Crowdsourcing based Room Localization on Smartphones

by

Yifei Jiang

B.S., Beijing University of Technology, 2002

M.S., University of Southern Denmark, 2005

A thesis submitted to the

Faculty of the Graduate School of the

University of Colorado in partial fulfillment

of the requirements for the degree of

Doctor of Philosophy

Department of Computer Science

2013

This thesis entitled:
Crowdsourcing based Room Localization on Smartphones
written by Yifei Jiang
has been approved for the Department of Computer Science

_____

Qin Lv

_____

Prof. Richard Han, Prof. Shivakant Mishra

_____

Prof. Li Shang, Prof. Michael Hannigan

Date _____

The final copy of this thesis has been examined by the signatories, and we find that both the content and the form meet acceptable presentation standards of scholarly work in the above mentioned discipline.

Jiang, Yifei (Ph.D., Computer Science)

Crowdsourcing based Room Localization on Smartphones

Thesis directed by Prof. Qin Lv

People spend approximately 90% of their time indoors, and human indoor activities are strongly correlated with the rooms they are in. Room localization, which identifies the room a person or smartphone is in, provides a powerful tool for characterizing human indoor activities and helping address challenges in public health, productivity, building management, etc.

Designing a room localization system that is practically useful in real-world environments is challenging. First, due to the complex multi-path propagation problem, Wi-Fi signals obtained by smartphones are dynamic and noisy. Such noise obscures the unique relationship between Wi-Fi signals and individual rooms. Second, existing room localization methods require labor-intensive manual annotation of individual rooms. The process is time-consuming and expensive, which is a key limitation of existing room localization applications. Third, knowledge of indoor floorplans is often required by room localization applications. However, indoor floorplans are either unavailable or obtaining them requires slow, tedious, and error-prone manual labor. In addition, the overhead of room localization, e.g., energy consumption, imposed on personal smartphones must be low.

To tackle those challenges, this thesis proposed a set of techniques: (1) an accurate temporal n-gram augmented Bayesian room positioning method that leverages the ordered sequence information of access points and users' daily motion pattern among rooms; (2) an automatic room fingerprinting approach that identifies in-room occupancy "hotspot(s)" using density of Wi-Fi signals, and then learns the inter-zone correlation – thereby distinguishing different rooms; (3) an automatic floorplan construction method that determines the geometries of individual rooms, as well as room adjacency information, and then constructs an indoor floorplan through hallway layout learning and force directed dilation; and (4) an energy-efficient trip detection framework

that consists of two modes: the deliberation mode learns cell-id patterns using GPS/Wi-Fi based localization methods, and the intuition mode only uses cell-ids and learned patterns for trip detection.

Acknowledgements

Completing this dissertation is a very enjoyable experience. This was due in large part to some wonderful people I worked with.

My advisor, Professor Qin Lv, provided constant advice about possible research avenues and was a constant guide during this process. I am very grateful for the opportunity to work with her. It has definitely sharpened my research ability and helped me grow as an individual and a professional.

I am also very grateful to my thesis committee members, Li Shang, Shivakant Mishra, Michael Hannigan, and Richard Han, for guiding my research and providing invaluable advice that helped me to improve my dissertation. I also would like to thank Professor Robert Dick who provided invaluable guidances for my research. I thank Ricardo Piedrahita, Kun Li, and Yun Xiang for collaborating on the mobile air quality sensing project.

During my internships at Nokia, I have collaborated with many great researchers there for the work of trip monitoring. I would particularly like to thank Du Li, Guang Yang, and Zhigang Liu. They provided invaluable advice, guidance, and friendship during my time at Nokia. In addition, I would like to thank Larry Shi, Jun Yang ,Yingen Xiong, and Feng Yang for their help and friendship.

Finally, I am especially grateful to Xin and my parents for being extremely patient with me throughout my Ph.D. study. Their love and support were instrumental in me finishing this thesis.

Contents

Chapter

Tables

Table

Figures

Figure

Chapter 1

Introduction

1.1 Background and Motivation

People spend approximately 90% of their time in buildings [51]. Understanding the indoor activities and environments of occupants will offer valuable insights to a wide range of pressing challenges, such as public health, employee productivity, building security and energy management. For instance, the economic consequences of poor indoor environments in U.S. commercial buildings were estimated at $40–160 billion per year in lost wages and productivity, administrative expenses, and health care costs [15].

A key observation of this thesis is that indoor environments are well structured – Buildings are organized into rooms with distinct functionalities, supporting and also defining the activities, interests, and social roles of their occupants. For instance, an office building is typically organized into offices, conference rooms, and lounges; a shopping center consists of a large number of shops selling different products; and a residential home is comprised of a living room, bedrooms, a kitchen, etc. Therefore, room localization, i.e., identifying which room an occupant is in, will help characterize the indoor activities of occupants, thereby providing valuable insights to address health, social, and economic related issues of indoor environments.

Recently, new applications and services that leverage indoor room information have started to emerge [16, 18, 33, 43]. Room localization, as a core functionality of those applications, consists of the following three techniques:

- **Room fingerprinting**, which measures and assigns unique RFS (radio-frequency signals, e.g., Wi-Fi) to rooms, is the de facto technique for room localization. The time-consuming manual annotation process is a key limitation of existing room fingerprinting techniques. They either rely on an expert surveyor [18, 33], or leverage a collaborative but less accurate effort [37] among volunteers [16, 43], to manually collect Wi-Fi fingerprints for each room and associate them with pre-defined room IDs;

- **Room positioning** is an operating phase for room localization. It matches each new or unknown radio-frequency signals (e.g., Wi-Fi) to the most similar fingerprints in the database in order to estimate indoor room location. Signal stability is the primary challenge to Wi-Fi based indoor location techniques, which is due to the fact that Wi-Fi signals experience significant run-time variation, due to multi-path fading, ambient environment change, as well as device heterogeneity [6, 16, 18, 36, 48]. These variations adversely affect the robustness of Wi-Fi fingerprints and Wi-Fi based localization accuracy; and

- **Room representation** in indoor maps, i.e., indoor floorplans, are often required by indoor localization techniques to answer the following question "*so... I'm here but where is here?*" [43]. Simply put, the floorplan is typically in the form of a building blueprint, defining the structure and functionality of a specific indoor environment. Given an indoor floorplan, the personal and social functions supported by the indoor environment are then defined. Acquiring indoor floorplan information is challenging – (1) many buildings do not have floorplans in easily-interpretable digital form; (2) even if available, building managers are often reluctant to share such information with the general public; furthermore, (3) buildings' internal structures and the corresponding functionalities often evolve over time, making the original floorplans outdated. Although it is possible to manually construct an indoor floorplan or correct an existing one, such process is slow, tedious, error prone, and difficult to scale.

Furthermore, most of the room localization services require a fundamental function, **trip**

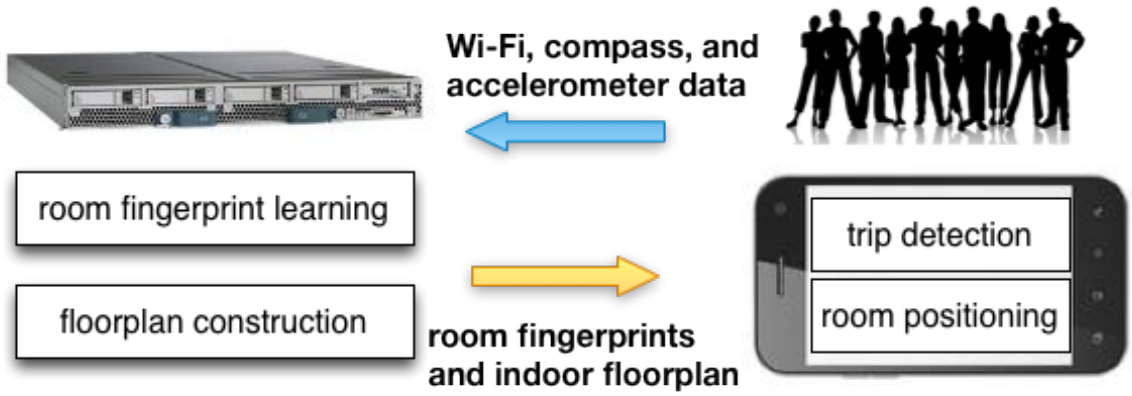


Figure 1.1: System Overview: Crowdsourcing Based Indoor Room Localization.

detection, to recognize significant places where a person stays or visits frequently and the trips between those places. This information is useful for an application to determine when to start and stop its localization service for energy efficiency and better user experience. Trip detection is a challenging task for the following three reasons: First, trip detection must be automated. We cannot leave it to the user as it would be annoying and error-prone. Second, it must be energy-efficient. Since the user may start or end a trip anytime and anywhere, the function will be running backstage all the time. Third, the detection must be accurate and timely for the services to be useful to the user.

1.2 System Overview

This thesis tackles the existing problems of room localization leveraging crowdsourcing mechanism. As illustated in Figure 1.1, our room localization system is composed of the following components:

Wi-Fi and motion data crowdsourcing. The daily activities of building occupants offer rich and up-to-date information of the personal and social functions of an indoor environment. Leveraging the built-in motion sensors of the mobile phones carried by occupants, "crowd sourcing" offers a potentially highly scalable solution for automatic room fingerprinting

and floorplan construction. The primary challenge is how to accurately extract stable and representative information from diverse and random occupant motion patterns. In addition, as a collaborative voluntary effort, the overhead of data collection, e.g., energy consumption, imposed on personal mobile phones must be low.

Room fingerprint learning. Designing an indoor room localization system that learns room fingerprints without manual annotation is challenging. First, due to signal reflection, refraction, diffraction, and absorption, indoor Wi-Fi signals are noisy. Such noise obscures the unique relationship between Wi-Fi fingerprints and individual rooms. Second, occupant-specific indoor activities directly affect the room fingerprinting process. For instance, in-room occupancy "hotspot(s)" are unevenly distributed both spatially and temporally. With such distribution, clustering algorithms may learn multiple fingerprints for a room. In response to these challenges, we have developed the following algorithms:

- A **zone-based clustering** algorithm that accurately and automatically identifies in-room occupancy hotspot(s) using Wi-Fi signature;
- A **motion-based clustering** algorithm to identify inter-zone correlation, thereby distinguishing different rooms; and
- An **energy-efficient motion detection** algorithm to minimize the noise of Wi-Fi signature.

Floorplan construction. To accurately construct an indoor floorplan, a set of indoor features are required, including (1) unique identification of individual rooms, (2) estimated room geometric information, e.g., length and width, and (3) geometric relationship among rooms. Accurate determination of these indoor floorplan features faces the following challenges:

- **Heterogeneous indoor environments:** An indoor environment consists of rooms with diverse sizes. Room connections through hallways also vary significantly. Such heterogeneous indoor environments make accurate floorplan construction challeng-

ing.

- **Noisy Wi-Fi fingerprints:** Indoor Wi-Fi infrastructure has been used by many indoor localization techniques. However, due to the complex multipath propagation problem, Wi-Fi fingerprints obtained by mobile phones are dynamic and noisy. The problem is particularly challenging when using Wi-Fi fingerprints to determine detailed floorplan features.

Leveraging Wi-Fi fingerprints and user motion information collected via mobile crowd sourcing, the automatic indoor floorplan construction system extracts indoor floorplan features, including room identity, geometry, and inter-room geometrical relationship, and then construct the indoor floorplan automatically. To tackle the aforementioned challenges, the proposed system uses the following main components.

- A **room adjacency graph construction** algorithm that identifies the adjacency of rooms and constructs a room adjacency graph that is robust to the spatial bias of room fingerprints and Wi-Fi noise;
- A **hallway layout learning** algorithm that determines the room arrangement along each hallway, e.g., room sizes and orders, using crowd-based motion sensing on smartphones; and
- A **force directed dilation** algorithm that adjusts the individual room structures globally to improve floorplan accuracy.

Room positioning. The challenge of room positioning is that Wi-Fi signals can vary due to the following factors.

- **Spacial variation**. Wi-Fi signals may be sampled at different positions within the same room, resulting in different RSS readings, since the power density of an electromagnetic wave is proportional to the inverse of the square of the distance from a point source in open space (inverse-square law).

- **Temporal variation**. Wi-Fi RSS is subject to reflection, refraction, diffraction, and absorption by structures and even human bodies. Therefore, Wi-Fi signals sampled at different time would vary due to temporal changes of the environment, such as communication channel dynamics, people moving around, etc.
- **Device heterogeneity**. Nowadays, people carry a wide variety of mobile devices. Wi-Fi signals sampled by different types of mobile devices at the same location and same time may be different due to device variations in antenna design, internal structure, and power source.
- **Room wall type**. The walls of a room can affect Wi-Fi signal propagation significantly, and different types of walls (e.g., load-bearing wall vs. drywall) would affect Wi-Fi signals differently.
- **AP deployment**. The deployment of Wi-Fi infrastructure may be centralized (e.g., multiple APs by a single organization on campus), decentralized (e.g., multiple APs by individuals in an apartment building), or mixed. As a result, the APs may have different uniformity with regard to models, spacial distribution, operation time, etc. All these would affect the Wi-Fi signals observed by mobile devices.

To deal with above noises, we proposed A temporal n-gram augmented Bayesian room positioning method that is accurate and requires few Wi-Fi fingerprints.

Trip detection. We propose a framework for automatic, accurate, timely, and energy-efficient trip detection on mobile devices. We focus on the detection of when a trip starts and ends with regard to **significant places** where the user spends considerable time, e.g., home and workplace. We synergistically combine the merits of GPS/WiFi based methods, which are more accurate yet energy-consuming, and cell id based methods, which are energy-efficient yet less accurate. Compared with previous GPS/WiFi based works, e.g., [25, 26, 27], our approach exhibits a clear trend of decreasing energy consumption as a user

visits the same places and repeats the same trips. Our framework is inspired from human decision theories in psychology [45, 23] with two distinct modes:

- In the **deliberation mode**, we learn cell id patterns from trace data collected by GPS, WiFi, and cell id sensors, and associate those patterns with places and trips learned using existing GPS/WiFi based localization methods. The energy expense will be comparable to previous approaches.
- In the **intuition mode**, we use collected cell id data and the learned cell id patterns for determining whether a user is at a place, entering a place, or departing a place. When it is a repeated place or trip, the energy expense will be minimum because only cell id data is collected.

1.3 Thesis Contributions

This thesis proposes several techniques to address the problem of crowdsourcing based room localization on smartphones. The research contributions of this thesis are as follows:

Wi-Fi fingerprint characterization and novel room positioning approach. We have designed and conducted comprehensive experiments to characterize how each of these factors affects Wi-Fi fingerprints and especially their capabilities of identifying and differentiating rooms. Based on the observed characters of Wi-Fi fingerprint and user's motion pattern, we proposed A temporal n-gram augmented Bayesian room positioning method that is accurate and requires few Wi-Fi fingerprints. The approach have been implemented on Android phone and deployed for user study with 17 participants over 12 weeks. The experiments results show that our room positioning can achieve the accuracy of 90%.

Automatic room fingerprinting approach. To accurately identify rooms without extensive manual annotation, we proposed and developed a number of novel techniques: 1) a zone-based clustering algorithm that accurately identifies in-room occupancy hotspot(s) using

Wi-Fi signatures; 2) a motion-based clustering algorithm to identify inter-zone correlation, thereby distinguishing different rooms; and 3) an energy-efficient motion detection algorithm to minimize the noise of Wi-Fi fingerprints. The system has been implemented and deployed for a 10-month study with 21 participants. The evaluation results demonstrate that our automated clustering algorithm generates clusters that are high representative of individual rooms and achieves high accuracy (95%) for room localization. The accuracy is comparable to existing techniques that require labor-intensive manual annotation.

Automatic indoor floorplan construction. Leveraging Wi-Fi fingerprints and user motion information, the automatic indoor floorplan construction method automatically constructs floorplan via three key steps: 1) room adjacency graph construction to determine which rooms are adjacent; 2) hallway layout learning to estimate room sizes and order rooms along each hallway, and 3) force directed dilation to adjust room sizes and optimize the overall floorplan accuracy. Deployment study in three buildings with 103 rooms demonstrates high floorplan accuracy. The system has been implemented as a mobile middleware, which allows emerging mobile applications to generate, leverage, and share indoor floorplans.

Energy-efficient trip detection approach. Learning from the human decision making process, we propose an energy-efficient trip detection framework that consists of two modes: The **deliberation** mode learns cell-id patterns using GPS/Wi-Fi based localization methods; the **intuition** mode only uses cell-ids and learned patterns for trip detection; transition between the two modes is controlled by parameters that are also learned. We evaluated our framework using real-life traces of six people over five months. Our experiments demonstrate that its energy consumption decreases rapidly as users' activities manifest regularity over time.

Chapter 2

Room Positioning

In this chapter, we designed and conducted in-depth experiments to arrive at a comprehensive characterization of different Wi-Fi fingerprints with regard to changes in aforementioned factors, showing different levels of robustness of Wi-Fi fingerprints under a wide variety of changes. We proposed a new room localization method using the ordered sequence of access point, since it maintains good same-room concentration and cross-room separation under different variations. We have also conducted a comparative evaluation of our new method, using data collected in controlled experiments and real-world user study. Our evaluation results demonstrate that our method achieves both high accuracy and low training cost under a wide variety of settings.

2.1 Related Work

In this section, we survey works that are most related to ours, focusing on Wi-Fi fingerprints and Wi-Fi based indoor localization techniques. We also discuss indoor localization techniques that leverage other types of signals.

Wi-Fi signals have been widely used for indoor localization. The basic approach is to construct Wi-Fi fingerprints from Wi-Fi signals and match new/unknown Wi-Fi fingerprints to reference fingerprints at known positions based on proximity or similarity. A training phase is usually required that maps reference fingerprints to specific indoor locations, which can be labor intensive. Different techniques have been proposed to construct Wi-Fi fingerprints using the occurrence, received signal strength(RSS), or subsequence of access points (APs), and most existing

systems use triangulation, k-nearest-neighbor (KNN), or probabilistic techniques to estimate location [6, 17, 18, 43, 16, 21].

Wi-Fi signals are subject to multi-path fading and ambient environment change in addition to device heterogeneity [18, 36, 16]. Such variations can reduce localization accuracy. Calibration of signal strength as a function of distance helps reduce the signal variation. Techniques using kernel density estimation, Pearson transformation, and radio propagation, have been proposed to estimate real-world signal strength at different distance [6, 18, 36, 48, 16]. The unsupervised learning method proposed by Tsui et al. can improve positional accuracy by up to 46% within a 100s learning period [48]. The work by Haeberlen et al. showed 20% increase in localization accuracy with calibration [18].

However, calibration can be labor intensive and does not scalable well. Similar localization accuracy may be maintained with slight reduction in sampling locations and sampling time [33, 5, 8]. Gwon and Jain proposed TIX (Triangular Interpolation and eXtrapolation), a calibration-free mechanism that used the three APs with the highest RSSIs to determine the centroid of the triangle [17]. A zero-configuration indoor localization method was proposed by Lim et al., which used online calibration and truncated singular value decomposition (SVD) to characterize the relationship between RSSI and geographical distance to anchors [36].

To further improve the localization accuracy of Wi-Fi signals, researchers have investigated techniques that combine Wi-Fi signals with other signals [9, 46]. Although GPS data may be obtained opportunistically indoors and used in conjunction with Wi-Fi signals (e.g., EZ model in [9]), the overall localization accuracy is limited by the ability to access GPS indoors. The work by Matic et al. combined Wi-Fi signals with FM signals, which are widely available and energy efficient. Using Gaussian process regression for classification, the combined method achieved localization accuracy that was comparable to the Wi-Fi only approach [40]. Tarzia et al. proposed Batphone, a system that uses a linear combination of Acoustic Background Spectrum (ABS) and Wi-Fi fingerprints for indoor room localization [46]. In SurroundSense [4], ambience fingerprints (Wi-Fi, sound, acceleration, color, and light) indexed by 2-hour time windows were used for mobile phone

localization. Users' room-level mobility has also been used to distinguish rooms with similar Wi-Fi signals [35, 21]. This is based on the observation that users' room locations are correlated with time and users' moving patterns across rooms are relatively stable. Augmenting Wi-Fi signals with other types of information can help improve indoor localization accuracy, especially when Wi-Fi is not available. At the same time, more energy is needed to capture sound, picture, and other sensor data. Also, depending on how a user carries his/her mobile phone (e.g., by hand vs. in pocket), the fingerprints and localization accuracy may be affected.

There are also indoor localization techniques that do not use Wi-Fi signals, and instead use FM, GSM, CDMA, or magnetic signals [32, 52, 50, 10]. Krumm et al. proposed a FM localization system based on the ranked signal strength from multiple FM stations [32]. Although FM radio towers provide wide coverage both indoors and outdoors and the ranked signal strength reduces measurement variation among devices, this method can only provide coarse localization to a suburb area and is therefore insufficient for indoor room localization. Using GSM signals, which have less temporal variation than that of Wi-Fi signals, similar localization accuracy may be achieved [52]. However, the accuracy depends on the density of the cellular network, and channel-to-cell allocation requires extensive configuration thus less flexibility. Since CDMA signal delay remains stable over time, it was used in the CILoS system and achieved over 87% accuracy, which was slightly higher than that of GSM-based approach, but lower than the 100% accuracy of Wi-Fi based approach [50]. Geo-magnetism was also used and achieved relatively high precision on position detection [10]. However, this approach requires a magnetic fingerprint map obtained beforehand, and any surrounding magnetic can affect the results. Moreover, the magnetic measurement tool is more expensive and may not be applicable for all users.

2.2 Methodology

Our high-level goal is to understand how variations in Wi-Fi signals affect the performance of Wi-Fi based room localization. Although it is possible to directly observe room localization performance under diverse Wi-Fi settings, we would like to seek answers to a deeper question –

Under different types of variations, what Wi-Fi features remain stable within the same room and have high differentiating power across rooms?

Answering this question would then allow us to design more effective Wi-Fi based room localization methods. Therefore, we conduct our study for characterization of Wi-Fi fingerprints. This section describes our methodology for that, as well as the data sets we used.

2.2.1 Characterization of Wi-Fi Fingerprint Variation

Each Wi-Fi sample (also referred to as Wi-Fi measurement) obtained by a mobile device consists of a list of APs along with the received signal strength value from each AP. Depending on how Wi-Fi fingerprints are constructed, two different types of information may be used: occurrence (OCC) and received signal strength (RSS).

• **Occurrence (**OCC**)**.

This refers to whether a specific AP occurs in a Wi-Fi sample or not. For a given mobile device at a place, it continuously scans all available APs with a certain frequency, for a total of N times. Let S be the set of all available APs. Consider a k**-combination** of APs $S_k \subseteq S$, if S_k occurred m times among the N samples, we define the **occurrence rate** of S_k to be m/N, i.e., how often do we see the APs in S_k occurring together in our Wi-Fi samples.

Different k values may be considered for k-combination of APs. When $k = 1$, we consider the occurrence rate of single APs. If an AP is more likely to occur in one room than another, we can then use occurrence rate of the AP to distinguish rooms. This is demonstrated in Figure 2.1, where the AP has a distribution of higher occurrence rates in Room A than that of Room B. Similarly, we can consider the concurrent occurrence of multiple APs (i.e., $k > 1$). The assumption is that the APs may have different correlations in different rooms, resulting in different distributions of occurrence rate. For example, when $k = 2$, for access points AP_i and AP_j, the probability of seeing both of them in a room $P(AP_i \cap AP_j) = P(AP_j|AP_i)P(AP_i) = P(AP_i|AP_j)P(AP_j)$. If this probability is relatively stable within the same room yet vary for different rooms, it could help our room localization problem.

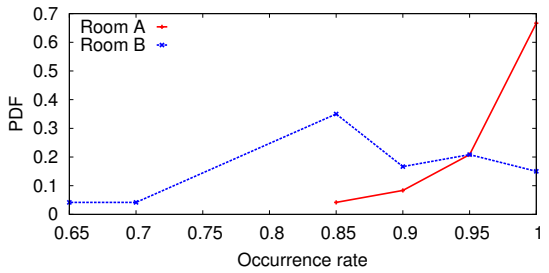


Figure 2.1: Occurrence rate in two adjacent rooms.

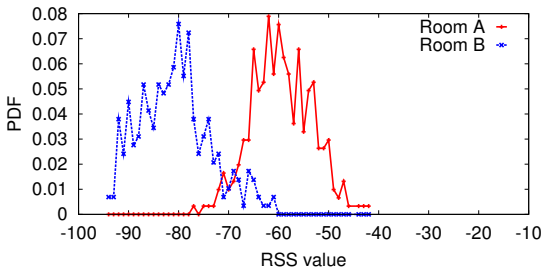


Figure 2.2: RSS distributions in two adjacent rooms.

Furthermore, instead of considering the k-combination of APs as a set, we could explore the **ordered k-combination** of APs. That is, given a Wi-Fi sample, we first order the set of APs by their corresponding RSS values, then generate k-combination using adjacent k APs. For example, consider a sequence of 5 APs $\{AP_3, AP_2, AP_5, AP_1, AP_4\}$ sorted in descending RSS values, then $\{AP_2, AP_5\}$ is an ordered 2-combination and $\{AP_5, AP_1, AP_4\}$ is an ordered 3-combination. Note that for the same number of APs, the number of ordered k-combination of APs is much fewer than that of regular k-combination of APs and can significantly reduce the computation overhead.

• **Received Signal Strength (RSS)**.

AP RSS is a commonly used feature in room localization. The RSS of each AP can be measured directly. Its value is relative stable within a room yet changes significantly when crossing walls. As shown in Figure 2.2, distribution of RSS value can be different in different rooms, even in adjacent rooms. RSS is thus a good indicator for room localization. However, there is still much variation of RSS value in the same room, and that variation becomes more significant when we consider ambient environment changes or device heterogeneity.

We consider not only the RSS of single APs, but also the difference between two APs' RSS values. The intuition is that APs' RSS values are usually correlated and change in similar fashion. Therefore, the difference between two APs' RSS values tends to be more stable than their raw RSS values. When choosing the pairs of APs, we consider both k-combination of APs (i.e., any two APs) and ordered k-combination of APs (i.e., adjacent APs in the sorted sequence based on descending RSS values).

2.2.2 Data Sets

In this work, we have collected two different data sets. The first data set has been collected in controlled experiments with wide variations and is used for characterization. The second data set has been collected in a real-world user study and is used for evaluation.

- **Controlled experiments**. We have carefully designed and conducted a set of controlled experiments to characterize variations of Wi-Fi fingerprints. Seven different phones, as listed in Table 2.1, were used in order to capture device heterogeneity. The experiments were conducted in three buildings with different AP deployment types, as listed in Table 2.2. We collected Wi-Fi fingerprints for a set of adjacent rooms in different buildings, as listed in Table 2.3. In addition, different spatial and temporal variations, as well as different room wall types are considered in the experiments.

- **Real-world user study**. We have also collected data via real-world deployment and user study. Data were collected from 6 users over 5 months, which are used to evaluate different room localization methods in people's real life. We developed a Wi-Fi sampling program on different smartphones, which can sample Wi-Fi fingerprints every 5 seconds. In addition, the program provided users an interface to log their room traces. This data set includes 2850 room visits and 131 unique rooms. Among those rooms, 29 rooms are visited by multiple users and 16 rooms are visited once. Note that, users did not record all their room visits during the 5 months.

Table 2.1: Different Phones Used for Data Collection

Phone	Manufacturer	Release Date
HTC Sapphire	HTC	2009-02
HTC Hero	HTC	2009-07
DROID	Motorola	2009-10
N900	Nokia	2009-11
Nexus One	Google	2010-01
Galaxy S2	Samsung	2011-02
HTC Sensation	HTC	2011-04

Table 2.2: Different Places for Data Collection

Building Type	AP Deployment
Educational	All APs are deployed and managed by university.
Commercial	Half of the APs are deployed and managed by building manager. Other APs are set up by occupants.
Residential	No centralized AP deployment or management. All APs are set up by occupants.

Table 2.3: Controlled Experiments

Building Type	# of Rooms	# of Experiments Per Room
Educational	18 adjacent rooms on three floors	10 times in 1 month
Commercial	12 adjacent rooms on three floors	10 times in 1 month
Residential	8 adjacent rooms on three floors	10 times in 1 month

2.3 Characterization of Wi-Fi Fingerprint Variation

In this section, we characterize how Wi-Fi fingerprints change under variations of each factor, including space, time, device, as well as room wall type and AP deployment type. Targeting on indoor room localization, we are particularly interested in the in-room stability (good same-room concentration) of Wi-Fi fingerprints and their capabilities of differentiating rooms (good cross-room separation). Table 2.3 summarizes the characteristics of different Wi-Fi fingerprints in terms of same-room concentration (con.) and cross-room separation (sep.) under different types of variations.

