p \in P_{t-1}} C_p$;
  **foreach** $p = \langle x_1 \rightarrow \cdots \rightarrow x_{t-1} \rightarrow y) \in C$ **do**
    $\phi_p \leftarrow \sum_{i=1}^{t-1} \phi(i, e(x_{i-1}, x_i)) + \phi(t, e(x_{t-1}, y))$;
  **end**
  $\langle x_1^* \rightarrow \cdots \rightarrow x_t^* \rangle \leftarrow \arg\min_{p \in C} \phi_p$;
  $l_t \leftarrow x_t^*$;
  Record the set of $s$ most likely candidate paths as $P_t$;
**end**
**return** $l_t$;

We use three models of smartphones for our data collection, including a Huawei Honor 8 (HH8), a Motorala One (MO), and a Google Pixel 6 (GP6). Table. 4.2 summarizes the configuration of each smartphone model. For each reference location, we record its coordinate and collect RSS measurement using all three smartphones at the same time. We then create three differential RSS fingerprint databases with one

for each smartphone.

We then use each of three smartphones to collect 60 movement traces. For each trace, we have one user carrying the smartphone walk in the indoor venue randomly for a duration of 28 seconds. The user stops every 1 second to record a test location and collects the corresponding RSS measurement. Note that the test locations may not be the same as any reference location.

Since CITS is mostly related to RADAR [2], the classical RSS-IPS that matches a user's RSS measurement to the reference location with the closest RSS fingerprint, we compare it with RADAR, i.e., the server estimates the user's location via RADAR on receiving the user's RSS measurement. Moreover, we use *error distance* as our performance metric, which is defined as the user's true location and the estimated reference location.

**Table 4.1:** Default Experiment Settings

| **Para.** | **Value** | **Description** |
|---|---|---|
| $n$ | 368 | # of reference locations |
| $m$ | 41 | # of APs |
| | 1.1 m | The size of each cell |
| | 1 m | User's maximum travel distance |
| $d$ | 2.2 m | Maximum distance between two vertexes connected by an edge |
| $s$ | 15 | # of candidate paths recorded |

**Table 4.2:** Smartphone Configurations

| Brand | Model | CPU | RAM |
|---|---|---|---|
| Huawei | Honor 8 | 2.3GHz Octa-core | 4G |
| Motorala | One | 2GHz ARM Cortex-A53 | 4G |
| Google | Pixel 6 | 2.8GHz Octa-core | 8G |

### 4.4.2 Experiment Results

We now report our experiment results.

#### 4.4.2.1 Validation of Device Diversity

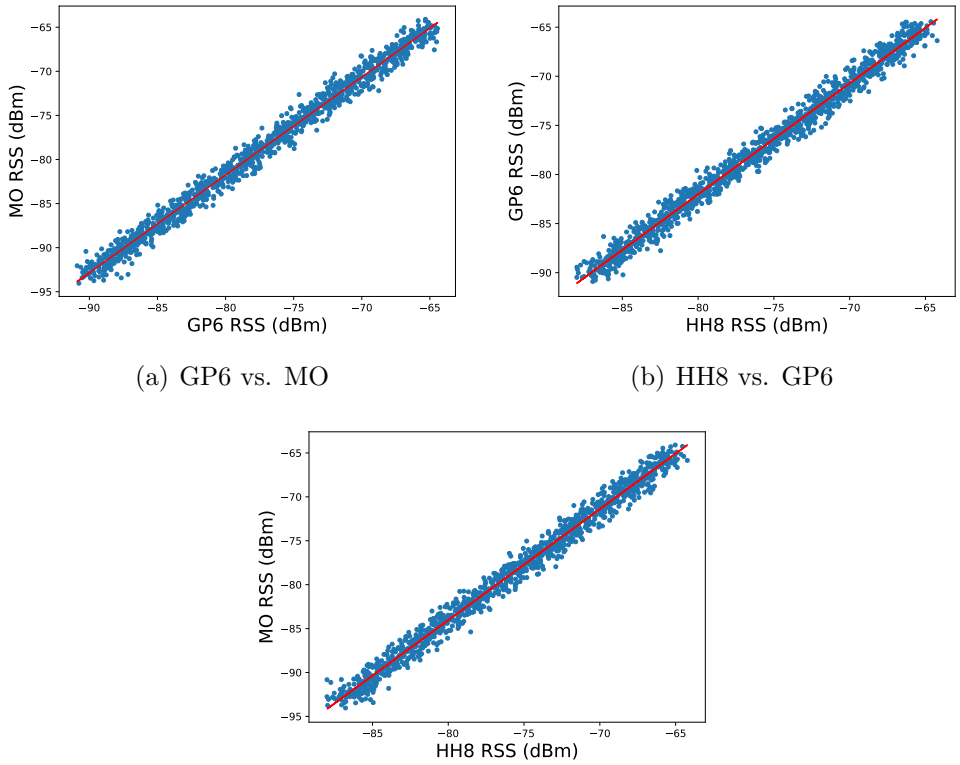
Since CITS relies on the assumption that there is a linear relationship between two devices' RSS measurements from the same AP at the same location and time and that the slope of the linear relationship is close to one, we first validate this assumption using the RSS measurements collected by different devices at the same location and time.

Figs. 4.3(a) to 4.3(c) plot the relationships between the RSS measurements collected by GP6 and MO, those collected by HH8 and MO, and those collected by GP6 and HH8, respectively. As we can see from all three figures, the RSS measurements collected by difference devices at the same locations and times exhibit a linear relationship, which is expected.

We further perform linear regression on RSS measurements collected by each pair of devices according to both Eq. (4.1) and Eq. (4.2). Tables 4.3 and 4.4 show the parameters obtained from the two types of linear regression, respectively. As we can see from Table 4.3, the slope of the linear equation under Eq. (4.1) ranges from 0.78 to 1.26 and the Mean Square Error (MSE) ranges from 0.74 to 0.94 for the six pairs of devices. In addition, Table 4.4 shows that the MSE ranges from 1.19 to 1.94 if we set the slope to one according to Eq. (4.2). While the MSE achieved under Eq. (4.2) is larger than that under Eq. (4.1), these results do indicate that it is reasonable to assume that the slope is one. As we will see shortly, CITS works well in practice under this assumption.

