Diss. ETH No. 20109

Robust Indoor Positioning through Adaptive Collaborative Labeling of Location Fingerprints

A dissertation submitted to
ETH ZURICH

for the degree of
Doctor of Sciences

presented by

Philipp Lukas Bolliger

Dipl. Informatik-Ing. ETH
born March 27, 1978
citizen of Küttigen (AG), Switzerland

accepted on the recommendation of

Prof. Dr. Friedemann Mattern, examiner
Prof. Dr. Marc Langheinrich, co-examiner
Prof. Dr. Kurt Rothermel, co-examiner

2011

To my parents

Abstract

Location-aware computing has become one of the most publicly visible results of ubiquitous computing research, with small, low-power GPS modules being incorporated in an ever increasing number of consumer devices. While GPS systems work well in outdoor environments, the limited propagation characteristics of GPS satellite signals require alternative solutions for positioning and navigating inside buildings. Radio location fingerprinting is one of the most promising indoor positioning mechanisms as it allows positioning using signal characteristics of existing wireless communication networks (e.g., a WiFi installation) and thus requires no dedicated localization infrastructure to be installed. However, location fingerprinting typically requires a costly setup phase, in which signal fingerprints are manually mapped to individual locations. Moreover, since radio signals change and fluctuate over time, map maintenance requires continuous recalibration.

In this thesis, we introduce the concept of user-contributed, collaborative fingerprint labeling to address the problems of radio map setup and map maintenance in location fingerprinting systems. Instead of manually creating an initial map prior to deployment, we propose a method to harness the inputs of all users to collaboratively create and subsequently maintain an accurate map of indoor radio fingerprints. We offer a novel user interface approach to simplify the solicitation of user-generated labels that rely on labeling long-term measurements, not just second-long snapshots, and provide algorithms that are able to accurately position a device based on such user-generated labels.

To alleviate accuracy degradation caused by signal variation, we introduce a new concept called "asynchronous interval labeling" that addresses these problems in the context of user-generated labels. By using an accelerometer to detect whether a device is moving or stationary, the system can continuously and unobtrusively learn from all radio measurements during a stationary period, thus greatly increasing the number of available samples. Movement information also allows the system to improve the user experience by deferring labeling to a later, more suitable moment. Experiments with our system show considerable increases in data collected and improvements to inferred location likelihood, with negligible overhead reported by users.

Zusammenfassung

Durch den immer weiter verbreiteten Einsatz von kleinen, energiesparenden GPS-Modulen in einer Vielzahl von mobilen Geräten sind ortsbezogene Anwendungen, die von der aktuellen Position eines Benutzers Gebrauch machen, immer mehr in den Fokus der wissenschaftlichen Forschung gerückt und wurden so zu einem der meistbeachteten Themen im Bereich Ubiquitous Computing. GPS-Module haben allerdings den inhärenten Nachteil, eine direkte Sichtverbindung zu Satelliten zu benötigen und funktionieren daher nicht, oder nur sehr schlecht, innerhalb von Gebäuden. Für den Einsatz innerhalb von Gebäuden ist also eine andere Lokalisierungsmethode nötig. Um dieses Problem zu lösen, gilt vor allem die Methode der Funkortung mittels "Fingerprinting" als sehr vielversprechend, da diese Methode Funksignale von bereits vorhandenen, drahtlosen Kommunikationsnetzen wie z.B. WiFi nutzen kann. Solche Systeme haben allerdings den Nachteil, dass sie typischerweise eine manuelle Erfassung der "Fingerprints", also der an einem Standort charakteristischen Funksignale, benötigen. Diese Arbeit ist zeitaufwendig und teuer. Erschwerend kommt hinzu, dass sich Funksignale über die Zeit ändern, was ein erneutes Erfassen nach sich zieht.

In der vorliegenden Arbeit stellen wir ein neuartiges Konzept des Fingerprinting vor, welches auf dem Prinzip der benutzergestützten, kollaborativen Erfassung von Funksignalen basiert. Anstatt die Abbildung von Funksignal-Charakteristik zu Ort manuell zu erfassen, schlagen wir vor, die Eingabe den Benutzern zu überlassen. Ohne zusätzlichen Aufwand für den Benutzer kann dieser damit während der Benutzung des Sys-

tems fortlaufend neue Messungen zur Abbildung beitragen. Um diese Eingabe für die Benutzer so einfach und unaufdringlich wie möglich zu gestalten, schlagen wir vor, die Funksignalmessungen, statt wie bisher während wenigen Sekunden, im Hintergrund und über mehrere Minuten zu machen.

Um das Problem der durch Signalschwankungen verursachten Abnahme der Genauigkeit zu schmälern, stellen wir ein neues Konzept vor, das wir "asynchronous interval labeling" nennen. Dieses Konzept erlaubt es, den Benutzer erst dann um die Zuweisung eines Funksignals zu einem Ort zu bitten, wenn die Wahrscheinlichkeit, ihn nicht bei einer aktiven Arbeit zu stören, am grössten ist. Hierzu verwenden wir den in vielen mobilen Geräten vorhandenen Beschleunigungssensor um festzustellen ob der Benutzer sich fortbewegt oder stationär an einem Ort bleibt. Dadurch weiss das System, wann der Benutzer an ein und demselben Ort bleibt und kann so im Hintergrund weitere Funksignalmessungen machen. Das Resultat ist eine Abbildung, welche eine um Grössenordnungen höhere Anzahl an Messungen enthält. Dies wiederum führt, in Kombination mit den von uns entwickelten Algorithmen, zu einer gesteigerten Genauigkeit und Präzision.

Acknowledgements

First and foremost, I would like to thank Prof. Friedemann Mattern. It was Friedemann who encouraged me to pursue a PhD and gave me the great opportunity to do so in his research group a few years ago. Throughout my doctoral studies, Friedemann gave me all the freedom and room to develop my own ideas while making sure that I was able to develop the required skills in several very interesting industry research projects. He supported me in all matters, be they academical or personal. In my personal experience after working several years with Friedemann, I got to know him as one of the most caring, generous and enabling persons. Dear Friedemann: Thank you!

I would also like to express my profound gratitude to Prof. Marc Langheinrich. Marc supervised and guided me. Marc was a true enabler for many of my works and encouraged me to follow-thru in times of disbelief. Often times, he pushed me to polish and publish my work, never without being very helpful, always leading to a successful publication. It was also Marc who made my summer internship at the Palo Alto Research Center (PARC) possible. This summer in Silicon Valley has been one of the most prolific and interesting times in my life. Besides all that, Marc has always been, and still is, a true friend and a great role model.

Very special thanks go to my past coworkers at the Distributed Systems research group, all of them determined and inspiring in their work. I felt at home as well as challenged from the very first day, a combination that made my fruitful endeavor possible in the first place. No matter how silly an idea looked at first sight, there was always someone interested in listening and willing to share criticism. In particular, I would like to thank Jonas Wolf and Benedikt Ostermaier for the many interesting and inspiring discussions we had when sharing an office, and also for being good sports and laughing at my sometimes silly jokes. Benedikt and Jonas where the first with whom I discussed the idea of smart plant care and it certainly was the immediate enthusiasm of these two that lead to the Koubachi project. A very special thank you goes to Moritz Köhler. It was Moritz who co-authored one of my more publicly discussed papers and it was Moritz who showed me the potential of the Koubachi project, a discussion which lead to the start-up we are proud to run today.

Part of the research that has been undertaken in this thesis was integral with my internship at PARC. I would like to thank Bo Begole, Kurt Partridge, and Maurice Chu for their insights and support! The rapid development and the success of the Redpin open-source project certainly wouldn't have been possible without the many lab, bachelor and master theses that I was lucky enough to supervise. I therefore would like to thank Pascal Brogle, Diego Browarnik, Andreas Kamilaris, Davide Spena, Simon Tobler, and last but not least Luba Rogoleva.

Further, I would like to thank my good friends—René Bachmann, Stefan Nägeli, Andy Sutter, and Yvonne-Anne Pignolet—for getting me into and being with me on the many stages of this journey. Finally, I am most grateful to my parents for supporting, encouraging and pushing me throughout this thesis and throughout my life. I could never have done what I did, never pursued my way without the love and support you gave me. Thank you!

Table of Contents

1 Introduction **1**

1.1 Motivation 3

1.2 Location Fingerprinting 8

1.3 Challenges of Indoor Positioning 10

1.3.1 Performance 10

1.3.2 Cost 12

1.3.3 Scalability 14

1.3.4 Signal Variation 16

1.3.5 Sensor Variation 18

1.3.6 Security and Privacy 19

1.3.7 User Interface 21

1.4 Goals and Hypotheses 21

1.5 Summary of Contributions 23

1.6 Thesis Overview 25

2 Background **27**

2.1 Location Information 28

2.1.1 Representation 28

2.1.2 Attributes 29

2.2 Location Models 32

2.2.1 Geometric Location Models 33

2.2.2 Symbolic Location Models 34

2.2.3 Hybrid Location Models 37

2.3 Positioning Technologies 39

2.3.1 Methods 39
2.4 Location Fingerprinting 45
2.4.1 Roles and Responsibilities 45
2.4.2 Estimation Methods 49
2.4.3 Training the Radio Map 50
2.5 Conclusion 51

3 WiFi Signal Characteristics 53
3.1 Controlled Study 54
3.1.1 Setup 54
3.1.2 Experiment 57
3.1.3 Results 58
3.2 User-Driven Study 61
3.2.1 Setup 62
3.2.2 Experiment 64
3.2.3 Results 65
3.3 Conclusion 73

4 Collaborative Labeling 75
4.1 Building Principles 77
4.2 Harnessing User Collaboration 80
4.2.1 Crowdsourcing 80
4.2.2 Wikipedia 82
4.2.3 Folksonomy 84
4.2.4 Games with a Purpose (GWAP) 86
4.2.5 Collaborative Mapping 87
4.2.6 Location Sharing 88
4.2.7 Discussion 89
4.3 Redpin 91
4.3.1 Redpin in Action 94
4.3.2 Architecture 97
4.3.3 Redpin Server 98
4.3.4 Mobile Clients 105
4.3.5 Preliminary Evaluation 109

4.4 Conclusion 112

5 Interval Labeling 115

5.1 Building Principles 116

5.2 Detecting Stationary State 119

5.2.1 Motion Detector 120

5.3 The PILS System 121

5.3.1 Hardware and Setup 123

5.3.2 Probabilistic Estimation Method 125

5.3.3 Evaluation 127

5.4 Optimizing Location Estimation 133

5.4.1 Method Comparison 134

5.5 Conclusion 139

6 Conclusion 143

Bibliography 151

Curriculum Vitae 169

The most exciting phrase to hear in science, the one that heralds the most discoveries, is not "Eureka!" but "That's funny..."

– Isaac Asimov

1 Introduction

Location has undeniably become a hot topic in the consumer market, with sales of GPS enabled smart phones skyrocketing. Around the simple service of positioning, an entire industry for *location-based* services is gradually taking shape, offering not only navigational services, but also shopping advice, tourism, and localization of friends and family members. Developments in both Asia and the US (E-911) have positioned mobile phones with integrated positioning technology at the forefront of location based service provisioning in many markets.

Determining the position of a user and her device respectively has been a hot topic in many different disciplines of computer science for decades [5, 65, 87, 141]. This was and still is particularly true for the field of *ubiquitous computing* (Ubicomp). It was this area of research that showed to the (enterprise) world how valuable location information can be. And yet it took many years for commercial software developers and solution providers to build upon this knowledge and provide practical solutions. Nevertheless, the plethora of services that provide, collect, analyze and augment location information can certainly be seen as proof for the promise made years ago. Google, Yahoo, Microsoft, Facebook — almost every big player in the Web 2.0 economy introduced a *location* or *places* service. Moreover, small start-up firms such as Foursquare, Dopplr, Gowalla or Brightkite all launched distributed, mobile solutions based on a user's location thus starting what infamously became known as the "location war".

However, all these systems, players and solutions work with and are built using systems that provide only very coarse positioning, which, in practice, allows for outdoor use only. Hence, the question remains:

- Why are systems that allow for accurate indoor positioning still used in labs and special purpose setups only?
- Why is it that what started in the area of Ubicomp until today has not prevailed where it was meant to be?

The answers to the above mentioned questions are of course neither simple nor obvious and there are many different aspects that need consideration. Yet, we believe and explain throughout this thesis that the core issue that hindered commercial solutions and hence adoption is high cost of providing indoor positioning information. We start by elaborating on the motivation of indoor positioning followed by a short overview of the problems and challenges in this first chapter. We also shortly introduce the main concepts and present our hypotheses. Finally, we will summarize our contributions and give an outlook of the following chapters.

1.1 Motivation

To our understanding, it was Mark Weiser who pioneered the idea of using location information to create a whole new user experience. In his many papers about Ubicomp [156, 158, 159, 160, 161, 162, 163], he imagined and sketched a world of what he called *Calm Computing*, a world where technology moves from center to background — it's there, it supports the user yet he doesn't notice. All with the one goal to *"put us at home, in a familiar place"*, or as he put it: *"[Ubicomp] will bring information technology beyond the big problems like corporate finance and school homework, to the little annoyances like 'Where are the car-keys?', 'Can I get a parking place?', and 'Is that shirt I saw last week at Macy's still on the rack?"'*[162].

When Mark Weiser outlined "Some Computer Science Issues in Ubiquitous Computing" in his seminal paper of 1993 [160], he insisted that applications are "the whole point of ubiquitous computing" and noticeably stated the ability to locate people as one of the defining examples. Previous work by Olivetti Research Labs in Cambridge [155] has already shown the feasibility of building indoor positioning systems and it was Mark Weiser who saw the potential of this technology and its many uses [160]: from video annotation to updating dynamic maps, controlling locks and lights, automatic phone forwarding, locating an individual for a meeting, and watching general activity in a building to feel in touch with its cycles of activity — just to name a few.

Consequently, location-aware computing has become one of the publicly most visible results of Ubicomp research and ever since, the location of a user or a device is a very meaningful and significant information for many applications in Ubicomp [71, 140]. It certainly is the most prominent contribution when it comes to determining a user's context or activity. Starting with the idea of collecting data to determine a user's context, another field of research was born: *context-aware computing applications* [1, 16, 44, 125]. In their paper "Towards a Better Understanding of Context and Context-Awareness"[45], Dey and Abowd explicitly list location information as one of the four primary data categories that contribute to a user's context — alongside with time, identity, and activity. Thus, as a user's activity and location became fundamental for many Ubicomp applications, research has been focused more deeply in the fields of activity recognition [8, 12, 84, 150] and of course location awareness.

One recent driver of this development has been the emergence of small, low-power Global Positioning System (GPS) modules being incorporated in an ever increasing number of consumer devices. But, as we will see in more detail later on, while GPS systems work well in outdoor environments, the limited propagation characteristics of GPS satellite signals require alternative solutions for positioning and navigation inside buildings. Evidence of this development can also be found in the sudden rise of web-based positioning services like *Navizon*[1] or *Loki*[2] that allow to determine your location using GSM and WiFi readings. Even more so, many big players in the consumer market, from Apple to Google, provide location-based services for their mobile platforms, developing and running the required software in-house.

Although Marc Weiser's vision of Ubicomp is finally starting to become reality, the missing of one key technology still hinders a broad emergence of such applications, namely an easily available positioning system. But why? A quick search on Google Scholar[3] reveals more than ten thousand papers on indoor location and positioning respectively. Yet, only very few of the proposed and prototypically built systems have been implemented in publicly available systems. Of course, one of the reasons for this must be that no commercial enterprise or start-up has found a feasible business model. But why was no one trying to come up with one? After evaluating the built systems in detail, we came to the conclusion that the main reason for this is that the proposed solutions, although being quite accurate, have one thing in common: it is very challenging to deploy them at reasonable cost.

As we will show in detail, there are basically three factors that drive

[1]http://www.navizon.com/
[2]http://www.loki.com
[3]http://scholar.google.com

costs: First, every system that is built to be used ubiquitously has to be compatible with a broad range of different hardware devices, each having very different characteristics and running different operating systems. Second, unlike outdoor use, where in most cases at least basic map data exists in some or another form, in indoor use it is often very complicated and time consuming to get map data or floor plans. Third and most important, in order to get accurate results, almost every indoor positioning system requires an extensive set of data points to train locator algorithms. Thus, despite the fact that commercial systems like Ekahau[4] or UbiSense[5] are very accurate, the costs of installation, maintenance, and in consequence ownership are very high. Another class of indoor localization systems that have been demonstrated to be very accurate are systems that use special hardware (for example, RFID [62], infrared [155], or ultrasound [65]). Although being very accurate, such systems usually require the installation of dedicated hardware that is needed for the localization. The same holds for most commercial systems, as they require one to purchase and install specific hardware, i.e., they cannot be used with portable devices already at hand.

In most indoor environments, GPS does not work for one very simple reason: it is just not possible to receive the signal broadcasted by the GPS satellites. The receivers used in todays devices are not sensitive enough while the building structure is quite simply to strong and thus absorbs the data signal. Consequently, indoor positioning systems have to use a different signal source. This basically leaves two choices: either install a new signal source, like for example ultra wide band (UWB) radio signals tags [145], or design the indoor positioning system such that it can make use of (radio) signals that can already be found. Obviously, if low-cost is a concern, only the latter of these two approaches is an option.