Fingerprint Type	Space		Time		Device		Wall		AP	
	con.	sep.	con.	sep.	con.	sep.	con.	sep.	con.	sep.
OCC 1-AP	X	√	√	X	—	√	n/a	—	n/a	—
OCC 2-AP	—	—	√	X	√	√	n/a	X	n/a	—
OCC 3-AP	√	X	√	X	√	X	n/a	X	n/a	—
OCC Ordered 2-AP	√	—	√	X	√	X	n/a	X	n/a	X
OCC Ordered 3-AP	√	X	√	X	√	X	n/a	X	n/a	X
RSS 1-AP	X	X	X	X	X	X	n/a	√	n/a	√
RSS 2-AP	√	√	√	√	X	√	n/a	√	n/a	√
RSS Ordered 2-AP	√	√	√	√	—	√	n/a	√	n/a	√

Table 2.4: Summary of Characterization Results: Different types of fingerprints perform differently under different types of variations. "con." stands for same-room concentration and "sep." stands for cross-room separation. "√", "—", and "X" mean good, partial, and bad, respectively.

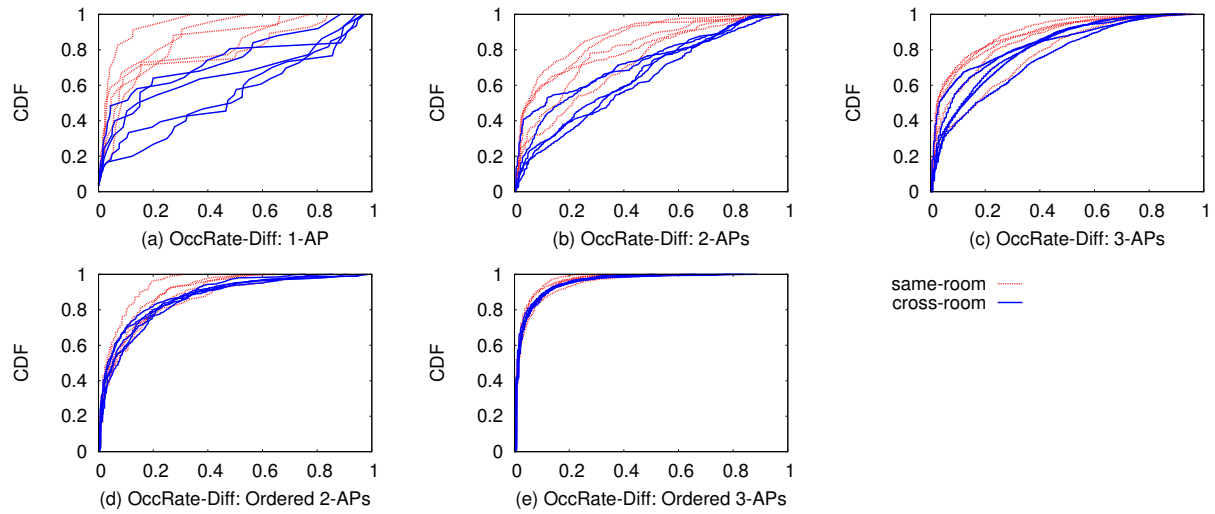


Figure 2.3: Spacial variation: CDF of OccRate-Diff using different occurrence based Wi-Fi fingerprints.

2.3.1 Spatial Variation

To characterize Wi-Fi fingerprints under different spatial variations, we conducted 10 experiments in each pair of adjacent rooms in three different buildings. Each pair of rooms have similar size and layout. Two phones were placed $d(= 0, 1, 2, 4, 8)$ meters apart, either in the same room (without obstacle) or in adjacent rooms (across wall). In each experiment, two phones collected Wi-Fi signals for 10 minutes, obtaining one sample every 5 seconds. In order to eliminate the impact of device variation in the experiments, we used two phones with the same model and one phone was calibrated to the other in our post data processing. All experiments were conducted in the same rooms during late night when the ambient environment was relatively stable, and during the same time period in order to reduce the impact of temporal variation.

• **Occurrence based Wi-Fi fingerprints.**

In each experiment, using the Wi-Fi samples obtained by each phone, we can calculate the occurrence rate for a given k-combination or ordered k-combination of APs S_k, and the **occurrence rate difference (OccRate-Diff)** for S_k between the two phones is defined as the absolute difference between one phone's occurrence rate and the other phone's occurrence rate. An occurrence rate

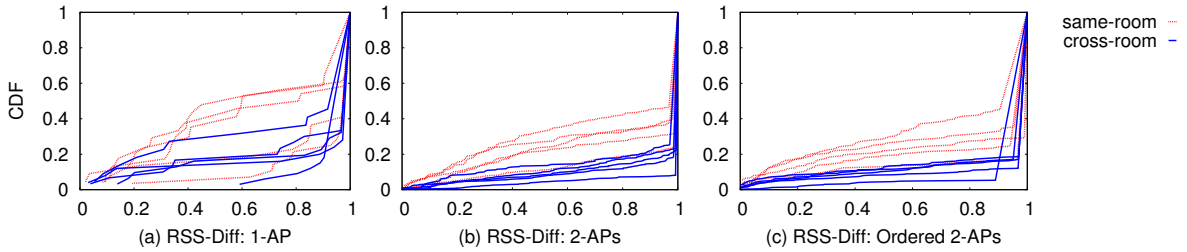


Figure 2.4: Spacial variation: CDF of RSS-Diff using different RSS based Wi-Fi fingerprints.

of 0 is assigned for any S_k that is not observed by a phone.

Figure 2.3 shows the CDF curves of OccRate-Diff between two phones at different distances d, within the same room or cross room. Figure 2.3(a)-(c) are plots for k-combination of APs with $k = 1, 2, 3$, respectively. Figure 2.3(d)-(e) are plots for ordered k-combination of APs with k = 2,3 respectively. Note that when $k = 1$, k-combination and ordered k-combination are the same. Curves that are closer to the top left in each plot indicate smaller OccRate-Diff values and more similar fingerprints. It is expected that increasing spacial distance and going across wall would increase the OccRate-Diff between two phones, in effect moving the curves toward the bottom right in each plot.

As shown in Figure 2.3(a), for 1-combination (i.e., single APs), there is a clear separation between the same-room OccRate-Diff distributions (red curves) and the cross-room OccRate-Diff distributions (blue curves). This means that we could use this fingerprint to distinguish rooms. The curves in the same room are not very similar, meaning that the same-room fingerprints are not very stable given different space variations. Increasing k from 1 to 3 (Figure 2.3(a)-(c)) makes the same-room curves closer to each other, i.e., better concentration and more stable fingerprints for each room despite spatial variations. However, as k increases, the distribution curves for same-room and cross-room start to mix with each other, making it difficult to distinguish rooms.

When ordered k-combination of APs is used ($k = 2$ for Figure 2.3(d) and $k = 3$ for Figure 2.3(e)), we see good concentration (i.e., stable) of same-room fingerprints despite spacial variations, as compared with their unordered k-combination counterparts (Figure 2.3(b)-(c)). Unfor-

tunately, such fingerprints have very little differentiating power across rooms as the same-room and cross-room distribution curves are mixed and have much overlap.

• **RSS based Wi-Fi fingerprints.**

Given a specific k-combination or ordered k-combination of APs S_k, let vectors A and B be the histograms of observed RSS (for 1-combination, i.e., 1-AP) or RSS difference (for 2-combination or ordered 2-combination, i.e., 2-APs) on two phones, respectively. The **RSS difference (RSS-Diff)** between two phones is calculated using Tanimoto Distance, which is defined as follows:

$$TanimotoDistance(A, B) = \frac{A \cdot B}{|A|^2 + |B|^2 - A \cdot B} \tag{2.1}$$

Figure 2.4 shows the CDF curves of RSS-Diff under different space variations (different distance and across wall). Three different RSS based fingerprints are considered: 1-AP, 2-APs, and ordered 2-APs. In Figure 2.4(a), we can see that there is much spread among same-room curves as well as much overlap between the same-room curves (red) and cross-room curves (blue). Therefore, under different spatial variations, RSS-Diff with 1-AP is not a good fingerprint for room localization. In Figure 2.4(b), curves for the same room are better concentrated and there is much better separation between same-room and cross-room curves. The same can be observed in Figure 2.4(c), with slightly better same-room concentration and cross-room separation. Therefore, RSS with 2-combination or ordered 2-combination of APs are good fingerprints to use for room localization under spatial variations. The results are also shown in Table 2.3.

2.3.2 Temporal Variation

To characterize Wi-Fi fingerprints under temporal variations, we used one phone to collect Wi-Fi signals in one room during different time slots of a day: **morning, noon, afternoon, evening, and night** and then collected Wi-Fi signals with the same phone on another two days in the same room and adjacent room at a position that is 2 meters apart from the first experiment position. We chose the 2-meter distance in order to maintain the same distance for same-room and cross-room experiments. In each experiment, we collected Wi-Fi signals over a time window of 10 minutes,

obtaining one sample every 5 seconds. Then, to calculate OccRate-Diff and RSS-Diff for different (ordered) k-combinations, we consider Wi-Fi samples collected at time periods, such as morning vs. afternoon or noon vs. evening.

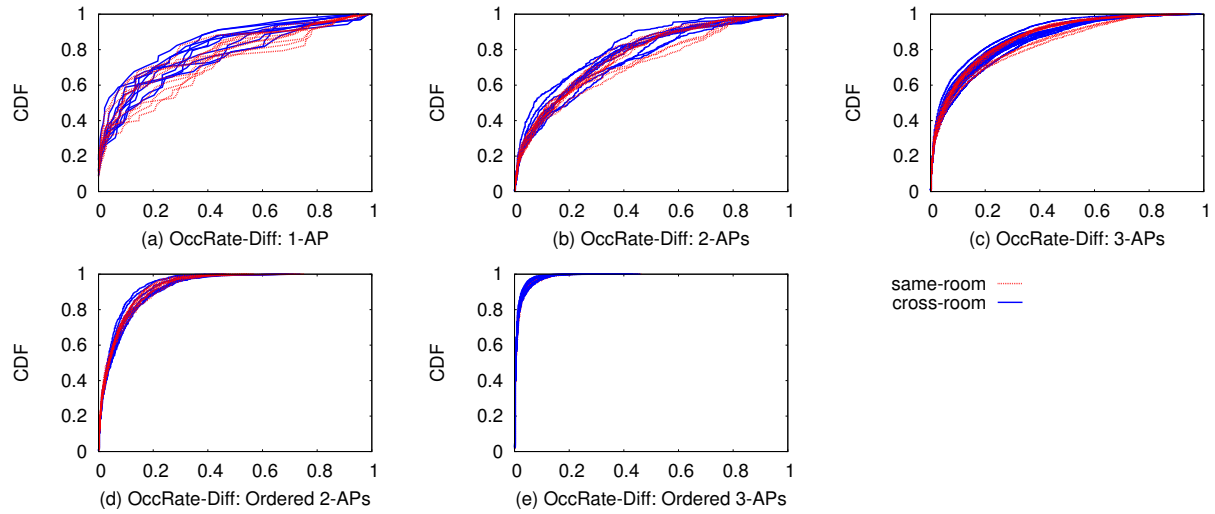


Figure 2.5: Temporal variation: CDF of OccRate-Diff using different occurrence based Wi-Fi fingerprints.

• **Occurrence based Wi-Fi fingerprints.**

Figure 2.5 shows the CDF curves of OccRate-Diff under different temporal variations, using different k-combination of APs ($k = 1, 2, 3$) and ordered k-combination of APs ($k = 2, 3$) as the fingerprint. The red (blue) curves represent same-room (cross-room) scenarios under different time variations. First, we can observe that in all five plots, the same-room and cross-room curves have much overlap and are difficult to separate. Therefore, none of the occurrence based Wi-Fi fingerprints have good separation power across rooms. Our results also confirmed the observation in [18] that real-time signal distributions often differ from those in the training phase. On the other hand, despite different temporal variations, we do see good concentration of same-room curves especially when ordered k-combination is used.

• **RSS based Wi-Fi fingerprints.**

Figure 2.6 shows the CDF curves of RSS-Diff under different temporal variations, using 1-

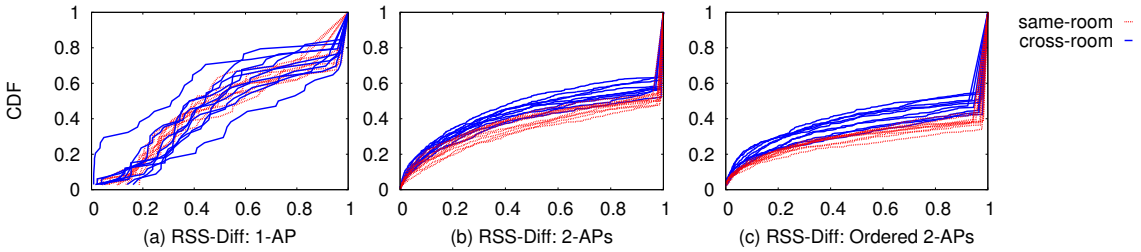


Figure 2.6: Temporal variation: CDF of RSS-Diff using different RSS based Wi-Fi fingerprints.

AP, 2-APs, and ordered 2-APs as fingerprints. Again, the red (blue) curves represent same-room (cross-room) scenarios under different time variations. From Figure 2.6(a), we see that there is much overlap between same-room and cross-room curves, i.e., low separation power. There is also a large spread among same-room curves, indicating poor concentration. The good news is that both 2-APs and ordered 2-APs perform well under temporal variations. As shown in Figure 2.6(b)-(c), we can clearly see that the same-room and cross-room curves are well separated, indicating high differentiating power for room localization. Moreover, both types of fingerprints also have good concentration for same-room scenarios.

2.3.3 Device Heterogeneity

To characterize Wi-Fi fingerprints under the impact of device heterogeneity, we used 7 different smartphones to collect Wi-Fi signals in a room and its adjacent room and then compared both OccRate-Diff and RSS-Diff RSS distributions for each pair of different phones, such as HTC Sensation vs. Samsung Galaxy S2. In each experiment, the 7 phones were placed together at one position. We then moved the phones to a different position (in the same room or in the adjacent room) that was 2 meters away from the original position. Each experiment was conducted for 10 minutes. To minimize temporal variation, each experiment was conducted right after the other and during night when the surrounding environment was very stable.

• **Occurrence based Wi-Fi fingerprints.**

Figure 2.7 shows the CDF curves of OccRate-Diff under different device heterogeneity, us-

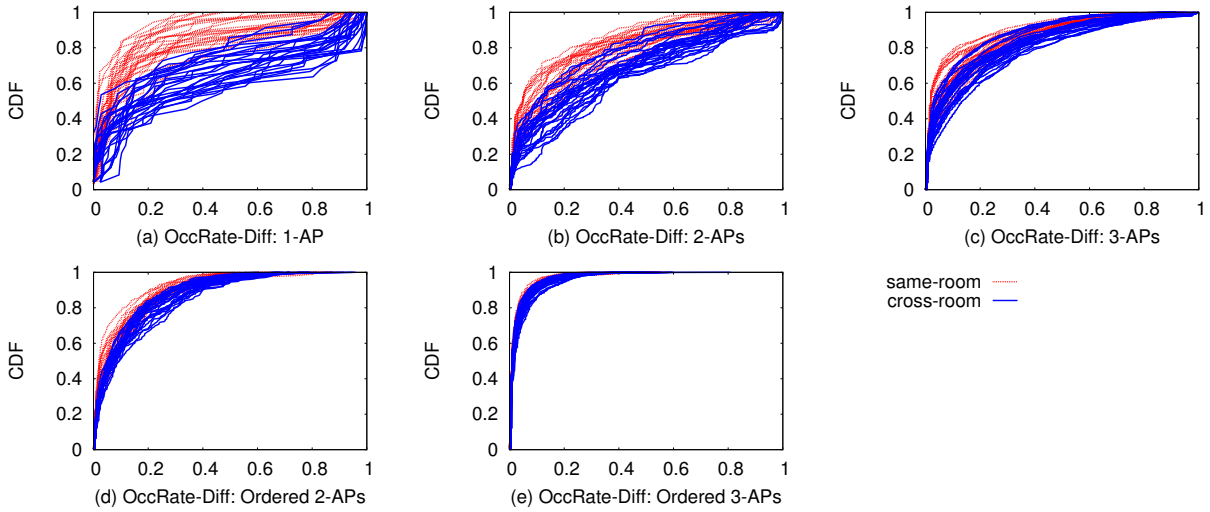


Figure 2.7: Device variation: CDF of OccRate-Diff using different occurrence based Wi-Fi fingerprints.

ing different k-combination of APs ($k = 1, 2, 3$) and ordered k-combination of APs ($k = 2, 3$). The same-room and cross-room Wi-Fi curves can be separated, especially when k =1 and 2 (see Figure 2.7(a)-(b)). However, the distributions within the same room (the blue or red curves) are not well concentrated. This shows the variation among devices. With the increase of k (see Figure 2.7(c)), although the same-room distributions are getting more similar (i.e., better concentration), there is also increased overlap between same-room distributions and cross-room distribution, i.e., reduced separation power. In Figure 2.7(d)-(e), we can see that the distribution curves within a room have good concentration, which makes room recognition easier. However, most of the curves are mixed up with the cross-room curves. As a result, it is difficult to use the ordered k-combination of APs for room localization.

• **RSS based Wi-Fi fingerprint.**

To understand how device heterogeneity impacts RSS based fingerprints for room localization, we draw the CDF of RSS-Diff between any two different devices, when using different k-combination ($k = 1, 2$) and ordered 2-combination. As shown in Figure 2.8, in all three plots, there is a large spread among the same-room curves. In other words, when different devices are

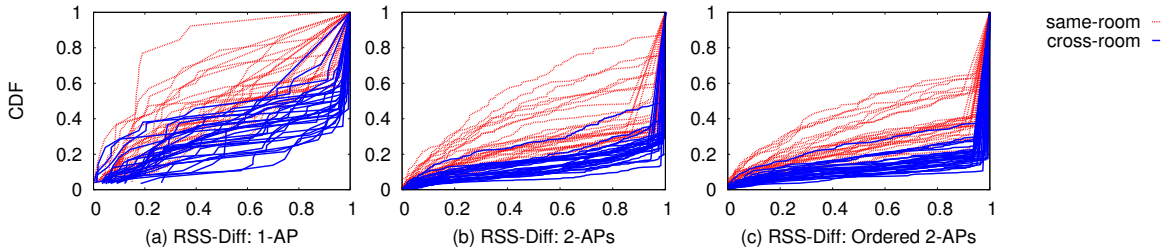


Figure 2.8: Device variation: CDF of RSS-Diff using different RSS based Wi-Fi fingerprints.

used, the RSS based Wi-Fi fingerprints can vary significantly within the same room. Such poor concentration makes it difficult to identify a specific room using the RSS based fingerprints. On the other hand, there is good separation between the same-room and cross-room curves, especially with 2-combination APs and ordered 2-combination APs (see Figure 2.8(b)-(c)). Also, ordered 2-combination APs slightly outperforms unordered 2-combination APs. This is due to the fact that there is better concentration of same-room curves when ordered 2-combination APs are used, thus increasing the ability to identify specific rooms and distinguish different rooms.

2.3.4 Room Wall Type

To characterize how Wi-Fi fingerprints may be affected by different room wall types, we first categorized walls into four types: **load-bearing wall, drywall, glass wall, and low wall**. Load-bearing wall provides a significant structural support in a building, and is normally made of concrete, blocks, or bricks. Wi-Fi signals passing through a load-bearing wall may be significantly affected. Drywall is non-structural and merely functions as decoration or for room separation. Glass wall and low wall function as room separation as well, such as separating a large room into smaller rooms or cubicles. The diverse materials used to constructing the walls can impact Wi-Fi signal measurements.

In this experiment, we used two phones to sample Wi-Fi signals across each of the 4 aforementioned types of walls. Wi-Fi measurements will be collected on both sides of the wall at positions that are 2 meters apart. Same as other experiments, a 10-minute sampling window is used

for each configuration with a sampling frequency of 1 sample every 5 seconds. Since it is not feasible in practice for us to "change" the type of wall for the same room, we are not going to consider same-room concentration in this set of experiments, and instead focus only on cross-room separation.

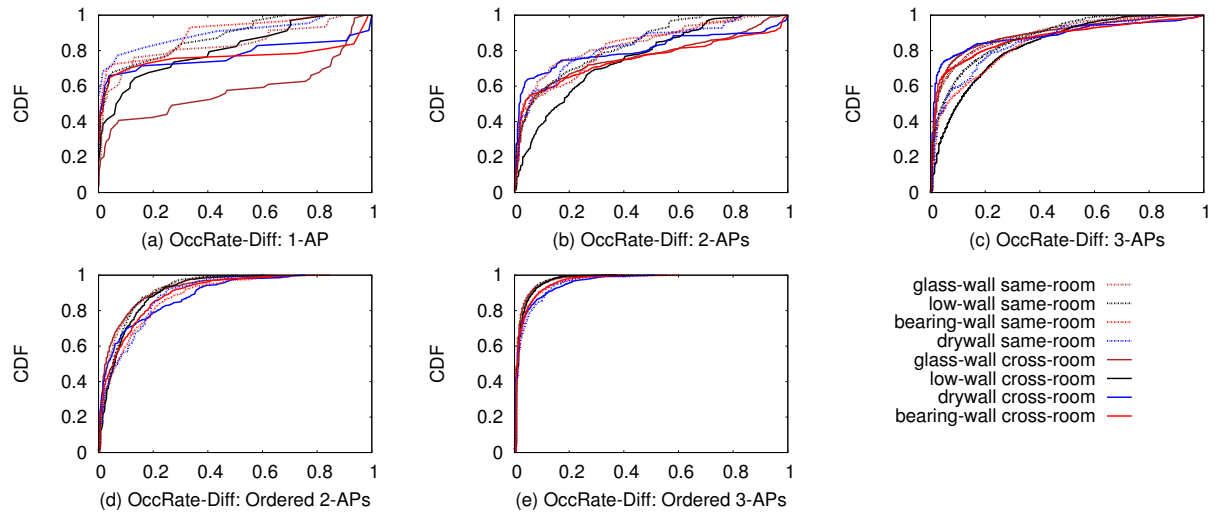


Figure 2.9: Room wall type: CDF of OccRate-Diff using different occurrence based Wi-Fi fingerprints.

• **Occurrence based Wi-Fi fingerprints.**

Figure 2.9 shows the CDF curves of OccRate-Diff when different types of walls are present and different occurrence based Wi-Fi fingerprints are used. In each of the plots, each pair of dotted and solid curves of the same color indicate cross-room settings for a specific type of wall. We can see in the plots that OccRate-Diff with 1-AP (Figure 2.9(a)) has some separation power, while others (Figure 2.9(b)-(e)) have substantial mix and cannot effectively distinguish rooms.

• **RSS based Wi-Fi fingerprints.**

Figure 2.10 shows the CDF curves of RSS-Diff when different types of walls are present, for three different RSS based Wi-Fi fingerprints: 1-AP, 2-APs, and ordered 2-APs. Our goal is to separate dotted and solid curves of the same color (i.e., a specific wall type), which is evident in all three plots. Combining 2 APs' RSS performed better than single AP RSS, and ordered 2-

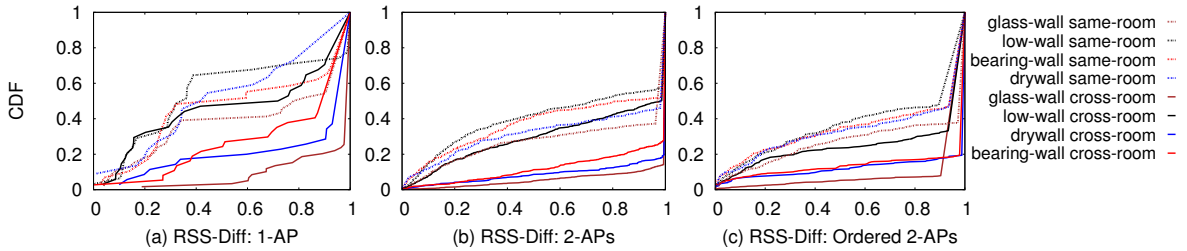


Figure 2.10: Room wall type: CDF of RSS-Diff using different RSS based Wi-Fi fingerprints.

combination APs outperformed the unordered case. Moreover, all four types of walls, including low walls used for cubicle separations, have significant impact on RSS fading. Therefore, we should be able to use RSS based Wi-Fi fingers to separate rooms even when different room types may be present.

2.3.5 AP Deployment Type

We define three different types of AP deployment: **centralized, decentralized, and mixed**. Examples of different types are shown in Table 2.2. To characterize the impact of AP deployment type on Wi-Fi measurements, we conducted experiments at three different buildings with different AP deployment types. In each building, two phones were first placed 2 meters apart in the same room, then one phone was moved to the adjacent room, still 2 meters apart from the other phone. Wi-Fi measurements were collected for 10 minutes during each experiment. To reduced the device variation, the two phones are calibrated. To reduce the impact of wall type, we chose the same type of wall for the experiment. OccRate-Diff and RSS-Diff are calculated based on the Wi-Fi measurements in adjacent rooms from the sam AP deployment type, namely from the same building. Similar to the wall type experiments, since we cannot "change" the AP deployment type for the same room, we only consider Wi-Fi fingerprint separation across room for each specific AP deployment type.

- **Occurrence based Wi-Fi fingerprints.**

Figure 2.11 shows the CDF curves of OccRate-Diff under different AP deployment types

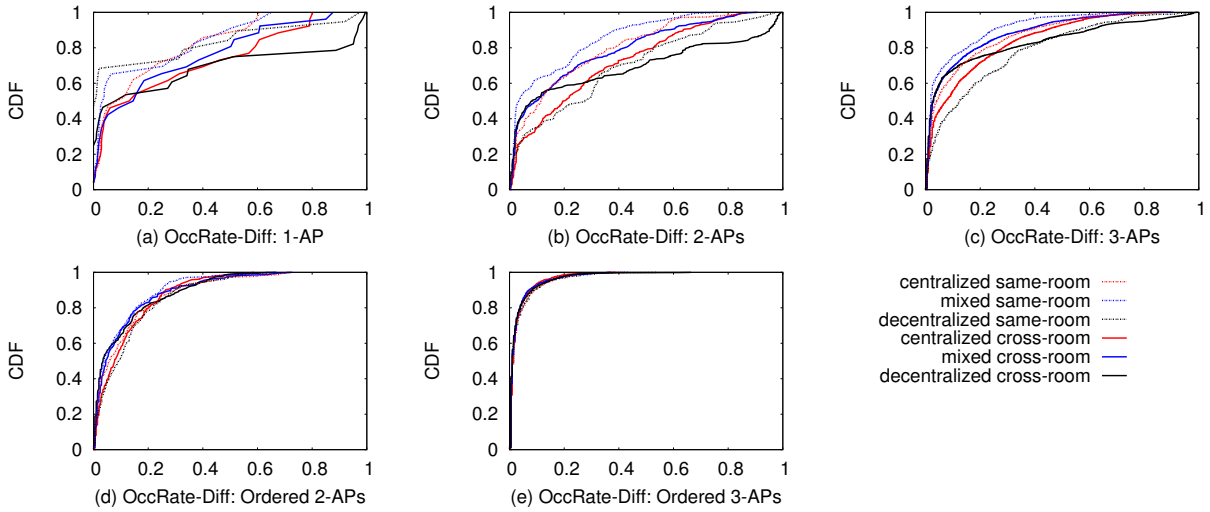


Figure 2.11: AP deployment type: CDF of OccRate-Diff using different occurrence based Wi-Fi fingerprints.

and using different k-combination ($k = 1, 2, 3$) and ordered k-combination of APs ($k = 2, 3$). If the dotted and solid curves of the same color are well separated in a plot, then the corresponding fingerprint has good cross-room separation. As shown in the figure, the unsorted k-combination ($k = 1, 2, 3$) schemes have some separation but not very good, while ordered k-combination ($k = 2, 3$) leads to curves that are highly mixed and difficult to separate different rooms.

• **RSS based Wi-Fi fingerprints.**

Figure 2.12 shows the CDF plots of RSS-Diff under different AP deployment types when different RSS based Wi-Fi fingerprints are used, including 1-AP, 2-APs, and ordered 2-APs. Again, our goal is to separate dotted and solid curves of the same color, which we can see in all three plots. This meanings that all three RSS based Wi-Fi fingerprints have good separation power across rooms under different AP deployment types. Moreover, as we can see in Figure 2.12(c), there is better separation for decentralized AP deployment than that for mixed or centralized AP deployment. Intuitively, when other setting stay the same, the non-uniformness of decentralized AP deployment leads to more unique Wi-Fi fingerprints in different rooms and better room separation.