#### 4.4.2.2 Error Distance Over Time

Figs. 4.4(a) to 4.4(c) compare the average error distances of CITS and RADAR at each time with the user's RSS measurements collected by HH8, MO, and GP6, respectively, where the differential RSS fingerprint database is constructed from the




(a) GP6 vs. MO

(b) HH8 vs. GP6

(c) HH8 vs. MO

**Figure 4.3:** Relationship between RSS measurements collected by different pairs of devices

**Table 4.3:** Linear Regression under Eq. (4.1)

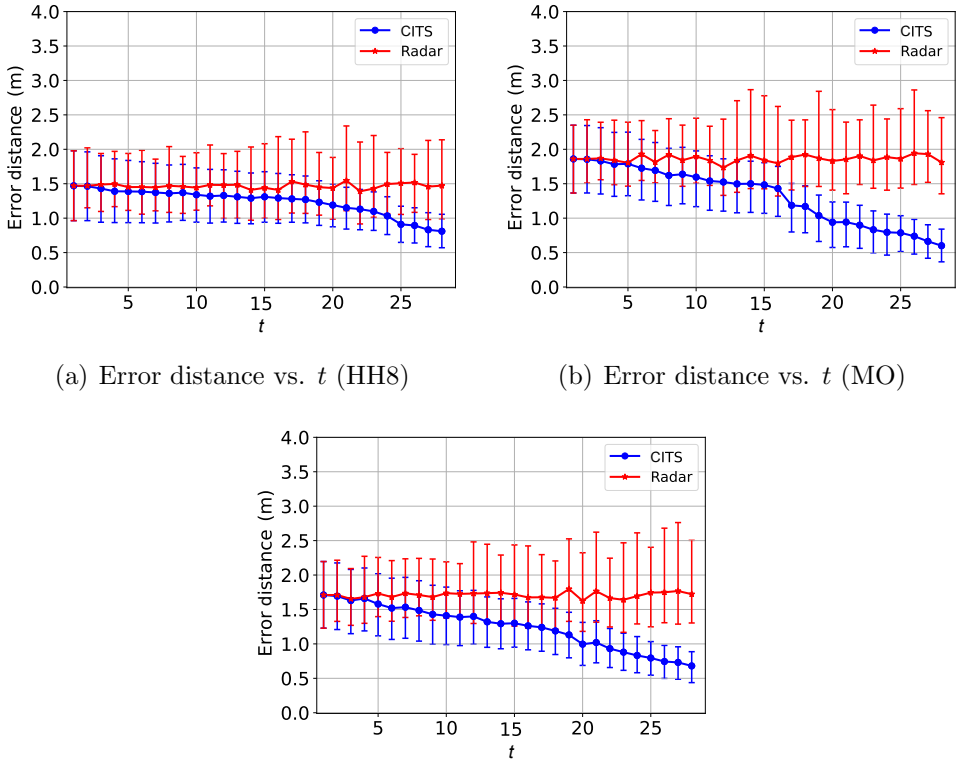
| Device $A$ | Device $B$ | $\theta$ | $\delta_{AB}$ | MSE |
|---|---|---|---|---|
| GP6 | MO | 1.11 | 7.29 | 0.89 |
| HH8 | GP6 | 1.12 | 8.34 | 0.86 |
| HH8 | MO | 1.26 | 17.21 | 0.94 |
| GP6 | HH8 | 0.87 | -8.37 | 0.76 |
| MO | HH8 | 0.78 | -14.44 | 0.74 |
| MO | GP6 | 0.88 | -7.42 | 0.80 |

RSS measurements collected by HH8 and every bar represents the 70% confidence interval.

We can see from Figs. 4.4(a) to 4.4(c) that the average error distance under

**Table 4.4:** Linear Regression under Eq. (4.2)

| Device $A$ | Device $B$ | $\theta$ | $\delta_{AB}$ | MSE |
|---|---|---|---|---|
| GP6 | MO | 1 | -1.48 | 1.21 |
| HH8 | GP6 | 1 | -1.50 | 1.19 |
| HH8 | MO | 1 | -2.99 | 1.94 |
| GP6 | HH8 | 1 | 1.50 | 1.19 |
| MO | HH8 | 1 | 2.99 | 1.94 |
| MO | GP6 | 1 | 1.48 | 1.21 |



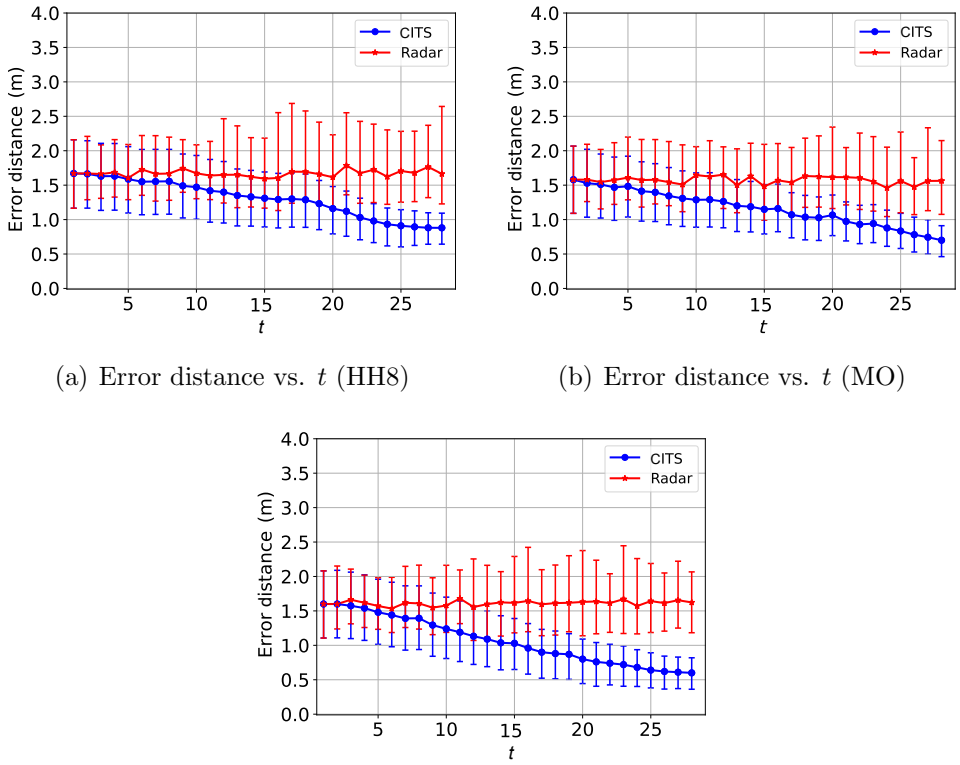

(a) Error distance vs. $t$ (HH8)