As WiFi became a quasi standard for wireless local area networks over the last decade, with ever more handheld devices such as netbooks, smartphones or tablet computers having WiFi network access by de-

[4] http://www.ekahau.com

[5] http://www.ubisense.net

fault, most modern indoor positioning systems proposed over the last years make use of 802.11, i.e., WiFi signals to localize devices. This seems feasible as radio signals from at least a few WiFi access points can almost always be measured where people work and live (e.g., [143] and [32]). In addition, WiFi signals can be used to estimate a users position indoors with an accuracy that is generally sufficient for most location-based systems [88]. In this respect, WiFi localization has shown great promise for indoor positioning, yet has not achieved ubiquitous commercial success. One difficulty has been the construction of an accurate mapping between signal strength patterns and physical locations. As we will show in more detail later on, the signal strength patterns depend not only on the distances between WiFi radios, but also on other factors such as the positions of physical objects that reflect, partially absorb, or even block signals. This complication may be overcome, at least to some extent, by either performing calculations with detailed models of the environment, or by collecting a dense dataset of fingerprints and their associated true locations [5].

As we will explain in the next section, research in the past few years has shown that radio location fingerprinting, a mechanism where location is determined by comparing received signal strength to a set of known patterns, i.e., the *fingerprint*, is the most promising approach to determine the location of a mobile device in various indoor settings with very different signal propagation characteristics. Hence, a lot of research focused on solving the problems that arise when using the received signal strength (RSS) to fingerprint a location, such as detecting and modeling line-of-sight obstructions [118], absorption by humans, or reflection on walls. In addition, a lot of effort was spent on finding accurate and robust algorithms to select a known fingerprint given a current RSS measurement, for example [61, 87, 95, 108, 112]. We elaborate on these challenges later in this chapter and cover related work more extensively in the next chapter.

Although having many advantages, *location fingerprinting* has one big drawback. In order to get accurate results, it is necessary to train the system with as many radio signal readings as possible. This training phase is often described as *offline phase*, as most systems only allow to perform this task before actual use or within designated maintenance phases. Naturally, these systems are only as accurate as this offline phase has been detailed. Moreover, collecting labeled fingerprint samples can be tedious. Signal readings must be collected every few meters or so, with pauses of tens of seconds at each position to get an accurate reading. This process must be repeated if the infrastructure or environment changes substantially. Commercial deployments usually conduct such surveys as part of deployment, however in some installations, such as private homes, consumers may not have the patience for this process.

Academic systems that have been made publicly available like Place Lab [36, 143] on the other hand are not easy to setup[6] and require one to train the system afterwards. In addition, as these systems try to optimize the accuracy of the localization, which increases with the quality of the trained fingerprints, the offline phase is typically very time consuming. The COMPASS system for example is able to determine the position with an average error distance of less than 2.05 meters [87] using WiFi RSS. Yet, in order to achieve this accuracy, it was necessary to measure at grid-aligned points with a spacing of only 1 meter and take measurements in 8 different directions at each point. Even in a very small building with a floor area of, for example, 125 m^2, the training phase would take more than 4 hours[7].

The biggest issue with having a designated training phase is that is has to be repeated whenever the environment changes, for example due to a replaced access point. However, such accuracy is only feasible when the measured signal strengths fluctuate only very slightly. Our own measurements (see chapter 3 for details) showed that the RSS of GSM signals can change up to 30% in only a few dozen seconds and the RSS of WiFi access points can even slip more than 50% within only one hour. Furthermore, the RSS of WiFi access points depends heavily

[6]It took one of our students almost two days to get the system running on just one mobile phone.

[7]This is, if we account 20 seconds per measurement, which is about the amount of time we experienced in our own experiments.

on whether humans are in the line of sight as the human body absorbs electromagnetic radiation quite well [101]. Hence, in rooms where the number of people is high and changes frequently, it seems unlikely that an accuracy of under 2 meters can be achieved. Lastly, second-by-second signal fluctuations mean that the fingerprint stored with a label may not match future measurements. Subsequently, a labeled fingerprint would need to be collected over an interval of several tens of seconds, much as it is done during formal calibration stages.

Before discussing these issues and challenges in detail, we first identify and explain the basic building blocks of any location fingerprinting system.

1.2 Location Fingerprinting

Radio location fingerprinting is one of the most promising indoor positioning mechanisms, as it allows positioning using signal characteristics of existing wireless communication networks (e.g., a WiFi installation) and thus requires no dedicated infrastructure to be installed. In recent research, it was Mikkel Baun Kjærgaard who studied the many issues and advantages of this approach in his thesis "Indoor Positioning with Radio Location Fingerprinting" [92]. In particular his work on a taxonomy for radio location fingerprinting [91] helped to understand and define the methods and components involved.

Figure 1.1 illustrates that every location fingerprinting system basically consists of two main components: the *radio map* and the *estimation method*. The radio map consists of a database of known fingerprints. In its most basic form, this can be a list of measurement tuples associated with a location. The measurement tuple contains the identifier of the signal source, for example the MAC address of a WiFi access point, along with the received signal strength (RSS) observed when recording the measurement. The estimation method is any algorithm that allows to map an observed measurement to the corresponding location in the radio map. As most estimation methods use mechanisms and algorithms known from machine learning, it can be said that the more measurements a radio map contains, the more accurate the estimation method is going to work.

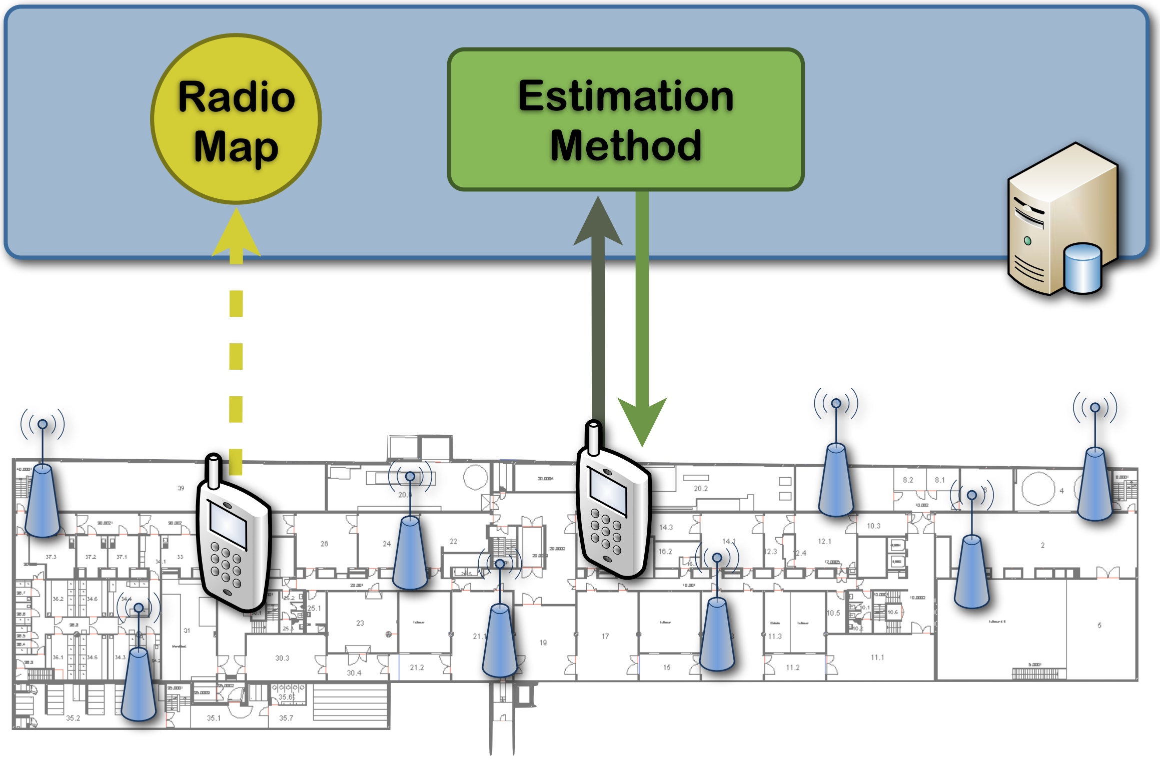


Figure 1.1: Basic elements and principle of location fingerprinting.

Hence, the method of location fingerprinting using radio signals assumes that the pattern of *mean signal strengths* received in one location differs from the pattern observed in another location. Unfortunately, various effects, including interference from interposing static objects as well as reflections off neighboring objects, make the relationship between the signal means and location difficult to predict in practice [74, 96, 109, 128]. Less well-documented are sources of variance in the signal, although there has been some work studying these effects over a one day period [83]. We cover these issues in more detail in Chapter 3.

Usually, the radio map is setup and organized on a central server. However, it may as well be distributed, as we will explain in the next chapter. In every case however, measurements are taken using mobile devices, may this be a laptop, a smartphone, or even a tiny little sensor node. As all these devices have very different antennas and (WiFi) chipsets, the RSS values observed (usually reported as power ratio in decibels of the measured power referenced to one milliwatt, or dBm in short) differs vastly between the different devices (see Section 1.3.5 for details). Although this issue has been addressed [89], most proposed systems use only one specific device.

In comparison to other (indoor) positioning mechanisms, like using the time or angel of arrival of a received signal, location fingerprinting requires a training phase, i.e., the radio map must be established. This inevitable learning process requires that measurements are taken in every place or room. And as more measurements yield better results, it is usually not sufficient to only measure once. Ideally multiple measurements per location are taken at different times of the day over many days. Which brings us to the challenges of indoor positioning.

1.3 Challenges of Indoor Positioning

When discussing location systems for indoor positioning, a broad set of issues and challenges arise. Consequently, many survey papers have been written over the last few years that deal with the different characteristics and issues of indoor positing systems. Most of these papers, as for example [60, 70, 71, 91], propose a classification or taxonomy of location systems. In the following, we summarize the most prominent and prevailing challenges of indoor positioning systems. We will focus on issues that are typical for systems that make use of location fingerprinting. The following analysis shall offer a coarse overview of current challenges and issues. We cover related work in more detail in Chapter 2. To clarify the many different aspects, we classify the issues by those attributes mostly used to assess and evaluate indoor positioning systems.

1.3.1 Performance

Depending on the type of system and its main purpose, there are different attributes to be considered when evaluating the performance of an indoor positioning system. For example, a system that was built to track fast moving objects has to be, first an foremost, responsive, i.e. the delay of measuring and calculating positions of the estimated target must be short. Yet, in order for an indoor positioning system to be considered good, we expect it to report locations accurately and consistently from measurement to measurement [71]. In this respect, accuracy and precision are the two main performance parameters used to evaluate an indoor positioning system, where accuracy usually means the average error distance, and precision means the success probability of position estimations with respect to predefined accuracy. For example, inexpensive GPS receivers are capable of locating positions within 5 meters for approximately 90 percent of measurements. Thus, the distance, 5 meters, denotes the accuracy of the position while the percentage, 90 percent, denotes precision. We elaborate on these issues in Section 2.1.2.

Usually, there is a trade-off between the performance of a positioning system and its cost: the higher the performance requirements, the higher the costs. For instance, one can improve the accuracy of infrared-based positioning systems by adding special filters to lower the influence from florescent light [60]. But adding these filters increase the price of the whole system. Or consider motion-capture systems that support high resolution, real-time target acquisition, as for example trakSTAR by Ascension[8]. Such systems allow for centimeter-level spatial positioning and precise temporal resolution. On the other hand, a system that provides personalized weather forecasts can do with an accuracy of a few kilometers. Consequently, we must evaluate the performance of location-sensing systems by determining whether they are suitable for a particular application [70, 71]. Therefore, the challenge is to get sufficient accuracy at reasonable cost. Regarding applications for Ubicomp, we are interested in queries such as “Where is that meeting that I am supposed to attend at 4 o’clok?” or “How do I get to Walter’s office?”. Thus, although it is possible to achieve accuracy of about 2 meters [87], it turns out that for almost all applications in Ubicomp that involve persons it is sufficient

[8] http://www.ascension-tech.com/

to provide room-level precision. Moreover, as Hightower found in 2004 [69], it is beneficial for most Ubicomp applications to use what he called *place*, i.e., a human-readable label of a position.

Although this finding favors location fingerprinting systems, it also reveals one of the biggest issues of fingerprinting. In order to associate a position with a place, the exact location of a place has to be known. Moreover, all users must have the same perception of a specific place. This holds particularly for fingerprinting systems that require or allow for manual collection of fingerprints. If the collector and the user of the system do not share their perception of a place, the user will almost always be disappointed by the performance of the system. Hence, a location fingerprinting system that employs manual collection of location labels is per-se error prone and its performance will vary. Usually, this problem is overcome by taking multiple measurements for the same place at slightly different positions or with slightly different orientation. But although taking more and more measurements might improve the performance of a location fingerprinting system, the cost of training and maintaining the radio map will increase.

1.3.2 Cost

One of the factors that make up the cost of an indoor positioning system, in particular a system that uses location fingerprinting, is the time needed to build and maintain the radio map. Other time factors are the effort required to install and administer a system or the battery lifetime of the used devices. Space costs involve the extent and complexity of installed infrastructure and the used hardware's size and form factor. Capital costs on the other hand include factors such as the price of the devices used and the required infrastructure. In addition, capital costs also include the salaries of support and maintenance personnel [70, 71]. The best-known positioning system GPS for example relies on a very large and pricy infrastructure, which is expensive and complicated to install and to maintain.

When assessing the cost of indoor positioning systems, it is crucial to consider the above listed factors over the lifetime of the system. For a detailed analysis, we suggest to break-down the cost into three phases: total cost of installation, total cost of use, and total cost of maintenance. As explained, one of the primary factors that make up the total cost of installation is given by the choice of hardware. If a system requires special hardware, for example special tags [155], antennas [145] or even WiFi access points with special capabilities [34], the acquisition costs are very high. Another factor that is often disregarded is the provisioning of maps. Unlike systems build for outdoor use, where maps have been created for many purposes already hundreds of years ago, appropriate indoor maps and floor plans are often not available.

The total cost of use comprises the cost of all resources required during use. Hence, it mainly depends on the technology chosen. For example, the cost of using GPS is relatively high as getting positioning information consumes a lot of energy and usually takes more than 10 seconds. In consequence, the user will have to wait for the GPS system to deliver the result and, on top of that, has heavily reduced battery lifetime. Or consider a location fingerprinting system that uses WiFi radio signals. In order to get a measurement that can then be compared to the radio map, the system has to scan the signal environment. Although being more resource-saving than GPS, this scan also requires both time and energy. In addition, if the scan is executed actively, the network interface may not be used to transfer data while scanning the network, i.e., the user faces additional opportunity cost and is forced to choose between either having a fast and accurate positioning system or transferring data. In fact, the concurrent use of the network subsystem is one of the biggest challenges of using WiFi for location fingerprinting. Consequently, most proposed systems do not deal with this problem, with the exception of [86].

To assess the total cost of maintenance is difficult as many systems, which have been proposed for indoor positioning, have not been used long enough to put a number on this cost factor. Still, the different mechanisms allow some general observations. Systems that rely on special tags and antennas like Ubisense for example, require recalibration about every six months. As this calibration process can only be executed by trained personnel, the cost of maintenance is very high. When it comes to systems that make use of fingerprinting, the by far biggest cost factor is the time and resources required to keep the radio map up-to-date. As we will show, the signal environment is subject to long- and short-term fluctuations. Thus, the radio map has to be update continuously. If the training of the radio map is to be done manually or worse, by special personnel, the costs for updating the radio map will be very high.

Another cost factor when using WiFi for location fingerprinting that adds to both the total cost of installation and the total cost of maintenance is the device-inherent difference of reported signal strength. Although most devices report the RSS in dBm, the reported value is not standardized and depends on the combination of antenna, network adapter, and operating system. For example, while a smartphone reports $-47dBm$ at a certain position, a laptop may report $-65dBm$. We elaborate on this factor and possible solutions in Section 1.3.5. In short, the solution to this problem that yields the best accuracy is to manually take measurements at reference positions and create profiles for every device. As said before, there is always a trade-off between the performance of a positioning system and its cost.

1.3.3 Scalability

As only very few indoor positioning systems have been deployed on a large-scale, the issue of scale and scalability is been discussed scarcely. Gu and Lo for example define scalability as *"the number of objects that an IPS [indoor positioning system] can locate with a certain amount of infrastructure devices and within a given time period"* [60]. Hightower et al. [71] on the other hand use the term scale to describe the coverage area per unit of infrastructure and the number of objects a system is able to locate per unit of infrastructure. Although being different, both notions capture the limiting factors of time, required or available infrastructure and number of devices. In this respect, time is, once more, of crucial importance. As we have seen before (see Section 1.3.2), the bandwidth available for sensing objects or devices is limited. Any radio-frequency-based system is only able to tolerate a maximum number of connections before the channel becomes congested [70]. Beyond that threshold limit either a loss of accuracy will occur or the latency in determining the position will increase as the system is forced to scan and calculate the device's position less frequently.