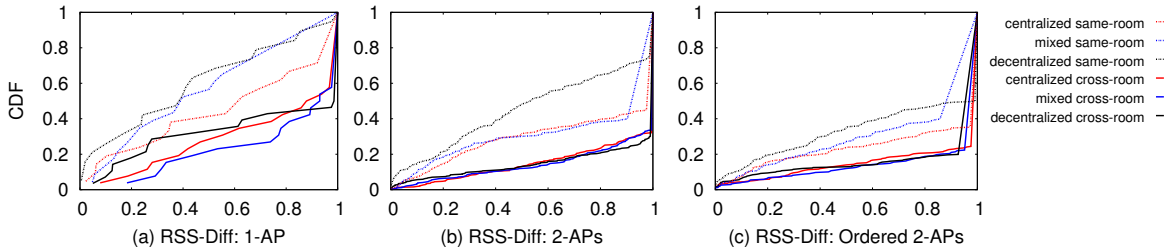


Figure 2.12: AP deployment type: CDF of RSS-Diff using different RSS based Wi-Fi fingerprints.

Table 2.3 summarizes the characteristics of different Wi-Fi fingerprints in terms of same-room concentration (con.) and cross-room separation (sep.) under different types of variations. A good fingerprint scheme for room localization should maintain both properties under different types of variations. As we can see in the table, both RSS 2-APs and RSS ordered 2-APs perform well. The latter has slightly better performance in almost all the settings, which leads to our new room localization method.

2.4 Room Positioning

Researchers have proposed room localization techniques based on Wi-Fi access point received signal strength (RSS) [43, 18]. These methods share two common stages. In the first (training) stage, a database that associates ambient Wi-Fi RSS fingerprints with physical rooms is constructed. In the second (operating) stage, the system identifies the stored Wi-Fi fingerprint that most similar to the one currently being measured, and returns the associated room.

The first stage of our room localization technique is similar to that of Park et al. [43]. All users contribute their Wi-Fi RSS and room information to create a shared database of room fingerprints. This is beneficial as it (1) eliminates the deployment cost for fingerprint pre-sampling and (2) reduces individual users' effort to build the database.

In the second stage, Bayesian room localization models are commonly used [18, 43]. Given a database of fingerprinted rooms R and a Wi-Fi RSS fingerprint represented by a set of access point (AP)-specific signal strengths (w_i for the i-th AP), the mobile device (and user) is most likely

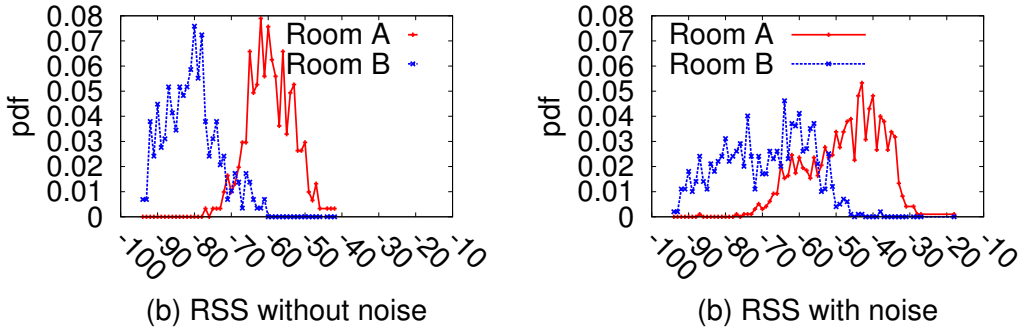


Figure 2.13: Wireless RSS distributions in two adjacent rooms: (a) without noise and (b) with environment and device noise.

in room $\hat{r}$:

$$\hat{r} = \arg\max_{r \in R} \left[\prod_i P(w_i|r)P(r) \right] \tag{2.2}$$

This model is based on the assumption that the signal strengths observed by the mobile device from different APs are conditionally independent. However, the model fails to address the following two challenges: (1) **device heterogeneity** – different devices may be used for gathering RSS fingerprints and devices might be held differently (e.g., in hand, pocket, or bag) and (2) **environment heterogeneity** – the wireless environment of a room may change over time, due to motion of people and other room contents, influencing the RSS fingerprints gathered by mobile devices. As demonstrated in Figure 2.13, noise induced by device and environment heterogeneity significantly increase the RSS overlap between adjacent rooms, leading to much lower room localization accuracy. To address these problems, we propose a novel temporal n-gram augmented Bayesian room localization method, which is robust to both environment and device noises.

2.4.1 N-gram Augmented Bayesian Room Localization

Our key observation is that, although the exact RSS values of each AP may change substantially for different devices and environments, **the ordered sequence of APs based on their RSS values tends to be similar for the same room and inconsistent among adjacent rooms**. For example, the ordering may be $[ap_1, ap_2, ap_3, ap_4, ap_5, ap_6]$ at one time, and $[ap_2, ap_1, ap_3, ap_4, ap_5, ap_6]$

at another time, for the same room. The ordered AP sequences of adjacent rooms are less similar, especially when different APs are observed in these rooms. Intuitively, the ordered AP sequence is useful for room localization because: (1) it captures the inherent correlations among APs, which are stable for the same room yet different for adjacent rooms and (2) it uses the order of RSS instead of their exact values, allowing many sources of device variation and wireless environment variation to be tolerated.

Based on the observations above, we propose an n-gram augmented Bayesian room localization model, which works as follows. Let s be a sequence of N APs ordered in descending RSS values:

$$s = (ap_1, ..., ap_N) \quad w_i \geq w_j (1 \leq i < j \leq N), \tag{2.3}$$

where w_i is the RSS value of ap_i observed by the mobile device. An n-gram is then defined as a subsequence of length n extracted from the sequence s at position i:

$$ngram_i(s) = (ap_i, \cdots, ap_{i+n-1}). \tag{2.4}$$

The most likely room $\hat{r}$ is determined as follows:

$$\hat{r} = \arg\max_{r \in R} \left[\prod_i P(ngram_i(s)|r) P(r) \right]. \tag{2.5}$$

In other words, $\hat{r}$ is the room with the highest probability of having the same ordering of APs in subsequence $ngram_i(s)$.

2.4.2 Temporal User Mobility for Room Localization

As shown in the experimental results, our n-gram augmented Bayesian room localization model achieves high accuracy when the room has enough fingerprints (more than 50 or 100). However, when the number of fingerprints is low, the model accuracy is poorer and users tend to be misclassified into nearby rooms. To remove such spatial errors, we propose to incorporate temporal user mobility information. This is motivated by the following observations.

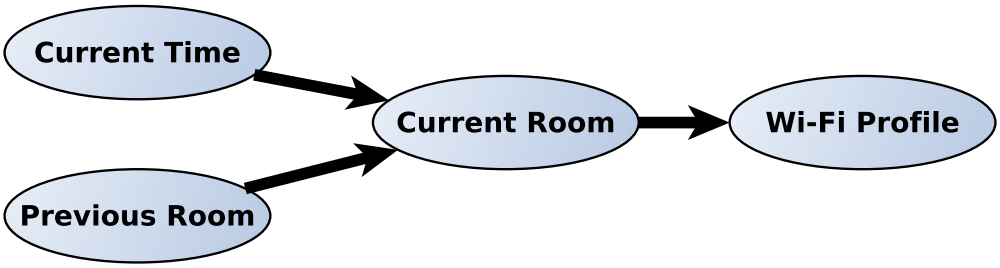


Figure 2.14: Bayesian network for room localization.

- A user's current room is closely related to time and weekday, e.g., the user has weekly meetings in the conference room on Tuesday mornings.
- Users can only move among adjacent rooms, and their paths tend to contain patterns. For example, a user usually goes to the conference room from her office instead of from a classroom.

Based on these observations, a user's current room can be predicted based on current time and previous room. As shown in Figure 2.14, the Bayesian network has three layers: current time and user's previous room (first layer) indicate the user's current room (second layer), and user's current room determines the observed Wi-Fi RSS fingerprint. We also define a set of values to represent some semantic concepts of time, including "day of week", "morning", "afternoon", and "evening". Given a Wi-Fi scan observation s, the user's previous room r', and current time t , the user is most likely in room

$$\hat{r} = \arg\max_{r \in R} \left[P(s, r, t, r') \right] \tag{2.6}$$

$$= \arg\max_{r \in R} \left[P(s|r) P(r|t, r') P(t) P(r') \right]. \tag{2.7}$$

$P(s|r)$ can be computed using our n-gram augmented Bayesian room localization model. $P(r')$ and $P(r|t, r')$ are calculated from the user's mobility history. $P(t)$ can be ignored since it is the same for any room r.

2.5 Evaluations

To evaluate our temporal n-gram augmented Bayesian room localization method, we determined the room localization accuracy as a function of the number of Wi-Fi fingerprints. A good room localization method should achieve high accuracy with few fingerprints. Specifically, we conduct experiments to answer the following questions: (1) How does the n-gram length affect overall room localization performance? (2) How does the n-gram augmented Bayesian model perform compared with existing approaches? (3) To what extent does temporal user mobility information improve room localization accuracy?

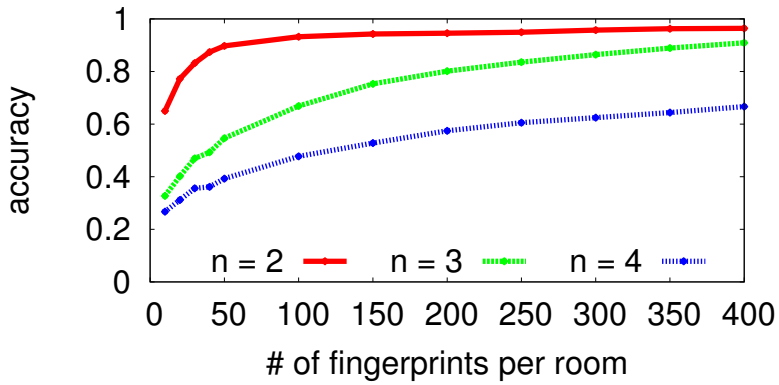


Figure 2.15: Comparison of different n for n-gram augmented Bayesian model.

The subsequence length n of n-gram has an important impact on room localization performance. If n is too small, adjacent rooms may have many similar subsequences and be difficult to distinguish. For example, when $n = 1$, our n-gram augmented model is similar to the original Bayesian model, and the only difference is the binary discretization of signal strength: AP equals 1 (or 0) if its signal can (or cannot) be observed by the device. If n is too big, then the room signature may change for each scan, since longer subsequences have a low probability of being repeated in each scan. For example, when n is set to the total number of APs that can be scanned in a room, it is very unlikely to receive the same sequence for each scan. Therefore, a good n value should make the n-gram signature for each room stable for different scans, and make the n-gram signa-

tures of adjacent rooms distinguishable. Figure 2.15 shows the accuracy of our n-gram Bayesian model with different n values and different number of fingerprints per room. As shown in the figure, when $n = 2$, our model achieves much better accuracy with fewer fingerprints.

We now compare our n-gram augmented Bayesian room localization model with four state-of-the-art algorithms: (1) Bayesian room localization [18, 43], (2) Delta signal Bayesian room localization, which is similar to Bayesian room localization but uses the difference of signal strength (instead of RSS values directly) between each pair of APs to calculate probability, (3) vector-based room localization [6], which uses AP RSS vector as the room signature and Euclidean distance to locate the nearest room, and (4) Delta signal vector-based room localization, which is similar to vector-based room localization but uses the difference in signal strength between each pair of APs to build the vectors. As shown in Figure 2.16(a), our n-gram augmented Bayesian model achieves the best accuracy, especially when the number of fingerprints per room is above 50.

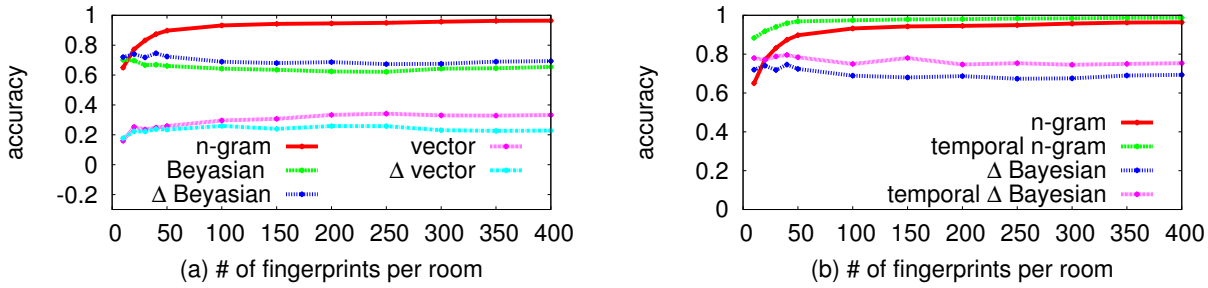


Figure 2.16: Performance comparison of room localization methods: (a) n-gram model and (b) temporal n-gram model.

By incorporating the temporal user mobility information, our temporal n-gram augmented Bayesian model can achieve better accuracy even when the number of fingerprints per room is limited. As shown in Figure 2.16 (b), our temporal n-gram model further improves the room localization accuracy over our n-gram model, especially when the number of fingerprints per room is less than 50. The Delta Bayesian room localization model, which performed second best in Figure 2.16(a), can also benefit from the use of temporal user mobility information, but it is still not comparable to our temporal n-gram model.

2.6 Summary

Variations in device, space, time, room wall type and AP deployment type can significantly impact the same-room concentration and cross-room separation of Wi-Fi fingerprints, which in turn affect the effectiveness of Wi-Fi based indoor room localization, a problem that has become increasingly important in people's daily lives. In this chapter, we have designed and conducted in-depth experiments to arrive at a comprehensive characterization of different Wi-Fi fingerprints with regard to changes in aforementioned factors, showing different levels of robustness of Wi-Fi fingerprints under a wide variety of changes. We proposed a new room localization method using RSS with ordered 2-combination, since it maintains good same-room concentration and cross-room separation under different variations. We have also conducted a comparative evaluation of state-of-the-art room localization methods, along with our new method, using data collected in real-world user study. Our evaluation results demonstrate that different methods performed differently in terms of accuracy and training cost, all methods had lower accuracy under more variations, and our new method achieves both high accuracy and low training cost under a wide variety of settings.

Chapter 3

Room Fingerprinting

This chapter presentes a fully automated indoor room fingerprinting approach. To accurately identify rooms without extensive manual annotation, we proposed and developed a number of novel techniques: 1) a zone-based clustering algorithm that accurately identifies in-room occupancy hotspot(s) using Wi-Fi signatures; 2) a motion-based clustering algorithm to identify inter-zone correlation, thereby distinguishing different rooms; and 3) an energy-efficient motion detection algorithm to minimize the noise of Wi-Fi fingerprints. The approach has been implemented and deployed for a 10-month study with 21 participants. The evaluation results demonstrate that our automated clustering algorithm generates clusters that are high representative of individual rooms and achieves high accuracy (95%) for room localization. The accuracy is comparable to existing techniques that require labor-intensive manual annotation.

3.1 Related Work

In this section, we survey works that are most related to ours, focusing on indoor localization techniques based on Wi-Fi signals. We also discuss works that aim to reduce the effort of fingerprint collection for indoor localization.

Indoor localization has been a topic of active research, some focusing on indoor positioning while others (similar to our work) aiming to determine rooms rather than the exact indoor positions. Indoor positioning systems [6, 17] can be easily extended to room-level localization but require extensive effort for fingerprint sampling or the pre-knowledge of AP positions, which are

not scalable. Previous room location methods either rely on manual room fingerprint collection, which imposes a high cost [18], or leverage user feedback to reduce deployment and maintenance cost [7]. Issues such as determining when user input is actually required, and discounting erroneous and stale data are addressed by the work of Park et al. [43, 16]. However, their approach requires a floor plan to deal with input errors and the inconsistent room naming preferences of different users. Furthermore, the input effort is non-negligible, given the small screens and keyboards of mobile phones. Most of these systems use Wi-Fi signals as room or position fingerprints.

To reduce fingerprint collection efforts, some works focused on reducing sampling locations and sampling time while maintaining similar localization accuracy [33, 8]. Other works proposed calibration-free approaches to improve scalability but require AP infrastructure information. Gwon and Jain proposed TIX (triangular Interpolation and extrapolation), a calibration-free mechanism that used the three APs with the highest RSSIs to determine the centroid of the triangle [17]. Lim et al. proposed a zero-configuration indoor localization method, which used online calibration and truncated singular value decomposition (SVD) to characterize the relationship between RSSI and geographical distance to anchors [36]. More recently, unsupervised indoor positioning systems [9, 53] have been proposed by leveraging GPS signals near a window or internal landmark that can be easily identified. However, these approaches rely on indoor GPS signasl, indoor landmarks, and/or floor plans. Our solution is fully automated and requires no manual annotation or prior knowledge of APs or floor plans, yet achieves high room localization accuracy.

Density-based clustering [13, 14] is a popular method and attractive for spatial identification. It clusters objects based on neighborhood density, which is defined by a given radius (Eps) and a minimum number of objects ($MinPts$). It can identify clusters of arbitrary shapes and is robust to noise. Density-based clustering is thus appropriate for our problem and achieves high performance as demonstrated by the evaluation results.

3.2 Problem Formulation and System Overview

This section formulates the Wi-Fi based room fingerprinting problem for indoor localization and summarizes the automatic room fingerprinting system architecture.

3.2.1 Room Fingerprinting and Localization

Room fingerprinting: Given a building with a set of rooms R and K Wi-Fi stationary access points (APs), Wi-Fi based room fingerprinting is to assign, to each room $r \in R$, a unique Wi-Fi signature consisting of m Wi-Fi signal vectors, $\mathbb{V}_r = \{V_{rj} | j \in \mathbb{Z}\ and\ 1 \leq j \leq m\}$, and

$$V_{rj} = \{ap_1 : rss_1, ..., ap_k : rss_k\}, \tag{3.1}$$

where each access point ap is identified by its MAC address, rss is the received signal strength (RSS) value of ap observed by a mobile phone. Note that k ($k \leq K$), the number of access points observed, could vary for each Wi-Fi scan, and m ($m \geq 1$), the number of Wi-Fi signal vectors, also varies by room. Automatically learning the Wi-Fi signature for each room is the key challenge addressed in this work.

Room localization: Given a run-time h-dimensional ($h \leq K$) Wi-Fi signal vector V_o observed by a user's mobile phone, indoor room localization determines the room that the user is in by measuring the similarity between V_o and each Wi-Fi signal vector $V_{rj} \in \mathbb{V}_r$ in each room r's fingerprint. We adopt the n-gram augmented Bayesian room localization method, which can achieve high accuracy, even when data are gathered using heterogeneous mobile phones [21]. Let Q_o be the sequence of access points sorted in descending RSS values:

$$\begin{aligned} Q_o &= seq(V_o) = (ap_{q_1}, \cdots, ap_{q_h}) \quad \text{with} \\ & rss_{q_i} \geq rss_{q_j} \quad (1 \leq q_i < q_j \leq h). \end{aligned} \tag{3.2}$$

An n-gram of Q_o is defined as a subsequence of length n extracted from the sequence Q_o at position i ($1 \leq i \leq h - n + 1$):

$$ngram_i(Q_o) = (ap_{q_i}, ap_{q_i+1}, \cdots, ap_{q_i+n-1}). \tag{3.3}$$

In the set of all rooms R, the most likely room $\hat{r}$ of occupation is determined as follows:

$$\hat{r} = \arg\max_{r \in R} \left[\prod_i P(ngram_i(Q_o)|r) P(r) \right]. \tag{3.4}$$

In other words, $\hat{r}$ is the room with the highest probability of producing the same ordering of access points as in subsequence $ngram_i(Q_o)$. Note that each V_{rj} $(1 \leq j \leq m)$ in room r is also converted to an n-gram subsequence based on Equation 3.2 and Equation 3.3. The signature of room r, $\mathbb{V}_r = \{V_{rj} | j \in \mathbb{Z}\ and\ 1 \leq j \leq m\}$, is therefore converted to

$$\{ngram_i(seq(V_{rj})) \quad | \quad i, j \in \mathbb{Z}, 1 \leq j \leq m \\ and\ 1 \leq i \leq |seq(V_{rj})| - n + 1\}.$$

$P(ngram_i(Q_o)|r)$ is the probability of $ngram_i(Q_o)$ appearing in the converted signature of room r.

3.2.2 System Overview

The system supports automatic indoor room fingerprinting and room localization using collaborative Wi-Fi signature analysis based on personal mobile phones carried by occupants. Figure 3.1 illustrates the overall system architecture, which is comprised of components on the mobile phone side and server side.

On the mobile phone side, the system performs the following operations.

- The run-time Wi-Fi signal vectors observed by each mobile phone are collected and delivered to the server to support room fingerprinting and room localization.
- The Wi-Fi signal vector stream is further annotated with motion data from build-in accelerometer, i.e., either collected when the occupant is in motion or stationary. Such information is also delivered to the server to improve Wi-Fi fingerprint identification of in-room occupancy hotspot(s) and inter-hotspot correlation.

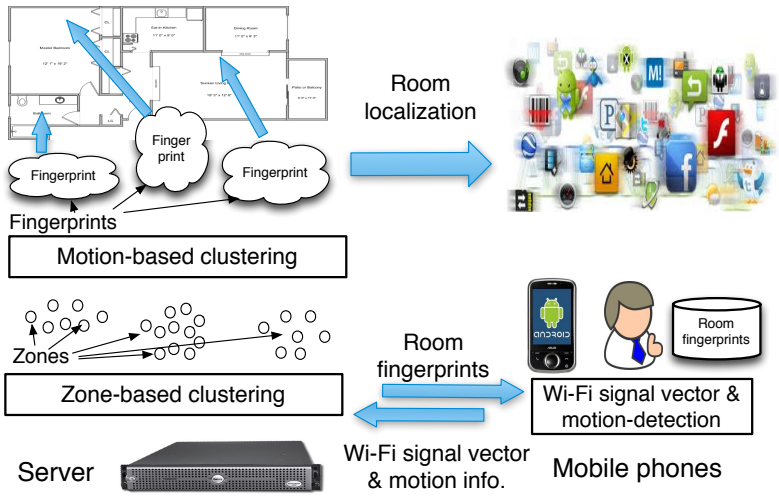


Figure 3.1: Automatic room fingerprinting system architecture.

- Each mobile phone also maintains a local database storing the fingerprints[1] of the rooms that the user has visited before, which serves as a local cache, enabling run-time on-device room localization without engaging the server.

- A system software module provides room localization APIs to support high-level applications & services.

On the server side, the system performs room fingerprinting and localization through an incremental process.

- Given the streams of Wi-Fi signal vectors and the corresponding motion information collected from mobile phones, the system uses the zone-based clustering algorithm to incrementally identify in-room occupancy hotspot(s), or zone(s). Meanwhile, inter-zone correlations are identified by the motion-based clustering algorithm, then zones belonging to the same room are merged into a new cluster. Each cluster is assigned a room ID and the Wi-Fi signal vectors in the cluster form the room fingerprint.

[1] These are the converted room fingerprints, i.e., n-gram subsequences of APs and corresponding probabilities of seeing each of them in a room.

- Using the n-gram augmented Bayesian room localization method, run-time room localization services are then offered to the occupants. A room fingerprint database maintains room IDs, room fingerprints, and the converted room fingerprints (n-gram AP subsequences and corresponding probabilities). The converted room fingerprints are selectively synchronized to each user's mobile phone based on the user's room visit history and predicted room visits in the future.

3.3 Zone-based Clustering

Our first step leverages Wi-Fi signals collected in stationary sessions (i.e., when a user is stationary in a room). We propose a zone-based clustering algorithm to accurately identify within-room stationary occupancy hotspot(s), i.e., zone(s). Each zone is then identified by a unique Wi-Fi signature, consisting of a set of Wi-Fi signal vectors which are typical for that zone.

Stationary occupancy hotspot (zone) identification is critical for accurate room fingerprinting and localization due to the following reasons. Within a room, human activities are highly nonuniform. One or multiple stationary occupancy hotspots typically exist, such as the couch area in a living room, cashier desk in a store, and desks in an office. Leveraging the mobile phones carried by occupants, more Wi-Fi signal samples are naturally collected from these zones, offering more robust fingerprints for a room. Vice versa, since occupants spend more time around these zones, their fingerprints usually better match the Wi-Fi signal vectors reported by occupants at run-time, improving accuracy.

The zone-based clustering algorithm uses the Wi-Fi signal vectors reported by the mobile phones when the occupants are stationary, e.g., sitting or standing (detected by our motion detection algorithm described in Section 5.1). We further define the set of Wi-Fi signal vectors collected during a stationary period of an occupant as a **Wi-Fi session**. Next, giving a large collection of Wi-Fi sessions reported from multiple occupants, the zone-based clustering algorithm aims to determine the distances between the collected Wi-Fi sessions and partition these Wi-Fi sessions into one or more clusters, each corresponding to one of the stationary occupancy hotspots (zones).

3.3.1 Distance of Wi-Fi Sessions

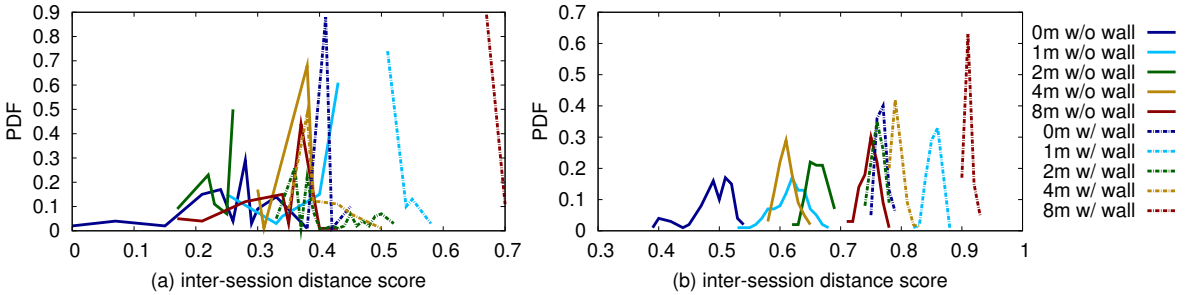


Figure 3.2: Within room or across wall: Distance function comparison between (a) direct RSS-based distance measure and (b) our method.

The primary challenge of automatically identifying indoor stationary occupancy hotspots comes from the high noise of the collected Wi-Fi signal vectors, which is mainly due to i) high noise of indoor Wi-Fi signals from reflection, diffraction, and absorption; and ii) heterogeneity of the occupants' mobile phones. Therefore, directly measuring inter Wi-Fi session distance is error prone (see Figure 3.2(a)).

Our zone-based clustering method consists of a novel de-noising procedure that is applied to the collected Wi-Fi sessions. Given a Wi-Fi session S containing m Wi-Fi signal vectors V_{Sj} ($1 \leq j \leq m$), our de-noising procedure works in three steps.

1) Computing the 2-grams of Wi-Fi session signals, which are robust to phone heterogeneity. Specifically, each V_{Sj} is converted to an $ngram_i(seq(V_{Sj}))$, with $n = 2$, $1 \leq i \leq |seq(V_{Sj})| - n + 1$, $1 \leq j \leq m$, using Equations 3.2 and 3.3.

2) Averaging Wi-Fi RSS values to minimize signal noise. Specifically, for each $ngram_i(seq(V_{Sj}))$, $n = 2$, we define APS_S as the set of all 2-gram AP sequences obtained from S and RSS_S as the corresponding set of RSS differences between the two APs in each 2-gram. Let APS^*_S be the set of **unique** 2-grams in APS_S (note $APS^*_S \subseteq APS_S$), for each unique 2-gram $APS_{Sk} \in APS^*_S$, we identify all occurrences of APS_{Sk} in APS_S and compute the average of their corresponding RSS

difference values. That is, for $1 \leq i \leq |APS_S|$,

$$RSS_{Sk} = avg\{RSS_{Si} | APS_{Si} = APS_{Sk}\}. \tag{3.5}$$

3) Weighting each unique 2-gram APS_{Sk} based on its occurrence rate, with more frequent 2-grams having heavier weights. The **occurrence rate** of each unique access point sequence APS_{Sk} is equal to the total number of occurrences of that sequence in the session S divided by the number of Wi-Fi signal vectors in that session. For $1 \leq i \leq |APS_S|$,

$$OCC_{Sk} = |\{APS_{Si} | APS_{Si} = APS_{Sk}\}| / m. \tag{3.6}$$

Then, the weighted RSS vector for session S is defined as follows:

$$WR(S) = \{APS_{Sk} : OCC_{Sk} \times RSS_{Sk}\} \tag{3.7}$$

for all $APS_{Sk} \in APS_S^*$.