(b) Error distance vs. $t$ (MO)

(c) Error distance vs. $t$ (GP6)

**Figure 4.4:** Error distance with different user devices and differential RSS fingerprint database constructed from RSS measurements collected by HH8

RADAR is relatively stable across different times. This is expected, as RADAR treats every RSS measurement from the user as an independent location query and does not consider the user's past location. In addition, the average error distance under



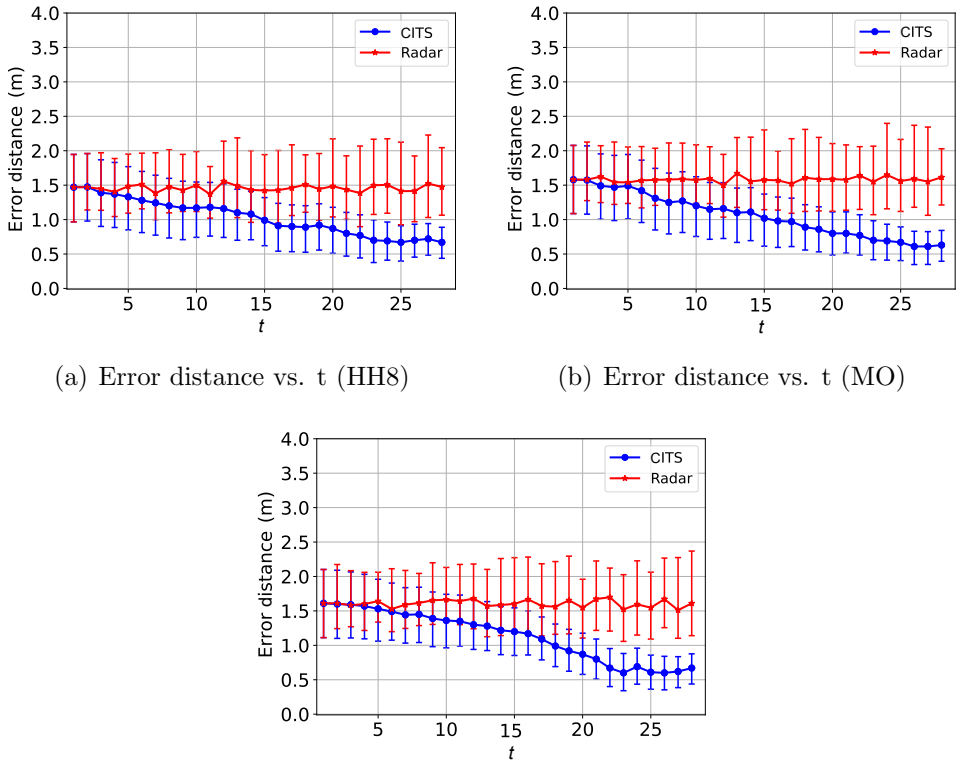

(a) Error distance vs. $t$ (HH8)

(b) Error distance vs. $t$ (MO)

(c) Error distance vs. $t$ (GP6)

**Figure 4.5:** Error distance with different user devices and differential RSS fingerprint database constructed from RSS measurements collected by MO

RADAR is 1.49 m, 1.88 m, and 1.75 m for HH8, MO, and GP6, respectively. Among them, the average error distance for HH8 is the lowest, which is anticipated because the differential RSS fingerprint database is constructed from the measurements collected by the same device. These results further confirm the negative impact of device diversity on positioning accuracy. In contrast, we can see from all three figures that the error distance of CITS matches that of RADAR at time $t = 1$ initially and decreases over time for all three different user devices. This is also anticipated, because CITS estimates the user's location at time $t = 1$ according to RADAR and thus has the same error distance as RADAR. In addition, CITS locates the user through differential RSS fingerprint matching and path matching at time $t \geq 2$ that can effectively mitigate the impact of device diversity and achieve higher positioning accuracy over time. For example, we




(a) Error distance vs. t (HH8)

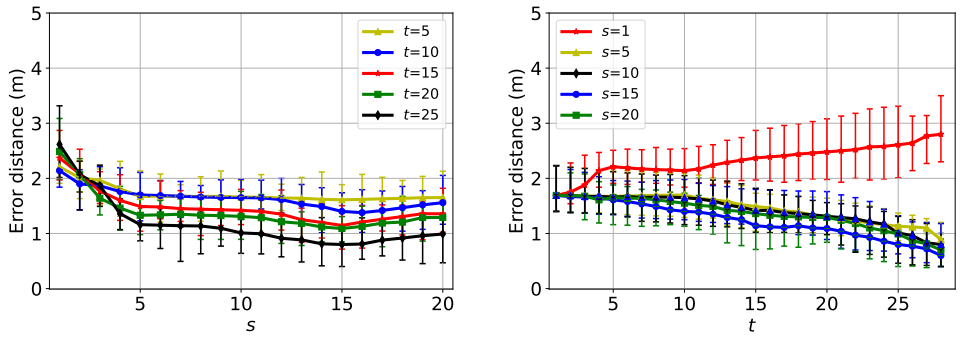
(b) Error distance vs. t (MO)

(c) Error distance vs. t (GP6)

**Figure 4.6:** Error distance with different user devices and differential RSS fingerprint database constructed from RSS measurements collected by GP6

can see from Fig. 4.4(b) that the average error distance decreases from 1.49 m at time $t = 1$ to 0.53 m at time $t = 28$.

Figs. 4.5 and 4.6 show the average error distances of CITS and RADAR at each time with the user's RSS measurements collected by HH8, MO, and GP6, respectively, where the differential RSS fingerprint database is constructed from the RSS measurements collected by MO and GP6, respectively. Similar to what we have seen from Fig. 4.4, the average error distance under RADAR is relatively stable across different times for all cases. Moreover, the average error distance is the lowest when the user's device is the same as the device used for collect RSS fingerprints. Last but not the least, the average error distance of CITS decreases over time. These results clearly demonstrate the advantages of CITS over Radar.