An indoor positioning system may be built to work all over the world, within city limits, throughout a campus, just in a particular building, or even just in one room and systems can often expand to a larger scale by increasing the infrastructure. For instance, a simple tag system like the Active Badge location system [141, 155], which locates tags in a single room, can be used on a campus by equipping all buildings with the required infrastructure. But barriers to scaling a positioning system do not only include infrastructure but also middleware complexity and finally computing power requirements demanded of the necessary servers.

In respect of systems that employ location fingerprinting, the issue of scaling is predominantly a problem of server performance. This is for systems that are designed to use a predefined server for storing the radio map and to execute the estimation method. This said, it is possible to distribute the radio map to the client devices. The consequence of this is that either every device has to learn the mappings and thus every place itself, or the system has to provide a mechanism that enables the terminal devices to exchange and propagate the mappings. Although both approaches have been applied [2, 100, 127, 143] with valuable results, using the device to store the radio map and to execute the estimation method entails two problems: First, as a device used for indoor positioning is mobile, it is not as powerful, i.e., it only has limited resources. Consequently, the radio map can only grow to a certain size and the execution of locally run localization algorithms may take a long time. Second, the exchange of radio mappings between these devices has to be taken care of by the wireless network adapter and is thus slow. Moreover, using the wireless network adapter for data communication implicates that it can not be used to scan the network. Hence, the device can either use the adapter for positioning or for data communication. Consequently, most systems rely on a (central) server infrastructure to store the radio map and to perform the positioning. Storing the former does usually not pose too much of a problem, as today's high-performance, distributed database systems are capable enough to handle even very big radio maps and thousands of concurrent users. Providing location lookup (i.e., actual position calculations) that operate on very large radio maps within fractions of a second is a big challenge, however.

As we will explain in this thesis, many different algorithms have been proposed for position estimation. These methods have very different characteristics. For example, the well-known and often used k-nearest-neighbor method allows to add new mappings, i.e., adding new measurements to a fingerprint, without significant delay. The position estimation on the other hand may take long, as the algorithm must access all entries in the radio map. Hence, the bigger the radio map, the bigger the delay. Another often-used estimation method, the support vector machine (SVM), shows opposite characteristics. While calculating the position estimate is dealt with in very short time, adding mappings to the radio map potentially takes very long. This is because SVM is a machine-learning algorithm, which supports multi-class classification. Thus, it is necessary to retrain the classification model every time a new mapping is added to the radio map. Ideally, an estimation method features the favorable characteristics of both of these algorithms, no matter how big the radio map is. Consequently, one of the biggest challenges, which, to the best of our knowledge, has not been tackled until today, is to integrate or combine known and proven algorithms in a server infrastructure able to scale for world-wide use.

1.3.4 Signal Variation

Positioning systems designed and built for specific purposes or applications, such as tracking of fast moving objects or augmented reality often have high demands on accuracy. Such systems require the installation of specific and expensive hardware that, for example make use of ultra wide-band radio signals (UWB). However, for most applications a lower accuracy suffices. Consequently, these systems make use of standard WiFi radio signals instead of UWB. However, WiFi radio signals are subject to absorption, reflection, refraction, multi-path, humidity, temperature, and many other factors. As a result, the signal strength measured fluctuates over time and attenuation correlates poorly with distance, which results in inaccurate and imprecise distance estimates [71]. In addition, when the device measuring RSS is moved, we can observe spatial variations in both large and small scale. Moreover, the signal strength depends on how the signals propagate, i.e., the further away from the source the smaller the measured signal strength. Thus, good systems rely on large-scale spatial variations for accurate indoor positioning [89]. Moreover, for systems that use location fingerprinting, the fact that walls absorb and reflect radio signals yields greater difference in RSS which allow for better separation. And there are many more factors that cause radio signal variations such as, for example, changing weather. To better understand the cause of fluctuations, we thus conducted different long-term studies. We elaborate on setup, procedure and results in Chapter 3.

Small-scale spatial variations, which can be observed when moving the measuring device by as little as one centimeter and in particular temporal variations cause many problems as well. Such small-scale fluctuations are mainly caused by people. The human body is an excellent sink for radio signals. Thus, the presence or even worse movement of only a few people cause heavy signal strength fluctuations. This is best illustrated with a little example: Imagine a location fingerprinting system where the radio map is trained by professional personnel. One of these specialist is assigned to map the RSS in a meeting room. In order not to disturb the working force, the specialist takes the measurements after office hours, when the meeting room is not in-use. The next day, people are informed about the installation of the new positioning system and are eager to try it out during a meeting in the very same room. Despite trying several times, users will never get an accurate position estimate. Unlike last night, when the meeting room was empty, there are now six people sitting at the table, which causes the RSS to fluctuate. Consequently, the measurements taken during the meeting will rarely match the fingerprint recorded by the specialist.

In addition to the causes of signal variations discussed above, we can also observe changes in the radio signal environment over long periods. For example, considering several months, it is very likely that new WiFi access points will appear while others are not in operation anymore. Another example of long-term changes is the fluctuations caused by weather. Depending on the climate zone, seasonal changes will naturally lead to significant changes in measured RSS.

The challenge of coping with all these causes and effects of signal variations are therefore manifold and have hence been a subject matter of research for many years. Summarizing, it can be said that for any location fingerprinting system making use of radio signals, it does not suffice to train the system once. In order to guarantee a certain level of accuracy and precision over time, it is necessary to train the system over and over again, i.e., measurements for the same place must be taken at different times of day, over many days and ideally in every situation. Moreover, to alleviate the effects of short-term fluctuations, it seems necessary to measure longer periods of time. This raises the question: How can the radio map be trained and updated at such high frequency while avoiding escalating costs?

1.3.5 Sensor Variation

For terminal-based indoor positioning system, i.e., systems where the mobile device is given the task of measuring the radio signal, the different characteristics of these devices cause different same-place, same-time measurements [90]. Moreover, different standards of wireless communication also affect the RSS measurements. According to Kjærgaard [89], large scale variations are variations between different radios, antennas, as well as firmware and software drivers. Small scale variations on the other hand are the variations between instances of the device model from the same manufacture. With the exception of [61] and [165], most systems do not explicitly address these variations.

In particular the handling of large scale sensor variation has not been addressed. Hence, today's indoor positioning systems require the provider to manually profile and configure each new device type. Given the potentially huge number of radio, antenna, firmware, and software combinations, this is less than ideal. And yet, the manual collection of measurements at certain calibration positions and the attempt to find mappings between signal strengths reported by different clients is still the most common solution for handling signal strength difference caused by sensor variations. Obviously, such manual solutions are both time consuming and error prone. More sophisticated solutions that avoid manual measurement collection by learning from online-collected measurements have been proposed by Haeberlen et al. [61] or Kjærgaard [89]. However, both of these solutions require a training phase and perform considerably worse in terms of accuracy than the manual approach. Kjærgaard also suggested to record fingerprints as signal strength ratios between pairs of base stations instead of absolute RSS values [94]. This approach solved the problem of signal strength differences to some extent. However, as Kjærgaard concludes, it is unclear how sensitivity affects the RSS measurements recorded at the same position from different access points across different clients.

1.3.6 Security and Privacy

As with most systems and application in the Ubicomp domain, security and privacy are also a core issue in indoor positioning systems [27, 42, 119, 142, 167]. As Langheinrich explained, *"knowing a persons location at a specific point in time often allows a substantial number of inferences to be drawn, e.g., regarding his or her hobbies, friends, political inclinations, or even sexual preferences"* [106]. Regarding privacy, the challenge in designing an indoor positioning system lies in providing anonymous position estimates. This is necessary, as the potential of data mining is very high [106]. A user's location is not only valuable for context-aware applications, but can also be exploited with relatively low effort. Using suitable heuristics, as for example correlating a person's often visited locations, even anonymized position data can be correlated [106]. In addition, people have a natural understanding of location-related privacy, they care if someone tracks them or records a history of all past whereabouts. Basically, controlling access to one's location information [13] or its distribution [70] can improve location privacy to some point. This can either be realized from the software side or by design of the architecture [30].

But as Langheinrich pointed out, the principle of data minimization becomes predominant in providing location security and privacy [106]. With location fingerprinting systems, this is most easily implemented at the RSS collection level. Beresford and Stajano for example propose a concept called mix zones, where a user's identity is anonymized by restricting the positions where users can be located [14]. Another approach is Gruteser et al.'s k-anonymity that allows to adjust the resolution of location information to meet specified anonymity constraints [59]. Both solutions try to reduce the amount of location information disclosed to applications. Other systems like those proposed by Rodden et al. [137] or Hauser and Kabatnik [66] provide some level of anonymity by the means of a proxy. While such systems manage to hide the true identity of a user, they fail to address the vulnerability of pseudonyms to correlation attacks [13, 106].

A third and very promising approach is self-localization [70, 130]. Here, the position estimation is being carried out by the target device itself. In consequence, no one can access the location information unless the target device discloses its location [60]. However, more and more mobile applications rely on data from remote providers or make use of web-services, i.e., it is almost impossible not to disclose one's location without degrading the use of the application. Thus, it often does not matter whether location information is obtained through a positioning service or self-positioning, location privacy must, first and foremost, be addressed at the application level.

1.3.7 User Interface

Mostly, location information is used and processed by higher level applications. As such, it doesn't have to be the concern of the indoor positioning system provider to offer a graphical user interface or any other human computer interface that is appealing or useable. However, as we maintained when discussing the issue of location privacy, it is sometimes necessary for the user to interact with the positioning system, for example to set the resolution level of location information that is disclosed. As we will show in the next section, one of the key concepts of our work is to include the user in the process of training the radio map and allow for user-contributed location labels. This concept has been used in different form in previous work [4, 15, 51, 69]. Studying these works, we learned that the positioning system must provide a way of user input that is appealing, easy-to-use and, most importantly, unobtrusive. Users will only contribute if the system is adding value to their work and not if the system turns out to be work itself. Thus, the challenge in designing such a user interface is to identify the incentives for contributing to the common radio map and to make this process as simple and transparent as possible. Regarding the issue of signal variations (see Section 1.3.4), we found that the radio map must be updated continuously. Hence, another challenge in designing the user interface is the question of how to motivate people to contribute location labels over a long period of time.

1.4 Goals and Hypotheses

Having analyzed many of the proposed, designed or built indoor positioning systems and having investigated the stated open research questions and suggestions for future work, we came to the conclusion that the two main problems of RSS-based location fingerprinting systems are:

- *Signal Variation* The received signal strength fluctuates due to many different factors. These signal variations occur both short and long-term (see Section 1.3.4).

- *Radio Map Training* An indoor positioning system that uses location fingerprinting is dependent on a large radio map. In general we can say that the more measurements a fingerprint comprises, the better the accuracy and precision. These measurements must be taken in place and ideally several times. In consequence, this (off-line) training process is very time consuming and thus costly.

Our goal was to improve location fingerprinting by tackling these two problems first an foremost. Our first objective was to understand the causes and effects of signal variations. Building on the findings of this analysis, our goal was to build a location fingerprinting indoor positioning system that is cost saving, easy to setup and install, and which would work over a long time period. For that, we built upon best practices and used algorithms, collection methods and other building blocks that have proven to work well. In addition, we created new concepts of use, training and location estimation that would alleviate the problems caused by signal variation and training costs. In an effort to encourage involvement and speed-up development, we decided to bundle the resulting source code and release the product under an open-source license[9].

The main concept of our work is *collaborative fingerprinting*. Instead of employing specialized personnel to train the radio map, our system enables users to add measurements and correct fingerprints themselves, thus avoiding a potentially time consuming and costly off-line training phase. As users are encouraged to correct fingerprints, this approach can also help to cope with long-term signal variations and changes in the radio signal environment. Moreover, instead of recording RSS only once, we created a mechanism that we call *adaptive collaborative fingerprinting*. This mechanism allows to record measurements over long time periods. Leveraging the accelerometer, a sensor that can be found in most mobile phones today, we determine the device's movement. This way we are able to deduce a user's activity and stationary status. As long as a device is not being moved, the system may continue to record and add measurements to the current place's fingerprint. Thus, our work is based

[9] The resulting project can be found at http://www.redpin.org

on the following hypotheses:

- Relying on user-contributed location labels is a feasible approach to location fingerprinting.
- Extending user-provided labels from an instant to an interval, i.e., a period of time over which the device is stationary, can greatly improve positioning accuracy.

We present our realization of these hypotheses in Chapter 4 and 5 respectively. In Chapter 6, we discuss our results and evaluate our hypotheses.

1.5 Summary of Contributions

In this thesis, we introduce the concept of user-contributed, collaborative fingerprint labeling to address the problems of map setup and map maintenance in location fingerprinting systems. Instead of manually creating an initial map prior to deployment, we harness the collaborative inputs of all system users to collaboratively create and subsequently maintain an accurate map of indoor fingerprints. In addition, end-user labeling allows labels to be added as needed for the places users visit most frequently. We offer a novel user interface approach to simplify the solicitation of user-generated labels that relies on labeling intervals instead of instants, and provide algorithms that are able to accurately position a device based on such user-generated labels. The contributions of our work are:

- *A long-term study of WiFi signal characteristics* To better understand the cause and effects of signal variations that can be observed when using 802.11 (WiFi) radios, we conducted two long-term studies. While we used stationary laptops to record measurements in the first study, we explicitly focused on end-user's activity and usage patterns in the second study. In summary, we found that WiFi signals vary substantially in both long- and short-term. We also found that the causes for these variations are manifold and can thus not be predicted or modeled in order to improve accuracy.

- *A novel method of collaborative fingerprinting* Our approach to end-user labeling allows the collection and correction of location fingerprints in the places that users most frequently visit. This way, we are able to train and update the radio map according to the user's needs while avoiding the high cost of offline training. By incorporating the training of the system into its usage, we are able to make the training an ongoing process that allows to quickly adapt to changes in the environment. Moreover, our approach allows to collect dense datasets of measurements for each fingerprint and their associated true locations, which alleviates the problem of signal variation. This yields more accurate results than other methods used, such as performing calculations with detailed models of the environment.

- *A mechanism to adapt fingerprinting based on device movement* In our thesis, we explore a technique that extends the applicability of a user-provided label from an instant to an interval over which the device is stationary. The stationary state is detected using an accelerometer, which allows to detect location changes autonomously, and consequently collect stationary interval measurements without explicit user intervention. Using intervals enables a different kind of labeling. By detecting intervals of device immobility, the system can also defer location labeling to a more appropriate time, and refer to longer time periods that are easy for users to remember (e.g., "Where were you between 9:15 am and 10:05 am today?"). This greatly improves the user experience as users do not need to provide labels while being at the location to which the label applies. Thus, they are more likely to provide meaningful labels.

- *An adaption of estimation methods to improve accuracy and latency* As our technique of adaptive collaborative fingerprinting yields very large radio map datasets, using common methods for position estimation would result in degraded performance in terms of look-up latency. To cope with that issue, we propose a new estimation mechanism that allows to combine different estimation methods and classifiers during runtime.

1.6 Thesis Overview

After having presented our case in Chapter 1, we explain the main concepts of indoor positioning in more detail. We first establish the main terms used as well as a common terminology that servers as a basis for presenting and evaluating positioning systems. Throughout Chapter 2 we analyze related work in more detail. In this chapter we also identify the main attributes of location and discuss their role and importance in evaluating positioning performance. An overview of location models and positioning technologies concludes this background analysis. The last section of this chapter is devoted to an in-depth analysis of different methodologies and concepts used for location fingerprinting.

Before discussing our approach of adaptive collaborative fingerprinting, we present our results on two long-term WiFi signal characteristics studies in Chapter 3. These studies have been conducted to better understand the causes and effects of signal and sensor variations. The focus of our work is to build an indoor positioning system that works well in real-world situations and over a long time period. Thus, the WiFi studies have been designed to capture signal variations for both stationary and mobile terminal devices. Particularly the second study presented in Chapter 3 has been designed with the use case of end-user labeling in mind. We conclude this chapter with a summary of findings and recommendations that need to be taken into account when building an indoor positioning system based on WiFi radio signals.

Chapter 4 illustrates the concepts and terminology of user collaboration. The usefulness, advantages and disadvantages of user collaboration is explained by means of different systems that have successfully employed end-users to contribute content. We then discuss the building principles of user labeling in location fingerprinting systems in detail and present the design and implementation of a reference system. Lastly, we present and discuss an evaluation that provides a detailed look at our prototype implementation while discussing its benefits.

Building on the principles of collaborative location fingerprinting, Chapter 5 illustrates the importance and advantages of interval labeling. Since end-users might not be willing to train the system as expected, it is crucial to have the ability to defer the labeling process to a time that is convenient for the user. In addition, we show how interval labeling can be used to further alleviate the problem of short-term signal variations and in consequence improve accuracy. In addition, we present an extensive evaluation of the different estimation and fingerprinting methods. In particular, we compare the estimation methods used in our systems to other well-known estimation methods. In addition, we take a closer look at the benefits of interval labeling regarding the resulting accuracy. Finally, our achievements are summarized in Chapter 6 where we review the contributions made in this work.

The important thing in science is not so much to obtain new facts as to discover new ways of thinking about them.