Finally, the distance function, based on Tanimoto distance, for two Wi-Fi sessions S_u and S_v is defined as

$$d(S_u, S_v) = 1 - \frac{WR(S_u) \cdot WR(S_v)}{|WR(S_u)|^2 + |WR(S_v)|^2 - WR(S_u) \cdot WR(S_v)}. \tag{3.8}$$

For each access point sequence APS_k that appears only in $WR(S_u)$, we set the corresponding weighted RSS value to zero in $WR(S_v)$, and vice versa.

Figure 3.2 compares the accuracy of inter-session distance measure using RSS directly [27] (Figure 3.2(a)) with our method (Figure 3.2(b)). In this study, two Wi-Fi sessions are gathered under different physical distances (0 to 8 meters), either within the same room (w/o wall) or in different rooms (w/ wall). It shows that our method can differentiate Wi-Fi sessions in different physical scenarios.

3.3.2 Density-based Wi-Fi Clustering

Given the inter Wi-Fi session distance measure described above, our system incorporates a clustering procedure to partition the collected Wi-Fi sessions into clusters. Each cluster corre-

sponds to one of the stationary occupancy hotspots.

We propose to use density-based clustering [13], which features a well-defined cluster model called "density-reachability". It connects points within a specific distance threshold (Eps) and a specific density criterion: minimum number of points ($MinPts$) within that distance threshold. Setting parameters Eps and $MinPts$ is discussed in Section 6. The rationale behind using density-based clustering is as follows. First, the stationary occupancy hotspots may have arbitrary shapes, which can be properly handled by density-based clustering, but not some other techniques, e.g., k-means clustering. Second, our distance measure may still leave residual noise and density-based clustering can effectively filter out such noise.

We now describe our density-based Wi-Fi session clustering technique. Given a set of Wi-Fi sessions D, we define a Wi-Fi session S_p as **density-reachable** from S_q, if there exists a chain of sessions $S_1, ..., S_n$ ($S_n = S_p$ and $S_1 = S_q$) that for $1 \leq i \leq n-1$,

$$S_{i+1} \in N_{Eps}(S_i) \text{ and } |N_{Eps}(S_i)| \geqslant MinPts, \tag{3.9}$$

where $N_{Eps}(S_i) = \{S_j | S_j \in D \wedge d(S_i, S_j) \leqslant Eps\}$.

Given the specific Eps and $MinPts$, we also define a Wi-Fi session S_p as **density-connected** to S_q if there exists a session S_k such that S_p and S_q are density-reachable from S_k. A **zone** z is defined as a non-empty subset of D satisfying the following conditions:

(1) $\forall S_p, S_q \in D$: if $S_p \in z$ and S_q is density-reachable from S_p, then $S_q \in z$; and

(2) $\forall S_p, S_q \in z$: S_p is density-connected to S_q.

Each zone, which can be in arbitrary shapes, consists of a set of Wi-Fi sessions and is robust to the outliers/noise. In order to identify zones incrementally as more Wi-Fi data are collected, the clustering process is conducted as follows.

Given a newly collected Wi-Fi session S, we first identify a set of Wi-Fi sessions within distance Eps from S, i.e., $D_S = \{S_k | S_k \in D_S \text{ and } d(S, S_k) \leqslant Eps\}$. Then we define a set D_{upd},

which includes Wi-Fi sessions that need to be updated:

$$D_{upd} = \{S_k | S_k \in D_S \cup \{S\} \text{ and } N_{Eps}(S_k)| \geqslant MinPts\}. \tag{3.10}$$

The new session S is categorized as follows:

- If D_{upd} is empty then there is no new cluster formed and no change to the existing clusters after the insertion of session S. Session S is marked as a noise and will be considered later.
- If none of the sessions in D_{upd} belongs to any zone before the insertion of S, then a new zone z is created that contains D_{upd} and S.
- If all the sessions in D_{upd} are members of the same zone z, then session S and possibly some $S_k \in D_{upd}$ are absorbed into zone z.
- If sessions in D_{upd} belongs to multiple zones, then all those zones and session S are merged into one zone.

3.4 Motion-based Clustering

In the motion-based zone clustering phase, we aim to identify zones (stationary occupancy hotspots) existing in the same room, then combine them together as the room's fingerprint. This is a difficult task because occupants may stay at any zone in a room and those zones can be far from, or close to, each other. Our goal is to combine zones in the same room, but not across rooms.

We describe a novel motion based-clustering algorithm based on an important observation – user mobility patterns are different in different rooms within a certain time period, because of the room functions or user habits and activities. For example, users may move more frequently in an office than in a meeting room and two users may have different mobilities in their own offices. Therefore, if two nearby zones are in the same room, they are likely to be surrounded by similar *moving Wi-Fi scans*, which are gathered when users are moving (discussed in Section 5). However, that is less likely to happen for two zones in different rooms. Our motion-based clustering algorithm determines whether two zones, z_a and z_b, are in the same room as follows.

First, we describe a process to select moving Wi-Fi scans for each zone. For a given zone z and K moving Wi-Fi signal vectors V_k ($1 \leq k \leq K$), we want to select m ($m \leq K$) moving Wi-Fi scans as motion profile of z, defined as $\mathbb{V}_z$, that can well represent how users move around near z within a time period. The selection is based on the probability of V_k belonging to z as follows:

$$\prod_{1 \leq i \leq |seq(V_k)|+n-1} P(ngram_i(seq(V_k))|z) > \tau_m. \tag{3.11}$$

We use n-grams of signal vectors because they are robust to noise and produce more accurate probabilities. The calculation of $ngram()$ is based on Equation 3.3. The parameter τ_m determines the closeness of the moving Wi-Fi signal vector V_k to zone z. If τ_m is close to 1, then the moving Wi-Fi scans in $\mathbb{V}_z$ are very close to z. $\mathbb{V}_z$ can only represent motion features of that zone, not the room, and are not helpful for identifying the inter-zone correlation. On the other hand, if τ_m is close to 0, the moving Wi-Fi scans in $\mathbb{V}_z$ surrounding the zone within a large area (e.g., cross multiple rooms) are selected. In that case, $\mathbb{V}_z$ may present motion features of multiple rooms, which may cause the merging of zones in different rooms. In our system, we set

$$\begin{aligned} \tau_m = 0.85 \times \max_k \{ \prod_i P(ngram_i(seq(V_k))|z\} \\ \text{for } 1 \leq k \leq K, 1 \leq i \leq |seq(V_k)| + n - 1, \end{aligned} \tag{3.12}$$

where $\max_k\{\prod_i P(ngram_i(seq(V_k))|z\}$ is the highest probability that a moving Wi-Fi scan V_{max} belongs to z. Namely, V_{max} is gathered at a position that is nearest to z, e.g., when users are leaving or arriving at z. We set τ_m a value slightly lower (0.85 $\times$) than the highest probability in order to get the maximum number of moving scans for z in the same room. Note that higher values of τ_m can result in failure to identify the correlation between two zones far from each other in a large room. In that case, we need to leverage middle-point zones in between or expand the size of the zones to shorten the distance between them.

Second, given two zones z_a and z_b, and corresponding motion profiles $\mathbb{V}_{z_a}$ and $\mathbb{V}_{z_b}$, we calculate the motion profile similarity of z_a and z_b using Jaccard similarity:

$$J_{z_a,z_b} = \frac{\mathbb{V}_{z_a} \cap \mathbb{V}_{z_b}}{\mathbb{V}_{z_a} \cup \mathbb{V}_{z_b}}. \tag{3.13}$$

If J_{z_a,z_b} is close to 1, its indicate that z_a and z_b are most likely in the same room because they are surrounded by almost the same moving Wi-Fi scans. Note that at least one of the zones has none empty $\mathbb{V}$ in calculation of J_{z_a,z_b}.

Calculating the value of J_{z_a,z_b} based on data gathered during a short time period may result in accuracy because the motion patterns in two adjacent rooms may be briefly similar. In response to that problem, the system calculates the value of J_{z_a,z_b} multiple times over different time periods. It then calculates the average value of J_{z_a,z_b} in order to eliminate above noise. Setting threshold τ_J for J_{z_a,z_b} is discussed in evaluation section.

3.5 Wi-Fi Sensing on Mobile Phones

In this section, we present an energy-efficient Wi-Fi sensing technique that automatically and intelligently collects representative stationary Wi-Fi sessions and moving Wi-Fi scans from mobile phones. The Wi-Fi collection process is controlled by two modules: 1) motion detection to determine a user's current status (moving or stationary), and 2) duplication checking to determine if Wi-Fi collection is needed at current location.

3.5.1 User Motion Detection

The key challenge of user motion detection on mobile phones is to achieve high accuracy and energy efficiency. Since a user's motion status can change at any time, the detection process needs to be constantly active. Previous work on motion detection has used either high-frequency (32 Hz) accelerometer readings to accurately detect all types of user activities [39], or medium-frequency (20 Hz) accelerometer readings to detect user's motion status [47]. The average power consumption of accelerometer is 330 mW for 30–50 Hz, 290 mW for 20–25 Hz, and only 50 mW for 4–7 Hz [27]. The first two frequencies result in unacceptably short battery lifespans if used continuously.

We describe a novel motion detection algorithm that can accurately determine if a user is moving or stationary using only 5 Hz acceleration sampling. The algorithm is based on our ob-

servation that when a person is moving (mainly walking), the average of **absolute** acceleration changes (m_{abs}) within a short period (e.g., 3 seconds) is large but the average of acceleration changes (m) is small. This is because when a user is walking, the body is in oscillation. As a result, the sign of acceleration readings also oscillate from positive and negative values, canceling each other in the average acceleration change but not the average absolute acceleration change. However, when user is not walking (e.g., no position change), the above property of acceleration data (m is large and m_{abs} is small) rarely occurs. For example, when a user types on phone, both m and m_{abs} will be small. And when a user picks up phone from desktop, both m and m_{abs} will be large.

Given acceleration readings (X, Y, Z) in a 3-second window with sampling frequency of 5Hz, and X, Y, and Z are acceleration readings in three axis, we first calculate the acceleration changes (i.e., difference of adjacent readings) on each axis, $(\Delta X, \Delta Y, \Delta Z)$. We then calculate the average acceleration change m as

$$m = \|(\text{avg}(\Delta X), \text{avg}(\Delta Y), \text{avg}(\Delta Z))\| \tag{3.14}$$

and the average absolute acceleration change m_{abs} as

$$m_{abs} = \|(\text{avg}(\Delta X_{abs}), \text{avg}(\Delta Y_{abs}), \text{avg}(\Delta Z_{abs}))\| . \tag{3.15}$$

We conducted a set of experiments on 10 users of different gender and age to collect acceleration data. Those experiments include stationary scenarios: completely stationary, playing with phone, and standing with some body movements; and moving scenarios: walking with phone in pocket (both in jacket and pants) and in a bag (on shoulder and in hand), walking on stairway, and with various walking speed. Figure 3.3 shows a scatter plot of m_{abs} and m/m_{abs} values we have obtained in different moving and stationary scenarios. We observe that moving instances are concentrated in the upper left region. The outliers are due to users making turning or pausing. Based on this figure, we determine a user is moving when $\tau_{m_{abs}} \geqslant 300$ and $\tau_{m/m_{abs}} \leqslant 0.15$, otherwise the user is stationary.

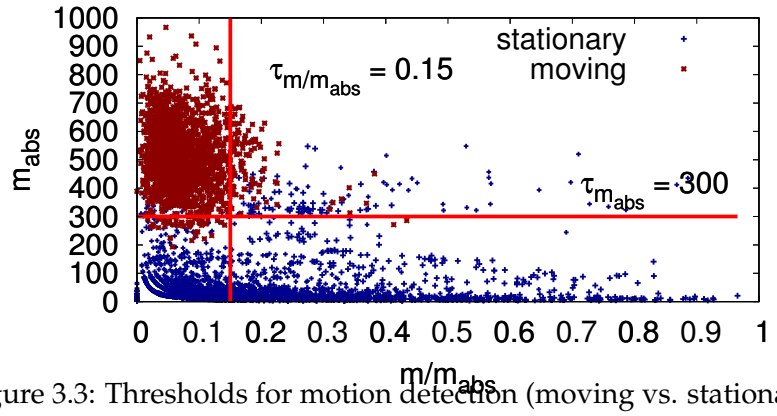


Figure 3.3: Thresholds for motion detection (moving vs. stationary).

Once a user is determined to be stationary, the system starts stationary Wi-Fi session sampling till the user changes to the moving status. When the user is determined to be moving, the system starts moving Wi-Fi scans sampling continuously for 5 minutes or till the user changes to the stationary status.

3.5.2 Wi-Fi Duplication Check

Our automatic Wi-Fi collection mechanism is passive and highly dependent on user motion patterns. Thus, we may collect a lot of redundant Wi-Fi data at places that a user visits often, such as home and office. Such redundant data increase the energy use on phone and computation overhead on server without improving the clustering results. We describe a Wi-Fi duplication check mechanism to stop Wi-Fi collection at a place when sufficient data have been collected.

Our duplication check is performed at the zone level. The server maintains a status for each zone – "full" if the zone contains enough Wi-Fi sessions (we use a threshold of 50) and "not full" otherwise. This status information is distributed to phones periodically. On the phone side, the application first samples one Wi-Fi signal V, and calculates the most likely zone $\hat{z}$ based on Equation 3.4. If $P(ngram_i(seq(V))|\hat{z})$ is larger than a given threshold (0.8 in our system) and zone $\hat{z}$ is full, then the system skips the stationary scan and following moving scans. Otherwise the Wi-Fi data will be collected.

3.6 Evaluations

Our automatic room fingerprinting system has been implemented and deployed for real-world evaluation. This section presents the experimental setup, system accuracy, and efficiency analysis results. We also explore system's parameter settings and discuss the problems encountered during user study.

3.6.1 Experimental Setup

The system is implemented on mobile phones and a server. The zone-based Wi-Fi clustering algorithm and motion-based zone clustering algorithm are implemented in python on the server side. The motion detection algorithm is implemented on the Android mobile platform. In addition, the Wi-Fi signal vector and room fingerprint database is built using MySQL. A web server receives Wi-Fi signal vectors from mobile phones, delivers room localization services, and synchronizes room fingerprints to mobile phones.

Over a period of 10 months, a total number of 21 participants, including faculty members and graduate students, have participated in the user study. Overall, we have collected Wi-Fi data for 193 rooms, in which 85 have at least one adjacent room and 61 rooms have been visited by at least two participants. The user study covered a wide range of building environments, including hospitals, supermarkets, restaurants, a university campus, apartments, and houses.

For evaluation purpose, during the user study, each user was asked to provide related room information, e.g., room name, room entry/departure time, and activity in the room, through a user interface integrated in our mobile application. Weekly meetings with the users were held to verify the accuracy of the user input data. The user data was also verified using collected GPS data (for the rooms in which GPS signals were available) and Wi-Fi data in building level. This data set is used to establish ground truth between each Wi-Fi session and the corresponding room. Each cluster automatically identified by the system is mapped (via majority voting) to the room with most Wi-Fi sessions in the cluster.

3.6.2 Accuracy

This section first evaluates the overall accuracy of the room localization service provided by the system, and then investigates the accuracy of the fingerprinting techniques.

Our system automatically builds room fingerprints through an incremental process using the Wi-Fi signal vector samples gathered by personal mobile phones, and then delivers room localization service back to the individual occupants. Therefore, the accuracy of the room localization service depends on the accuracy of room fingerprints, which in turn is a function of the amount of Wi-Fi signal samples collected for each room. A good room localization system should achieve high accuracy with a reasonable number of Wi-Fi signal samples.

Figure 3.4 shows the accuracy of the room localization service as a function of the number of Wi-Fi signal vector samples. Given the user study including 21 participants and 193 rooms, it shows that, first, when the Wi-Fi samples collected per room are less than 200, the system is unable to generate fingerprints. This is due to the fact that, given the default system setting, the clustering algorithm cannot build up meaningful clusters using such limited number of Wi-Fi samples. Next, as the collected Wi-Fi sample count increases, the accuracy of room localization quickly improves, and the variance of service quality decreases. For 400 or more samples per room, it achieves 95% accuracy, which is comparable to past works that require time-consuming manual annotation [16, 21]. Note that obtaining 400 Wi-Fi samples per room requires collection of 15 Wi-Fi sessions, and each session takes less than two minutes.

Figure 3.5 shows room localization accuracy as a function of numbers of users. It shows that the accuracy of our approach improves as the number of users per room increases. When more users visit the same room, a better spatial coverage in terms Wi-Fi signal samples will be obtained during room fingerprinting. On the other hand, even if room fingerprinting is conducted based on data from a single user, ARIE is still able to localize with over 90% accuracy.

We next evaluate the room fingerprinting techniques, and focus on evaluating the performance of the clustering algorithms. Specifically, our approach uses clustering techniques to con-

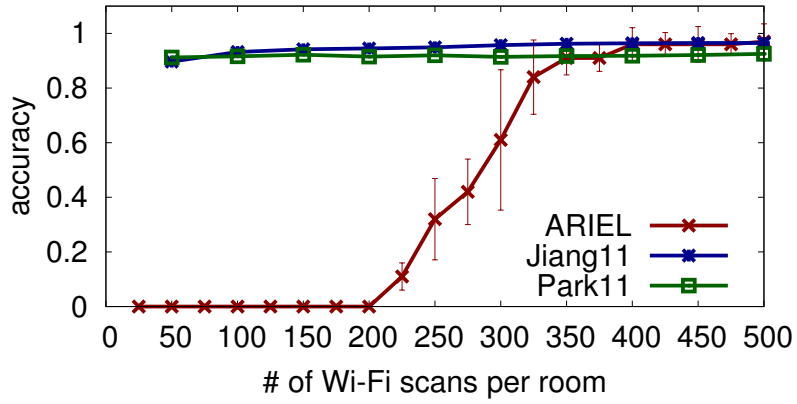


Figure 3.4: Room localization accuracy vs. number of Wi-Fi sessions.

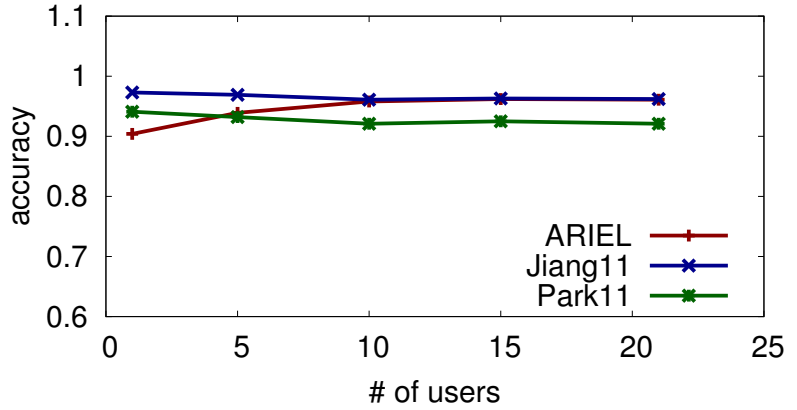


Figure 3.5: Room localization accuracy vs. number of users.

struct room-level clusters consisting of Wi-Fi sessions belonging to each room. Two potential errors introduced by the clustering methods include 1) Wi-Fi sessions collected in room A are incorrectly assigned to a cluster belonging to room B and 2) the clustering algorithm is unable to merge all the zones belonging to the same room into one cluster, resulting in multiple clusters per room. To this end, we introduce the following two performance metrics.

1. Purity measures the percentage of Wi-Fi sessions with correct room assignment divided by the total number of assigned Wi-Fi sessions.

2. Unity measures the quality of room-level zone merging, i.e., it is the reciprocal of the total number of zones assigned to each room, a average over all the rooms. The unity of one is ideal.

Figure 3.6 shows the accuracy of the clustering methods as a function of number of Wi-Fi sessions. Our system starts to produce clusters when the number of collected Wi-Fi sessions is greater than 200. The unity measure improves as the number of Wi-Fi sessions increases, because more Wi-Fi session collections form more connections between in-room stationary hotspots, thus improving clustering quality. The value of unity converges when the total number of Wi-Fi sessions reaches 400. The purity measure slightly degrades as the number of Wi-Fi sessions increases, which introduces more Wi-Fi variations.

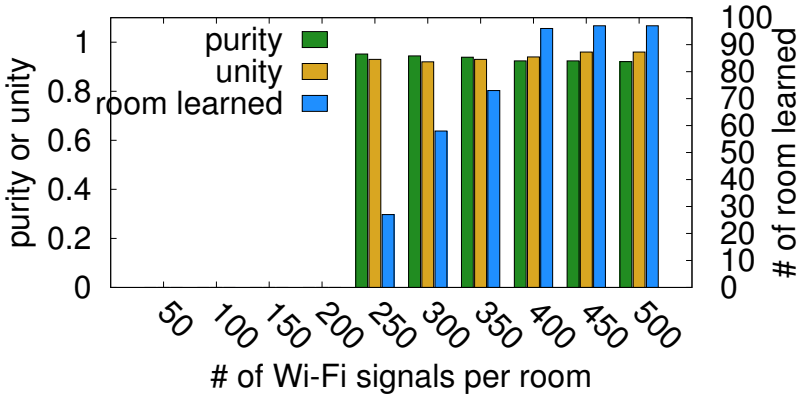


Figure 3.6: Clustering accuracy vs. number of Wi-Fi sessions.

Figure 3.7 shows the clustering accuracy as a function of the number of users. When the system has more users, the purity of the clusters slightly degrades as more variations are introduced by different users and different phones. Meanwhile, the unity of the clusters improves as different motion patterns of users within a room help bridge in-room hotspots, thus improving the clustering quality.

Overall, this study shows that our approach offers high clustering quality.

3.6.3 Energy Efficiency

The energy overhead imposed by the system on the mobile phone has two main parts: Wi-Fi sensing and motion detection. We calculate the average power consumption based on power measurement data from Android HTC G1 [1]. The average power consumption on reading accel-

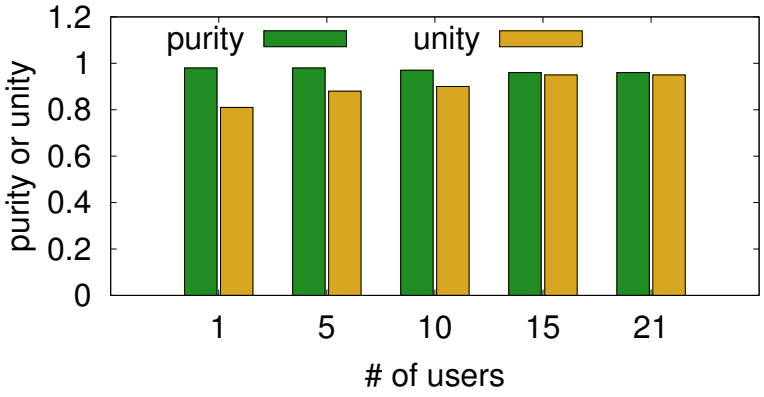


Figure 3.7: Clustering accuracy vs. number of users.

eration at 4–7Hz is 50mW, at 20–25Hz is 285mW, and at 30–50Hz is 325mW. The average power consumption of Wi-Fi scan at 1/5Hz is 108mW.

We first compare our motion detection method with two other approaches: 1) frequent acceleration sampling [39], which targets general user activity recognition including stationary and walking and 2) moderate acceleration rate sampling [47], which focuses on motion detection. As shown in Figure 3.8, given the same data set, our detection algorithm achieves the highest energy efficiency, nearly 5× better than that of the other two approaches. In addition, the average motion detection accuracy of our approach is 94%, which is higher than the medium-frequency approach and slightly lower than the high-frequency approach.

We also study the power consumption of Wi-Fi sensing. We found that the active time duration of Wi-Fi sensing varies among users. On average, the Wi-Fi active time over a day ranges from 20 minutes to 2 hours with an average sampling frequency of 1/5Hz. The corresponding average power consumption is 4.5mW.

3.6.4 Parameter Settings

Several parameters in the system need to be determined: 1) density parameters Eps and $MinPts$, 2) threshold τ_J for motion-based clustering, and 3) size of Wi-Fi stationary session (number of Wi-Fi scans in a session). These three parameters directly impact the purity and unity of the

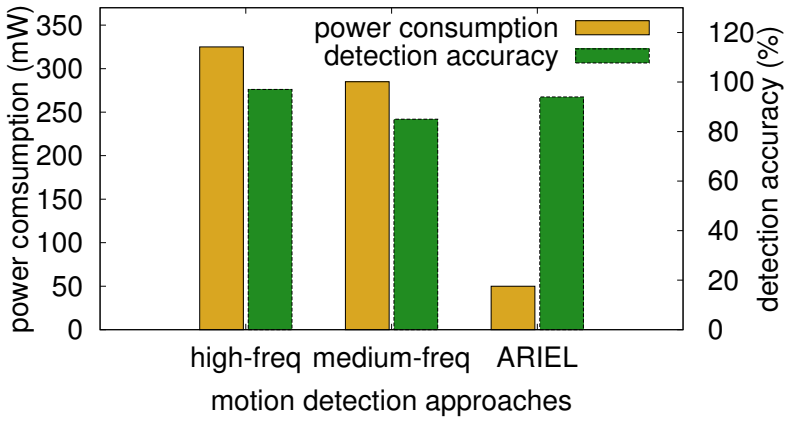


Figure 3.8: Comparison of power consumption and detection accuracy of different motion detection methods.

room fingerprints.

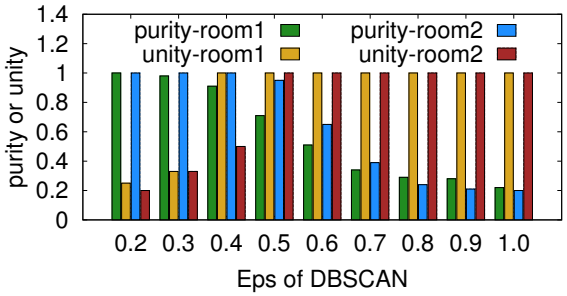


Figure 3.9: Impact of different Eps values in two different rooms.

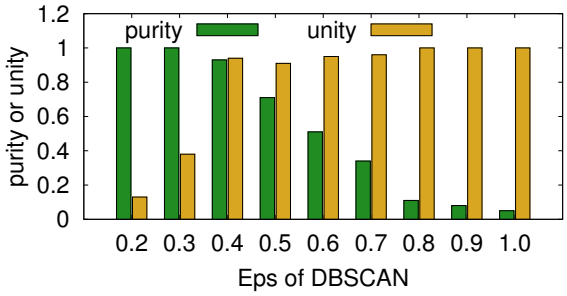


Figure 3.10: Average clustering accuracy as a function of Eps.

In our density-based clustering algorithm, we set $MinPts$ to 5 for all data because 1) in our experiments, we found no difference among clustering results when $MinPts \geq 5$; and 2) larger $MinPts$ requires more computation effort and more Wi-Fi sessions per room. Figure 3.9 shows the purity and unity of two rooms with different Eps values. The two rooms have different wall types: room 1 shares a thin drywall with adjacent rooms, while room 2 shares a thick concrete wall with adjacent rooms. Generally, larger Eps can increase the unity of cluster but reduce the purity, vice versa. The two rooms with different wall types have different clustering results. Without prior knowledge of the wall type, it is difficult to choose the right value for Eps. As shown in

Figure 3.10, the average purity and unity of all data change with the value of *Eps*. We set *Eps* conservatively to 0.32, which ensures high purity for all the clusters but relatively low unity, i.e., a room may have multiple clusters but each cluster has high purity. The low unity of the clusters will be compensated for in the merging phase.

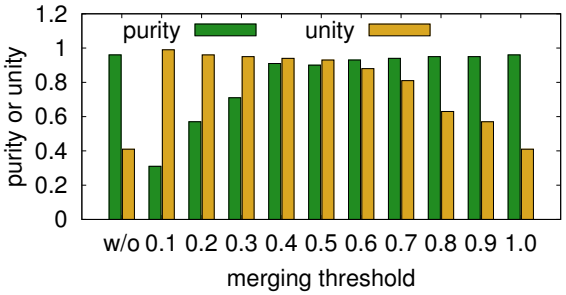


Figure 3.11: Average clustering accuracy as a function of merging threshold τ_J.