(a) Error distance vs. $s$ (b) Error distance vs. $t$

**Figure 4.7:** The average distance error of CITS over time under different $s$.

#### 4.4.2.3 The Impact of Parameter $s$

Fig. 4.7 shows the average error distance of CITS at different times and under different $s$, i.e., the number of most likely candidate paths that the server records at every time. Specifically, Fig. 4.7 shows the average error distance of CITS at time $t = 5, 10, 15, 20,$ and 25 as $s$ increases from 1 to 20. We can see from Fig. 4.7(a) that the average error distance of CITS first decreases quickly as $s$ increases from 1 to 5, then decreases relatively slowly as $s$ further increases from increases from 5 to 15, and finally remains relatively stable or slightly fluctuates as $s$ further increases from increases from 15 to 20. Consider $t = 25$ as an example. The average error distance decreases from 2.6 m to 0.80 m as $s$ increases from 1 to 15, and then slightly increases to 0.99 m as $s$ further increases to 20. The reason for the initial decrease in the average distance error is the more most likely candidate paths that the server stores, the higher possibility that the user's true path is among them. Therefore, the average error distance decreases initially. Moreover, having the server records too many most likely candidate paths at each time $t$ could negatively affect positioning accuracy. The reason is there is non-negligible probability that a candidate paths ranked lowly earlier may become a top ranked path at some point due to the fluctuation of RSS and thus mislead the server into choosing an incorrect path. Therefore, parameter $s$ needs to be

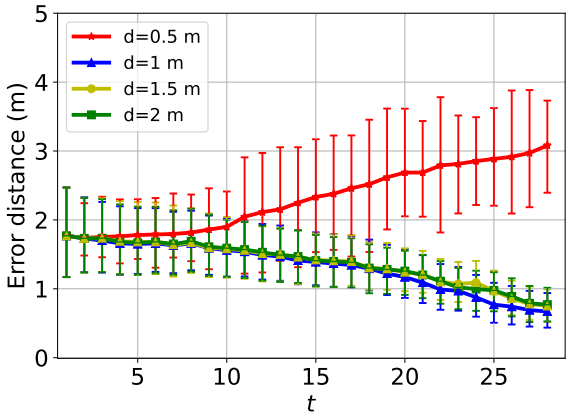



**Figure 4.8:** The average distance error of CITS over time under different $d$.

carefully chosen. Our experiment results show that CITS achieves the best positioning accuracy when $s = 15$.

Fig. 4.7(b) shows the average error distance of CITS over time for $s = 1, 5, 10, 15$, and 20. We can see that the average error distance of CITS increases over time when $s = 1$. This is expected, as the true path may not be the most likely path determined via path matching. Once an incorrect path is selected by the CITS server, subsequent location estimations will be affected by previous positioning error, which would result in increased error distance over time. Moreover, we can also see that the error distance of CITS decreases over time if $s$ is larger than 5. This is expected and also confirms the effectiveness of the path matching.

#### 4.4.2.4 The Impact of Parameter $d$

Recall that the differential RSS fingerprint database requires us to construct a graph $G = (V, E)$, in which any two vertexes are connected by an edge if the minimal distance between the two cells they represent is smaller than $d$. Ideally, parameter $d$ should be set sufficiently large so that it is impossible for a user to travel from one cell to another cell that are not connected by any edge between two consecutive RSS measurements.

Fig. 4.8 shows the the average error distance of CITS over time with $d$ varying from 0.5 m to 2 m where the true maximum travel distance of the user is 1 m. We can see that when $d = 0.5$ m that is smaller than the true true maximum travel distance of the user, the average error distance of CITS increases from 1.77 m at $t = 1$ to 3.08 m at time $t = 28$. This is expected because if $d$ is set too small, the graph $G$ does not contain the user's true path. As a result, CITS is unable to find the user's path and correctly estimate the user's location. Moreover, we can also see that when average error distance of CITS decreases over time for $d = 1$ m, 1.5 m, and 2 m. This is because $d$ is sufficiently large and the user's true path can be captured and identified by CITS in these cases. Last but not the least, we can see that the average error distance of CITS at time $t > 25$ when $d = 2$ m is slightly larger than that when $d = 1$ m. This is also expected, because the larger $d$, the more candidate paths needs to be examined at each time, the higher the probability that an incorrect path is chosen by the CITS server, which would result in slightly higher error distance.

### 4.4.3 Summary

We summarize the experiment result as follows.

- Devices of different models could collect different RSS measurements at the same time and location, but there is a linear relationship between the RSS measurements collected by different devices. It is also reasonable to assume that the RSS measurements collected by different devices differ by a device-specific constant.
- CITS achieves higher positioning accuracy than RADAR in the presence of device diversity and the average error distance of CITS decreases over time.
- CITS achieves higher positioning accuracy when the user's smartphone is of the same model as the one used for collecting RSS fingerprints.
- Parameters $s$ and $d$ both need to set sufficiently large to ensure that the user's true path is always included in the candidate path set at each time.
- Setting parameters $s$ and $d$ too large have a negative impact on the positioning accuracy of CITS due to the introduction of too many candidate paths.

## 4.5 Summary

In this chapter, we have introduced the design and evaluation of CITS, a novel continuous indoor tracking system. CITS achieves high positioning accuracy in the presence of RSS fingerprint ambiguity and device diversity through differential RSS fingerprinting and path matching. Detailed experiment studies based on real prototype implementation have confirmed the significant advantages of CITS over prior alternative solutions. As our future work, we plan to investigate extending CITS by removing the assumption of $\theta = 1$ by gradually estimating $\theta$ as more RSS measurements are received.

# Chapter 5

# CONCLUSION

This dissertation tackles several key secure and usable challenges in indoor positioning and tracking systems. First, to cope with the existing trade-off between indoor positioning accuracy and attack resilience, we proposed a novel weighted multi-vote indoor positioning system(MV-IPS). Inspired by bonda count voting scheme, MV-IPS treats APs as voters and reference points as candidates. Extensive simulation studies based on real RSS measurements have confirmed the accuracy and the attack resilience of MV-IPS.