– Sir William Bragg

2 Background

Indoor positioning is being developed and used in many different domains from asset tracking in logistics to navigation systems in robotics. In this chapter we present an overview of different techniques and mechanisms used for indoor positioning. As such systems have their origin in different areas of research, there is little common ground regarding nomenclature. We establish and define the main terms and concepts as used throughout this thesis at the beginning of this chapter, followed by an overview of location models and positioning technologies. While discussing the latter, we will focus on location fingerprinting as the goal of our work is to improve this promising technique. Over the course of this chapter, we will also investigate and examine related work where it is appropriate.

The first section of this chapter presents and clarifies the notion of location information, discussing different forms for representation as well as its attributes. The next section deals with the many different location models that have been proposed for indoor positioning systems. Using the different forms of location representation established in the first section, the location models section is structured according to the type of location information it is based on. We will focus on symbolic location models as this is the type of model we use in our work. Lastly, we present the different positioning technologies used in indoor positioning along with their beneficial advantages and drawbacks.

2.1 Location Information

2.1.1 Representation

Given the very different use of indoor positioning, the adequate definition of location information is nontrivial. At its core however, location information always describes a specific place. For example, regarding maps or floor plans, a location can be described as a reference point in a two dimensional space. Some indoor positioning systems even allow for and provide location information in three-dimensional space. Such geometric location representation has many advantages, as we will se in Section 2.2.1. However, it also brings serious drawbacks and often comes at very high cost. Consequently, many indoor positioning systems only use descriptive labels to specify location information. Therefore, we distinguish two classes of location representation: *geometric* and *symbolic* [10, 47]. Accordingly, we can classify indoor positioning systems based on the kind of location representation used.

Geometric positioning systems determine a device's position as a geometric figure using coordinates relative to a global or local reference system. GPS for example returns the position of a client in reference to the World Geodetic System (WSG84) [113] as a tuple of *latitude*, *longitude* and *altitude*. Local reference systems are inherently used by indoor positioning systems like Active Bat [157].

Symbolic positioning systems on the other hand return a symbolic identifier. This may be an ID, for example the cell ID in GSM systems, a simple label, or in *semantic positioning systems* even a concrete name. The Active Badge [155] system, for example, determines the position of a client by identifying the sensors which are within sight of the badge.

Although readable names are often more meaningful to users, geometric attributes are needed in order to calculate distances or for example areas. Thus, most of the common known and successful indoor localization system such as ActiveBat [65], COMPASS [87], or SpotOn [72] provide location information in terms of geometric coordinates. However, two of the most prominent localization systems, namely RADAR [5] and Place Lab [143], provide mechanisms to output both, geometric coordinates as well as symbolic location identifiers. However, given application specific requirements other systems such as [33] or [61] only provide symbolic location identifiers.

On top of the very basic information required to describe a location, many systems provide and use ancillary attributes like containment or hierarchy, temporal attributes such as freshness, the extent of a place, or even the exact geometric description of a room or building. The inclusion of ancillary attributes requires a sound description and mapping considering the amount of data needed as well as the information system or database to be used. This again is a nontrivial task as the requirements are application specific. Including more attributes allows for more powerful operations. In turn, having more attributes means higher costs for actually providing the location information. For example, while having an exact three-dimensional geometrical representation of a whole building with all its rooms, stairways, elevators and inspection chambers is beneficial, the effort required to provide this data is enormous. Location models are created to provide the right level of abstraction. A location model defines the representation of location information along with ancillary attributes. We will be discussing the different types of location models in Section 2.2.

2.1.2 Attributes

Besides the primary position data, location information naturally comprises qualitative attributes like *freshness*, i.e. how old the data is, *accuracy*, i.e., how accurate the information is with regards to the world truth [52], or *reliability*, i.e. how certain can we get location information and how good it can be reproduced. In the following, we will explore these attributes in more detail and analyze their implications on modeling location information.

Freshness

It may seem that the age of location data, i.e., the time that has elapsed since the acquisition of the reading, is not per-se crucial to the functionality of location-based applications. However, it might be beneficial if the time when a positioning system measured a particular location is part of the location data. This is particularly relevant for systems using symbolic labels, which may change over time. Here, all sensor readings come with an expiration time, beyond which a reading is no longer valid. A location model may also employ a temporal degradation function that reduces the confidence of the location information from a particular sensor with time, as described in [132]. From this perspective, knowledge about the freshness of location information can be used to increase or decrease the level of reliability associated with it.

Accuracy and Precision

Naturally, the location information should be as precise as possible. However, since every positioning method inherently determines the location with a certain error, the user of this information wants to know how big this error actually is. Speaking of location information, we must distinguish between accuracy, error, resolution, and precision [97, 157, 166]. As illustrated in Figure 2.1, the *error* denotes the difference between the position estimated by the positioning system and the actual position. *Accuracy* usually denotes the same measure. With *resolution*, we denote the minimal difference between two measurements, while *precision* denotes the distribution of all measurements.

With GPS for example, it is possible to get a report that describes uncertainty. Thus, GPS vendors provide location uncertainty values that are more indicative of the errors experienced by the end-user [56, 73],

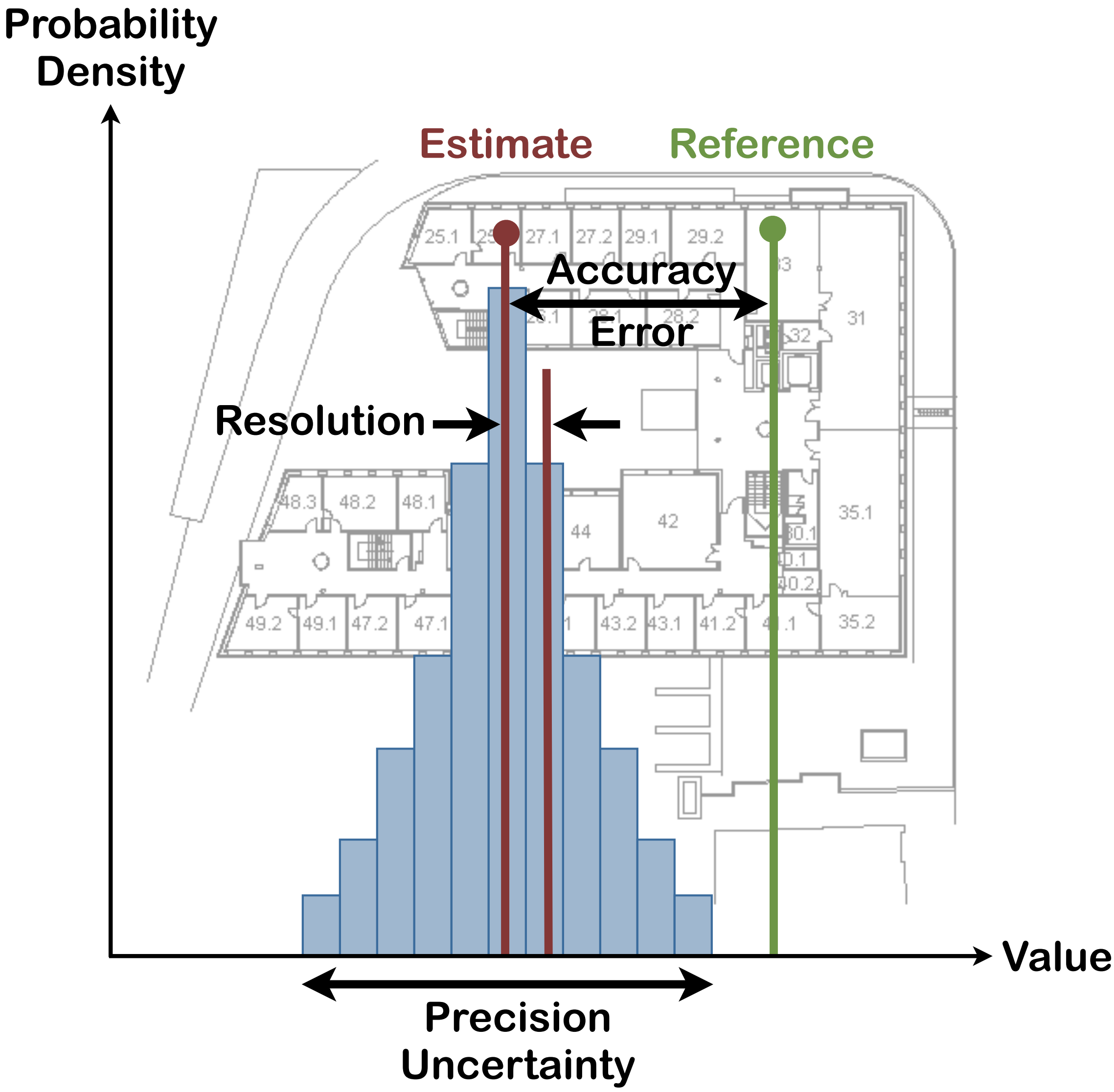


Figure 2.1: Location Error, Accuracy and Precision (based on [23]).

which again makes a common abstraction necessary. Moreover, for (indoor) positioning systems using symbolic labels, it is unclear how accuracy or error can be expressed, as no exact notion of distance exists. Nevertheless it is possible to trade less precision for increased accuracy [73]. Consequently, these two attributes must be processed in a common framework in order to compare and rate them. For example, Hightower [73] suggests that the fusion of different positioning sensors can improve both accuracy and precision by integrating many readings and forming hierarchical levels of resolution.

Reliability

Another attribute associated with location information is reliability. Even more than freshness and accuracy, reliability is a qualitative attribute. The idea to qualify the reliability of a positioning system by how reproducibly it returns values was formulated by Anderson in [3]. Anderson uses *zones*, i.e., a portion of space distinguished by others by signal strength, to represent the finest granularity where reliable positioning is possible. By appropriately choosing the size of such zones, it is possible to produce a similarity of up to 87% between the actual path of a user and the measurements of the positioning system.

Again, by using several different positioning systems and fusing the measured data, both the accuracy and the coverage can be increased [144]. Moreover, *sensor fusion* can be used to calculate and compare different readings and thus to quantify location data [56, 73, 143]. However, being able to qualify the error and the coverage of a positioning system is not necessarily sufficient to qualify the reproducibility of a system, i.e. with which certainty does a positioning system return the same readings when a user follows exactly the same path.

2.2 Location Models

As we have shown in Section 2.1.1, location information needs to be represented accordingly in order to process and store it. For this purpose, several location models have been proposed with different attributes and objectives. In the following, we will give an overview along with the classification that is commonly used to describe location models. Derived from the different types of results of geometric and symbolic positioning systems, we distinguish between geometric, symbolic, and hybrid location models according to [9, 10, 46].

Figure 2.2: Example of a simple geometric location model using a local coordinate reference system.

2.2.1 Geometric Location Models

Geometric location models use global or local coordinate reference systems (CRS) to describe a location. Such systems typically output Cartesian coordinates, which, for indoor settings, are often mapped to rooms based on available map data [87]. Most geometric models provide support for multiple CRS and hence include mechanisms to translate locations between different systems. Geometric systems are particularly well suited to calculate exact distances or other spatial properties, like the size of an area (e.g. a country). Figure 2.2 illustrates the use of a simple geometric location model with local coordinate reference. Based on the floor plan of the building, the lower left corner has been chosen as point of origin. Accordingly, the coordinates of the green location are `(24, 29)` in respect to the axis while `(48, 18)` denotes the blue location. In order to calculate the exact distance between these two locations, simple vector algebra suffices. By the same means, we can easily calculate the area of the yellow or blue area in Figure 2.2.

2.2.2 Symbolic Location Models

Symbolic models, in contrast, use *identifiers* such as *labels* instead of geometric coordinates to describe locations. Based on the grouping by Becker and Dürr [10], we classify symbolic models into the following categories: *unstructured*, *set-based*, *hierarchical*, *graph-based* and *combined* symbolic location models .

Figure 2.3: Different variations of symbolic location models (based on [10]).

Unstructured Location Models

In its most basic form, symbolic location models comprise simple location identifiers. Using labels as identifiers in particular has great advantages, as human readable labels are already used to denote locations. Moreover, in publicly accessible buildings like office buildings or schools, rooms are almost always labeled using a scheme. In Figure 2.3 we find five blue labels. One of these rooms for example is labeled *C33*. If we used the very same label as identifier in an unstructured model, everybody familiar with the labeling scheme used in a building knows where to find this room. Such schemes are often designed to reflect the building's layout. Using this knowledge, we can deduce more information from a label. In the case of our example label from above, which in its totality is *IFW C33*, *IFW* is the code for the building, *C* is the level or our floor whereas the number *33* denotes the exact room on this floor. As such numbers are mostly assigned in sequential manner, we can deduce the *neighborship* relation of two locations. For example, given the scheme used in our building, we known that the rooms *IFW A32* and *IFW A33* are adjacent just by looking at the label. However, as there are often discontinuities in the labeling of rooms, this deduction is not always correct.

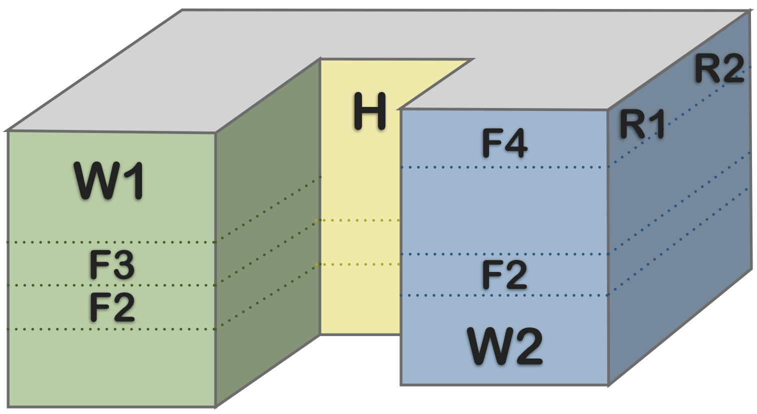
Although being very simple, unstructured location models thus allow to deduce more information such as *connected-to* or *contained* relations, provided that the labels are assigned using a scheme. In practice, it is often not necessary to explicitly model additional information at all. Regarding modeling effort, additional information can be added quite easily to elevate the unstructured models to graph-based, set-based or hierarchical location models.

Set-based Location Models

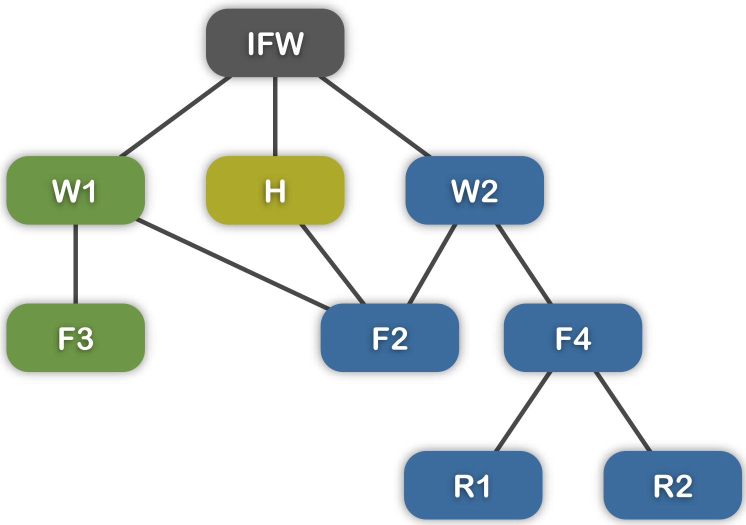
Set-based location models consist of a set of symbolic identifiers, e.g. all the room numbers of a building (note the yellow and green labels in Figure 2.3). Thus, a location is defined as a subset of identifiers. Location *A* in Figure 2.3, for example, comprises the identifiers *C45.1* and *C47.1*. As it is straightforward to determine overlapping locations, set-based location models are very well suited to answer range queries such as *"return the locations of all printers located on the second floor"*.

Graph-based Location Models

Graph-based location models represent symbolic location identifiers as a set of vertices. Direct connections, e.g., doors between rooms or elevators between floors, are consequently represented as edges in the graph, as shown in the upper left corner of Figure 2.3. Vertices in such a graph can be given a weight, which can be used as a notion of distance. As graph-based models naturally support the definition of the relation *"connected to"*, they are very well suited for nearest neighbor queries and navigation purposes.



(a)



(b)

Figure 2.4: Example of a hierarchical lattice-based location model (based on [47]).

Hierarchical Location Models

Hierarchical location models consist of a set of locations ordered according to given criteria. Mostly, the spatial containment is used as criteria to order the locations. If the locations do not overlap each other, this leads to a *tree*. For example, the root of a hierarchical location model is a building whereas the different floors are modeled as child nodes to the root node and rooms are leave nodes. However, as locations may overlap, the resulting data structure must be modeled as a *lattice*, as illustrated in Figure 2.4. Because hierarchical location models are mostly based on the *containment* relation, they are very well suited to answer range queries such as *"return all rooms of building A"*.

2.2.3 Hybrid Location Models

To benefit from the advantages of geometric models, namely the ability to calculate exact distances, while keeping the advantages of symbolic models, namely the support of range and nearest neighbor queries, *hybrid location models* are created that comprise both symbolic and geometric information. Figure 2.6 shows a simple example of a hybrid location model. The symbolic part is represented using a graph that interconnects the rooms on two floors. The spatial expanse of these rooms is geometrically modeled using polygons.