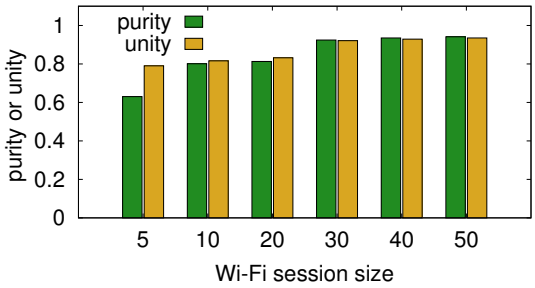


Figure 3.12: Average room localization accuracy vs. session size.

Another important parameter in our system is the zone merging threshold τ_J introduced in Section 4. As shown in Figure 3.11, if we set the threshold too low, the purity decreases a lot. On the other hand, if we set the threshold too high, clusters in the same room may not be merged. In the tests, we empirically set τ_J to 0.5.

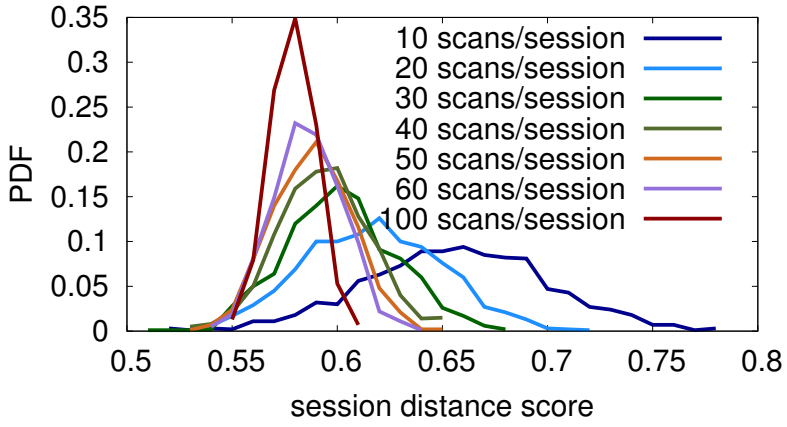


Figure 3.13: Session distance distribution vs. session size.

Session size also impacts system performance. A session with more Wi-Fi scans gives more

accurate distances, but requires more energy for Wi-Fi scan. Figure 3.13 shows that, as session size increases, the distance between two sessions become more stable. We evaluated our system with different session sizes; the results are shown in Figure 3.12. When session size is larger than 30, the room purity and unity do not change much. Although the session distance is not perfect with a size of 30 as shown in Figure 3.13, our density-based clustering algorithm is robust to outliers and achieves high accuracy even with some noise.

3.7 Summary

In this chapter, We present a room localization system that automatically learns room fingerprints based on occupants' indoor movements. The system consists of (1) a zone-based clustering algorithm that accurately identifies in-room occupancy "hotspot(s)" using Wi-Fi signatures; (2) a motion-based clustering algorithm to identify inter-zone correlation, thereby distinguishing different rooms; and (3) an energy-efficient motion detection algorithm to minimize the noise of Wi-Fi signatures. The system has been implemented and deployed for real-world testing with 21 users over a 10-month period. Our studies show that it supports room localization with higher than 95% accuracy without requiring labor-intensive manual annotation.

Chapter 4

Indoor Floorplan Construction

This chapter present an automatic floorplan construction approach targeting indoor environments. The proposed solution uses on-site Wi-Fi infrastructure. It also uses information on the motion of occupants gathered via crowd sourcing. Our analysis and optimization techniques determine the identities and geometries of individual rooms, as well as room adjacency information, and then construct an indoor floorplan through hallway layout learning and force directed dilation. The system functions for a variety of indoor environments, and is robust to Wi-Fi noise and variation in occupant motion patterns. Deployment at three building sites with 103 rooms yields an average room position accuracy of 90%, room area estimation error of 31% and room geometric aspect ratio error of 19%, using on average only 20 data points per location for the system to converge.

4.1 Related Work

In this section, we survey related work spanning the following areas: Wi-Fi fingerprint based indoor localization, indoor mapping techniques, and user behavior sensing techniques.

Wi-Fi signals have been widely used for indoor localization. The basic approach is to construct reference Wi-Fi fingerprints from a training set of Wi-Fi signal measurements and use them to match new/unknown Wi-Fi fingerprints at unknown positions based on proximity or similarity of the fingerprints. A number of techniques have been proposed to construct Wi-Fi fingerprints using the occurrence, received signal strength (RSS), or subsequence of access points (APs), and

apply triangulation, k-nearest-neighbor (KNN), or probabilistic techniques to estimate indoor locations [6, 18, 21, 28]. Most of those approaches either assume the floorplan is available or use room names for user interaction. However, in real-world applications, the floorplan is unavailable most of the time and creating floorplans is a nontrivial task. Using room names works fine when the number of rooms is small. However, for a large number of rooms, it becomes difficult for users to manage all the room names. Moreover, unlike a floorplan, using room names cannot provide the relative positions and room dimensions information to users, which are crucial in many applications.

Simultaneous Localization and Mapping (SLAM) is a technique to build a map in an unknown environment, while simultaneously tracking the current location [11]. This technique is widely used in robotics. However, the map constructed in SLAM consists of a set of significant points, instead of floorplan. SmartSLAM applies the SLAM approach on the smartphones [44]. It utilizes the Wi-Fi signals collected when users are walking to generate maps of a building. However, SmartSLAM focuses on just hallway layout instead of the floorplan of the whole building. For example, it cannot derive the room dimension and position information. Alzantot et al. have proposed a floorplan construction approach based on user motion traces in the building [2]. Their technique requires user traces covering the whole room to learn the shape of each room. Moreover, to accurately track users' motion pattern, their technique relies upon landmarks (e.g., elevator and stairway), which are not always available for all buildings.

There is a rich literature on user behavior sensing technologies [30, 29, 22]. In this work, we implemented the walking detection technology proposed by Jiang et al. [22] because it has good energy efficiency. Turn detection has been studied in previous work [29] using accelerometer and compass on smartphones. However, their technique is targeted for outdoor scenarios and thus does not consider the impact of indoor environment on the accuracy of compass.

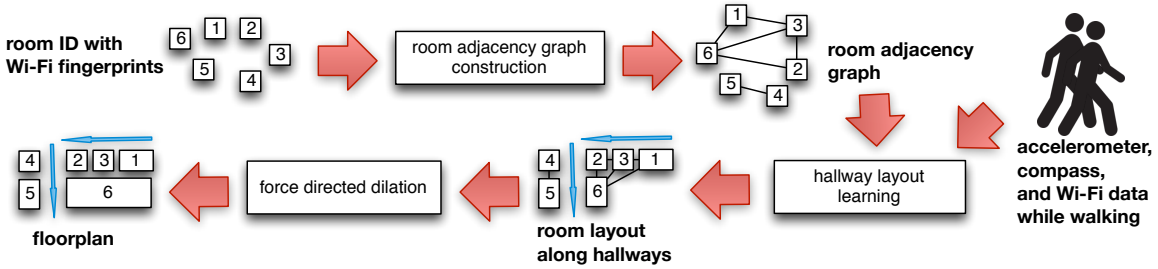


Figure 4.1: Automatic indoor floorplan construction system overview.

4.2 System Overview

This section formulates the problem of indoor floorplan construction, and provides an overview of the proposed system.

A floorplan graphically displays the dimensions and positions of rooms inside a building. A floorplan includes three types of information: the number of rooms, dimensions of rooms, and relative position of rooms. In this work, we focus on rectangular-shaped rooms typically seen in most buildings. Each room can be represented by the coordinates of its center point (e.g., x and y values on a 2-D plane), along with its width and length information. The indoor floorplan construction problem can be defined as follows: Given a room set R containing n distinct rooms on the same floor in a building, determine the relative center coordinates, width, and length for each room $r_i (1 \leq i \leq n, r_i \in R)$ in a two-dimensional space.

In our system, the input data includes room Wi-Fi fingerprint r_i for each room, passively collected motion sensor data (accelerometer and compass), and Wi-Fi scan data when users are walking in the building. A number of approaches exist for collecting room Wi-Fi fingerprints, such as manual room fingerprint collection [18], leveraging user feedback [31, 43] , and automated learning techniques [22]. Our work leverages the outputs of the aforementioned approaches and we assume that each room has been associated with a unique room ID and the corresponding Wi-Fi fingerprint. Note that the room IDs are used for identification and are necessarily related to real room names or numbers.

When users are walking, the motion sensor data and Wi-Fi scan data are collected by our system running on users' smartphones. The accelerometer data are collected at a 5 Hz frequency and used to determine if a user is walking. When a user is walking, the compass and Wi-Fi data are collected at 5 Hz and 1 Hz frequencies, respectively. Note that this system only needs to be run in the training phase. Once the floorplan is constructed, the system no longer needs to be active, and the produced floorplan will be provided to other indoor mobile applications and services, thus saving smartphone energy.

Figure 4.1 illustrates the overall process of our indoor floorplan construction solution. The system consists of three main components: (1) room adjacency graph construction, (2) hallway layout learning, and (3) force directed dilation.

Room Adjacency Graph Construction. This component identifies adjacencies among rooms and constructs a room adjacency graph. In a room adjacency graph, the vertices represent individual rooms and the edges between vertices indicate adjacency between two rooms. We used k-means clustering of the Wi-Fi signals collected at the corners of rooms, based on the observation that if two rooms are adjacent, the Wi-Fi signals measured at the corners of the rooms are most likely grouped into the same cluster. This approach is robust to Wi-Fi noise and unevenly distributed Wi-Fi scans within a room.

Hallway Layout Learning. Based on the room adjacency graph and users' walking traces in hallways, the layout learning algorithm first identifies rooms and their orders along each hallway, and then assembles the hallways based on the similarity score of Wi-Fi signals. This component includes a novel and accurate indoor turn detection algorithm, which leverages compass sensor data and a noise-robust room layout detection algorithm.

Force Directed Dilation. This step optimizes the overall floorplan accuracy via global adjustment of the estimated room dimensions. We use a force directed dilation algorithm, in which the learned floorplan is defined as a mechanical system with springs between adjacent rooms, and the dimensions of rooms are automatically adjusted based on the forces among rooms.

4.3 Room Adjacency Graph Construction

Room adjacency graph construction is a process to determine whether each pair of rooms is adjacent for a given set of rooms, R. The adjacency information is represented using an undirected graph. For example, if two rooms are adjacent, the corresponding nodes in graph are connected by an edge as shown in Figure 4.1. The adjacency graph is a fundamental information for floorplan construction, and is used later in hallway layout learning and force directed dilation.

To determine the adjacency of two rooms, a straightforward approach is to compare the similarity of their fingerprints. However, such similarity can be affected by the relative positions of two rooms, the type of wall between rooms, and the distribution of fingerprints within rooms. Yet, Wi-Fi signal noise increases the uncertainty. Moreover, this approach requires the definition of a global threshold applicable to diverse room conditions, which is very difficult to derive.

In this work, we propose a clustering based approach to determine the relationship between pairs of rooms. It is based on the observation that Wi-Fi signals in adjacent rooms are similar, especially for those signals collected near the partition wall. Even though wall can decrease Wi-Fi signal strength, allowing differentiation of adjacent rooms, noise can blur the boundary between two rooms. On the other hand, signals collected near the wall can be differentiated by extreme RSS (received signal strength) values. The extreme RSS values for an AP (access point) are close to the minimum or maximum RSS value of that AP. Those signals are most likely at the edges (corners) of rooms. By leveraging such Wi-Fi signals, we can cluster adjacent rooms into the same group.

Based on the above observations, for each room fingerprint (containing a set of Wi-Fi scans collected in that room), we use the following procedure to remove Wi-Fi scans in which APs contain few extreme RSS values. First, for each access point ap_i that can be observed in a room r, we extract its RSS value from the room fingerprint. Second, we normalize the RSS values using min-max normalization to the range of $[0, 1]$. Given a Wi-Fi scan w, where $w = \{rss_{ap_1}, rss_{ap_2}, ..., rss_{ap_m}\}$,

we calculate the average normalized RSS value nr_w as follows:

$$nr_w = \sum_{i=1}^{m} (rss_{ap_i})/m \qquad (ap_i \in w). \tag{4.1}$$

We remove all Wi-Fi scans whose nr_w values satisfy $0.2 \leq nr_w \leq 0.8$, and only keep Wi-Fi scans with more extreme RSS values.

After removing those Wi-Fi scans, we run the k-means clustering algorithm on the remaining Wi-Fi scans from all rooms. Based on our evaluation results, we set k to 4 times the total number of rooms. For each cluster, we calculate the total number of Wi-Fi scans n_{scan}, the number of unique rooms n_{room}, the number of Wi-Fi scans for each room $n_{scan/room}$, and the ratio n_{scan}/n_{room} as the minimum number of Wi-Fi scans for each room. If there are two rooms in one cluster that each satisfies $n_{scan/room} \geq n_{scan}/n_{room}$, these two rooms are detected as adjacent rooms.

4.4 Hallway Layout Learning

A hallway is defined as a straight path in a building that is adjacent to rooms. The end of a hallway can be a wall or an outlet. When a user walks through a straight hallway, he/she will pass all the rooms along the hallway sequentially. This indicates how rooms are arranged along the hallway.

Rooms vary by size and pairs of rooms can be positioned in many possible orientations. Based on these observations, our layout learning technique includes three sub-components: (1) **hallway motion detection**, which detects walking in straight hallways; (2) **layout learning along hallway**, which identifies the lengths of rooms and sequences along the hallway; and (3) **hallway assembling**, which is based on room overlap information on intersecting hallways.

4.4.1 Hallway Motion Detection

The goal of hallway motion detection is to identify user activities, such as walking straight and making a turn. Figure 4.2 shows a path that starts inside a room and crosses two hallways. Our detection algorithm first distinguishes traversing hallways (green and blue lines) from walk-

ing within rooms (red line) by leveraging Wi-Fi signals while walking and room fingerprints. It then identifies activities in different hallways via turn detection.

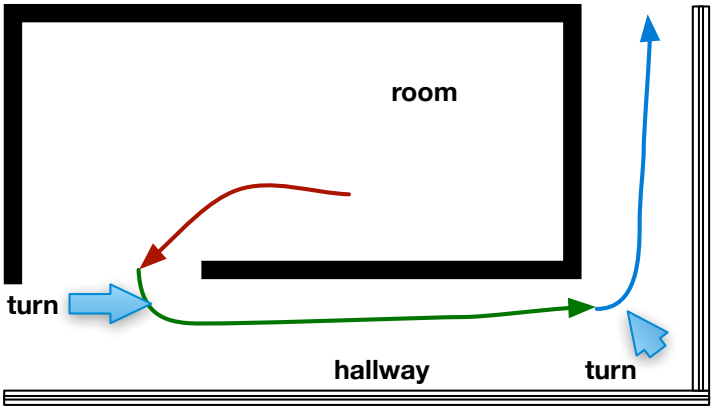


Figure 4.2: Hallway walking activities.

In order to detect walking activities, we use the technique proposed by Jiang et al.[22], which has been shown to achieve good accuracy and energy efficiency. Their technique uses accelerometer data with 5 Hz sampling rate to compute two parameters, i.e., the average absolute acceleration change m_{abs} and the average acceleration change m for every 3 seconds. If $m_{abs} \geqslant 300$ and $m/m_{abs} \leqslant 0.15$, the user is detected as walking. In our system, the walking detection function is running continuously on users' smartphones until the floorplan is constructed.

4.4.1.1 Hallway Traversal Detection

In order to determine whether a user is walking in hallway, our system periodically scans Wi-Fi signals at around 1 Hz. When a user is moving, the Wi-Fi RSS for access points change. However, as obstacles between phone and access points mainly affect Wi-Fi signal strengths, Wi-Fi signals within a room are relative stable (even for moving phones) compared to those during hallway traversal.

Based on the observation, we determine whether a moving user is in a hallway or room by

comparing two consecutive Wi-Fi scans w_a and w_b. Due to Wi-Fi signal noise, the similarity score between two single Wi-Fi scans is quite unpredictable. In order to deal with this challenge, we first convert the single Wi-Fi scan w_i to a set of similarity scores d_{ij} for each room $r_j (r_j \in R)$. d_{ij} is defined as [21]

$$d_{ij} = \prod_k P(ngram_k(w_i)|r_j)P(r_j), \tag{4.2}$$

where

$$ngram_k(w_i) = (ap_k, ap_{k+1}). \tag{4.3}$$

That is, ap_k and ap_{k+1} are two APs in Wi-Fi scan w_i and are consecutive when ordered by RSS values. $P(r_j)$ is the probability of room r_j appearing in the system. We set all $P(r_j)$ equal to 1.0 since rooms are treated equally in our system.

The Wi-Fi scan w_i is then converted to a vector $D_{w_i} = d_{i1}, ..., d_{in}$, where n is the total number of rooms in R. The similarity computation between w_a and w_b is converted to the similarity computation between D_{w_a} and D_{w_b}. This conversion makes the similarity more reliable because d_{ij} is calculated using room fingerprint, which contains a set of Wi-Fi scans.

The similarity function between D_{w_a} and D_{w_b} is defined by Tanimoto Distance:

$$\begin{aligned} & simi(D_{w_a}, D_{w_b}) \\ = & \frac{D(w_a) \cdot D(w_b)}{|D(w_a)|^2 + |D(w_v)|^2 - D(w_a) \cdot D(w_b)}. \end{aligned} \tag{4.4}$$

We set a threshold τ. If $simi(D_{w_a}, D_{w_b}) \leq \tau$, this suggests motion within a room. $simi(D_{w_a}, D_{w_b}) > \tau$ suggests motion within a hallway. Based on our experimental data, we set $\tau = 0.3$.

4.4.1.2 Hallway Turn Detection

Compass-based turn detection [47] has been well studied in outdoor environments. However, in indoor environments, the accuracy of digital compasses on smartphones can be significantly affected by electrical cables and devices, construction materials, and other metallic objects nearby.

To deal with these challenges, we propose a technique combining accelerometer, compass, and Wi-Fi signals to detect turns in hallways. This is accomplished in three key steps: (1) coarse-grained turning point detection using accelerometer-based and compass-based turn detection; (2) clustering of turning points using Wi-Fi signals; and (3) fine-grained turning point detection when a user is close to a cluster of turning points. The intuition is that, we want to identify stable turning point clusters and utilize fined-grained (i.e., more sensitive) turning point detection only when a user is close to a turning point cluster. Next, we describe the three steps in detail.

Step 1: **Coarse-grained turning point detection**. Here, a turning point is detected whenever either accelerometer or compass detects a turn. Accelerometer-based turn detection is inspired by the walking detection approach [22], which assumes that when a user is walking, his body is in oscillation. However, when the user is turning, the oscillation is broken. If $m/m_{abs} \geqslant 0.15$ and $m_{abs} \geqslant 300$, we consider the user to be turning. Compass-based turn detection uses a 3 second long sliding window and calculates the mean value for each window. If the change of two consecutive values is larger than $\tau_{compass}$ (set to 30 in this work), the user is determined to be turning.

Step 2: **Clustering of turning points**. The individual turning points detected using either accelerometer or compass can be noisy. Given the physical constrains of hallways, users' turning points should be centered around certain regions such as the end of a hallway or hallway intersection. To identify these regions where turning is likely to occur, we leverage Wi-Fi scans collected at individual turning points and use density-based clustering to group Wi-Fi scans into clusters representing these turning regions. The similarity of two Wi-Fi scans is computed based on RSS value using Euqation 4.4. The two parameters for density clustering algorithm, Eps and $MinPts$, are set to 0.8 and 15 respectively. Note that although the similarity of two single Wi-Fi scans are noisy, the density based clustering algorithm is robust to such noise and can produce stable Wi-Fi based turning regions.

Step 3: **Fine-grained turning point detection**. Given the stable turning regions, we can determine whether a user is close to any of these regions by calculating the similarity (Equation 4.2)

between the current Wi-Fi scan and each of the Wi-Fi clusters produced in Step 2. If the user is close to a turning region, we lower the thresholds of accelerometer- and compass-based turning point detection. Specifically, m/m_{abs} is set to 0.11 and $\tau_{compass}$ is set to 20. The lower thresholds make turning point detection more sensitive near the turning regions. If both accelerometer- and compass-based approaches detect turning activity, the user is classified as turning.

4.4.2 Layout Learning Along Hallway

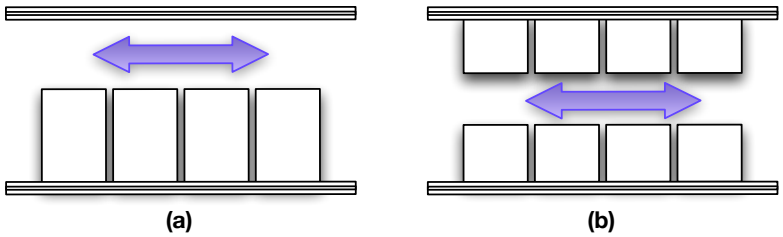


Figure 4.3: Room layout along hallway: (a) Rooms on one-side only and (b) rooms on both sides.

Given a specific hallways, there are two different types of room layouts along this hallway (Figure 4.3): (a) rooms on one side of the hallway only and (b) rooms on both sides of hallway. Not only do we need to identify each hallway and the rooms along the hallway, we also need to determine the rooms' orders and lengths along the hallway.

The hallway motion detection algorithm identifies the Wi-Fi scan sequences collected in a particular hallway. Those sequences may cover a whole hallway or segments of it. The hallway identification algorithm then clusters those Wi-Fi sequences into groups so that each group represents a unique hallway. All Wi-Fi sequences in one group are collected in the same hallway.

Given two Wi-Fi sequence s_a and s_b, where
$s_a = \{w_{a1}, w_{a2}, \ldots, w_{ai}\}$ and $s_b = \{w_{b1}, w_{b2}, \ldots, w_{bj}\}$ (each w is a single Wi-Fi scan), a straightforward method is to compare Wi-Fi signals in s_a and s_b directly to determine the similarity of the two sequences. However, as we mentioned before, Wi-Fi signals are noisy in indoor environments. Therefore, the similarity measurement for two Wi-Fi scans can be very unreliable. In order

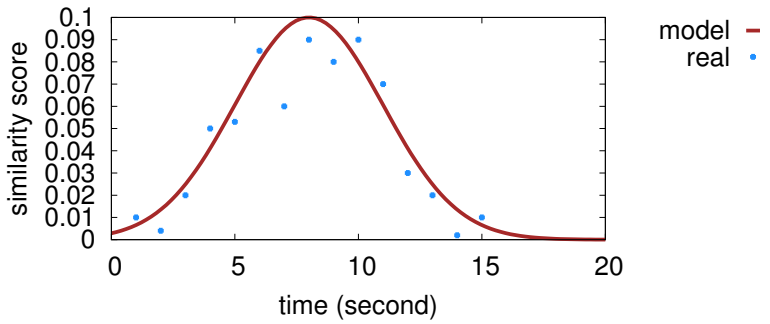


Figure 4.4: Hallway room similarity fitting.

to reduce the uncertainty, we map each w to the room r_{s1} whose room fingerprint is most similar to signal w. Again, the similarity is calculated based on Equation 4.2. If the second most similar room r_{s2} has a similarity score that is similar to that of room r_{s1} ($<$ 20% difference), then w will be mapped to both rooms. Whether w maps to one or two rooms corresponds to one of the two layouts shown in Figure 4.3, i.e., one-sided or two-sided rooms.

After removing duplicate rooms in each sequence, we get two room sets $s'_a = \{r_{a1}, r_{a2}, \ldots, r_{ai'}\}$ and $s'_b = \{r_{b1}, r_{b2}, \ldots, r_{bj'}\}$. Each Wi-Fi sequence is mapped to rooms along the hallway or hallway segment it is collected in. Using rooms to represent a hallway for clustering is much more robust than Wi-Fi signals because (1) the similarly score for a Wi-Fi scan and room fingerprint is more accurate than that of two single Wi-Fi scans and (2) even if w is mapped to a room r that is not physically closest to the location where w is collected, it is likely to be in the same hallway.

After the conversion, we iteratively merge each pair of s'_a and s'_b with the following two steps: (1) if $s'_a \subset s'_b$, we merge the two sets and cluster s_a and s_b into the same group and (2) if s'_a, s'_b, and s'_c are not subset of each other, and $s'_c \subset (s'_a \cup s'_b)$, we merge s'_a, s'_b, and s'_c, and cluster them into the same group. The algorithm stops when no two sets can be merged.

When the iteration is finished, for each hallway h, we get a room set s'_h and a group of Wi-Fi scan sequences S_h. To identify room order and length of rooms along the hallway, we first group rooms on the same side of hallway together based on the room adjacency graph – if rooms are

on the same side of the hallway, their shortest path length to the group equals 1. Then for each side of the hallway, the following calculation is performed. For each Wi-Fi scan sequence $s_h \in S_h$, $s_h = \{w_1, \ldots, w_j\}$, we calculate the similarity between w_i $(1 \leq i \leq j)$ and each room r along the hallway based on Equation 4.2. Then we can get a time series shown as real dots (blue) in Figure 4.4, where x is the time point at which w_i was collected and y is the Wi-Fi similarity score. The time series tend to follow Gaussian distribution, and fitting the time series with a Gaussian model can help reduce noise in the Wi-Fi similarity calculation.

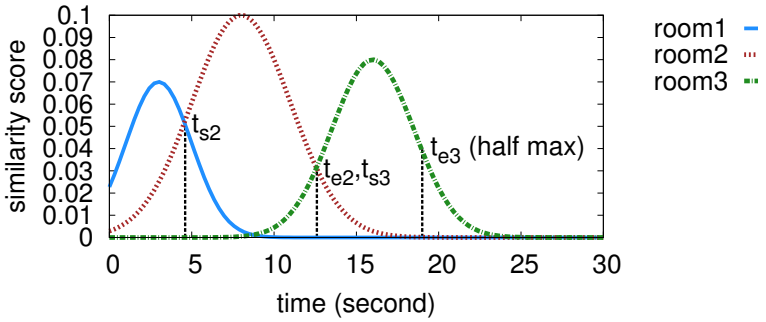


Figure 4.5: Hallway room order and size.

We repeat this process for all rooms on the same side of the hallway, producing results as shown in Figure 4.5. We compute the orders of rooms by compare the timestamps at the means of the Gaussian distributions. The time duration for crossing each room is also computed using intersection of adjacent distributions. For example, in Figure 4.5, the start time t_{s2} of crossing room 2 is the intersection of room1 (blue) and room2 (red) curves and the end time t_{e2} is the intersection of room2 (red) and room3 (green) curves. The time duration of crossing room 2 is thus $t_{s2} - t_{e2}$. The room length along the hallway is computed by multiplying the duration with average walking speed, which is set to be 1.4 m/s. The final room length is the average of room lengths computed from all s_h in S_h.

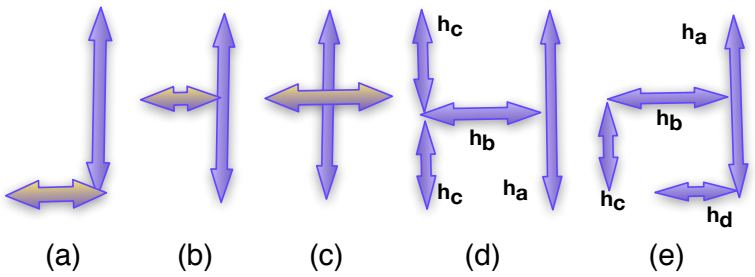


Figure 4.6: Hallway assembling: Connecting multiple hallways.

4.4.3 Hallway Assembling

So far, we have a set of hallways H and the information of room order and lengths along each hallway h ($h \in H$). Hallway assembling aims to identify connecting and/or intersecting hallways. The assembling algorithm starts with two hallways h_a and h_b ($h_a, h_b \in H$). If h_a and h_b share common rooms, they are assembled as one of the structures in Figure 4.6 (a), (b), and (c). The choice of structure is determined by the positions of the shared rooms in each hallway. For example, if the shared rooms are at one end of both hallways, the two hallways are assembled as the structure in Figure 4.6(a). It should be noted that we assume the angle of two intersecting hallways is always 90°. This assumption holds in most cases and simplifies algorithm design, although we admit that relaxing it would improve generality.

Next, we pick another hallway h_c that intersects with h_a or h_b. The constraint of hallway intersection alone is insufficient for the placement of h_c. For example, in Figure 4.6(d), if h_c intersects with h_b at one end, h_c can be placed either above h_b or below h_b. In order to determine that, we select a room r_{h_c} from the side that does not intersect with h_b and another two rooms r_{above} and r_{below} which are above and below h_b, e.g., two rooms at each side of h_a. If the similarity between fingerprint r_{h_c} and r_{above} is greater than that of r_{h_c} and r_{below}, h_c is placed above h_b. Otherwise, it is placed below h_b. We define the similarity of two room fingerprints to be the average similarity of each Wi-Fi scan in one room to the fingerprint of the other room based on Equation 4.2.