Second, we introduced a novel image-based IPS (ImPos) for indoor venues where WiFi signals are unavailable or unreliable. Realizing the setup and processing cost for 3D model construction image-based IPS and the low accuracy property for geometry image-based IPS, our ImPos improves the locating accuracy by correcting angle measurements through image analysis. We further enhance the performance of ImPos by considering varies number of landmarks. Trace-driven experiments based on real landmarks and images have confirmed the accuracy of ImPos and the effectiveness of locating the user with varies number of landmarks.

Last but not least, we developed a novel continuous WiFi-based indoor tracking system(CITS) which can localize a user with high accuracy in the presence of RSS fingerprint ambiguity and device diverstiy. By exploiting a novel differential WiFi fingerprint database and and efficient path matching, CITS localize the user with periodic location queries by matching a sequence of RSS measurements from user to a movement path. Detailed experiment studies based on real RSS-data confirm the significant advantages of CITS over prior RSS-based solutions.

# BIBLIOGRAPHY


[1] "Global indoor positioning and navigation system market – industry trends and forecast to 2028," 2021. [Online]. Available: https://www.databridgemarketresearch.com/reports/global-indoor-positioning-and-navigation-system-market

[2] P. Bahl, V. N. Padmanabhan, and A. Balachandran, "Enhancements to the radar user location and tracking system," *Microsoft Research*, vol. 2, no. MSR-TR-2000-12, pp. 775–784, Feb. 2000.

[3] M. Youssef and A. Agrawala, "The horus wlan location determination system," in *International Conference on Mobile Systems, Applications, and Services (MobiSys'05)*, Seattle, WA, June 2005, pp. 205–218.

[4] K. Wu, J. Xiao, Y. Yi, D. Chen, X. Luo, and L. M. Ni, "Csi-based indoor localization," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 7, pp. 1300–1309, 2013.

[5] G. Fischer, B. Dietrich, and F. Winkler, "Bluetooth indoor localization system," 01 2004.

[6] Y. Wang, Q. Ye, J. Cheng, and L. Wang, "Rssi-based bluetooth indoor localization," in *11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN'15)*, 2015, pp. 165–171.

[7] R. Giuliano, G. Cardarilli, C. Carlo, N. Carlo, Di, F. Luca, F. Francesca, M. Rocco, R. Franco, V. Marco, and Alessandro, "Indoor localization system based on bluetooth low energy for museum applications," *Electronics*, vol. 9, no. 6, 2020.

[8] C. Fischer, K. Muthukrishnan, M. Hazas, and H. Gellersen, "Ultrasound-aided pedestrian dead reckoning for indoor navigation," in *Mobile Computing and Networking(MOBICOM'08)*, 2008.

[9] S. P. Tarzia, P. A. Dinda, R. P. Dick, and G. Memik, "Indoor localization without infrastructure using the acoustic background spectrum," in *Proceedings of the 9th International Conference on Mobile Systems, Applications, and Services(MobiSys'11)*, 2011, p. 155–168.

[10] R. Armin, B. Marcel, and K. Reiner, "Precise self-calibration of ultrasound based indoor localization systems," in *International Conference on Indoor Positioning and Indoor Navigation(IPIN'11)*, 2011, pp. 1–8.

[11] Y. Chen, K. Kleisouris, X. Li, W. Trappe, and R. P. Martin, "A security and robustness performance analysis of localization algorithms to signal strength attacks," *ACM Trans. Sen. Netw.*, vol. 5, no. 1, pp. 2:1–2:37, Feb. 2009.

[12] Z. Li, W. Trappe, Y. Zhang, and B. Nath, "Robust statistical methods for securing wireless localization in sensor networks," in *International Symposium on Information Processing in Sensor Networks (IPSN'05)*, Los Angeles, CA, April 2005, pp. 1–12.

[13] L. Yuan, Y. Hu, Y. Li, R. Zhang, Y. Zhang, and T. Hedgpeth, "Secure rss-fingerprint-based indoor positioning: Attacks and countermeasures," in *IEEE Conference on Communications and Network Security (CNS'18)*, Beijing, May 2018, pp. 1–9.

[14] A. Kushki, K. N. Plataniotis, and A. N. Venetsanopoulos, "Sensor selection for mitigation of rss-based attacks in wireless local area network positioning," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP'08)*, Las Vegas, NV, March 2008, pp. 2065–2068.

[15] J. Yang, Y. Chen, V. B. Lawrence, and V. Swaminathan, "Robust wireless localization to attacks on access points," in *IEEE Sarnoff Symposium*, Princeton, NJ, March 2009, pp. 1–5.

[16] S. Fang, C. Chuang, and C. Wang, "Attack-resistant wireless localization using an inclusive disjunction model," *IEEE Transactions on Communications*, vol. 60, no. 5, pp. 1209–1214, May 2012.

[17] W. Martin, K. Moritz, and M. Chadly, "Indoor positioning using smartphone camera," in *2011 International Conference on Indoor Positioning and Indoor Navigation*, 2011, pp. 1–6.

[18] G. Ruipeng, T. Yang, Y. Fan, L. Guojie, B. Kaigui, W. Yizhou, W. Tao, and L. Xiaoming, "Sextant: Towards ubiquitous indoor localization service by photo-taking of the environment," *IEEE Transactions on Mobile Computing*, vol. 15, no. 2, pp. 460–474, 2016.

[19] S. Yang, P. Dessai, M. Verma, and M. Gerla, "FreeLoc: Calibration-free crowdsourced indoor localization," in *INFOCOM'13*, April 2013, pp. 2481–2489.

[20] F. Dong, Y. Chen, J. Liu, Q. Ning, and S. Piao, "A calibration-free localization solution for handling signal strength variance," *Mobile Entity Localization and Tracking in GPS-less Environnments*, vol. 5801, pp. 79–90, 2009.

[21] A. Goswami, L. E. Ortiz, and S. R. Das, "WiGEM: a learning-based approach for indoor localization," in *CoNEXT '11*, Tokyo, Japan, 2011, pp. 3:1–3:12.

[22] J. geun Park, D. Curtis, S. Teller, and J. Ledlie, "Implications of device diversity for organic localization," in *INFOCOM'11*, April 2011, pp. 3182–3190.