As it combines the advantages of both geometric and symbolic loca-

		Unstructured	Graph-Based	Hierarchical	Combined	Set-Based
Modeling Effort		Very Low	Medium	Medium	Medium	High
Supported Queries	Position	Good	Good	Good	Good	Good
	Nearest Neighbor	Limited	Good	Basic	Good	Basic
	Range	Limited	Basic	Good	Good	Good
Distance Support		Very Limited	Very Limited	Good to Very Good	Good to Very Good	Limited
"Connected to" Relation Support		No	No	Yes	Yes	Yes
"Containment" Relation Support		Limited	Good	Limited	Good	Good

Figure 2.5: Properties of symbolic location models (based on [10]).

Figure 2.6: Example of a hybrid location model, combining symbolic location identifiers with spatial properties. The simple tree in the top left corner represents the symbolic subset hierarchy.

tion models, hybrid location models are used in many Ubicomp applications [28, 47]. One simple representative of a hybrid location model is the *RAUM* model proposed by Beigl et. al [11]. The RAUM location model describes locations of artifacts relative to the environment and in relation to each other. A main design goal of RAUM was to capture significant features of human perception in order to make the model relatively easy to read for humans. Locations are represented by symbolic identifiers and structured in a tree to reflect organizations and rooms, for example. A little more complex and powerful is the hybrid location model introduced by Jiang and Steenkiste for Carnegie Mellon's AURA project [53, 81]. This model combines the advantages of symbolic and geometric location models while clearly separating the model and its representation. Jian and Steenkiste proposed to use a formatted Universal Resource Identifiers (URI) compliant string to represent all the above concepts. The proposed syntax allows to combine symbolic (e.g. the name of a room) and geometric (e.g. the base area and height of a building) within a single URI. Thus, geometric attributes like the exact expanse are contained within the symbolic representation.

2.3 Positioning Technologies

To determine the position, i.e. to detect the current location, many *positioning systems* exist for both outdoor and indoor applications. While the application of outdoor positioning systems like GPS or GLONASS[1] is very common these days (e.g., GPS-enabled mobile phones, car navigation systems), there is no equivalent standard for indoor location systems. Moreover, many applications, especially in the area of Ubicomp, require more accurate positioning in both dimensions, space and time [70]. Such applications mostly imply distributed services, like messaging based on the current location of the user [48], or adapting the settings of a device [28]. As discussed in Chapter 1, indoor positioning systems are thus built to match specific application requirements and used where the high costs of an installation can be justified, for example in hospitals where such systems are used to track doctors and patients. In result, the many specific and different requirements and demands made on indoor positioning systems in regard to accuracy, freshness, and reliability lead to the development of very different mechanisms and techniques. In the following, we will introduce the main concepts and clarify the terms used to describe the process of location acquisition and its result.

2.3.1 Methods

For special purpose positioning systems such as high-speed tracking or high-resolution positioning, special sensing technologies have been developed using ultrasound, light, or electro-magnetic field strength [143]. However, most indoor positioning systems utilize the physical properties of radio signals to determine the location of a device. Illustrated in Figure 2.7 is a classification of positioning methods used in indoor posi-

[1] GLONASS, like GPS, is a radio-based satellite navigation system. Constructed with the same goals as GPS (and for the same reasons), it is operated by the Russian Space Forces.

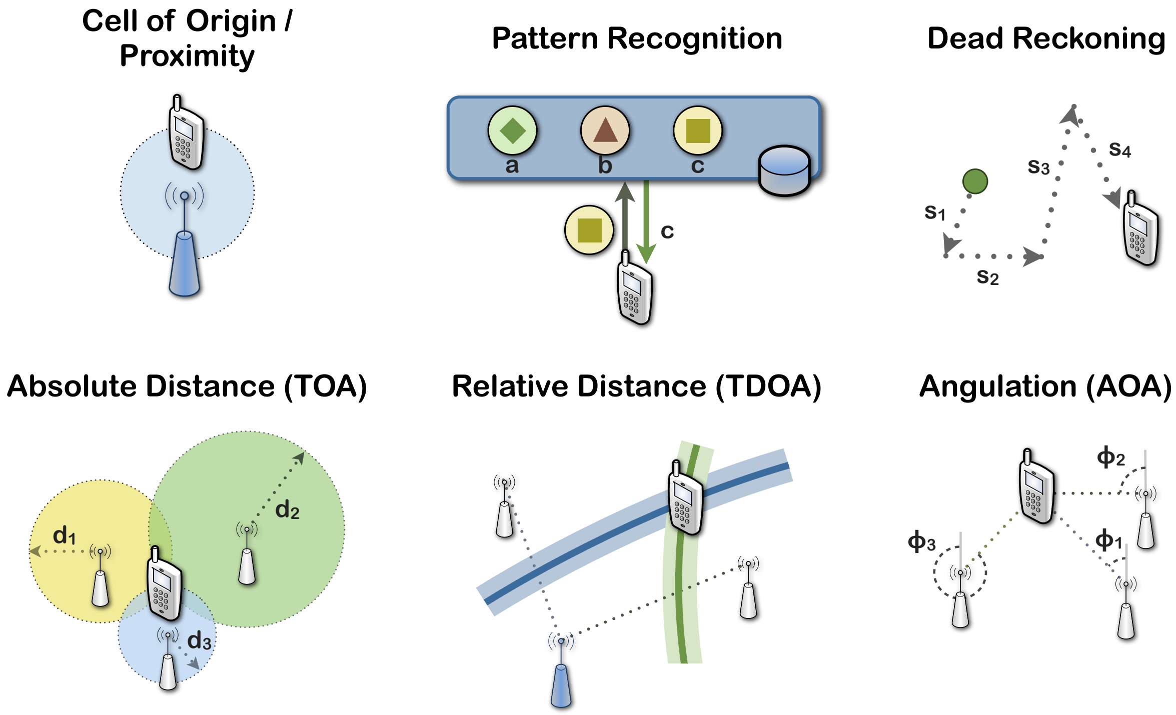


Figure 2.7: Overview of different positioning methods (based on [91]).

tioning. With the exception of *Dead Reckoning*, all of these methods can make use of radio signals. In the following, we will discuss the different methods in detail.

Proximity

Proximity, or *cell of origin* as it is sometimes referred to, is a very basic and very simple method of positioning. It acts on the assumption that tags or signal sources of a certain kind are distributed throughout the environment. In addition, these sources or tags are assumed not to change their location. The mobile device to be located can sense these sources only when within close range however. As the location of the source is known to the system, devices can determine position simply from the fact that they can sense the source. Bohn for example used an RFID tag infrastructure as source for a proximity-based positioning system [17]. Other proposed systems [67, 120] used Bluetooth devices to achieve the same goal.

While being very simple in design, proximity-based systems only allow for very coarse accuracy. In addition, without combining this method with more sophisticated techniques, it is only capable of providing symbolic identifiers. The greatest advantage of this method on the other hand is its unmatched precision when used with short range signals. Once we can sense the source signal, we can be almost absolutely sure about our location.

Dead Reckoning

Dead reckoning is a method that allows to estimate one's current location based on a previously known position. Dead reckoning is mostly used in robotics [21]. Given a fix starting point, dead reckoning can determine the current position by advancing based on speed and heading over elapsed time. This method is heavily used in automotive navigation systems to guarantee accurate positioning where GPS fails, for example in tunnels.

Unlike cars, where measuring speed and heading is relatively simple, it is technically very challenging to measure these parameters with wearable or portable devices. Still, Ojeda and Borenstein for example built a measurement unit that can be attached to a shoe and still manages to measure six degrees-of-freedom [123] with good results. However, the cost for such a device is very high as it makes use of sophisticated hardware. The application is thus limited to use-cases that justify the high cost. Randell et al. [131] use inexpensive pedometers and accelerometers to perform simple step sensing and step length estimation with satisfactory result. Even heading information could be determined using only a two-axis compass. However, it was necessary to maintain close coupling to the body.

The big advantage of dead reckoning is its independency of beacons or landmarks. In order to obtain accurate results however, it requires complex and costly sensing hardware. Consequently, dead reckoning is often used to improve the reliability of indoor positioning systems where requirements justify costs, as Borenstein et al. explain [22].

Absolute Distance

A more complex positioning method is trilateration by means of *absolute distance*. Conceptually based on the idea of triangulation, positioning with absolute distance measures is a method for calculating the intersections of (at least) three spheres given the position of the centers and the respective radii of the spheres, as illustrated in Figure 2.7. For indoor positioning purposes, radio signals from known sources are often used as transmitters. Hence, the measure of distance is defined by the relation between the speed of light and the absolute time of arrival at a certain base station, i.e. the time it takes for the signal to travel from transmitter to receiver.

The most prominent positioning system making use of absolute distance is certainly GPS. However, this method has also been used for indoor positioning with great success, for example by Bahl and Padmanabhan in their RADAR system [5]. Using laptop computers as mobile devices, Bahl and Padmanabhan achieved a median resolution in the range of two to three meters. Given the characteristics of radio channels indoors, this certainly is a satisfactory result. Nevertheless, using absolute distance, or TOA, works best in open spaces. This method assumes the radio signal to travel at constant speed, which is very rarely the case in indoor environments. Moreover, to accurately perform trilateration, the clocks of all senders and receivers must be synchronized precisely, something that is complex, erroneous, and thus difficult to achieve.

Relative Distance

Like the method of *Absolute Distance*, *Relative Distance* makes use of lateration. This method, also known as *Time Difference of Arrival (TDOA)*, uses the relation between the distances from fixed sources to a mobile sensor as measurement. The respective relative distance from a mobile sensor to two fixed sources form a hyperbola of possible positions. In figure 2.7 for example, the relative distance of the mobile sensor to the blue cell towers in the lower left and the cell tower in the upper left form the blue hyperbola with a given uncertainty due to clock synchronization errors. The relative distance between the blue cell tower and the cell tower on the right form a second, green hyperbola. The intersection of these hyperbolas indicate the position of the mobile sensor.

An indoor position system that uses the relative distance of 802.11 radio signals has been proposed by Yamasaki et al. [164]. The proposed system measures relative distance as the difference in propagation time for pairs of 802.11 access points. As these access points are time synchronized by default, the difference can be computed in their respective clock time. This however requires special software to be run on the access points.

Similar to the above presented method of absolute distance, this method can be used for systems with high precision requirements. However, it also shares the same drawbacks, namely the need for special sensors and the prerequisite of knowing the exact position of the fixed sources. Moreover, the precision can be severely degraded by signal reflection, absorption, or effects caused by multipath signals.

Angulation

The method of *Angulation* determines position from angle measurements in respect to fixed sources at known locations. In the example in Figure 2.7, each of the angles Φ_1, Φ_2, and Φ_3 describes a line of possible positions. The position can then be estimated by selecting the most probable intersection of all three lines.

A good example of an indoor position system using angulation is the *VHF Omnidirectional Ranging (VOR)* system proposed by Niculescu et al. [40, 41]. Based on extended 802.11 access points, VOR allows to make angle measurements and by doing so is capable of positioning with an average error of about one meter.

Angulation basically shares the advantages and disadvantages with the lateration methods using absolute or relative distance. It is however even more sensitive to the effect of multipathed signals. This method is thus often used in combination with TOA or TDOA, for example to solve problems with ambiguity [39].

Pattern Recognition

The term *Pattern Recognition* is used to describe different mechanisms for indoor positioning [93]. In general, pattern recognition describes methods that estimate positions by looking for patterns in measurements. This may be a 802.11 radio signal but may also be a video stream. An example of a system using the latter is Cantag [135]. Cantag uses video streams from distributed cameras to estimate the position of visual markers. The most prominent, and for the purpose of this thesis most relevant, method of pattern recognition however is *location fingerprinting*. We introduced this method in Section 1.2. A detailed analysis and description will be given in Section 2.4. Using location fingerprinting, systems have been proposed that use GSM signals (for example [124]), WiFi signals (for example [5, 87]), and also Bluetooth signals [7]. Extending this idea, LaMarca et al. [143] are using multiple wireless technologies simultaneously to increase both the robustness as well as the accuracy of localization.

Compared to other methods of positioning, location fingerprinting falls short when it comes to delivering centimeter precision. In addition, the fact that the radio map must be trained makes this method costly in terms of maintenance. However, of all methods discussed, location fingerprinting is the only positioning method that does not require special hardware to be installed or custom software to be deployed to access points. Because cost is a key factor when it comes to the decision which method to use, this is a clear advantage for location fingerprinting. Moreover, surveys of the most prominent challenges and issues in Ubicomp have shown that for almost all existing applications in this domain it is sufficient to localize a user with room-level precision [6, 35, 57, 70, 82, 160]. Consequently, we used WiFi fingerprinting in our own work. As this is our method of choice, we examine this method in more detail.

2.4 Location Fingerprinting

Location fingerprinting can theoretically use any physical phenomenon that differs between locations, even light or temperature. It is of course beneficial to use sources that are temporally more or less stable. Hence, most indoor positioning systems use radio signals such as GSM, Bluetooth or 802.11, i.e., WiFi. In particular WiFi fingerprinting [5] has been very popular for indoor positioning, because it requires no new hardware infrastructure for sites that already have WiFi. We introduced the main concepts of WiFi fingerprinting, namely the *radio map* and the *estimation method* in Section 1.2. In the following, we will elaborate on the roles and responsibilities of all devices required for location fingerprinting. Following Kjærgaard [93], we present the terminology used throughout this thesis. As in particular the training of the radio map hinders broad deployments of location fingerprinting systems, we analyze the problem of training in more detail in the last part of this section. In doing so, we identify possible solutions to this problem.

2.4.1 Roles and Responsibilities

Infrastructure-based location fingerprinting is per-se a distributed system with many entities involved, from wireless clients to base stations and servers. In this respect, roles denote the assignment and division of responsibilities between these entities. The manner of how these roles are assigned inherently affects the implementation of the system as well as the complexity required to provide security and privacy properties. According to Küpper [99] and Kjaergaard [91], infrastructure-based system can be divided into three categories: terminal-based, terminal-assisted, and network-based systems. The difference between these categories is the assignment of roles, i.e., who initiates the measurement process, who is responsible to observe radio signals, and who takes care of storing the measurements in the end. Moreover, the different categories assign the task of storing the radio map and executing the estimation method to different roles, as illustrated in Figure 2.8.

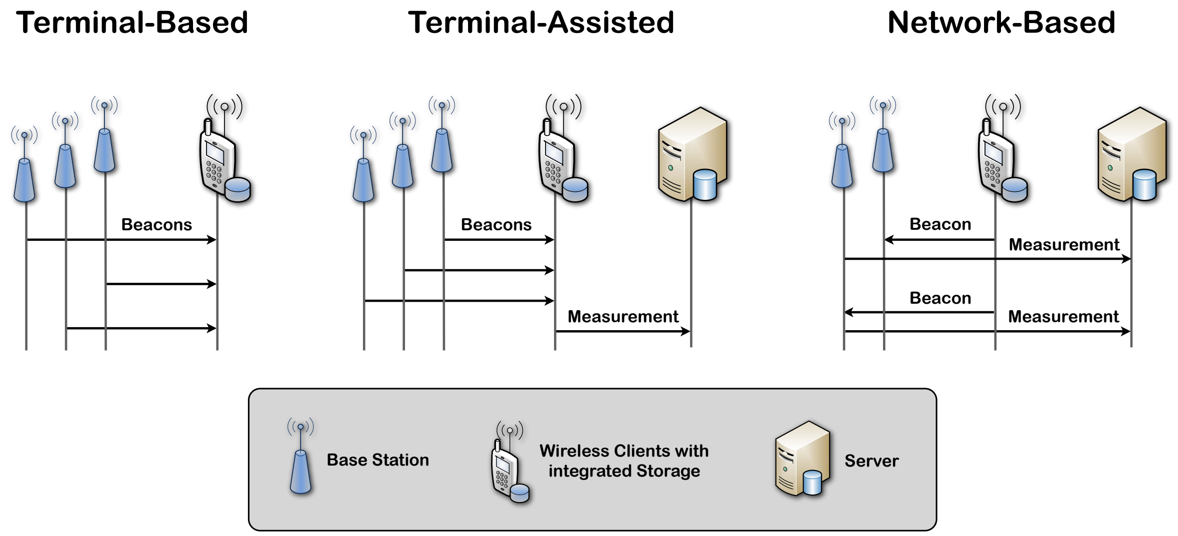


Figure 2.8: Classification of different assignments of responsibilities (based on [91]).

Terminal-Based

With terminal-based positioning systems, the measurement of the received signal strength (RSS) and the location estimation are executed by the mobile terminal, i.e., the user-operated device. Thus, the radio map has to be stored on the mobile device. Because the terminal is not required to transmit or share any measurement, it is almost impossible to detect its location. As it is quite simple to guarantee a high degree of privacy with this approach, many systems have been built as infrastructure- and terminal-based systems [126, 143]. In addition, terminal-based positioning allows to include WiFi access points, Bluetooth devices, or GSM towers that are not controlled by the location server [143]. However, storing the radio map on the mobile device naturally prevents a simple sharing of radio map entries between devices. If the devices should use the same radio map, the terminal is required to synchronize with all other devices every time it updates its radio map. Obviously, this can cause serious problems when it comes to scalability. Thus, the biggest advantage of terminal-based positioning, namely that the radio map is stored at the device, is also its biggest drawback. Although being favorable from a privacy point of view, terminal-based positioning does not really work with resource-constrained devices such as feature phones or smart cards. This also means that the algorithms used for the estimation method must be relatively simple as sophisticated algorithms might overstrain the terminal.