Figure 4.6(e) shows one more scenario that needs to be addressed in hallway assembling.

Here, h_d intersects with h_a and h_c, but the length of h_d is shorter than the distance already learned between h_c and h_d. In that case, we dilate hallway h_d to satisfy the conditions, and all rooms along hallway h_d are dilated by the same proportion.

4.5 Force Directed Dilation

Our hallway layout learning algorithm is able to estimate the room dimensions along each hallway (i.e., room length) but not the width (or depth) of these rooms. To estimate the perpendicular dimensions, e.g., room 105 and 150 shown in Figure 4.11(c), another technique is required.

One might consider using user motion within rooms to estimate room dimensions [2]. However, this approach can be unreliable because it requires that users move to all edges and corners of the room. In reality, users are generally concentrated within part of a room. Thus, we propose to use a force directed dilation technique to estimate and automatically adjust the sizes of rooms.

Force directed algorithms [49, 12] have been applied to the problem area of graph drawing. The idea is to place graph nodes in 2D or 3D space with as few crossing edges as possible. The algorithm assigns an attractive and repulsive force to each edge in a graph. The values of the two forces are determined by the distances between the nodes. Once the forces have been defined, the algorithms gradually adjust the positions of nodes until each node experiences net zero force.

Our floorplan optimization algorithm responds to attractive F_s and repulsive F_r forces, between rooms. As shown in Figure 4.7, if two rooms are connected in the graph, we assign a spring system between their centers. The force F_s exerted by a spring is defined as [12]. Note that F_s also exists between hallways and rooms.

$$F_s(d) = c_1 * log(d/c_2), \tag{4.5}$$

where d is the length of the spring, and c_1 and c_2 are constants. As shown in Figure 4.7(a), each room receives the same amount of force F_s but in opposite directions. We define F_r as the force generated by walls between two adjacent rooms. Thus, if walls are not against each other, as shown in Figure 4.7(a), the force $F_r = 0$. Otherwise, F_r is equal to the opposite force on the wall,

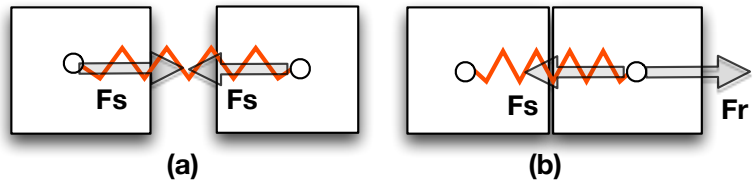


Figure 4.7: Spring system and the forces.

e.g., $F_r = -F_s$ in Figure 4.7(b).

Having defined the attractive force F_s and repulsive force F_r, we use the force directed dilation algorithm, as sketched in Algorithm 1, to determine the dimensions of the rooms. The inputs of the algorithm are the graph G and direction set D. G is learned in the first two components and nodes have been placed as learned/individual rooms. D represents the direction of force and consists of four directions: left, right, up, and down. A force F towards any other direction will be decomposed into these four directions.

The equilibrium state of graph is defined as follows: (1) the net forces F on all vertices are zero or (2) no vertex in the graph can be dilated. If the graph is not in equilibrium state, we compute the sum of forces for each vertex F_v and decompose it to each direction in D as F_{vd}. Then we find the largest force F_{vd} and dilate vertex v in direction d. Note that if there are multiple vertices with the same largest force F_{vd}, all those vertexes will be dilated. The algorithm repeats the above steps until the graph is in an equilibrium state.

The reason to dilate a room with the largest force F_{vd} is that when a vertex has a larger force, it means that the room has less similarity to the actual shape in the floorplan. For example, in Figure 4.11(c), room 150 has the largest force on the left direction, which is generated by the attractive forces from room 105, 116 and 137. The large force is caused by the deviation of room 105's currently assigned size from its actual size.

We assign higher priorities to rooms like 150 and 105 because (1) adjusting those rooms will make the learned floorplan look more like the real one and (2) avoiding block among rooms while

Algorithm 1 Force-Directed-Dilation(**input** G, D)

1: **input** G; ▷ graph after hallway learning & assembling
2: **input** D; ▷ directions set: left, right, up, and down
3: **while** G is not in equilibrium state **do**
4: Compute F_{vd} on each v in d ▷ $v \in G$ and $d \in D$
5: $F_{v_j d_k}$ = Max(F_{vd})
6: Stretch v_j in direction d_k with Δ meters
7: **end while**

dilation. For example, room 116 in Figure 4.11(c) also receives force in the down direction. If we change it first, it will block the dilation of room 150 in the following steps.

4.6 Deployment Study

The system has been implemented as a mobile middleware on Android. A pilot indoor mobile application has also been developed built upon the floorplan construction middleware. Six users have been recruited to carry mobile phones with the mobile application, and conducted a deployment study at three different building sites: classrooms, research facilities, and libraries, with a total of 103 unique rooms. We implemented a state-of-the-art room fingerprinting algorithm [22] to generated the Wi-Fi based room fingerprints that are required by the system as inputs.

4.6.1 Room Adjacency Graph Construction

We first evaluate the proposed room adjacency graph construction algorithm, which provides fundamental information for use in floorplan construction. We adopt the precision-recall measure. More specifically, given the ground truth graph G and the produced graph G' of our algorithm, the **recall** is the number of links in G divided by the sum of the number of links in both G and $G's$. The **precision** is the number of links in G' divided by the sum of the number of links in both G and $G's$. The precision-recall measure is evaluated under different k-means clustering setting k. As shown in Figure 4.8, as k increases, recall increases and precision decreases. This is due to the fact that, with more clusters, the algorithm becomes more sensitive to the change of adjacency between rooms. However, as the corresponding size of each cluster decreases, the

classification process becomes less robust. The deployment study shows that a k value equaling to 4 times of the total number of rooms provides a good precision-recall tradeoff.

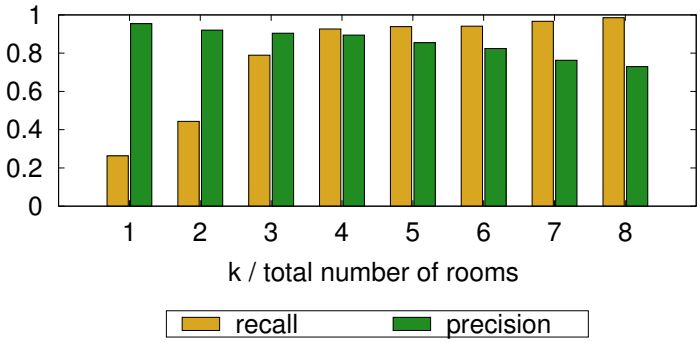


Figure 4.8: Quality of room adjacency graph with different k.

4.6.2 Motion-based Traversal/Turn Detection

We developed a motion detection technique for hallway traversal detection and turn detection. Appropriate thresholds need to be experimentally determined for high-accuracy motion detection.

Figure 4.9 evaluates the quality of hallway traversal detection using different thresholds. Using precision-recall measure, the **recall** is the total duration of correctly detected hallway traversals divided by the total duration of hallway traversals and the **precision** is the total duration of correctly detected hallway traversals divided by the total duration of detected hallway traversals. This study shows that the threshold settings for similarity between two consecutive Wi-Fi scans significantly affect the quality of motion detection. Small thresholds result in missing hallway traversals. Large thresholds lead to misclassification of in-room walking as hallway traversal. The deployment study shows that a threshold of 0.3 provides a good precision-recall tradeoff.

Figure 4.10 evaluates the quality of hallway turn detection using the precision-recall measure. The proposed method, which combines accelerometer, compass, and Wi-Fi fingerprint data, is compared against the following three techniques, accelerometer-only, compass-only, and accelerometer-

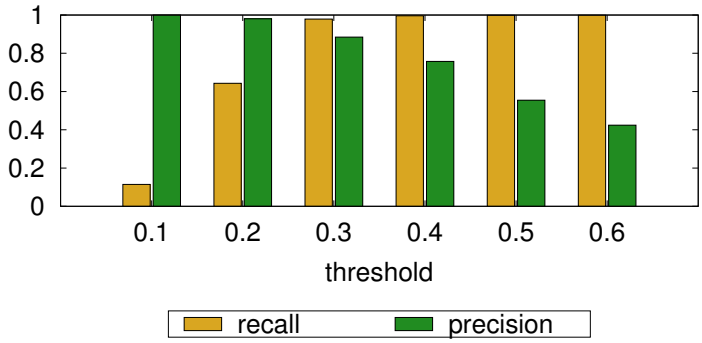


Figure 4.9: Traversal detection quality with different thresholds.

compass. It shows that, accelerometer-only based approach yields low recall and precision due to high dynamics and diversities of human motion. Compass-only based approach yields high precision but suffers from low recall when metallic objects are nearby. The combined accelerometer-compass approach further improves precision with the cost of low recall measure. On the other hand, leveraging the proposed Wi-Fi fingerprint-based turning point detection, our proposed solution offers both high precision and recall.

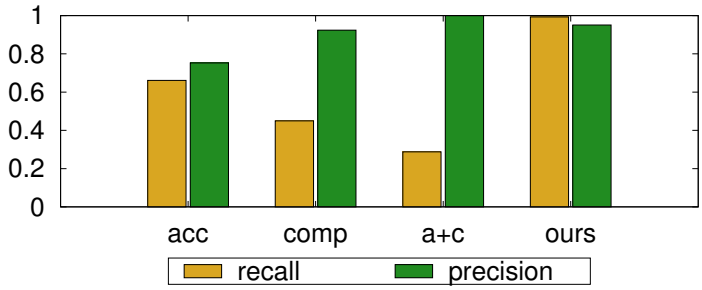


Figure 4.10: Turn detection quality comparison.

4.6.3 Floorplan Construction

Floorplans are the final output of the proposed system. Figure 4.11 and 4.12 show two case studies of automatically constructed floorplans, a classroom building and a research lab building.

The library building floorplan is not shown due to limited space, but its results are included in the following quantitative analysis results.

Figure 4.11 provides a step-by-step walk through of the floorplan construction process for the classroom building – (a) is the ground truth floorplan; (b) is the room adjacency graph; (c) is the initial floorplan after hallway layout learning; and (d) is the final floorplan after force directed dilation. The results show that the automatically constructed floorplan offers excellent match to the ground truth.

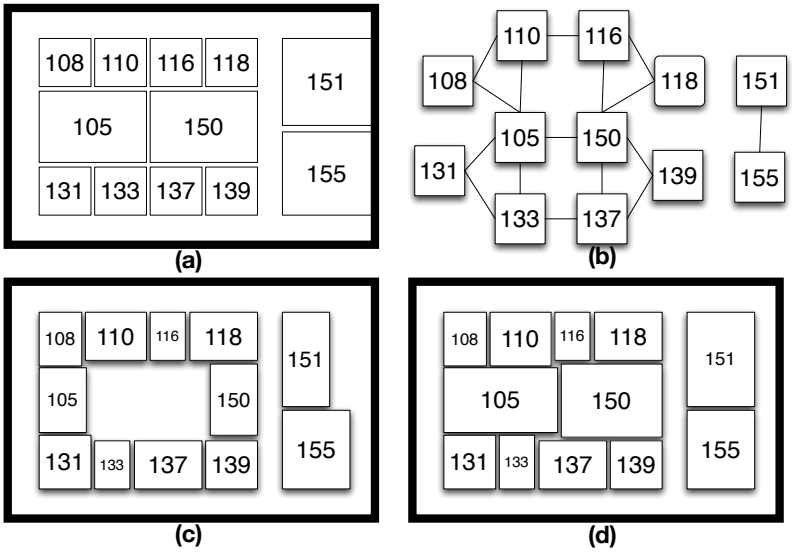


Figure 4.11: Classroom building floorplan case study.

Figure 4.12 shows the research lab building case study, which also shows excellent match to the ground truth. In addition, detected floorplan mismatches are indicated. The following room pairs (indicated in red) had their positions swapped: 19–20, 40–41, and 49–50. For rooms 44, 45, 46 (marked in blue), the width is estimated incorrectly, because their widths are different from the rest of the rooms along the same hallway. Room 10 and 11 are elevators and emergency exits, which are not accessible to our study and were therefore not detected and lead to overestimates of the sizes of rooms 8 and 9.

To quantitatively measure the quality of the constructed floorplan, the following three met-

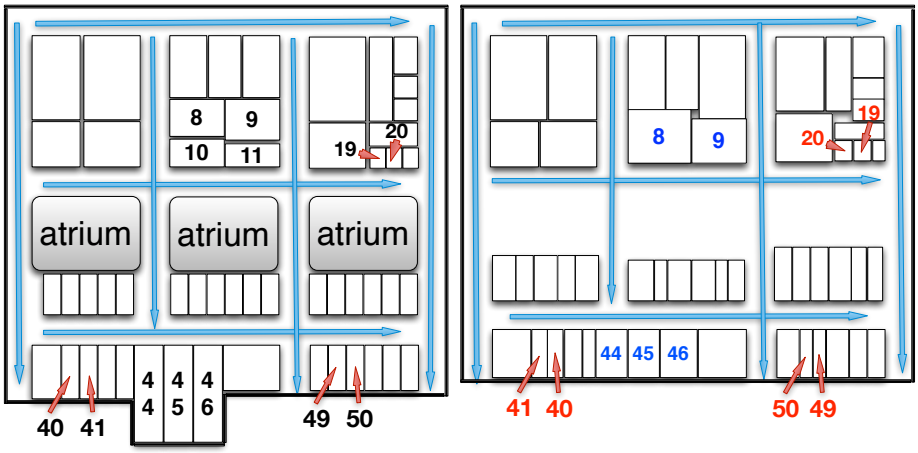


Figure 4.12: Research lab building floorplan case study.

rics are introduced:

- Position: A binary measurement for the relative position of two adjacent rooms with four directions: up, down, right, and left. Given two rooms, r_a and r_b, if any part of r_b is on the left side of room r_a's left wall, we say that r_b is on the left side of r_a, similar conventions are used for other directions. Also the position may be a combination of two directions, e.g., up-left. The position accuracy is defined as the total number of room pairs with correctly learned relative position divided by the total number of room pairs.

- Room area: Area error is defined as the area difference between learned area and ground truth and divide by the ground truth.

- Aspect ratio: The room geometric length-width ratio, which reflect the shape of a room. Ratio error equals to the difference of learned ratio and ground truth divided by real the ground truth.

Figure 4.13 shows the average position accuracy measure. The system we have described uses crowd-sourcing based learning method. In general, the more user data collected, the better its accuracy. Given the targeted accuracy measure, a good learning method should converge quickly,

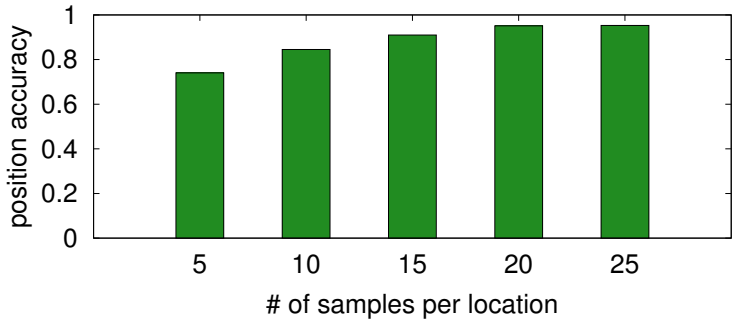


Figure 4.13: Position measure.

and therefore have a low crowd-sourcing data requirement. As shown in this figure, given different numbers of data points collected per location, the yield floorplan accuracy is shown. When 20 data points are collected at each location, the position accuracy converges to 90%.

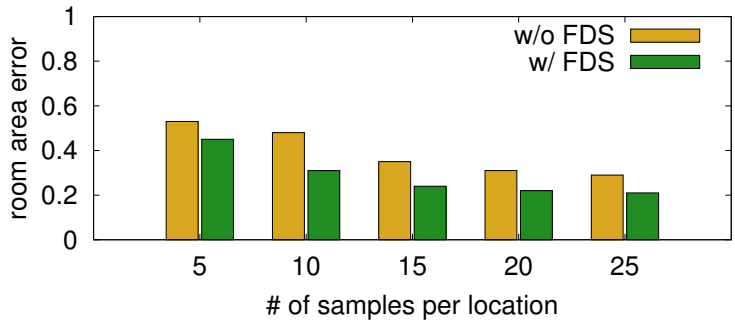


Figure 4.14: Area measure.

Figure 4.14 shows the area error measure with and without the forced directed dilation. The results show that force directed dilation consistently improves room area estimation accuracy by 10%, achieving a stable error measure of 31% with 20 data points per location.

Figure 4.15 shows the average aspect ratio measure. This study shows that, force directed dilation consistently helps reduce the error measure, converging to a 19% error rate with 20 data points per location.

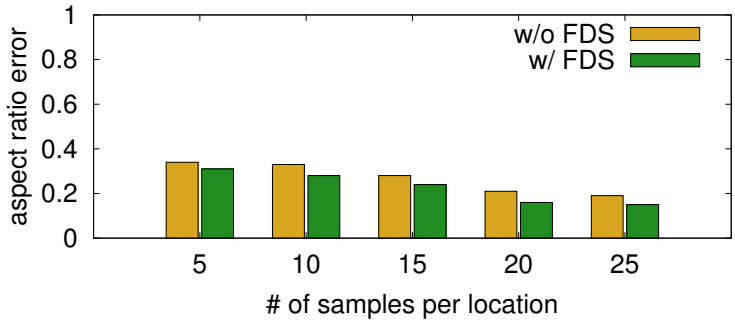


Figure 4.15: Aspect ratio measure.

4.7 Summary

This chapter describes an automatic indoor floorplan construction system. Leveraging Wi-Fi fingerprints and user motion information, this system automatically constructs floorplan via three key steps: (1) **room adjacency graph construction** to determine which rooms are adjacent; (2) **hallway layout learning** to estimate room sizes and order rooms along each hallway, and (3) **force directed dilation** to adjust room sizes and optimize the overall floorplan accuracy. Deployment study in three buildings with 103 rooms demonstrates high floorplan accuracy. The system has been implemented as a mobile middleware, which allows emerging mobile applications to generate, leverage, and share indoor floorplans.

Chapter 5

Trip Detection

Trip detection is a fundamental issue in room localization application on mobile devices. It aims to automatically recognize significant places and trips between them, then automatically start to stop the room localization services. The key challenge is how to minimize energy consumption while maintaining high accuracy. Previous works that use GPS/Wi-Fi sampling are accurate but energy efficiency is low and does not improve over time. Learning from the human decision making process, we propose an energy-efficient trip detection framework that consists of two modes: The **deliberation** mode learns cell-id patterns using GPS/Wi-Fi based localization methods; the **intuition** mode only uses cell-ids and learned patterns for trip detection; transition between the two modes is controlled by parameters that are also learned. We evaluated our framework using real-life traces of six people over five months. Our experiments demonstrate that its energy consumption decreases rapidly as users' activities manifest regularity over time.

5.1 Background and Related Work

In this section, we survey works that are most related to ours, focusing on place learning, and recognition. We also discuss the energy efficiency issue in location sensing and the difference of our work with others.

5.1.1 Place Learning

There are generally two approaches to place learning – geometry and fingerprint. [3] and [25] are geometric, in which geo-coordinates belonging to the same meaningful places are clustered. [3] considers locations where GPS signal is lost for some time as potential place candidates, exploiting the fact that GPS reception is poor indoors and around the so-called urban canyons. [25] uses time-based clustering and conceptually can work with any indoor or outdoor positioning technologies as long as they produce geo-coordinates.

Among fingerprint-based approaches, [19][34][54] assume that radio fingerprints of places are stable and unique. By comparing similarity between two fingerprints, we can tell whether they are close or far apart. The BeaconPrint algorithm [19] periodically scans WiFi base stations and GSM cells to form fingerprints. According to [19], the visibility of a beacon is often a better metric than the received signal strength when constructing a fingerprint, an idea inherited in later works [26][27]. Works in [34][54] address mass-market devices by making a harsher assumption that nothing but cellular radio is available. Both adopt simple algorithms to scan traces of timestamped cell ids and cluster those seen as close, using temporal correlations only. Compared to other more advanced methods, although they provide less accuracy, they consume less energy and run on any cellular device.

5.1.2 Place and Trip Recognition

Geometry-based algorithms recognize places by checking whether the device's current geolocation falls into the geometric shape of any place. Approaches in [34][54] work by comparing the current GSM cell id to each of the cell id clusters. PlaceSense [26] proposes to not only recognize if a user is at a place, but also whether she is arriving at or leaving a place, i.e., place entrance and departure. In addition to revising the original BeaconPrint algorithm in [19], PlaceSense reliably detects place departure by requiring all representative beacons associated with a place to have disappeared for some time, and detects place entrance by buffering scanned fingerprints for

parallel entrance/departure detection.

5.1.3 Energy Efficiency for Location Sensing

Location sensing methods, relying on GPS and/or WiFi, can provide accurate location information but are always power hungry. In order to address the problem, Kim et al. [26, 27] proposed an accelerometer based approach, which uses acceleration reading to detect a user's movement and stops location sensing when the user is stationary. The flip side is two-fold: (1) although accelerometers consume little energy compared to GPS or WiFi, it still incurs extra overhead when always on, and (2) movement detected by accelerometers does not necessarily indicate that the user is in place transition, e.g., she may be walking around in her office.

Lin et al. [38] proposed to select different location sensors to sense locations based on accuracy requirement, energy profile of sensors, and their accuracy at different locations. Paek et al.[42] proposed a rate-adaptive approach for GPS positioning, in which the GPS is adaptively turned on based on estimated velocity of user; further, their approach learns locations of GPS unavailability to avoid turning on GPS in those places. Zhuang et al. [55] proposed an energy efficient location sensing framework that leverages other lower-power location sensors and uses an accelerometer to avoid unnecessary GPS sensing; in addition, it synchronizes the location sensing requests from multiple applications and tunes sensing parameters based on battery level.

5.1.4 Comparisons

Our work aims to exploit cell ids for trip detection and use GPS/WiFi only for training. Its energy efficiency improves over time as places are re-visited, a trend not demonstrated in previous works to our best knowledge. It resembles [26] in addressing the similar problem of detecting begin and end of a trip; however, [26] resorts to periodic GPS/WiFi scan and does not improve its energy efficiency over time. As in [27, 55], we use accelerometer for energy-efficient data collection in the deliberation mode; however, we only use cell ids in the intuition mode while at familiar places, thus avoiding the overheads and accuracy problems discussed above. Techniques

in [**?**, 42, 55] are complementary to ours, which could be leveraged in our future work for further improvements. Our approach differs from [34][54] in that we use cell ids for recognizing begin/end of a trip, by computing the **time-variant** cell id flipping patterns (probabilities), instead of constructing **time-invariant** cliques as they do.

5.2 Design Overview

5.2.1 Motivating Observations

Our work is motivated by the following four observations that are related to cellular devices: First, each device is assigned to one active cell tower at any time through which its basic telephone capability functions (or none in absence of cellular signals). In a mobile phone application, reading the active cell id is as cheap as reading a system variable. In contrast, it incurs extra costs to read other sensory data such as GPS, WiFi, and Bluetooth that are often used for localization. If a level of accuracy comparable to using those alternative sensors can be achieved, it would be more energy-efficient to use cell id for trip detection.

Second, for reasons such as load balancing and signal attenuation, cells usually overlap. As a result, the active cell id of a device often oscillates between several alternatives even when the device remains still at one place. However, the patterns typically differ when the device is moving within a place, e.g., a building, and when it is moving between two places that are sufficiently far apart. That is, it is possible to use cell id patterns for trip detection.

Third, it is fast to read the active cell id information and detect its changes. Cell id is almost always available, even where WiFi and GPS information is unavailable or noisy. By comparison, WiFi is not always available and takes time to scan; GPS is noisy in the urban canyons and often takes from tens of seconds to a few minutes to warm up, especially after a period of sleep or signal loss, as confirmed in [25].

Last but not least, due to their coarse granularity, cell tower ids alone cannot be used for accurate localization. We must use cell ids in combination with more accurate measures, such as

GPS and WiFi beacons, or at least first use those measures to train the cell id based mechanism so as to achieve the desired level of accuracy.

From the above observations, it seems possible and advantageous to use cell id patterns for detecting places and place boundaries (i.e., begin and end of a trip). If this is feasible with high accuracy, then we can rely on cell ids for energy-efficient trip detection. The questions are (1) how to obtain the cell-id patterns for places and their boundaries, (2) how to use these patterns for trip detection, and (3) under what conditions we can or cannot use these patterns.

5.2.2 Human Decision Making Model

In pursuit of a solution to the trip detection problem, we are inspired by the dual-process theory for human decision making in the psychology literature [45][23]. According to the theory, individuals sometimes make decisions deliberately and sometimes rely on their intuition or gut feeling. Intuition (or associative reasoning) is directly proportional to the similarity of past experiences, relying on temporal and similarity relations to determine reasoning rather than an underlying mechanical structure. On the other hand, deliberation (the true reasoning) functions on logical structure and variables, basing upon rule systems to come to conclusions. In general, deliberation is slower and subject to conscious judgments, while intuition comes to mind quickly and effortlessly.

In spirit of this theory, we treat our mobile phone app as an intelligent agent, which distinguishes deliberation and intuition modes: In the deliberation mode, our system uses more energy-expensive information (e.g., WiFi, GPS, and accelerometer) with heavier computation to make decisions regarding trip detection. Meanwhile, the system will build "experiences," i.e., cell id patterns or probabilities. In the intuition mode, the system uses inexpensive information (the active cell id) and "past experiences" to make decisions through lightweight probabilistic reasoning. Wherever past experiences are insufficient, the system falls back to the deliberation mode to accrue more experiences by learning new patterns, or reinforcing or correcting learned patterns.

5.2.3 Overall Approach and Scope of Research

Figure 5.1 gives an overview of our trip detection approach as motivated above. Initially, our system determines the user's current state, whether she is at some significant place or in a trip. By definition, a significant place is where the user has stayed for more than some time (say 10 minutes) or where she frequently visits; a trip is the transition state between two places. In absence of knowledge such as accurate schedule, when the user is at a place, the system has to continuously detect whether the user is leaving the place for a trip to another place, and, when the user is already in a trip, the system has to continuously determine whether she is arriving at some place. This is the so-called problem of trip detection, i.e., to recognize begin and end of a trip [26].

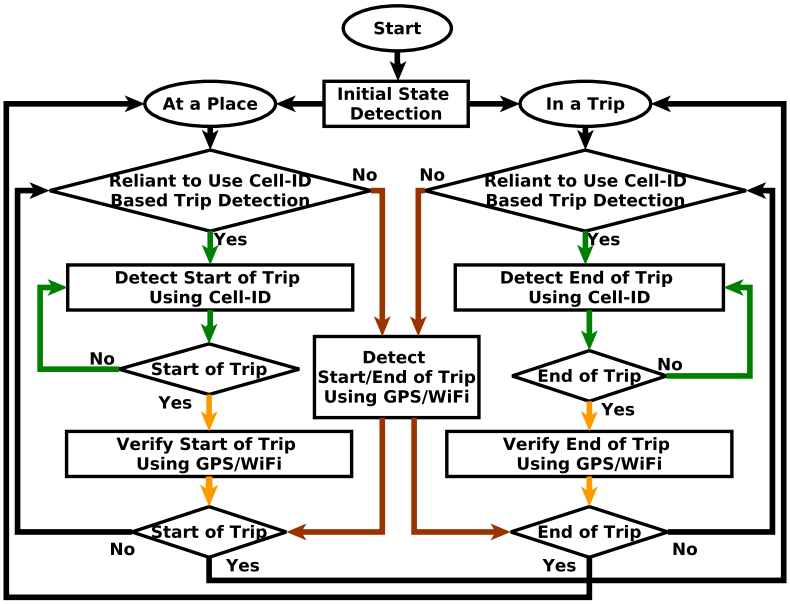


Figure 5.1: Detecting begin and end of a trip relative to a significant place: Use cell id patterns for "intuitive" detection when possible, and use GPS/WiFi for "deliberate" detection when necessary. Energy saving is achieved when a user visits the same places and repeats the same trips and accordingly the system works in the intuition mode.

In each of the states, at a place or in a trip, the system works in either a deliberation or an intuition mode. In the deliberation mode, sensory data such as GPS and WiFi beacons are collected and used for trip detection and place recognition. Many existing methods could be leveraged here,

e.g., [19][25][27]. The method in our system is extended from [25] and [27], which uses GPS/WiFi to infer places/trips and accelerometer based movement detection to reduce power consumption. After the significant places are recognized, we compute the patterns of cell ids at those places.