[23] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *IEEE International Conference on Computer Communications (INFOCOM'00)*, vol. 2, Tel Aviv, Israel, March 2000, pp. 775–784.

[24] "Wi-fi indoor location in retail worth $2.5 billion by 2020," 2015. [Online]. Available: https://www.abiresearch.com/press/wi-fi-indoor-location-retail-worth-25-billion-2020

[25] R. M. Adelsman and A. B. Whinston, "Sophisticated voting with information for two voting functions," *Journal of Economic Theory*, vol. 15, no. 1, pp. 145 – 159, 1977.

[26] S. He and S.-H. G. Chan, "Sect junction: Wi-fi indoor localization based on junction of signal sectors," in *IEEE International Conference on Communications (ICC'14)*, Sydney, NSW, June 2014, pp. 2605–2610.

[27] Y. Jiang, X. Pan, K. Li, Q. Lv, R. P. Dick, M. Hannigan, and L. Shang, "ARIEL: automatic wi-fi based room fingerprinting for indoor localization," in *ACM Conference on Ubiquitous Computing (UbiComp'12)*, Pittsburgh, PA, September 2012, pp. 441–450.

[28] C. Wu, L. Fu, and F. Lian, "Wlan location determination in e-home via support vector classification," in *IEEE International Conference on Networking, Sensing and Control (ICNSC'04)*, vol. 2, Taipei, Taiwan, March 2004, pp. 1026–1031, Vol.2.

[29] G. Nuno and J. Paez Borrallo, "A new location estimation system for wireless networks based on linear discriminant functions and hidden markov models," *EURASIP Journal on Applied Signal Processing*, vol. 2006, pp. 159–159, 01 2006.

[30] D. Madigan, E. Elnahrawy, R. P. Martin, W. Ju, P. S. A. Krishnanm, and A. S. Krishnakumar, "Bayesian indoor positioning systems," in *IEEE International Conference on Computer Communications (INFOCOM'05)*, vol. 2, Miami, FL, March 2005, pp. 1217–1227 vol. 2.

[31] A. Goswami, L. E. Ortiz, and S. R. Das, "Wigem: A learning-based approach for indoor localization," in *International Conference on emerging Networking EXperiments and Technologies (CoNEXT'11)*, December 2011, pp. 3:1–3:12.

[32] B. Ferris, D. Fox, and N. Lawrence, "Wifi-slam using gaussian process latent variable models," in *International Joint Conference on Artifical Intelligence (IJCAI'07)*, Hyderabad, India, January 2007, pp. 2480–2485.

[33] C. Yang, Y. Song, and G. Gu, "Active user-side evil twin access point detection using statistical techniques," *IEEE Transactions on Information Forensics and Security*, vol. 7, no. 5, pp. 1638–1651, October 2012.

[34] T. Li, Y. Chen, R. Zhang, Y. Zhang, and T. Hedgpeth, "Secure crowdsourced indoor positioning systems," in *IEEE International Conference on Computer Communications (INFOCOM'18)*, Honolulu, HI, April 2018, pp. 1034–1042.

[35] H. Li, L. Sun, H. Zhu, X. Lu, and X. Cheng, "Achieving privacy preservation in wifi fingerprint-based localization," in *IEEE International Conference on Computer Communications (INFOCOM'14)*, Toronto, Canada, April 2014, pp. 2337–2345.

[36] Z. Yang and K. Javinen, "The death and rebirth of privacy-preserving wifi fingerprint localization with paillier encryption," in *IEEE International Conference on Computer Communications (INFOCOM'18)*, Honolulu, HI, April 2018, pp. 1223–1231.

[37] M. Azizyan, I. Constandache, and R. Roy Choudhury, "Surroundsense: Mobile phone localization via ambience fingerprinting," in *Annual International Conference on Mobile Computing and Networking (Mobicom'09)*, Beijing, China, September 2009, pp. 261–272.

[38] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka, "You are facing the mona lisa: Spot localization using phy layer information," in *International Conference on Mobile Systems, Applications, and Services (MobiSys'12)*, Low Wood Bay, Lake District, UK, June 2012, pp. 183–196.

[39] X. Wang, L. Gao, S. Mao, and S. Pandey, "Deepfi: Deep learning for indoor fingerprinting using channel state information," in *IEEE Wireless Communications and Networking Conference (WCNC'15)*, 2015, pp. 1666–1671.

[40] P. H. Calamai and J. J. Moré, "Projected gradient methods for linearly constrained problems," *Mathematical Programming*, vol. 39, no. 1, pp. 93–116, Sepember 1987.

[41] L. Armijo, "Minimization of functions having lipschitz continuous first partial derivatives," *Pacific Journal of Mathematics*, vol. 16, no. 1, pp. 1–4, November. 1966.

[42] M. Liu, J. Du, Q. Zhou, Z. Cao, and Y. Liu, "Eyeloc: Smartphone vision-enabled plug-n-play indoor localization in large shopping malls," *IEEE Internet of Things Journal*, vol. 8, no. 7, pp. 5585–5598, 2021.

[43] S. S. Saab and Z. S. Nakad, "A standalone rfid indoor positioning system using passive tags," *IEEE Transactions on Industrial Electronics*, vol. 58, no. 5, pp. 1961–1970, 2011.

[44] F. Höflinger, J. Hoppe, R. Zhang, A. Ens, L. Reindl, J. Wendeberg, and C. Schindelhauer, "Acoustic indoor-localization system for smart phones," in *IEEE 11th International Multi-Conference on Systems, Signals Devices (SSD'14)*, 2014, pp. 1–4.

[45] S. Zhu and X. Zhang, "Enabling high-precision visible light localization in today's buildings," in *MobiSys'17*, Niagara Falls, NY, 2017, p. 96–108.

[46] I. Arnold, Z. Christopher, F. Jan-Michael, and B. Horst, "From structure-from-motion point clouds to fast location recognition," in *2009 IEEE Conference on Computer Vision and Pattern Recognition*, 2009, pp. 2599–2606.

[47] C. Tommaso, G. Stuart, L. Nicholas A., V. Julien, S. Luigi Di, and T. Philip H. S., "On-the-fly adaptation of regression forests for online camera relocalisation," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'17)*, 2017, pp. 218–227.