Network-Based

As the name already implies, in network-based positioning systems the complete procedure of locating a device takes place in the network. As illustrated in Figure 2.8, the RSS values are measured by the access points or base stations, which is then forward the readings to a central server. The main advantage of this approach is that all the heavy lifting is done by powerful devices having power supplies. This is very resource-saving and allows thus to use simple terminals such as smart cards or active badges. The downside of network-based positioning however is that the positioning software needs to be installed and maintained on the base stations and intermediaries, which usually comes at a high cost and does not allow easy operation in different organizations. Moreover, privacy is a huge issue with this approach, as the position of any terminal can be observed all times, giving almost no control to the mobile device's user.

Terminal-Assisted

Between terminal-based and network-based positioning lies the so called terminal-assisted positioning, where the workload is divided between the terminal and the server. While the terminal is used to observe and measure RSS, the server's job is to store the radio map and to execute the location estimation. The main reason to choose this approach is the ability to store the radio map on a central server, as this allows to easily share fingerprints. In addition, as the estimation method is executed on the server as well, this assignment of responsibilities allows to use resource-weak terminal devices, such as sensor nodes. Moreover, the radio map can be very large and the algorithms used for the estimation method can be very complex. Terminal-assisted positioning has many advantages and has thus been the method of choice for many indoor positioning systems that have been proposed, for example [15, 33, 87, 165], just to name a few. As we use a terminal-assisted approach in our work, we explain this type of role and responsibility assignment in more detail, thereby clarifying the terminology used throughout this thesis.

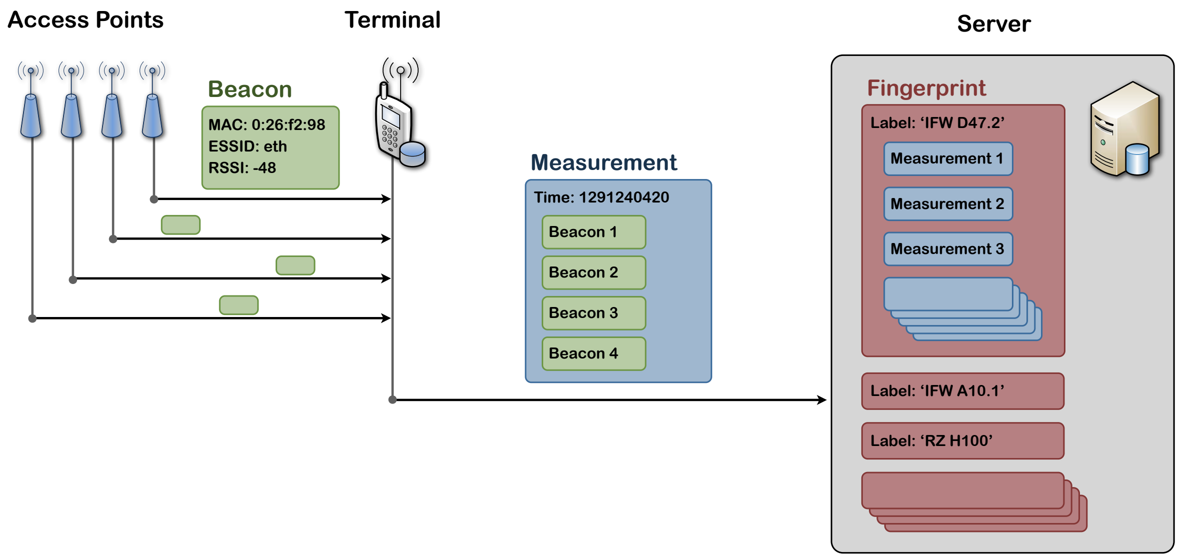


Figure 2.9: Basic concept of measuring as used in terminal-assisted location fingerprinting.

Illustrated in Figure 2.9 is the process of recording a measurement, which is subsequently stored in the radio map or used to estimate the position. This process is initiated by the terminal that scans the network for base stations in its vicinity. When using WiFi, this can either be done actively or passively [86, 93]. While active scanning is usually faster, passive scanning generally yields more results. We explain the differences in more detail in Chapter 3. While scanning, the terminal collects a *beacon* for every access point. The beacon contains at the least the following information:

- *MAC*: The access point's unique identifier for media access control. In infrastructured WiFi networks, this is also referred to as the basic service set identifier (BSSID).

- *(E)SSID*: The (extended) service set identifier of the network the access point is part of. This string is supposed to be human readable, for example "public" and is the same for all access points of this network.

- *RSSI*: The received signal strength indicator. It is a measurement of the power in the received radio signal.

Once a scan is complete, the terminal has a list of beacons, one beacon from each access point it could observe. This list is what we refer to as a *measurement*, as it contains all beacons that were observed at a certain point in time. Hence, a measurement always contains a timestamp. Once forwarded to the server, the measurement is then either added or compared to a *fingerprint* in the radio map. A fingerprint is the collection of all measurements taken at the same location. Hence, it representers a unique, impression of a location. Assuming simple symbolic location identifiers, a fingerprint must at least have a string label to indicate the location, as illustrated in Figure 2.9. As a fingerprint may contain multiple measurements, taken at different times and over a long period of time, it is crucial to choose an appropriate data structure. The entirety of all fingerprints makes up the radio map.

2.4.2 Estimation Methods

The estimation method denotes the algorithm used to find the "right" fingerprint given a measurement. In this section we present a very brief overview of the most popular estimation methods. We elaborate on the used and proposed estimation methods in more detail in Chapters 4 and 5. To estimate the location, location fingerprinting systems either employ a deterministic method like *Nearest Neighbor* or *Support Vector Machine*, or, on the other hand, a probabilistic method like a *Hidden Markov Model* [91]. One of the advantages of using a probabilistic method is that the probability of the result is a good indication of confidence. Moreover, this likelihood can be reused or fused with other methods and models. In this respect, a method that proved to work very well is *Bayesian Inference*, used for example by Castro et al. in their "Nibble" system [33], which can easily incorporate other contextual information such as the likelihood of a user going to a particular location. Much like Nibble, Ferris et al. [49] use gaussian processes to generate a likelihood model for signal strength measurements. Augmenting this idea, Madigan et al. use a hierarchical bayesian approach that can even provide location estimates without any location information in the training data [114]. Nonetheless, when using probabilistic positioning methods the performance and thus the accuracy can primarily be improved by adding more measurements to the fingerprint database [29, 34]. In our work we used both probabilistic as well as deterministic estimation methods. As the choice of method is decisive when it comes to achieving high accuracy and performance respectively.

2.4.3 Training the Radio Map

In theory, WiFi fingerprinting is capable of providing a resolution of only a few meters and can thus support room-level localization. And yet, as we discussed in the introduction, the biggest drawback of WiFi fingerprinting is the high cost that comes in the form of having to establish the radio map. As to achieve high accuracy from the noisy WiFi signal, WiFi fingerprinting systems require extensive calibration, mostly carried out manually and prior to use. For example, King et al. [87] were able to achieve an average error distance of less than 1.65 meters, but required prior calibration with 80 measurements every square meter (20 measurements each at 0°, 90°, 180°, and 270° orientations). Even though a single active WiFi scan takes only about 250 ms, the time needed to measure all four orientations and to move between locations quickly adds up to tens of seconds per reference point. In total, the training phase for an average 100 m^2 flat could take well over one hour. In addition, the training may miss longer-term variations. Systems have been proposed that omit the offline-training phase, for example the work of Lim et al. [109], that requires no training by automating the calibration of the effect of wireless physical characteristics on RSS measurements. But such automation requires very accurate RSS readings and thus the usage of sensitive WiFi network adapters. And while training time can be reduced by modeling the environment [80], this approach is less accurate and requires additional information (such as floor plans) that are not always available or easy to input.

An additional challenge is to keep the radio map up-to-date. In this respect, the biggest problems are long-term signal variations and changes in the radio signal environment, for example caused by newly installed or replaced access points. As we will show in Chapter 3, WiFi signal fluctuate substantially over the course of only a few days, let alone weeks. Consequently, it is necessary to continuously train the radio map in order to maintain good accuracy and precision.

2.5 Conclusion

Regarding existing location models, we can see that hybrid location models are best suited to realize the rather complex scenarios in the field of ubiquitous computing. However, most location models that have been proposed are still tightly coupled to the other components and integrated into a framework. However, to easily exchange and process location information, a common abstraction is a must. In addition, in order to natively use location information within a programming language, the corresponding model must be formally described. However, as room-level precision is generally sufficient for existing applications in the Ubicomp domain, we use an unstructured symbolic location model. In our systems we represent a room by a string like for example its name or number. This model can easily be extended should it become necessary to add additional information such as graph-based navigational information. However, this consequently requires work from the contributing users or administrator. From our experience, we recommend not doing this until a specific need arises, e.g. when an indoor navigation app should be built.

As one of our goals was to reduce the cost incurred by installing, maintaining, and using an indoor positioning system in order to boost proliferation, we choose a terminal-assisted approach. Only this method allows to install an indoor positioning system without having to change the installed network infrastructure and to share radio maps between terminals fast and easily. To achieve our goal of reducing the effort and thus cost of training the radio map, we believe that sharing fingerprints between users is crucial, as we will explain in the next chapters.

There's a way to do it better - find it.

– Thomas A. Edison

3 WiFi Signal Characteristics

Using WiFi radio signals for location fingerprinting is beneficial for many reasons, as we have shown in the previous chapters. The biggest problem, however, is that the received signal strength (RSS) changes over time. These fluctuations are caused by many different factors such as changing weather or nearby devices using the same frequency band, orientation of user and used device, or the presence of humans. To better understand the effects and significance of those long-term variations in signal strength, we performed two experiments. In doing so, we were particularly interested in patterns of signal separation, correlation between changes in RSS, access point visibility, as well as the effect of human presence. Understanding these properties is, as we will see in the next chapters, most important to guarantee an evenly high accuracy.

In our first study, we observed the RSS using 5 laptops, measuring at the same location for a duration of 20 days. These laptop computers had been placed in selected locations and were only used for the purpose of recording RSS, i.e., no human was using it during the study. However, some of them were placed in offices right next to or on top of an employee's desk. Each laptop collected a measurement every minute. As we didn't change the location of these laptops, we refer to this study as "controlled". For our second study, we developed an iPhone App that could be used to record RSS measurements. This App was given to volunteers, with limited instructions how to use it. Unlike on the first study, where measurements were taken automatically, we relied on the participants to start the App and record RSS measurements. Therefore, we refer to this study as "user-driven". During this study, 14 participants recorded measurements over a period of 6 weeks.

In the following, we will explain the setup, present the experimental procedure, and discuss the results for each study individually, starting with the controlled study. The second study, were we recorded RSS measurements over a much longer time, allows for interesting conclusions, because the pattern of how the participants recorded measurements resembles the use of an indoor positioning system to a high degree. We will therefore spend more time on analyzing the measurements of the user-driven study and present more detailed results. Finally, we conclude this chapter by summarizing our findings and listing guidelines for designing indoor positioning systems that must cope with long-term signal variations.

3.1 Controlled Study

3.1.1 Setup

This Section 3.1 is based on joint work with Kurt Partridge, Maurice Chu, Marc Langheinrich. While I was the main researcher on this topic, Kurt, Maurice and Marc supported my analysis and initial investigation into interval labeling. Together we published the results in our paper entitled "Improving Location Fingerprinting through Motion Detection and Asynchronous Interval Labeling", which was published in proceed-

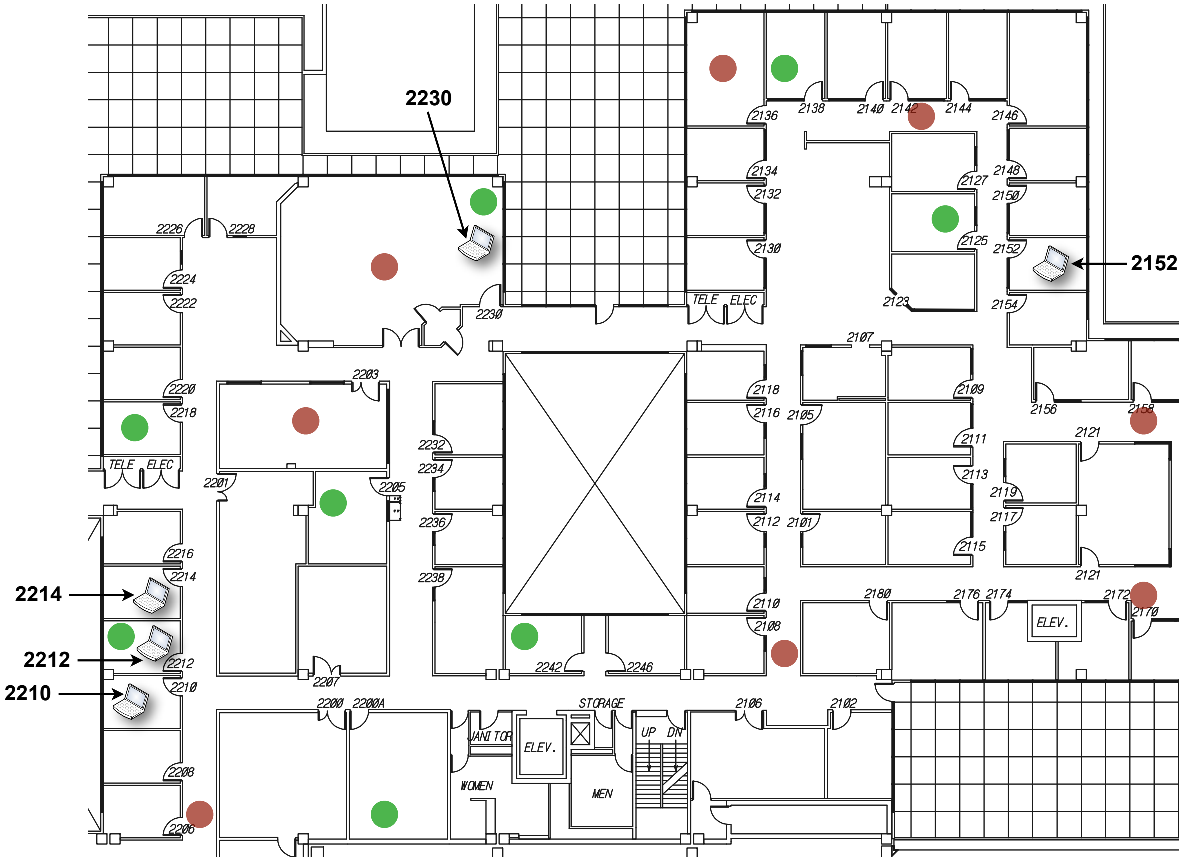


Figure 3.1: Setup of our controlled WiFi signal study at PARC. The red circles indicate the APs of the public WiFi network. The green circles indicate additionally APs we added for the purpose of the study.

ings of the Fourth International Symposium on Location- and Context-Awareness (LoCA) held in Tokyo, Japan, in May 2009 [19]. I was the main author of this paper and wrote the main parts, this chapter is based on, myself. I worked on this paper while I was visiting researcher at PARC. Maurice and Kurt advised me in my research and, together with Marc, helped me to improve the quality of the paper by giving it more structure and polishing my english.

The controlled study was conducted at the offices of Palo Alto Research Center (PARC) in California. As illustrated in Figure 3.1, we placed 5 MacBook Pro laptops in different rooms. We used different revisions of MacBook Pro with different network cards from either Atheros or Broadcom. For 20 days, each laptop did an active WiFi scan every minute and recorded the access points' unique identifiers (BSSID) and received signal strengths (RSS). In placing the laptops, we intentionally choose three adjacent offices. As one can see in the lower left of Figure 3.1, we placed a laptop in offices 2210, 2212, and 2214. The laptops in offices 2212 and 2214 were placed close to the wall having the same orientation.

The laptops used for observing the RSS were not used for other purposes than recording measurements, i.e., they were not used for work. With the exception of the laptop placed in room 2230, all devices were placed on the desk of an employee working in the respective office. Room 2230 on the upper left in Figure 3.1 is a meeting room that was either empty or, in case of a meeting, filled with as many as forty people. In this respect, we expected to get interesting comparisons, given that we expect the RSS to change substantially depending on how many people are in a room.