In the intuition mode, we use cell ids alone for trip detection (and place recognition). We keep a short history of recently observed active cell ids and compute the probability of the history with regard to the patterns learned in the deliberation mode. If the user was last known to be at a place, she is likely departing the place when the probability is low relative to that place, or remains at that place when the probability remains above some threshold. Similarly, if the user was last known to be in a trip, she is likely arriving at a place when the probability is high relative to some known place, or she is still in the trip when no such place is found. It could be a new place when new cell id observations are made; then the algorithm for place/trip learning will step in (Section 2.1).

The two modes are not independent: The intuition mode makes trip detection decisions based on the cell id patterns computed in the deliberation mode. The transition between modes is controlled by system parameters that gauge whether or not the patterns are sufficiently reliant. At a place or in a trip, when a new cell id observation is made that has not been learned, the system falls back to the deliberation mode to collect more data and learn new patterns.

5.3 Deliberation and Intuition

This section elaborates the deliberation and intuition modes (phases). Specifically, we will discuss how to compute cell id patterns in the deliberation phase, how to detect begin and end of a trip using cell ids in the intuition phase, and when to transit from the deliberation phase to the intuition phase to harvest the energy benefits without sacrificing accuracy. Key parameters will be identified in the discussions, which will be further determined by experiments in the next section.

5.3.1 The Deliberation Mode

This phase has three tasks: (1) learning significant places, (2) collecting cell id traces associated with places, and (3) learning cell id patterns. Since we leverage [25, 27] for task (1), we assume that the interesting places and trips are already learned accurately. Hence we focus on tasks (2) and (3).

5.3.1.1 Collecting Cell ID Traces

In principle, cell phones are able to see all available cells in the vicinity before selecting one to connect to, and some commercial phone models do provide such capability under the so-called "field test" mode. In reality, however, most phone OSs only provide APIs to reveal the active cell to which the phone is currently connected. We assume that only the active cell information may be retrieved, to accommodate mass-market devices that lack a field test mode. The active cell information usually includes mobile country code, mobile network code, local area code, and cell id. To protect user privacy, we hash each observed tuple into an internal id and refer to the hash code as the "cell id".

We associate a time duration to each cell id observation to indicate how long that cell id is continuously observed. For example, if connection to an observed cell id X has lasted δ_X seconds before the phone switches to another cell, we record this observation in a 2-tuple (X, δ_X). If that observation is immediately followed by an observation of cell id Y for δ_Y seconds, they will appear in the trace consecutively as (X, δ_X) (Y, δ_Y). Modern phone APIs allow an application to register a callback function that is called when the active cell changes. We can record the time difference as δ_X between the moment the active cell changes to X and the moment it changes to something else. In absence of such APIs, we can read the active cell id periodically say every 2 seconds and similarly record the duration of each observed cell id.

5.3.1.2 Learning Cell ID Patterns

Algorithm 2 sketches how cell id patterns are learned from collected traces. In the input, we assume that each recognized place in tab_plc is associated with a sequence of cell id observations, which is a time series. Each cell id observation is labeled with some place Z from tab_plc. This provides the basis for supervised learning of cell id patterns. We then calculate the following three sets of probabilities:

(1) Pr(Z), the prior probability of place Z, for each place Z;

(2) Pr($X, \delta_X|Z$), the conditional probability of observing (X, δ_X) at place Z, for each observation (X, δ_X) at each place Z;

(3) Pr($X, \delta_X \rightarrow Y|Z$), the conditional probability of observing cell id Y immediately following the observation of (X, δ_X) at place Z, for each observation (X, δ_X) and each cell id Y at each place Z, where $Y \neq X$.

Given the training set of tab_plc, Pr(Z) is calculated by dividing the total stay time at Z by the total stay time at all places. Pr($X, \delta_X|Z$) is calculated by dividing the sum of durations of all (X, δ_X) instances by the total stay time at place Z. Pr($X, \delta_X \rightarrow Y|Z$) is calculated by dividing the number of observation instances where Y appears immediately after some (X, δ_X), by the total number of (X, δ_X) instances at place Z. These probabilities are then saved to tab_ptn.

Note that the duration δ_X in each observation (X, δ_X) is a real number. As will be discussed in Section 5, the duration values will all be discretized into a small number of categories. The discretization scheme itself is a system parameter to be determined by experiments.

5.3.2 The Intuition Mode

Algorithm 3 outlines how we detect the start/end of a trip given the last known state s, either at-a-place or in-a-trip, and an up-to-date sequence q of cell id observations. The length of q

Algorithm 2 Supervised cell-id pattern learner

1: **global** tab_plc; ▷ places with cell-id traces – Input
2: **global** tab_ptn; ▷ cell-id patterns – Output
3: **for** each place Z in tab_plc **do**
4: $tab\_ptn(Z) \leftarrow \Pr(Z)$; ▷ prior probability
5: **for** each cell id instance (X, δ_X) in Z **do**
6: $tab\_ptn(Z, X, \delta_X) \leftarrow \Pr(X, \delta_X|Z)$;
7: **end for**
8: **for** each consecutive (X, δ_X) (Y, δ_Y) in Z **do**
9: $tab\_ptn(Z, X, \delta_X, Y) \leftarrow \Pr(X, \delta_X \rightarrow Y|Z)$;
10: **end for**
11: **end for**

is fixed and we keep a sliding window of $|q|$ most recently observed cell ids and their durations. Notation q_i denotes the i-th element in q, where $0 \leq i < |q|$.

When the last system state s is at place Z, Algorithm 3 constantly monitors the active cell id, and calculates $Pr(q|Z)$ every time a new cell id observation is made and added to q. If $Pr(q|Z)$ drops sufficiently low, say below a threshold $T_s(Z)$, a trip start is reported. As shown in Figure 5.1, GPS/WiFi scan will be turned on to verify this detection. The system state s is changed to in-a-trip if it is confirmed.

When the last state is in-a-trip, Algorithm 3 continues to monitor the active cell id, looking for potential arrival at another place $\hat{Z}$ with the following Bayesian equation:

$$\hat{Z} = \underset{Z \in all\ places}{\operatorname{argmax}} \ Pr(q|Z) \cdot Pr(Z)$$

If $Pr(q|\hat{Z})$ is above threshold $T_e(\hat{Z})$, a trip end is reported. Again GPS/WiFi will be turned on for verification. If it is confirmed, state s is changed to at place $\hat{Z}$.

Assuming that the sliding window size $|q|$ is 3 and the most recent cell id is C. Duration δ_C is not available until the device switches to yet another cell. Let q = $(A, \delta_A)(B, \delta_B)(C, \_)$. By the

Algorithm 3 Cell-id state predictor(**input** s, q)

1: **global** tab_ptn; ▷ cell-id patterns
2: **global** T_s and T_e; ▷ probability thresholds for places
3: **if** last state s is at-a-place(Z) **then**
4: Pr($q|Z$) $\leftarrow$ ▷ probability of observed sequence q
Pr($q_0, \delta_{q_0}|Z$) $\cdot$ Pr($q_0, \delta_{q_0} \rightarrow q_1|Z$) $\cdot$
$\cdots \cdot$ Pr($q_{n-1}, \delta_{q_{n-1}} \rightarrow q_n|Z$); ▷ n=$|q|$
5: **if** Pr($q|Z$) $\leq T_s(Z)$ **then** ▷ departing place Z
6: **return** start-of-trip;
7: **end if**
8: **else if** last state s is in-a-trip **then**
9: $\hat{Z} \leftarrow \text{argmax}_{Z \in all\ places}$ Pr($q|Z$) $\cdot$ Pr(Z);
10: **if** Pr($q|\hat{Z}$) $\geq T_e(\hat{Z})$ **then** ▷ arriving at place $\hat{Z}$
11: **return** end-of-trip;
12: **end if**
13: **end if**

Bayesian theorem, $Pr(q|Z)$ can be written as:

$$\begin{aligned} Pr(q|Z) &= Pr((A,\delta_A)(B,\delta_B)(C,\_)|Z) \\ &= Pr(A,\delta_A|Z) \times Pr(A,\delta_A \rightarrow B|Z) \\ &\quad \times Pr(A,\delta_A \rightarrow B,\delta_B \rightarrow C|Z) \\ &= Pr(A,\delta_A|Z) \times Pr(A,\delta_A \rightarrow B|Z) \\ &\quad \times Pr(B,\delta_B \rightarrow C|Z). \end{aligned}$$

Note that in the last step we assume a first-order Markov model, which is a widely made assumption. That is, each observation only depends on its immediate preceding observation. Each term on the right-hand side of the equation can be directly retrieved from tab_ptn, which is computed by Algorithm 2. We consider the sliding window size $|q|$ and, for each place Z, the two probability thresholds, $T_s(Z)$ (for detecting start of trip) and $T_e(Z)$ (for detecting end of trip), as parameters yet to be determined by experiments.

5.3.3 Transition between Two Modes

Obviously, we must deliberate the cell-id patterns before they could be used for intuitive trip detection. For the probabilities to be reliable, they must be computed from a sufficiently large

set of data. Otherwise we must fall back to the deliberation mode for more data. The question is how much data would be sufficient and when to switch between modes.

Unfortunately, there is no obvious answer to this obvious question. At any place, for example, the set of cell ids observed and their durations will be dependent on the local cell coverage as well as how active the user is. This has two practical implications: One is that the thresholds for the transition between deliberation and intuition are necessarily different from place to place. The other is that the value space is very large and it would be impractical to require the user to exhaust the value space for data collection.

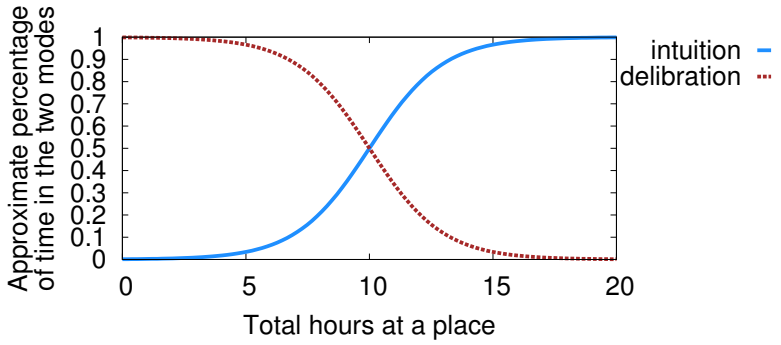


Figure 5.2: The energy-consuming deliberation phases out after sufficient experience has accrued for intuitive trip detection. The two modes coexist for some period of time for desired accuracy.

In order to seek a balance between energy cost and accuracy, our system takes an incremental approach, which is illustrated in Figure 5.2. Initially, the system works in a deliberation mode to aggressively collect data. After spending a few hours at one place, it computes the cell-id patterns for that place. Then, it starts trying to use the cell-id patterns for prediction, while staying in the deliberation mode. Hence there is a period (called dual mode) in which the two phases overlap. In this period, the results yielded from the intuitive predictor are compared to those from the deliberate predictor. When the accuracy of the intuitive predictor is sufficiently high (based on the results of the deliberate predictor), the deliberate predictor will phase out.

On one hand, there are heuristics by which we can estimate how long the user must stay before entering the dual mode. For example, it may deserve a trial if the user has accumulated

two hours of data at a place. On the other hand, exceptions may still arise even after it has demonstrated sufficient accuracy at some point and is already in a "pure" intuition mode. This may happen, e.g., when a new cell id X is observed at place Z, or the (discretized) duration δ_X of some X is not yet in the computed patterns. In such cases, our system will temporarily fall back to the dual mode.

5.4 Experiments and Performance Evaluation

In this section, we evaluate the performance of the proposed framework for energy-efficient trip detection. Specifically, we will answer the following two questions:

- How well does our approach perform trip detection in terms of accuracy, timeliness, and energy efficiency?
- How do the identified parameters affect system performance? How to choose the "best" parameter values?

5.4.1 Data Collection

We have developed a data collection program which collects GPS, WiFi, accelerometer, and cell id information. GPS and WiFi data are collected every 2 seconds, while cell id data are collected using the "callback" mode and the current cell id is recorded every time it changes. Accelerometer data are sampled at 30 Hz. The data collection program has been deployed on six Nokia N900 devices, each carried by a user in our study. Four users ran it each for four months, and the other two users ran it each for one month. To obtain the ground truth, we asked the users to take notes on all the trips during the data collection period. All trips are also verified on maps using the Google Map API and we have manually marked the start and end of each trip in the collected data.

In total, we have collected 1755 trips, among which there are 139 unique trips and 83 unique places. To demonstrate the advantages of our approach, we ignore casual trips and focus only on

regular trips, i.e., trips that have been repeated at least once. This leaves us with 1701 trips, which contain 85 unique trips and 43 unique places. On average, a trip is repeated 4.17 times, ranging from 2 to 62 times. The trips range from 1 mile to 32 miles in distance (average 12 miles) and include driving, cycling, and walking trips. Trip duration varies from 5 minutes to 53 minutes, and the average is 24 minutes. The amount of time that our users stay at a place varies from 8 minutes to 16 hours and the average is 5 hours. The regularity in our data set is consistent with [20].

5.4.2 Parameter Settings

There are three key parameters in our framework: the trip start and end thresholds, $T_s(Z)$ and $T_e(Z)$, for each significant place Z, the discretized duration length δ_X of each cell id connection X, and length $|q|$ of observed cell id sequence q on which we compute probabilities.

5.4.2.1 Length of Recent Cell ID Sequence

As shown in Algorithm 3, we compute the probability of a sequence q of recent cell ids. A larger $|q|$ value may yield more accurate probability estimation and better trip detection accuracy, but it may also reduce the timeliness of trip detection and incur higher computation overheads. As will be shown (Figure 5.6), when $|q| = 3$, the algorithm demonstrates a desired performance tradeoff, and there is no obvious improvement on accuracy when $|q| > 3$. Therefore, we set $|q|$ to 3 in the other experiments and our system.

5.4.2.2 Discretization of Cell-ID Connection Durations

Figure 5.3 shows the probability distribution of cell id connection durations in users' places and trips. We see that the connection durations vary significantly from several seconds to a few hours. Equally treating all the possible duration values would result in many zero probabilities. Instead, we need to discretize the duration values into multiple categories which best capture the ranges of connection durations that occur frequently. Our analysis reveals that (1) connection

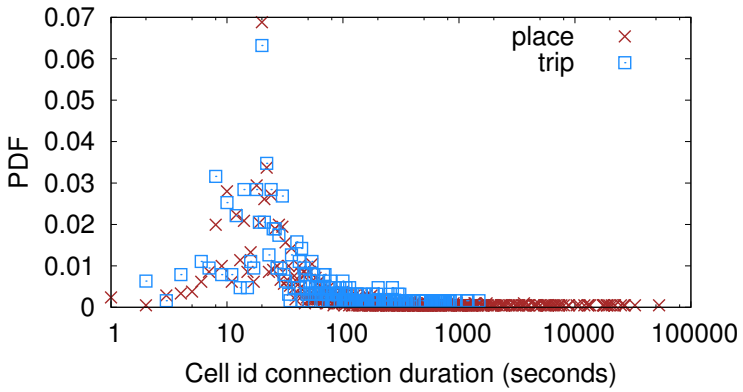


Figure 5.3: Distribution of cell id connection duration (δ) when at place and in trip. Accordingly, we define 6 discrete categories, i.e., the class is 6 if $\delta > 243$, and $\lceil log_3 \delta \rceil$ if $0 < \delta \leq 243$.

duration goes above 1000 seconds only when a user stays at a place (not during a trip), and (2) connection duration is most frequently around 20 seconds for both trips and places.

Let δ be the duration value of some cell-id connection. We define the discretization function as follows: If $\delta > 243$, the class is 6; or otherwise if $0 < \delta \leq 243$, the class is $\lceil log_3 \delta \rceil$. Then we have finer classes for more frequent durations and coarser classes for less frequent connections. This helps to distinguish start/end of trips from places.

5.4.2.3 Trip Start/End Thresholds

For each place Z, the thresholds $T_s(Z)$ and $T_e(Z)$ determine at what values of probability $Pr(q|Z)$ should a user be considered at a place or at start/end of a trip. To decide their values, we studied the probability distributions of all users staying, departing and arriving at their significant places and found that they demonstrated similar patterns.

As an example, Figure 5.4 plots the probability distributions when one user staying at some place Z, leaving place Z (from the user crossing the place boundary till $Pr(q|Z) = 0$), and arriving at place Z (from $Pr(q|Z) > 0$ till user crossing the boundary into Z). When the user stays at place Z, the curve shows higher probabilities for larger values of $Pr(q|Z)$; there is a long tail, however, when the user walks around the boundary of the place and/or some rare cell id observations are

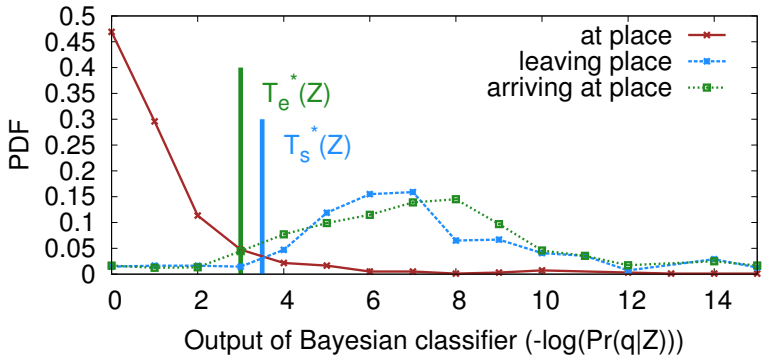


Figure 5.4: Determination of the probability thresholds $T_s^*(Z)$ and $T_e^*(Z)$ for place Z. Note that the probability $Pr(q|Z)$ decreases as $(-log(Pr(q|Z)))$ increases along the x-axis. To handle small probability values, we compute $-log(Pr(q|Z))$ instead of $Pr(q|Z)$.

made. On the other hand, when the user is leaving or approaching place Z, the probabilities of $Pr(q|Z)$ are relatively low as compared to those when stay at place Z. By the two curves, values of $-log(Pr(q|Z))$ are almost normally distributed around 4-10, where the user is nearby place Z. A few high probabilities happen when user is at the place boundary and a few very low probabilities happen when some rare cell id observations are made.

The goal is to find the optimal threshold $T_s^*(Z)$ or $T_e^*(Z)$ that separates the cases of staying at place Z and leaving (or arriving at) place Z. In spirit of the clustering threshold tuning approach proposed in [43], our optimal threshold $T_s^*(Z)$ or $T_e^*(Z)$ corresponds to the point at which the two corresponding probability distributions intersect. Our experimental results show that $T_s^*(Z)$ and $T_e^*(Z)$ may take different values for the same place Z. For each place Z, our system will recalculate the threshold periodically since both user behavior changes (e.g., switching office room to another side of the building) and cell id changes (e.g., a new cell id observation is made) may entail adjustment to the thresholds.

Monotonicity of $p(q|Z)$ when user is leaving or approaching Z is important because the verification would be triggered many times and thus not "energy-efficient" if value of $p(q|Z)$ changed drastically. However, we found that 85% of $p(q|Z)$ are monotonic when user is leaving or approaching Z. 94% of the 15% non-monotonic values happen below (when leaving Z) or above

(when approaching Z) the threshold, which does not need to verify by GPS/WiFi.

5.4.3 Performance Metrics

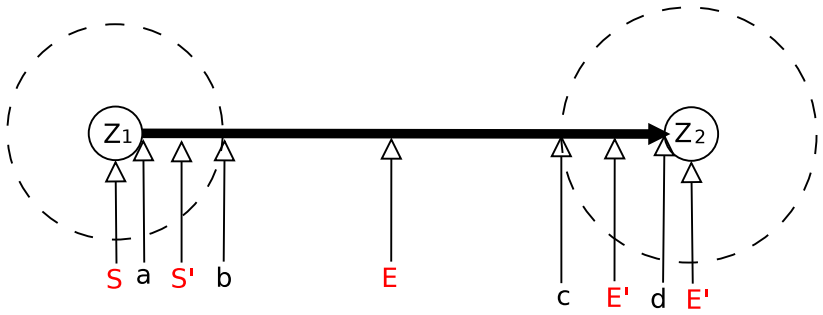


Figure 5.5: A trip from place Z_1 to place Z_2: a/d is the start/end of trip by ground truth; b/c is where the observation probability $Pr(q|Z_1)/Pr(q|Z_2)$ turns zero when moving away from place Z_1/Z_2.

Figure 5.5 shows a typical trip in which a user travels from place Z_1 to place Z_2. Consider the following four phases:

- Phase 1: Before a trip starts, the system should report that the user is at place Z_1. It is a false positive if the system reports a trip start (S). Although this type of error will be corrected by verification, a high false positive error will cause frequent verification and thus high energy cost.

- Phase 2: A trip has started, which should be detected (S'); otherwise it is a false negative error if the system still reports at place Z_1. Ideally, the trip start should be detected before $Pr(q|Z_1)$ drops to zero so as to reduce the latency. This is where threshold $T_s(Z_1)$ comes into play.

- Phase 3: During a trip, the system should report in trip. Otherwise, if the system reports end of trip (E), it is a false positive error, which costs energy for verification. Practically, the report of E is allowed to happen anywhere between when S' and E' are reported.

- Phase 4: When arriving at place Z_2, the system should report end of trip (E'). Ideally, to have a timely report, E' should be detected after $Pr(q|Z_2)$ turns greater than zero but before it reaches its maximum or equilibrium. This is where threshold $T_e(Z_2)$ comes into play. Reporting E' either right before d or right after d is considered acceptable (accurate). However, if the system still reports in trip after passing point d, it is a false negative error.

The performance of our trip detection solution can be measured using three key metrics: (1) **accuracy** measures how often our system correctly detects a trip start or end; (2) **timeliness** measures how quickly our system detects a trip start or end; and (3) **energy efficiency** measures the energy overhead incurred by our trip detection system. Ideally, our system should be able to detect trip starts and ends with high accuracy, in a timely manner, and with high energy efficiency. However, these three metrics impact each other and it is our goal to seek an appropriate performance tradeoff.

5.4.4 Accuracy

Since we resort to existing localization methods in the deliberation mode, here we only evaluate the accuracy of our approach in the intuition mode. For each trip and place, we compare our decision with the ground truth in each of the four phases. Our system makes a trip detection decision every time the active cell id changes. We define accuracy in any phase y as the number of correct detections in phase y divided by the total number of detections in phase y.

Figure 5.6 compares the trip detection accuracy when different cell id sequence length values are used. We make the following observations:

- Our chosen sequence length $|q| = 3$ achieves the best accuracy overall, and it represents a good balance among accuracy, energy efficiency, and timeliness.

- Longer sequences help improve the accuracy of Phase 1, since people normally stay at a place longer and testing more cell ids will help increase the confidence of decision.

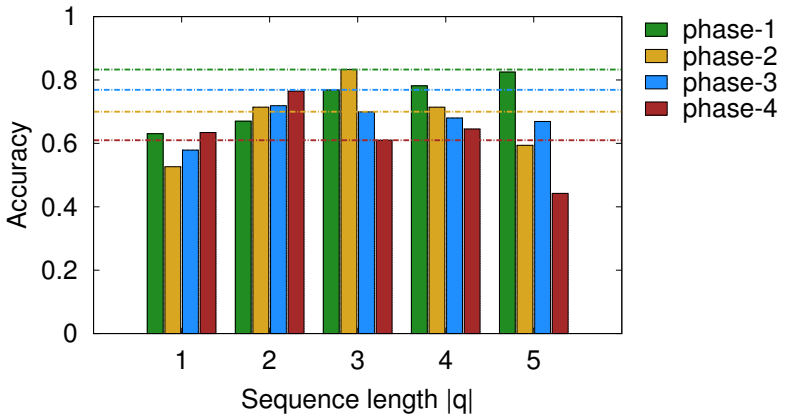


Figure 5.6: Trip detection accuracy in four phases using different cell id sequence lengths: $|q|$=3 represents a desired performance tradeoff.

- Accuracy drops for Phase 2 and Phase 4 when $|q| > 3$, since longer sequences are more conservative and make it harder to detect trip start and end in a timely manner.
- Longer sequences result in a slight decrease of accuracy in Phase 3, since people's trips tend to be shorter than when staying at a place, and increasing sequence length may introduce more noise and reduce detection accuracy.

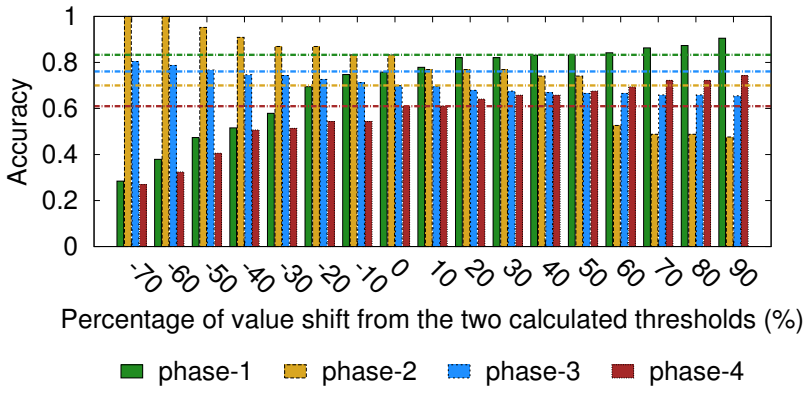


Figure 5.7: Trip detection accuracy by different classification thresholds.

Figure 5.7 shows the accuracy of the four phases using different classification thresholds. The x axis is the percentage of shift from the two thresholds automatically calculated as explained

in Section 5.2.3: zero means the original thresholds, 10 means that both thresholds are incremented by 10%, and so forth. We can make the following observations:

- Our automatically-determined thresholds balance the false positive and false negative for both start and end of trip detection. This is because the accuracy of Phases 1 and 4 increases while that of Phases 2 and 3 decreases when lowering the thresholds.
- Under our automatically-determined thresholds, higher accuracy is achieved for Phase 1 and Phase 2, i.e., detecting a trip start is stabler than detecting a trip end. This is because the cell id pattern of staying at a place is stabler, while trips may differ in the means of transportation, speed, traffic conditions, and routes.
- For Phase 1 and Phase 2, lowering the thresholds leads to higher accuracy for Phase 2 (detecting trip start) but lower accuracy in Phase 1.
- For Phase 3 and Phase 4, lowering the thresholds leads to lower accuracy for detecting end of trip in Phase 4, but higher accuracy in Phase 3.
- In practice, we could adjust the thresholds to make trip detection more timely or more energy efficient.

Further, we also studied the overall accuracy of trip detection by using metrics of precision and recall. Recall is computed as the fraction of correctly detected trips among all trips that the users had, and precision is computed as the fraction of correctly detected trips among all trips that the algorithm believe to be correct. Based on our data set, we compared our method with four other methods: (1) GSM based, we use PlaceSense [26]; (2) GPS based, we use Kang et al. [25]; (3) GPS + Accelerometer (or GPS+), we use SensLoc [27]; (4) WiFi + Accelerometer (or WiFi+), we use SensLoc [27]. The experimental results are plotted in Figure 5.8.

As showed in Figure 5.8, WiFi and GPS based methods both have high precision and recall. For the WiFi based method, the errors in recall are because that several buildings and open places in our data are not covered by WiFi and hence trips from and to those places are not detected;

the errors in precision are caused by noise of WiFi signals, when user walked within a building, sometimes the algorithm treats it as a trip start. For GPS based methods, the errors are due to the poor performance in urban canyon and indoor areas. GSM based methods have poor accuracy because single cell id can only work at a very coarse level and it may not be able to detect short trips; oscillation of cell ids at a place makes the precision low. Our method has better performance than WiFi and GPS based methods in recall because the cell id has better coverage than WiFi and GPS. Although the precision of our method is slightly lower, our verification process using WiFi or GPS compensates for the errors.