[48] H. Taira, M. Okutomi, T. Sattler, M. Cimpoi, M. Pollefeys, J. Sivic, T. Pajdla, and A. Torii, "Inloc: Indoor visual localization with dense matching and view synthesis," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR'18)*, 2018.

[49] G. Lu and J. Song, "3d image-based indoor localization joint with wifi positioning," in *Proceedings of the 2018 ACM on International Conference on Multimedia Retrieval(ICMR'18)*, 2018, p. 465–472.

[50] L. Nam Tuan and J. Yeong Min, "Photography trilateration indoor localization with image sensor communication," *Sensors*, 2019.

[51] H. Kawaji, K. Hatada, T. Yamasaki, and K. Aizawa, "Image-based indoor positioning system: Fast image matching using omnidirectional panoramic images," 01 2010.

[52] W. Ke, M. Lin, and T. Xuezhi, "An improvement algorithm on ransac for image-based indoor localization," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC'16)*, 2016, pp. 842–845.

[53] S. Hamed, V. Shahrokh, and S. Shahram, "A weighted knn epipolar geometry-based approach for vision-based indoor localization using smartphone cameras," in *2014 IEEE 8th Sensor Array and Multichannel Signal Processing Workshop (SAM'14)*, 2014, pp. 37–40.

[54] K. Guan, L. Ma, T. Xuezhi, and G. Shizeng, "Vision-based indoor localization approach based on surf and landmark," in *2016 International Wireless Communications and Mobile Computing Conference (IWCMC'16)*, 2016, pp. 655–659.

[55] Z. Yuanqing, S. Guobin, L. Liqun, Z. Chunshui, L. Mo, and Z. Feng, "Travinavi: Self-deployable indoor navigation system," *IEEE/ACM Transactions on Networking*, vol. 25, no. 5, pp. 2655–2669, 2017.

[56] N. Qun, L. Mingkuan, H. Suining, G. Chengying, S.-H. G. Chan, and L. Xiaonan, "Resource-efficient and automated image-based indoor localization," vol. 15, no. 2, 2019.

[57] D. Jiang, N. Marius, X. Yu, and Y.-J. Antti, "Vinav: A vision-based indoor navigation system for smartphones," *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1461–1475, 2019.

[58] B. Atreyi and C. Heng Foh, "A practical path loss model for indoor wifi positioning enhancement," in *6th International Conference on Information, Communications Signal Processing*, 2007, pp. 1–5.

[59] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf:speeded up robust features," in *Computer Vision(ECCV'06*, 2006, pp. 404–417.

[60] Y. Li, R. H. Kambhamettu, Y. Hu, and R. Zhang, "Impos: An image-based indoor positioning system," in *2022 IEEE 19th Annual Consumer Communications Networking Conference (CCNC)*, 2022, pp. 144–150.

[61] J. Guangyao, L. Xiaoyi, and P. MyongSoon, "An indoor localization mechanism using active rfid tag," in *IEEE International Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC'06)*, vol. 1, 2006, pp. 4 pp.–.

[62] F. Seco and A. R. Jiménez, "Smartphone-based cooperative indoor localization with rfid technology," *Sensors*, vol. 18, no. 1, 2018.

[63] T. Q. Wang, Y. A. Sekercioglu, A. Neild, and J. Armstrong, "Position accuracy of time-of-arrival based ranging using visible light with application in indoor localization systems," *Journal of Lightwave Technology*, vol. 31, no. 20, pp. 3302–3308, 2013.

[64] X. Guo, S. Shao, N. Ansari, and A. Khreishah, "Indoor localization using visible light via fusion of multiple classifiers," *IEEE Photonics Journal*, vol. 9, no. 6, pp. 1–16, 2017.

[65] L. Kanaris, A. Kokkinis, A. Liotta, and S. Stavrou, "Fusing bluetooth beacon data with wi-fi radiomaps for improved indoor localization," *Sensors*, vol. 17, 2017.

[66] W. Huang, Y. Xiong, X.-Y. Li, H. Lin, X. Mao, P. Yang, Y. Liu, and X. Wang, "Swadloon: Direction finding and indoor localization using acoustic signal by shaking smartphones," *IEEE Transactions on Mobile Computing*, vol. 14, no. 10, pp. 2145–2157, 2015.

[67] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Push the limit of wifi based localization for smartphones," in *Proceedings of the 18th Annual International Conference on Mobile Computing and Networking*, ser. Mobicom '12. New York, NY, USA: Association for Computing Machinery, 2012, p. 305–316.

[68] "Dead reckoning," 2021. [Online]. Available: https://en.wikipedia.org/wiki/Dead_reckoning

[69] F. Lei, P. Antsaklis, L. Montestruque, M. McMickell, M. Lemmon, S. Yashan, F. Hui, I. Koutroulis, M. Haenggi, X. Min, and X. Xiaojuan, "Design of a wireless assisted pedestrian dead reckoning system - the navmote experience," *IEEE Transactions on Instrumentation and Measurement*, vol. 54, no. 6, pp. 2342–2358, 2005.

[70] S. House, S. Connell, I. Milligan, D. Austin, T. L. Hayes, and P. Chiang, "Indoor localization using pedestrian dead reckoning updated with rfid-based fiducials," in *2011 Annual International Conference of the IEEE Engineering in Medicine and Biology Society*, 2011, pp. 7598–7601.

[71] V. Radu and M. K. Marina, "Himloc: Indoor smartphone localization via activity aware pedestrian dead reckoning with selective crowdsourced wifi fingerprinting," in *International Conference on Indoor Positioning and Indoor Navigation*, 2013, pp. 1–10.

[72] P. Alexey, G. Andrey, and S. Alexey, "Indoor positioning using wi-fi fingerprinting pedestrian dead reckoning and aided ins," in *International Symposium on Inertial Sensors and Systems (ISISS'14)*, 2014, pp. 1–2.

[73] G. Lu, Y. Yan, L. Ren, P. Saponaro, N. Sebe, and C. Kambhamettu, "Where am i in the dark:exploring active transfer learning on the use of indoor localization based on thermal imaging," *Neurocomputing*, vol. 173, no. P1, p. 83–92, jan 2016.

[74] L. Lin Shen and W. Wong Shung Hui, "Improved pedestrian dead-reckoning-based indoor positioning by rssi-based heading correction," *IEEE Sensors Journal*, vol. 16, no. 21, pp. 7762–7773, 2016.