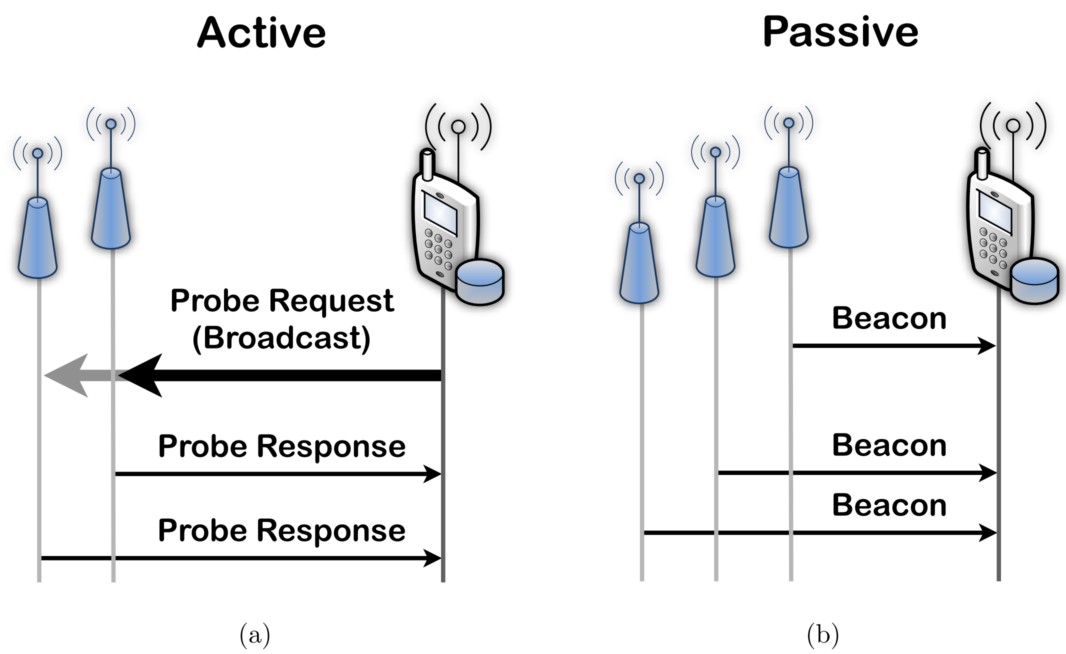


Figure 3.2: The two existing modes for IEEE 802.11 network discovery (based on [93]).

For scanning the WiFi network and recording the RSS we wrote our own software. As mentioned in Section 1.3.2, the concurrent use of the network interface card (NIC) for both scanning and data transmission can considerably degrade throughput as scanning almost always interrupts the data flow [86]. One easy way to alleviate this problem is to reduce the time required for scanning. While the NIC has to listen and wait for access points to send beacons in IEEE 802.11's *passive scanning* mode [79], *active scanning* forces access points to immediately send a beacon by actively broadcasting a probe frame. Although passive scanning generally allows to observe access points with very low signal strength and subsequently yields more access points than active scanning, it can take up to 80 seconds for a passive scan to finish. In contrast, an active scan does only very rarely take longer than 2 seconds. In our study, we used active scanning.

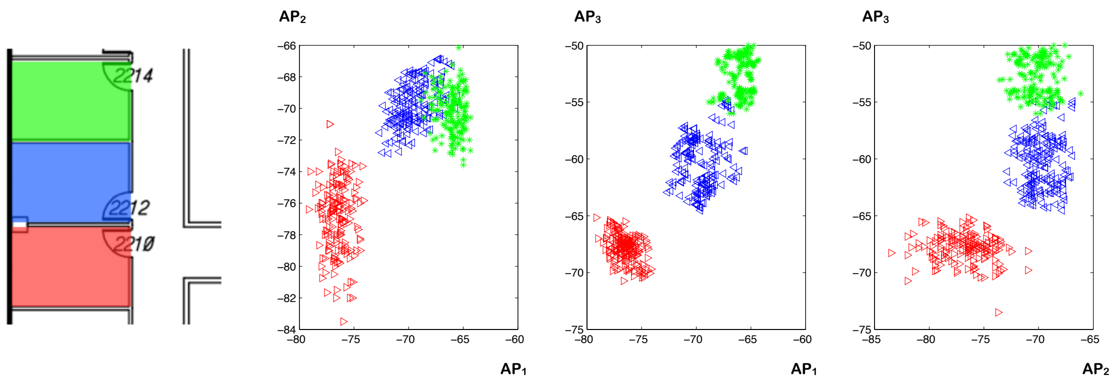
One problem we encountered with the WiFi setup at PARC was the visibility to the network used for internal purposes. For security reasons, this network was setup so that the SSID would not be visible to unconnected devices. As a consequence of this policy, we were only able to scan the public WiFi network setup for guests. The respective access points and their locations are depicted as red dots in Figure 3.1. To get a realistic picture nevertheless, we installed additional access points just for the purpose of the study. These access points are depicted as green dots in Figure 3.1.

3.1.2 Experiment

The controlled WiFi study at PARC lasted for 20 days. During this time, we had to exchange two laptops because of technical difficulties. The two laptops in question were at the end of their lifetime and crashed pretty often. They were replaced with MacBook Pro models from the same generation, having the same aluminum case, antenna and NIC. To better understand the effect of human presence, we relocated the laptop in the meeting room (2230) after 10 days. For the first 10 days, it was in the back of the room as illustrated in Figure 3.1. For the second 10 days of the study, we placed it to the opposite side of the room, right next to the speaker's desk. The only other change we made was to alter the orientation of some chosen laptops by 90 degrees. We expected to observe potential signal variations caused by interference.

3.1.3 Results

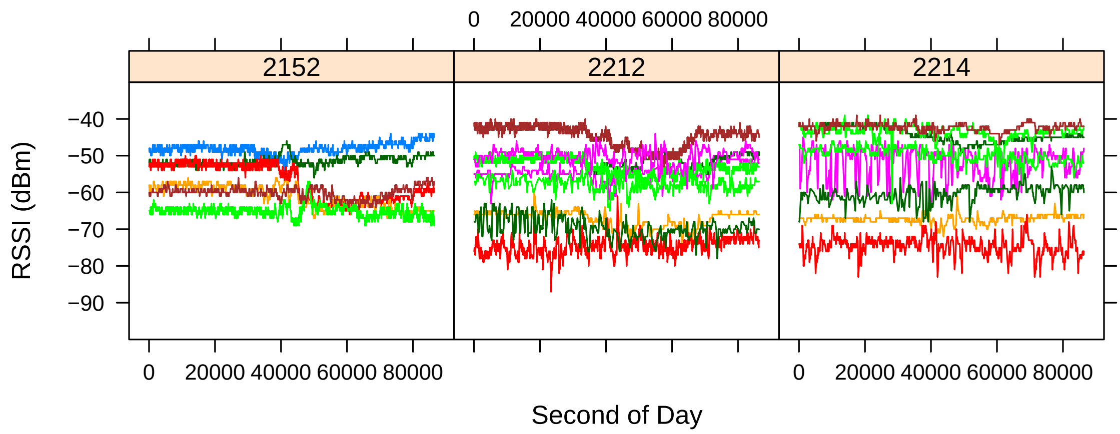
Figure 3.3 shows the signal separation for three selected access points measured over the course of one day. The red markers represent the readings as observed in room 2210, the blue markers these from room 2212 and the green markers those recorded in room 2214. The graphs in Figure 3.3 represent the RSS measured from two access points, each drawn on either the x or y axis and is thus a good depiction of signal separation. Of course, the clearer this separation is, i.e., the further apart the marker clouds in this figure are, the easier it is for any estimation method to make a decision. From Figure 3.3 we can see that access point AP_1 does not really help to tell readings from room 2212 and 2214 apart. Readings from AP_2 on the other hand are very beneficial when it comes to tell the difference between 2210 and the other two rooms. Finally, readings from AP_3 are most convenient to tell these three rooms apart. We can see that readings from different access points have different significance for different fingerprints.



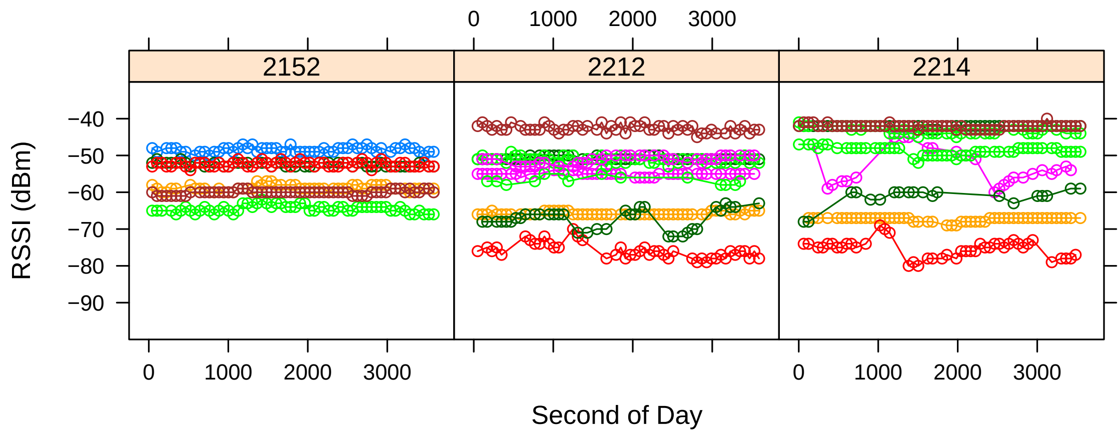
(a) (b)

Figure 3.3: Signal separation for 3 different access points as measured in 3 adjacent rooms.

Figure 3.4(a) shows the signal strength variation for three laptops over the course of a day. Different lines correspond to the signal strengths from different access points. While rooms 2212 and 2214 are adjacent to each other, room 2152 is further away. Room 2212 and 2214's patterns resemble each other much more than either of them do 2152, illustrating how these readings can be used to determine position. However, the graph also shows that there is much short-term variation from minute-to-minute as well as longer-term fluctuations. The short-term fluctuations arise not only from the motion of people, average per-access point variance on low-traffic weekends was still 68% of the variance during the week.



(a) RSSI measurements over the course of a day



(b) Detail from above showing the first hour

Figure 3.4: Signal strength variations from three laptops. Rooms 2212 and 2214 are adjacent to each other, and Room 2152 is further away. Signal variations happen on different timescales, ranging from a few minutes to several hours.

Additionally, different access points have different variances. Figure 3.4(b) shows the detail of the first hour, with individual scans now indicated by circles. This shows how readings can appear in scans at different rates independent of the mean received signal strengths. Analyzing these measurements, we found substantial differences in the variance across different access points. We also found these variances to be independent of receivers. However, we generally found little variation in the short-term variance, i.e., mean signal strength was more or less stable over several hours, yet fluctuated slowly day by day.

In addition to the short-term variance, we observed several sources of long-term variance between stationary senders and receivers. First, we noticed a temperature effect from a receiver that was exposed to sunlight. This change affected the received signals from all access points. However, this effect might not be a concern for a fingerprint-based location system as the relative ratios of all signal strengths did change only marginally. A second source of long-term variation was that of other receivers. As mentioned, we placed two laptops back-to-back with an office wall separating them (see rooms 2212 and 2214 in Figure 3.1). We believe that the antenna of the second receiver, which was tuned to the same radio frequencies, provided an exceptionally effective source of signal reflections. The long-term variance, which is especially noticeable during the day in Room 2212 (see Figure 3.4), shows that for nearby locations it may not suffice to build the radio map only once.

To our surprise we could not not find effects of significant level-off, i.e. the network adapter reported the same RSSI from the beginning and did not change significantly over the first measurements. When measured as fast as possible, the values from the first readings were the same as those about 1-2 seconds later, when the signal might change by 1dBm. Another finding we could establish is that the signal changes about every 15 seconds on average. It is thus sufficient to scan the network about every 20 seconds, even when collecting measurements over a long period of time.

Analyzing the whole dataset, we could also observe that signal fluctuation over time is substantial. Thus we conclude that the more measurements a fingerprint comprises, the better. Ideally these measurements are taken in different situations and at different times of the day. For preliminary testing, we used libSVM[1], a support vector machine library to simulate an actual position estimation method. Using these results, we found that at least 5 access points are needed to guarantee an accuracy of over 95%.

3.2 User-Driven Study

In the controlled study we took samples in a systematic way, as it was done in many past studies on WiFi signal characteristics before. Although giving first insights, results of these studies do not reveal all phenomena of signal degradation and variation as they occur when using WiFi for indoor positioning. For example, Kaemarungsi found in his study that RSSI is normally distributed [83]. As we will see in this section, this does not hold if measurements are taken collaboratively by the users of the indoor positioning system. As user-contributed, collaborative fingerprint labeling is a key concept of our work, we needed to get a clearer picture of signal variations as they occur when RSS measurements are taken by the users and thus in a non-systematic manner.

We believe that the approach of collecting measurements from users as opposed to experts is more realistic. Users collect data from locations they visit, for the time they spend in those locations, while they place their mobile device at arbitrary spots in these locations. Algorithms using fingerprints contributed by end-users would most likely have to deal with data collected in a similar fashion, as opposed to data collected in a systematic way such as an identical number of measurements taken in every room. Thus, we developed an iPhone application that allows to measure RSS values. To get as many fingerprints as possible we asked users to help and participate in collecting the data. The iPhone application was given to 14 users in two different groups who recorded measurements over a period of 6 weeks.

This section is based on joint work with Luba Rogoleva that was first

[1] http://www.csie.ntu.edu.tw/~cjlin/libsvm/

presented in her master thesis on "Crowdsourcing Location Information to Improve Indoor Localization" [138]. Luba implemented the iPhone app used to collect data and conducted the data collection under my supervision. With Luba's consent, in this section we present figures that were created using data and tools first used for her thesis.

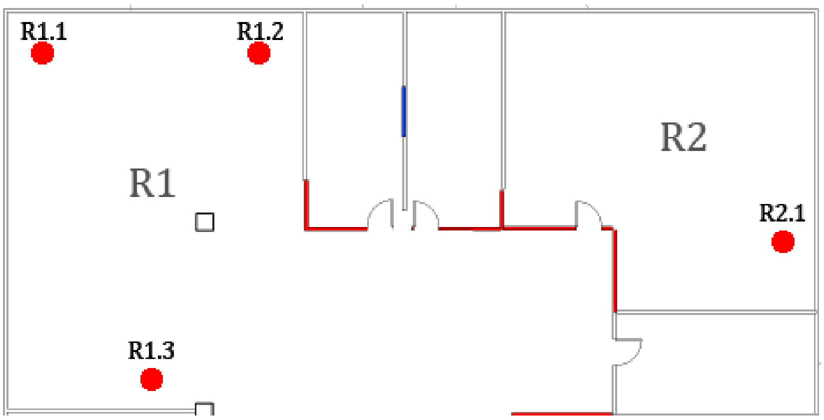


Figure 3.5: Floor plan of the office group. The red dots are the location of mobile devices while taking measurements.

3.2.1 Setup

To get as many RSS readings as possible and to get a broad picture, we recruited participants working in two different office setups. The first group, consisting of 4 people, had been working in the same open plan office in a company in Zurich. Their working places are illustrated in Figure 3.5. We refer to this group as the "office" group. The second group consisted of 10 researchers working at the offices of ETH Zurich. We refer to this group as the "eth" group. Participants in the later group had their offices on different floors in two adjacent buildings on the ETH campus, with the majority working on the "D"-floor[2] in the south wing of building "IFW", as illustrated in Figure 3.6. Two more participants were working on the "A"-floor of the same building while another two participants had their offices in the adjacent "RZ" building (see Figure 3.7 for a floor plan). Without giving any further instructions, participants

[2]The floor levels at ETH are labeled alphabetically. Thereby, "A" denotes the lowermost floor.

were asked to record measurements whenever possible. Although getting most measurements from the rooms our participants were working in, we collected readings from different rooms on all floors in both buildings.

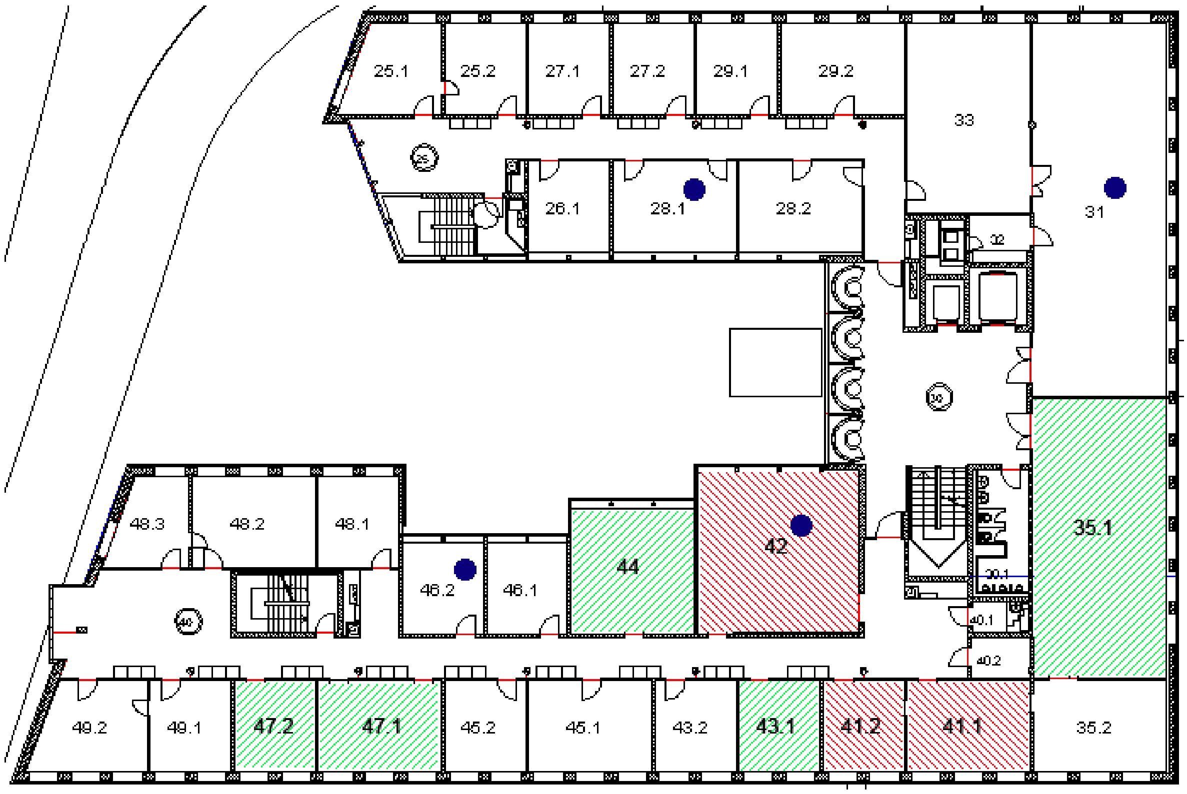


Figure 3.6: Floor plan of “D”-floor in building “IFW”. Blue dots mark rooms with known access points. Rooms where measurements were taken from both fixed and moving receivers are marked green while rooms where measurements were only taken from moving receivers are marked red.