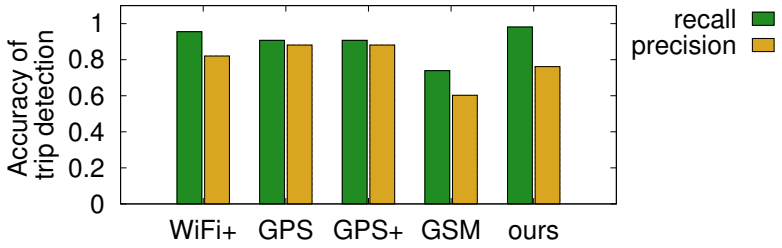
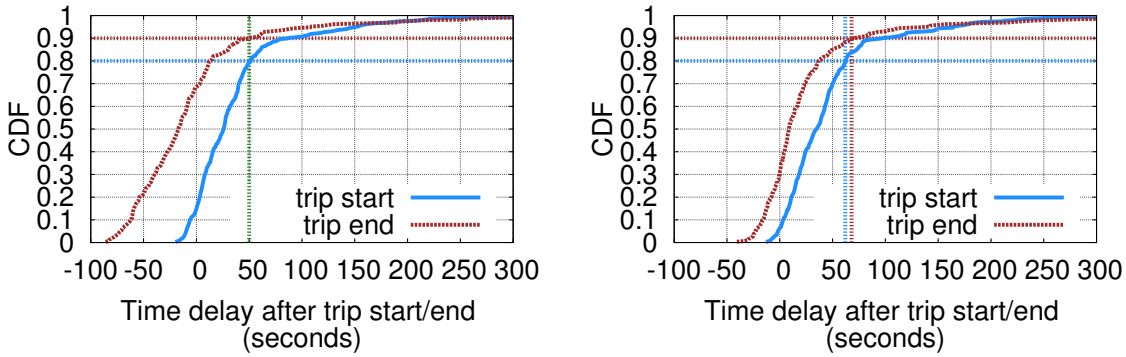


Figure 5.8: Comparison of overall accuracy of trip detection methods.

5.4.5 Timeliness

Not only do we want to detect trips accurately, the detection must also be made in a timely manner. That is, when a trip starts or ends, our system needs to detect it quickly. The timeliness of trip detection depends on two factors: The first is how fast the cell id changes when starting or ending a trip, and the second is how fast we make the decision. The ground truth of trip starts and ends is determined by the time when a user entered or left a building. For open places, such as parks, the trip starts and ends depend on user inputs.

To evaluate how fast the active cell id changes after a user starts or ends a trip, we examine all trips in the collected data. For each trip we extract time of trip start (or end) t from the ground truth, and calculate the time of first cell id change t_c after t. The difference between t and t_c is



(a) Delay in cell id changes

(b) Delay in trip detection

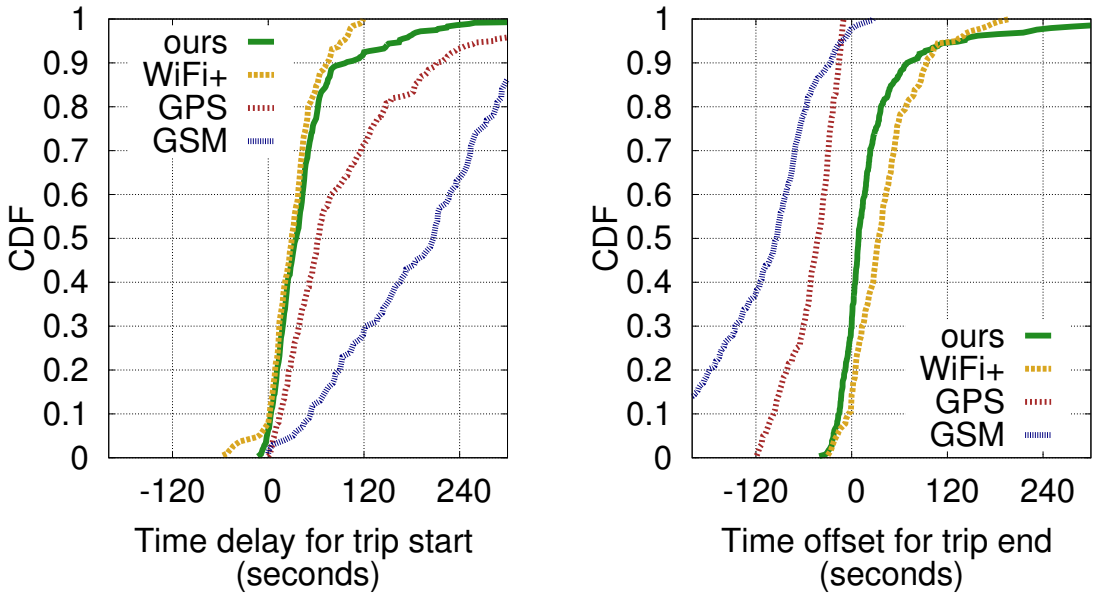
Figure 5.9: Delays in active cell id changes and intuitive trip detection.

the inherent delay in our trip detection process. Figure 5.9(a) plots the cumulative distribution function (CDF) of all $t_c - t$ values calculated from our collected data for trip start and end. It shows that, for 80% (90%) of the time, the cell id changes within 50 seconds after a user starts (ends) a trip.

We then evaluate the average delay in our intuitive trip detection. As shown in Figure 5.9(b), for 80% (90%) of the time, our system can detect the trip start (end) in less than 62 (68) seconds. It is clear that the actual delays, 12 or 18 seconds, are insignificant. They are caused by trading timeliness for accuracy in our work: Often another cell-id change is awaited for $Pr(q|Z)$ to meet the desired threshold with regard to place Z. Our results are consistent with those in [27] yet we use a more energy-efficient cell-id based method that is, however, insensitive to extra delays as possibly caused by the GPS/WiFi scan period in their work.

We compared timeliness of different methods in Figure 5.10. As shown in Figure 5.10(a), our method is comparable to the WiFi based method in that about 80% of trip starts can be detected within 60 seconds. The GPS and GSM based methods have longer delay on trip start detection because GPS takes time to find fixes when the user starts trip and GSM has a coarse granularity in place recognition.

Figure 5.10(b) compares timeliness of trip end detection in different methods. Similarly to the WiFi based method, which can detect 80% of trip ends within 60 seconds, our method can



(a) Compare trip start delays

(b) Compare trip end delays

Figure 5.10: Comparison of trip start/end delays in different methods.

detect 80% of trip ends within 40 seconds. The GPS and GSM methods can detect trip ends much earlier than our method – more than 90% trip ends are detected 20–240 seconds ahead of time. However, they may lead to premature decisions when the user just passes by the place, which lowers the trip detection accuracy.

In Figure 5.10, both CDF curves of our method have a long tail, which indicate that 7.5% (5%) of trip detections has a delay of more than 120 seconds. Those delays happened in remote areas due to sparse cellular coverage – the cell ids did not change that frequently. To address the delays, currently we resort to traditional GPS-accelerometer based methods (e.g., [27]) for energy saving at those places.

5.4.6 Energy Efficiency

To estimate the energy use of our approach, similarly to [27], we make the following assumptions: In the deliberation mode, (a) accelerometer is always on; and (b) GPS is used only when there is no WiFi; (c) neither GPS nor WiFi is on when the user is detected as stationary by

accelerometer readings. In the intuition mode, (a) only collect cell id information; and (b) WiFi and accelerometer are always off.

We calculate average power consumption based on the N900 power consumption specification [41]. Specifically, reading cell id costs 0.01mW each time, WiFi scan at 1/6 Hz costs 80mW, GPS reading at 1/2 Hz costs 325mW, and accelerometer reading at 30 Hz costs 25 mW. In addition, we have conducted in-field power consumption measurements. These numbers are slightly higher than the spec but generally consistent, which do not affect the overall energy efficiency improvement we report here.

It is our observation from the data that a user's mobility heavily affects power consumption. For instance, if a user is stationary most of the time, the system consumes very little energy even when running in the deliberation mode. If a user stays at a place all day and patterns at that place are already learned, then our system will run in the intuition mode most of the time and the power consumption will also be very low. However, if a user visits many new places, then the cell phone will run in the deliberation mode most of the time and the power consumption will be very high.

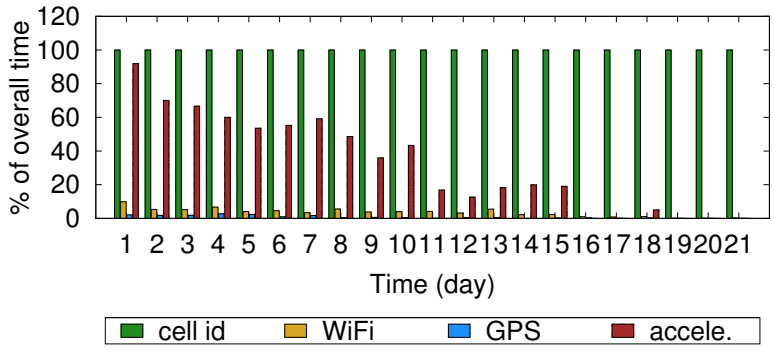


Figure 5.11: Average sensing time of cell id, WiFi, GPS, and accelerometer by 6 users in the first 21 days.

Figure 5.11 shows the average percentage of sensing time based on the total daily system running time of 6 users. We consider the sensing time of cell id, GPS, WiFi, and accelerometer over the first 21 days. During the first several days, the system runs mostly in the deliberation mode,

accelerometer and WiFi are turned on most of the time. After a few days, for some places, the system starts to work in the intuition mode, then the daily working time of WiFi and accelerometer decreases generally, but still go up when a user visits new places. The working time of GPS is dependent on how much time a user spends in places without WiFi coverage.

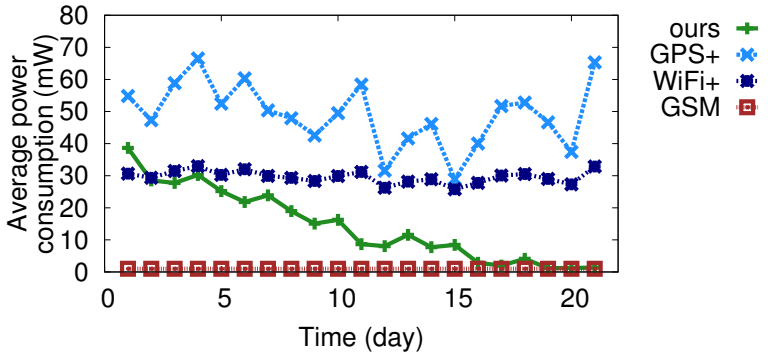


Figure 5.12: Average energy consumption of 6 users in the first 21 days.

Figure 5.12 shows the average power consumption of 6 users over the first 21 days. It shows that, overall, the average power consumption decreases quickly over time as our system learns about more places and spends more time in the intuition mode. In some days, the average power consumption increases, as a result of users making more trips or visiting new places. By comparison, other methods do not have a similar trend of decreasing power consumption over time.

Over the first 21 days, the average power consumption is 49.1 mW for the GPS based method, 29.6 mW for the WiFi based method, 1 mW for the GSM based method, and 14.5 mW for ours. Hence, our approach saves 70% energy compared to the GPS based method and 51% to the WiFi based method. Furthermore, the longer a user runs our system, the more energy they will save. In a long run, the power consumption of our system will be close to the GSM based approach because only cell id is used most of the time.

5.5 Summary

This chapter has presented a 2-phase framework for energy-efficient trip detection on mobile devices: The deliberation phase uses sensors such as GPS and WiFi as in previous work for training a cell-id learner, and the intuition phase solely uses cell-id patterns for predicting start/end of a trip. The goal is to save energy by intuition when a user revisits familiar places or repeats familiar trips. Thus the room localization can be automatically triggered at places that required.

We have collected real-life traces from six users over five months and used the data to evaluate the framework in terms of accuracy, timeliness, and energy efficiency. Our experiments demonstrated 51-70% energy savings over previous GPS/WiFi based methods and, moreover, an obvious trend in decreasing power consumption after a period of deliberation, with considerable accuracy and timeliness in intuitive trip detection. We also showed how the system parameters are tuned and how they impact the system performance tradeoffs among the three metrics.

Chapter 6

Conclusion and Future Work

6.1 Thesis Summary

This thesis has presented several techniques to address the problem of crowdsourcing based room localization on smartphones, including:

- A temporal n-gram augmented Bayesian room positioning method that leverages the ordered sequence of APs based on their RSS values and user's daily room visitation pattern.
- An automatic room fingerprinting approach that consist of three algorithms:1) a zone-based clustering algorithm that accurately identifies in-room occupancy hotspot(s) using Wi-Fi signatures; 2) a motion-based clustering algorithm to identify inter-zone correlation, thereby distinguishing different rooms; and 3) an energy-efficient motion detection algorithm to minimize the noise of Wi-Fi fingerprints.
- An automatic indoor floorplan construction approach with three key steps: (1) room adjacency graph construction to determine which rooms are adjacent; (2) hallway layout learning to estimate room sizes and order rooms along each hallway, and (3) force directed dilation to adjust room sizes and optimize the overall floorplan accuracy.
- An energy-efficient trip detection framework that consists of two modes: The **deliberation** mode learns cell-id patterns using GPS/WiFi based localization methods; the **intuition** mode only uses cell-ids and learned patterns for trip detection; transition between the two modes is controlled by parameters that are also learned.

In above techniques, our solutions addressed the following challenges:

- **Heterogeneous indoor environments:** An indoor environment consists of rooms with diverse sizes. Room connections through hallways also vary significantly. Such heterogeneous indoor environments make accurate floorplan construction challenging.
- **Noisy Wi-Fi fingerprints:** Indoor Wi-Fi infrastructure has been used by many indoor localization techniques. However, due to the complex multipath propagation problem, Wi-Fi fingerprints obtained by mobile phones are dynamic and noisy.
- **Mobile crowd sourcing:** The daily activities of building occupants offer rich and up-to-date information of the personal and social functions of an indoor environment. Leveraging the built-in motion sensors of the mobile phones carried by occupants, "crowd sourcing" offers a potentially highly scalable solution for automatic floorplan construction.
- **Energy overhead:** Those energy overhead imposed on individual smartphones includes two categories: 1) collaborative voluntary effort for Wi-Fi signals and inertial sensor data collection on smartphone; and 2) system energy consumption at places that do not require room localization services, e.g. in a trip or at home.

6.2 Future Work

In this work, we focused on Wi-Fi signals only and Wi-Fi based room localization techniques. As our future work, we would like to investigate other types of signals (such as audio and human mobility patterns), with a special focus on their variabilities under different application scenarios and their impacts on indoor room localization. Another interesting question to investigate is how to adapt dynamically under changing environments to achieve high accuracy, low training cost, and better energy efficiency. We also plan to implement the system as mobile middleware to support emerging indoor mobile applications, allowing them to generate, leverage, and share indoor room fingerprints and floorplans, thereby offering better indoor personal and

social services.

For the deliberation for intuition framework proposed in this thesis, we plan to look into the following directions in future research:

First, it is possible for people who live or work nearby to share the learned cell-id patterns and collaboratively construct a community database of patterns. This would be useful, e.g., for better coverage of data and faster bootstrap of system. It would be interesting to study the mechanisms for supporting this construction, reuse, and evolution process and related privacy issues.

Secondly, as is discussed in [24], the intuitive human decision process, although fast, is fundamentally biased and largely limited to the so-called availability heuristics. In the same light, a mobile app for human activity recognition like this and other works are also subject to the same limitation as imposed by the availability of data. Our approach performs well in urban areas where cellular coverage is typically dense and there are GPS and WiFi signals to make use of. In remote areas such as parks and other recreational places, however, those signals may not even be available. Other sensors such as accelerometer, gyro, and compass could be exploited in those areas.

Thirdly, in this thesis, we elaborate the framework for trip detection. The success seems attributable to the fact that different sensing methods (i.e., cell-id versus GPS/WiFi) have complementary strengths in energy efficiency and accuracy. It is interesting to explore further how this framework can generalize to other sets of sensing methods, different performance tradeoffs and application domains.

Bibliography

[1] G1 Android Power Measurement. http://urban.cens.ucla.edu/resources/g1power/.

[2] Moustafa Alzantot and Moustafa Youssef. Crowdinside: automatic construction of indoor floorplans. In Proceedings of the 20th International Conference on Advances in Geographic Information Systems, SIGSPATIAL '12, pages 99–108, New York, NY, USA, 2012. ACM.

[3] Daniel Ashbrook and Thad Starner. Using GPS to learn significant locations and predict movement across multiple users. Personal Ubiquitous Computing, 7(5):275–286, 2003.

[4] Martin Azizyan, Ionut Constandache, and Romit Roy Choudhury. SurroundSense: mobile phone localization via ambience fingerprinting. In Proceedings of the 15th annual international conference on Mobile computing and networking, MobiCom '09, pages 261–272, New York, NY, USA, 2009. ACM.

[5] P. Bahl, A. Balachandran, and V. Padmanabhan. Enhancements to the RADAR User Location and Tracking System. In Microsoft Research, MSR-TR-2000-12, 2000.

[6] P. Bahl and V.N. Padmanabhan. RADAR: an in-building rf-based user location and tracking system. In INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, volume 2, pages 775–784 vol.2, 2000.

[7] E.S. Bhasker, S.W. Brown, and William G. Griswold. Employing user feedback for fast, accurate, low-maintenance geolocationing. In Pervasive Computing and Communications, 2004. PerCom 2004. Proceedings of the Second IEEE Annual Conference on, pages 111–120, 2004.

[8] Xiaoyong Chai and Qiang Yang. Reducing the calibration effort for probabilistic indoor location estimation. Mobile Computing, IEEE Transactions on, 2007.

[9] Krishna Chintalapudi, Anand Padmanabha Iyer, and Venkata N. Padmanabhan. Indoor localization without the pain. In Proceedings of the sixteenth annual international conference on Mobile computing and networking, MobiCom '10, pages 173–184, New York, NY, USA, 2010. ACM.

[10] Jaewoo Chung, Matt Donahoe, Chris Schmandt, Ig-Jae Kim, Pedram Razavai, and Micaela Wiseman. Indoor location sensing using geo-magnetism. In Proceedings of the 9th international conference on Mobile systems, applications, and services, MobiSys '11, pages 141–154, New York, NY, USA, 2011. ACM.

[11] H. Durrant-Whyte and Tim Bailey. Simultaneous localization and mapping: part I. Robotics Automation Magazine, IEEE, 2006.

[12] Peter Eades. A Heuristic for Graph Drawing. Congressus Numerantium, 1984.

[13] Martin Ester, Hans P. Kriegel, Jorg Sander, and Xiaowei Xu. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise. In Evangelos Simoudis, Jiawei Han, and Usama Fayyad, editors, Second International Conference on Knowledge Discovery and Data Mining, pages 226–231, Portland, Oregon, 1996. AAAI Press.

[14] Martin Ester, Hans-Peter Kriegel, Jörg Sander, Michael Wimmer, and Xiaowei Xu. Incremental clustering for mining in a data warehousing environment. In Proceedings of the 24rd International Conference on Very Large Data Bases, VLDB '98, pages 323–333, San Francisco, CA, USA, 1998. Morgan Kaufmann Publishers Inc.

[15] W.J. Fisk. Health and productivity gains from better indoor environments and their relationship with building energy efficiency. Annu. Review Energy Environ, 25:537–566, 2000.

[16] Jun geun Park, D. Curtis, S. Teller, and J. Ledlie. Implications of device diversity for organic localization. In INFOCOM, 2011 Proceedings IEEE, pages 3182–3190, 2011.

[17] Youngjune Gwon and Ravi Jain. Error characteristics and calibration-free techniques for wireless LAN-based location estimation. In Proceedings of the second international workshop on Mobility management & wireless access protocols, MobiWac '04, pages 2–9, New York, NY, USA, 2004. ACM.

[18] Andreas Haeberlen, Eliot Flannery, Andrew M. Ladd, Algis Rudys, Dan S. Wallach, and Lydia E. Kavraki. Practical robust localization over large-scale 802.11 wireless networks. In Proceedings of the 10th annual international conference on Mobile computing and networking, MobiCom '04, pages 70–84, New York, NY, USA, 2004. ACM.

[19] Jeffrey Hightower, Sunny Consolvo, Anthony LaMarca, Ian Smith, and Jeff Hughes. Learning and recognizing the places we go. In Proceedings of the 7th international conference on Ubiquitous Computing, UbiComp'05, pages 159–176, Berlin, Heidelberg, 2005. Springer-Verlag.

[20] P.S. Hu and T.R. Reuscher. Summary of travel trends, 2001 national household travel survey. Technical report, U.S.F.H.A. U.S. Department of Transportation, 2004.

[21] Yifei Jiang, Kun Li, Lei Tian, Ricardo Piedrahita, Xiang Yun, Omkar Mansata, Qin Lv, Robert P. Dick, Michael Hannigan, and Li Shang. MAQS: a personalized mobile sensing system for indoor air quality monitoring. In Proceedings of the 13th international conference on Ubiquitous computing, UbiComp '11, pages 271–280, New York, NY, USA, 2011. ACM.

[22] Yifei Jiang, Xin Pan, Kun Li, Qin Lv, Robert P. Dick, Michael Hannigan, and Li Shang. ARIEL: automatic wi-fi based room fingerprinting for indoor localization. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp '12, pages 441–450, New York, NY, USA, 2012. ACM.

[23] D. Kahneman. A perspective on judgment and choice: mapping bounded rationality. American Psychologist, 58(9), September 2003.

[24] Daniel Kahneman. Thinking, Fast and Slow. Farrar, Straus and Giroux, October 2011.

[25] Jong Hee Kang, William Welbourne, Benjamin Stewart, and Gaetano Borriello. Extracting places from traces of locations. ACM SIGMOBILE Mobile Computing Communication Review, 9(3):58–68, 2005.

[26] Donnie H. Kim, Jeffrey Hightower, Ramesh Govindan, and Deborah Estrin. Discovering semantically meaningful places from pervasive RF-beacons. In ACM Ubicomp, pages 21–30, 2009.

[27] Donnie H. Kim, Younghun Kim, Deborah Estrin, and Mani B. Srivastava. SensLoc: sensing everyday places and paths using less energy. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10, pages 43–56, New York, NY, USA, 2010. ACM.

[28] M.B. Kjaergaard, M. Wirz, D. Roggen, and G. Troster. Mobile sensing of pedestrian flocks in indoor environments using wifi signals. In Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on, pages 95–102, 2012.

[29] Mikkel Baun Kjærgaard, Sourav Bhattacharya, Henrik Blunck, and Petteri Nurmi. Energy-efficient trajectory tracking for mobile devices. In Proceedings of the 9th international conference on Mobile systems, applications, and services, MobiSys '11, pages 307–320, New York, NY, USA, 2011. ACM.

[30] Mikkel Baun Kjærgaard, Martin Wirz, Daniel Roggen, and Gerhard Tröster. Detecting pedestrian flocks by fusion of multi-modal sensors in mobile phones. In Proceedings of the 2012 ACM Conference on Ubiquitous Computing, UbiComp '12, pages 240–249, New York, NY, USA, 2012. ACM.

[31] MikkelBaun Kjrgaard, Georg Treu, and Claudia Linnhoff-Popien. Zone-based rss reporting for location fingerprinting. In Anthony LaMarca, Marc Langheinrich, and KhaiN. Truong, editors, Pervasive Computing, volume 4480 of Lecture Notes in Computer Science, pages 316–333. Springer Berlin Heidelberg, 2007.

[32] John Krumm, Gerry Cermak, and Eric Horvitz. Rightspot: A novel sense of location for a smart personal object. In Anind K. Dey, Albrecht Schmidt, and Joseph F. McCarthy, editors, UbiComp 2003: Ubiquitous Computing, 5th International Conference, Seattle, WA, USA, October 12-15, 2003, Proceedings, volume 2864 of Lecture Notes in Computer Science, pages 36–43. Springer, 2003.

[33] John Krumm and John C. Platt. Minimizing Calibration Effort for an Indoor 802.11 Device Location Measurement System. In Microsoft Research, Technical Report MSR-TR-2000-12, 2003.

[34] Kari Laasonen, Mika Raento, and Hannu Toivonen. Adaptive on-device location recognition. In Pervasive, pages 287–304. 2004.

[35] Dik Lun Lee and Qiuxia Chen. A model-based wifi localization method. In Proceedings of the 2nd international conference on Scalable information systems, InfoScale '07, pages 40:1–40:7, ICST, Brussels, Belgium, Belgium, 2007. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[36] Hyuk Lim, Lu-Chuan Kung, Jennifer C. Hou, and Haiyun Luo. Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure. Wireless Netwroks, 2010.

[37] Jialiu Lin, Guang Xiang, Jason I. Hong, and Norman Sadeh. Modeling people's place naming preferences in location sharing. In Proceedings of the 12th ACM international conference on Ubiquitous computing, Ubicomp '10, pages 75–84, New York, NY, USA, 2010. ACM.

[38] Kaisen Lin, Aman Kansal, Dimitrios Lymberopoulos, and Feng Zhao. Energy-accuracy trade-off for continuous mobile device location. In Proceedings of the 8th international conference on Mobile systems, applications, and services, MobiSys '10, pages 285–298, New York, NY, USA, 2010. ACM.

[39] Hong Lu, Jun Yang, Zhigang Liu, Nicholas D. Lane, Tanzeem Choudhury, and Andrew T. Campbell. The jigsaw continuous sensing engine for mobile phone applications. In Proceedings of the 8th ACM Conference on Embedded Networked Sensor Systems, SenSys '10, pages 71–84, New York, NY, USA, 2010. ACM.

[40] Aleksandar Matic, Andrei Popleteev, Venet Osmani, and Oscar Mayora-Ibarra. FM radio for indoor localization with spontaneous recalibration. Pervasive Mob. Comput., 6(6):642–656, December 2010.

[41] N900 hardware power consumption. http://wiki.maemo.org/N900_Hardware_Power_Consumption.

[42] Jeongyeup Paek, Joongheon Kim, and Ramesh Govindan. Energy-efficient rate-adaptive gps-based positioning for smartphones. In Proceedings of the 8th international conference on Mobile systems, applications, and services, MobiSys '10, pages 299–314, New York, NY, USA, 2010. ACM.

[43] Jun-geun Park, Ben Charrow, Dorothy Curtis, Jonathan Battat, Einat Minkov, Jamey Hicks, Seth Teller, and Jonathan Ledlie. Growing an organic indoor location system. In Proceedings of the 8th international conference on Mobile systems, applications, and services, MobiSys '10, pages 271–284, New York, NY, USA, 2010. ACM.

[44] Hyojeong Shin, Yohan Chon, and Hojung Cha. Unsupervised construction of an indoor floor plan using a smartphone. Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on, 2012.

[45] Steven A. Sloman. The empirical case for two systems of reasoning. Psychological Bulletin, 119, 1996.

[46] Stephen P. Tarzia, Peter A. Dinda, Robert P. Dick, and Gokhan 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, pages 155–168, New York, NY, USA, 2011. ACM.

[47] Arvind Thiagarajan, Lenin Ravindranath, Hari Balakrishnan, Samuel Madden, and Lewis Girod. Accurate, low-energy trajectory mapping for mobile devices. In Proceedings of the 8th USENIX conference on Networked systems design and implementation, NSDI'11, pages 20–20, Berkeley, CA, USA, 2011. USENIX Association.

[48] Arvin Wen Tsui, Yu-Hsiang Chuang, and Hao-Hua Chu. Unsupervised learning for solving RSS hardware variance problem in wifi localization. Mob. Netw. Appl., 14(5):677–691, October 2009.

[49] W. T. Tutte. How to draw a graph. London Math, 1963.

[50] Waqas ur Rehman, Eyal de Lara, and Stefan Saroiu. Cilos: a cdma indoor localization system. In Proceedings of the 10th international conference on Ubiquitous computing, UbiComp '08, pages 104–113, New York, NY, USA, 2008. ACM.

[51] U.S. Environmental Protection Agency Green Building Workgroup. Buildings and their impact on the environment: A statistical summary, 2009.

[52] Alex Varshavsky, Eyal de Lara, Jeffrey Hightower, Anthony LaMarca, and Veljo Otsason. Gsm indoor localization. Pervasive Mob. Comput., 3(6):698–720, December 2007.

[53] He Wang, Souvik Sen, Ahmed Elgohary, Moustafa Farid, Moustafa Youssef, and Romit Roy Choudhury. No need to war-drive: unsupervised indoor localization. In Proceedings of the 10th international conference on Mobile systems, applications, and services, MobiSys '12, pages 197–210, New York, NY, USA, 2012. ACM.

[54] Guang Yang. Discovering significant places from mobile phones a mass market solution. In Richard Fuller and XenofonD. Koutsoukos, editors, Mobile Entity Localization and Tracking in GPS-less Environnments, volume 5801 of Lecture Notes in Computer Science, pages 34–49. Springer Berlin Heidelberg, 2009.

[55] Zhenyun Zhuang, Kyu-Han Kim, and Jatinder Pal Singh. Improving energy efficiency of location sensing on smartphones. In Proceedings of the 8th international conference on Mobile systems, applications, and services, MobiSys '10, pages 315–330, New York, NY, USA, 2010. ACM.