[75] N.-H. Ho, P. H. Truong, and G.-M. Jeong, "Step-detection and adaptive step-length estimation for pedestrian dead-reckoning at various walking speeds using a smartphone," *Sensors*, vol. 16, no. 9, 2016.

[76] R. Ban, K. Kaji, K. Hiroi, and N. Kawaguchi, "Indoor positioning method integrating pedestrian dead reckoning with magnetic field and wifi fingerprints," in *Eighth International Conference on Mobile Computing and Ubiquitous Networking (ICMU)'15*, 2015, pp. 167–172.

[77] L. You, Z. Yuan, L. Haiyu, Z. Qifan, N. Xiaoji, and E.-S. Naser, "A hybrid wifi/magnetic matching/pdr approach for indoor navigation with smartphone sensors," *IEEE Communications Letters*, vol. 20, 2016.

[78] J. Kuang, X. Niu, P. Zhang, and X. Chen, "Indoor positioning based on pedestrian dead reckoning and magnetic field matching for smartphones," *Sensors*, vol. 18, no. 12, 2018.

[79] J. Yang and Y. Chen, "A theoretical analysis of wireless localization using rf-based fingerprint matching," in *2008 IEEE International Symposium on Parallel and Distributed Processing*, 2008, pp. 1–6.

[80] M. Hoang, J. Schmalenstroeer, C. Drueke, D. Tran Vu, and R. Haeb-Umbach, "A hidden markov model for indoor user tracking based on wifi fingerprinting and step detection," in *21st European Signal Processing Conference (EUSIPCO 2013)*, 2013, pp. 1–5.

[81] A. W. S. Au, C. Feng, S. Valaee, S. Reyes, S. Sorour, S. N. Markowitz, D. Gold, K. Gordon, and M. Eizenman, "Indoor tracking and navigation using received signal strength and compressive sensing on a mobile device," *IEEE Transactions on Mobile Computing*, vol. 12, no. 10, pp. 2050–2062, Oct 2013.

[82] H. Liu, N. Xia, D. Guo, and P. Qing, "Csi-based indoor tracking with positioning-assisted," in *Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)'18*, 2018, pp. 1–8.

[83] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, "E-eyes: Device-free location-oriented activity identification using fine-grained wifi signatures," in *Proceedings of the 20th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '14. New York, NY, USA: Association for Computing Machinery, 2014, p. 617–628.

[84] K. Kleisouris, Y. Chen, J. Yang, and R. P. Martin, "The impact of using multiple antennas on wireless localization," in *2008 5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2008, pp. 55–63.

[85] X. Zheng, H. Liu, J. Yang, Y. Chen, R. P. Martin, and X. Li, "A study of localization accuracy using multiple frequencies and powers," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 8, pp. 1955–1965, 2014.

[86] Y. Shu, Y. Huang, J. Zhang, P. Coué, P. Cheng, J. Chen, and K. G. Shin, "Gradient-based fingerprinting for indoor localization and tracking," *IEEE Transactions on Industrial Electronics*, vol. 63, no. 4, pp. 2424–2433, 2016.

[87] H. Liu, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye, "Accurate wifi based localization for smartphones using peer assistance," *IEEE Transactions on Mobile Computing*, vol. 13, no. 10, pp. 2199–2214, 2014.

[88] J. Yang and Y. Chen, "Indoor localization using improved rss-based lateration methods," in *GLOBECOM 2009 - 2009 IEEE Global Telecommunications Conference*, 2009, pp. 1–6.

[89] N. Hernández, M. Ocaña, J. M. Alonso, and E. Kim, "Wifi-based indoor localization and tracking of a moving device," in *2014 Ubiquitous Positioning Indoor Navigation and Location Based Service (UPINLBS)*, 2014, pp. 281–289.

[90] G. Chandrasekaran, M. A. Ergin, J. Yang, S. Liu, Y. Chen, M. Gruteser, and R. P. Martin, "Empirical evaluation of the limits on localization using signal strength," in *2009 6th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks*, 2009, pp. 1–9.

[91] I. Mai, T. Marwan, and E. Mustafa, "Cnn based indoor localization using rss time-series," in *IEEE Symposium on Computers and Communications (ISCC'18)*, 2018.

[92] P. Tiwary, A. Pandey, S. Kumar, and M. Youssef, "Novel differential $r$-vectors for localization in iot networks," *IEEE Sensors Letters*, vol. 5, no. 6, pp. 1–4, 2021.

[93] J. Yang, Y. Chen, V. B. Lawrence, and V. Swaminathan, "Robust wireless localization to attacks on access points," in *2009 IEEE Sarnoff Symposium*, 2009, pp. 1–5.

[94] X. Li, Y. Chen, J. Yang, and X. Zheng, "Designing localization algorithms robust to signal strength attacks," in *2011 Proceedings IEEE INFOCOM*, 2011, pp. 341–345.

[95] Y. Li, Y. Hu, R. Zhang, Y. Zhang, and T. Hedgpeth, "Secure indoor positioning against signal strength attacks via optimized multi-voting," in *2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS)*, 2019, pp. 1–10.

[96] L. Li, G. Shen, C. Zhao, T. Moscibroda, J.-H. Lin, and F. Zhao, "Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service," in *MobiCom'14*, Maui, Hawaii, USA, 2014, pp. 459–470.

# Appendix A

# PERMISSION

Internal or personal use of IEEE/ACM copyrighted materials involved in this dissertation is permitted.

Chapter 2 is based on paper:

©IEEE/ACM Reprint, with permission, from Yunzhi Li, Yidan Hu, Rui Zhang, Yanchao Zhang and Terri Hedgpeth "Secure Indoor Positioning Against Signal Strength Attacks Via Optimized Multi-voting." In 2019 IEEE/ACM 27th International Symposium on Quality of Service (IWQoS), Phoenix, AZ, pp. 1-10

Chapter 3 is based on paper:

©IEEE. Reprint, with permission, from Yunzhi Li, Rajeswari Hita Kambhamettu, Yidan Hu and Rui Zhang "ImPos: An Image-Based Indoor Positioning System." In 2022 IEEE 19th Annual Consumer Communications  Networking Conference (CCNC), 2022, Las Vegas, NV, pp. 144-150