From our 14 participants, nine users had an iPhone 3GS, five had an iPhone 3G and one user had the second generation iPhone 2G. Before recording data, participants were asked to enter a label for their current location. We instructed people to enter the room number if applicable or any other label they would use to refer to the room in question. By the press of a single button, users could then start recording measurements from all access points in their vicinity. The application would continue recording until the user quit the application or the iPhone was picked up. Using the built-in accelerometer, we detected iPhone movement and would automatically quit recording as soon as the device was not stationary anymore. The application was recording a measurement every 30 seconds.

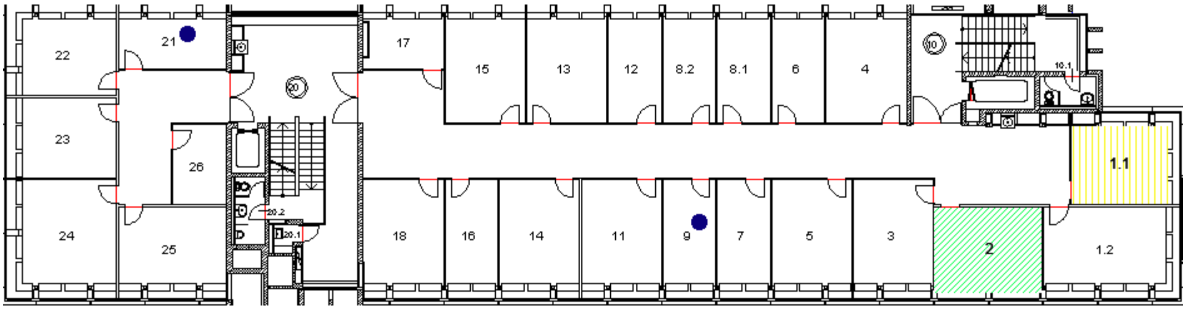


Figure 3.7: Floor plan of “H”-floor in building “RZ”. Blue dots mark rooms with known access points. Rooms where measurements were taken from both fixed and moving receivers are marked green while rooms where measurements were only taken from fixed receivers are marked yellow.

3.2.2 Experiment

We divided the six weeks of data collection into two phases of three weeks each. During the first phase, participants were told to have their iPhone at the exact same spot while recording. When discussing results, we will refer to this phase as “fixed”. For the second phase, we softened this rule and people were free to replace their devices while recording measurements, e.g. from one side of the desk to the other. We will refer to this phase as “moving” because participants recorded measurements for the same room in different locations within this room.

After 6 weeks of recording, we counted almost 70000 measurements that have been collect in 23 different locations. About 30000 measurements were taken during the “fixed” phase, recorded in 19 unique locations. During the “moving” phase, participants recorded almost 40000 measurements in 16 different locations. Note that the last week of our study took place during holidays. In this last week we did not receive measurements from participants in the “office” group since they were on vacation.

3.2.3 Results

While we did not have the problem of hidden SSIDs we encountered during our first study, we had to cope with a similar problem. Most access points in the IFW and RZ building on the ETH campus are used as different virtual access points to serve different networks and thus different SSIDs, but also different BSSIDs, i.e., MAC addresses. We conducted different statistical analyses to determine whether these virtual access points could be treated as the one physical AP it actually is. This simplification would be possible if all virtual APs transmitted the same signal strength. To our surprise, the RSS of different virtual APS being served by the same physical AP did not only differ most of the time but also varied significantly. Thus we decided to consider each virtual AP individually and hence equal to other physical APs.

	AP: 0:3:52:4d:e7:90 (IFW D42)		AP: 0:3:52:1c:31:60 (IFW D46.2)	
	Fixed	Moving	Fixed	Moving
Mean	-48.80	-58.30	-63.66	-63.40
Std Dev	4.91	7.25	3.29	3.32

Figure 3.8: Mean and standard deviation (in dBm) by fixed and moving terminals for two exemplary rooms where we got the most measurements during the whole period of the study.

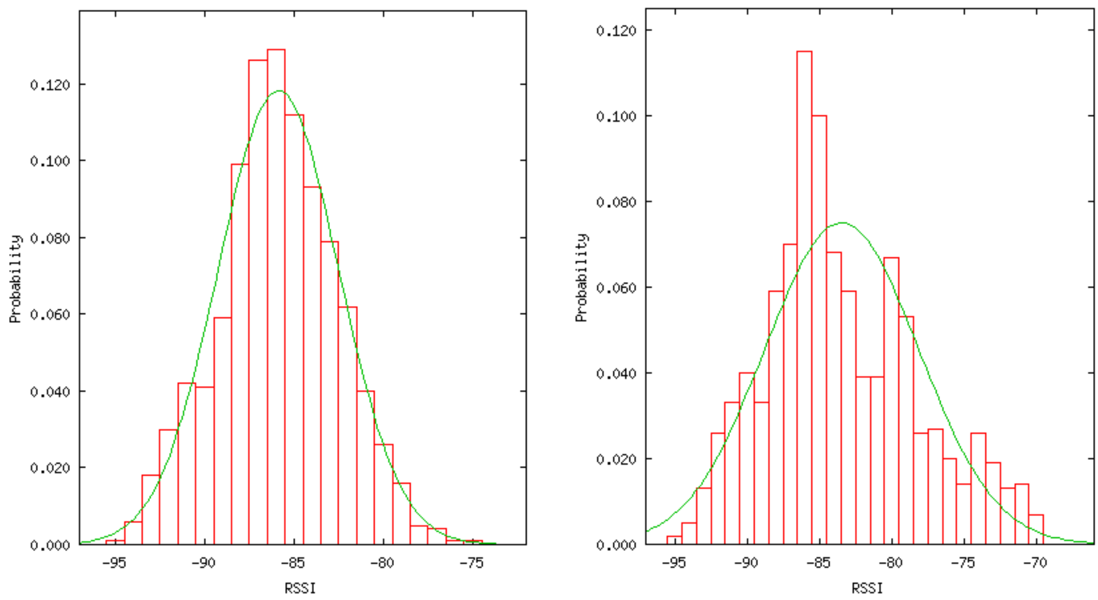
From what we learned during our first study, we expected that the observed RSS is lower the farther away a receiver is located from the AP. However, while first examining the recorded measurements we observed that the signal propagation is not only a function of distance between the transmitter and the receiver but also of different transmitting power of the specific access point and fading effects. In other words, just because a receiver is closer to an AP a than it is to AP b does not necessarily mean we will observe higher RSS values.

Another observation we made is that the standard deviation is significantly higher for moving receivers than it is for fixed receivers when comparing measurements from the same location. Considering all recorded measurements, we calculated a maximal standard deviation for moving receivers of 9.52dBm and 7.758dBm for fixed receivers. This finding meets our expectations as intuitively RSS varies more when the receiver is moved than when it is stationary due to the multi-path effect which can cause several fades in short duration.

Mean values are not varying much between fixed and moving receivers. As mentioned above, the mean of RSSI is influenced more by the large-scale fading effect caused by absorption of signals by obstacles such as walls and floors. However, in the example shown in Figure 3.8, the mean value of RSSI measured in room "IFW D42" from AP 0:3:52:4d:e7:90 for the moving receiver is almost 10dBm lower than for the fixed receiver. This can be explained by the relatively short distance between the AP located in room "IFW D43" and the receiver in room "IFW D42", since, as explained in the next paragraph, the closer the transmitter is to the receiver, the higher the fluctuations.

RSS Distribution

Although usually considered to be normally distributed, we expect the RSS value recorded by users over a long period of time to behave differently. Due to the known fading effect of human presence and the fact that the devices used to record measurements change position between measurements, we expect to see outliers in the RSS distribution. Figure 3.9(a) illustrates the histogram of RSS as measured by a fixed receiver at location "IFW A44" for a period of 21 days while Figure 3.9(b) shows the histogram measured by a moving receiver at the same location for a period of 17 days. As we can easily see, there is obvious irregularity in the RSS distribution of the moving receiver while the fixed receiver shows almost perfect normal distribution. RSS distribution can dramatically change.

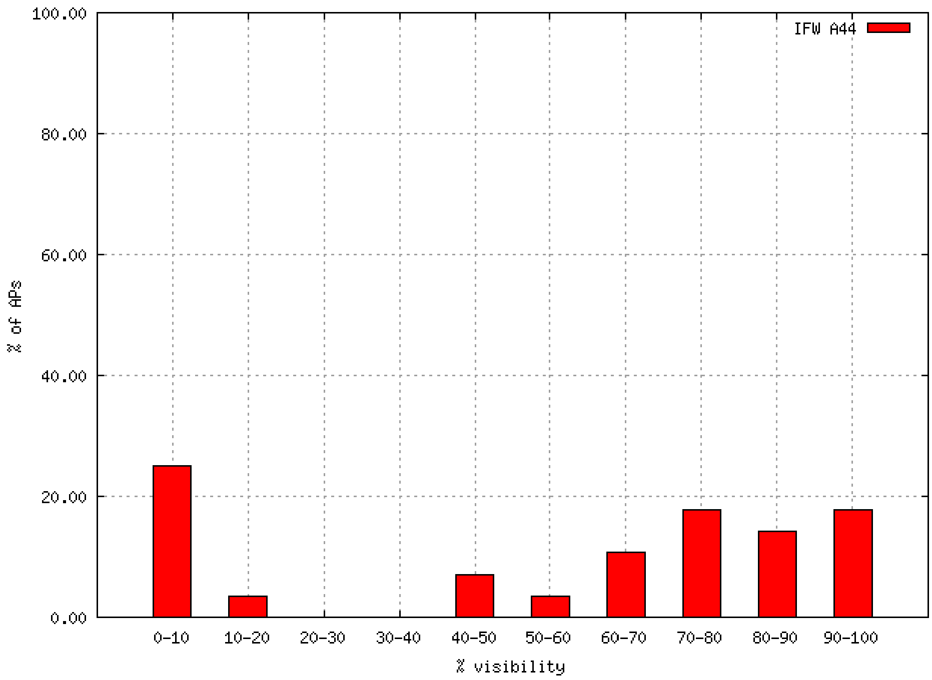


(a) (b)

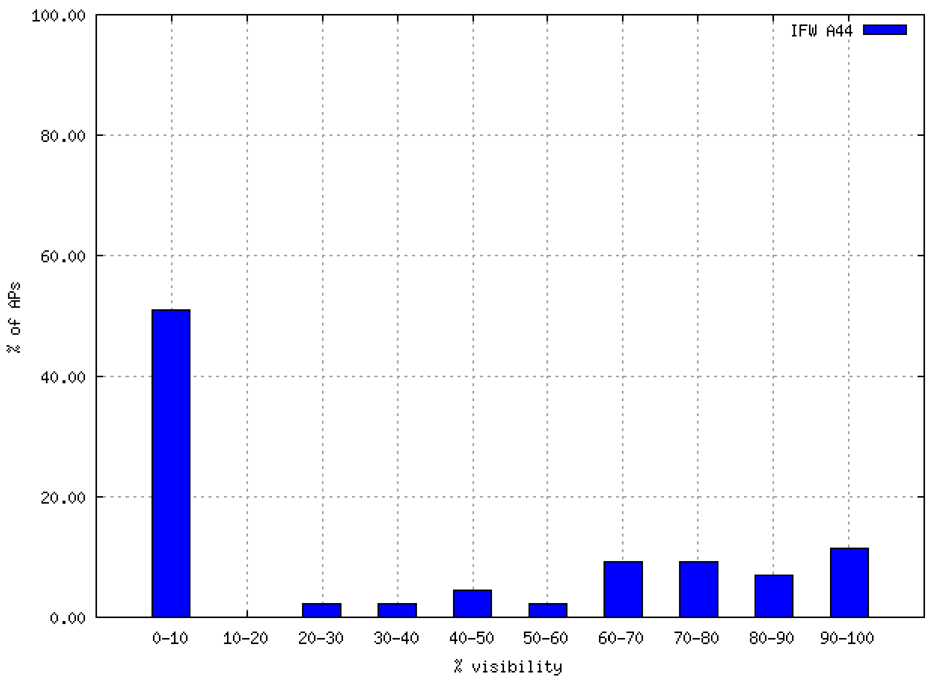
Figure 3.9: Histogram comparison of RSS for fixed (a) and moving (b) receivers.

Access Point Visibility

When recording RSS measurements in the same location, one would expect to observe the same list of APs. However, due to signal variations and fading effects certain access points may be captured only at certain times. Consequently, the number of APs observed by a mobile receiver at a certain location is not constant over time. For example, while the number of APs observed stays is almost constant during nighttime when the signal is relatively stable, it varies a lot over the course of a day. The extent of this finding is of particular importance for algorithms that use techniques such as filtering [110]. To investigate this effect, we considered ten classes of AP visibility. APs that can be observed 0%-10% of the time comprise the lowest class while APs that are observed 90%-100% constitute the highest visibility class. Figure 3.10 shows the histogram for a 5 day and a 16 day period respectively. The measurements used for this analysis have been taken in the "moving" phase. Comparing Figures 3.10(a) and 3.10(b) we can see that the number of APs that fall into the class of 0%-10% increases with longer observation time. From this we can deduce that this kind of "noise" is well captured by measuring over a long period of time. Moreover, this is an indication that filtering APs with low visibility before training the radio map is a viable



(a) Visibility over a period of 5 days.



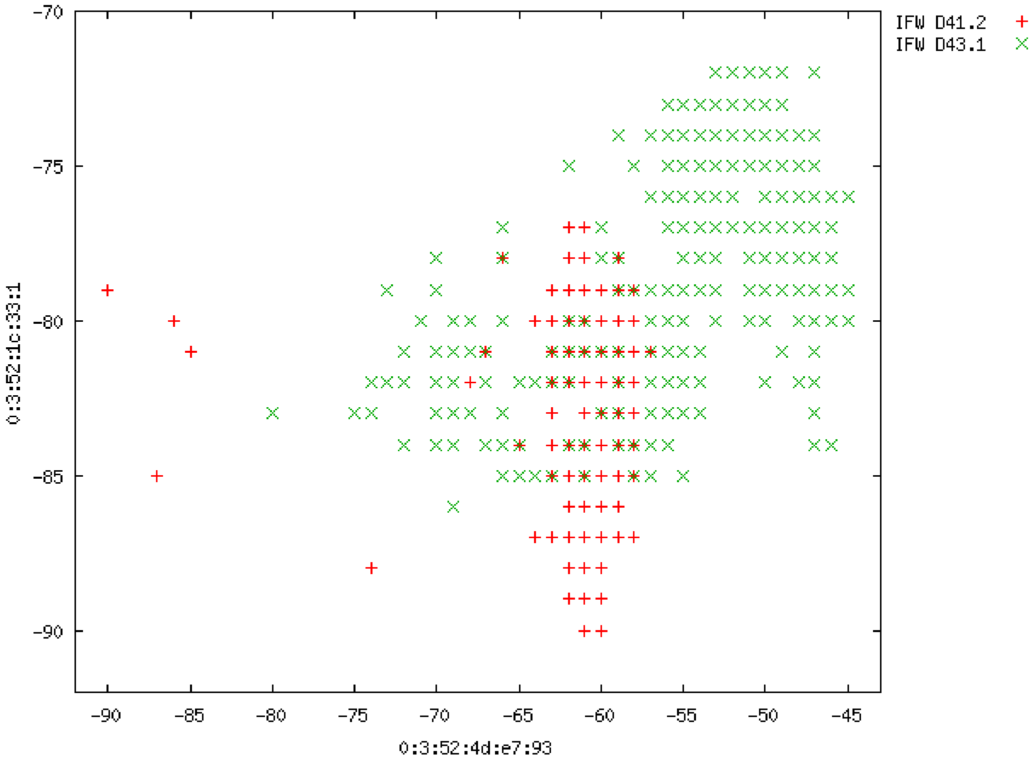
(b) Visibility over a period of 16 days.

Figure 3.10: Visibility observed by moving receivers in room "IFW A44".

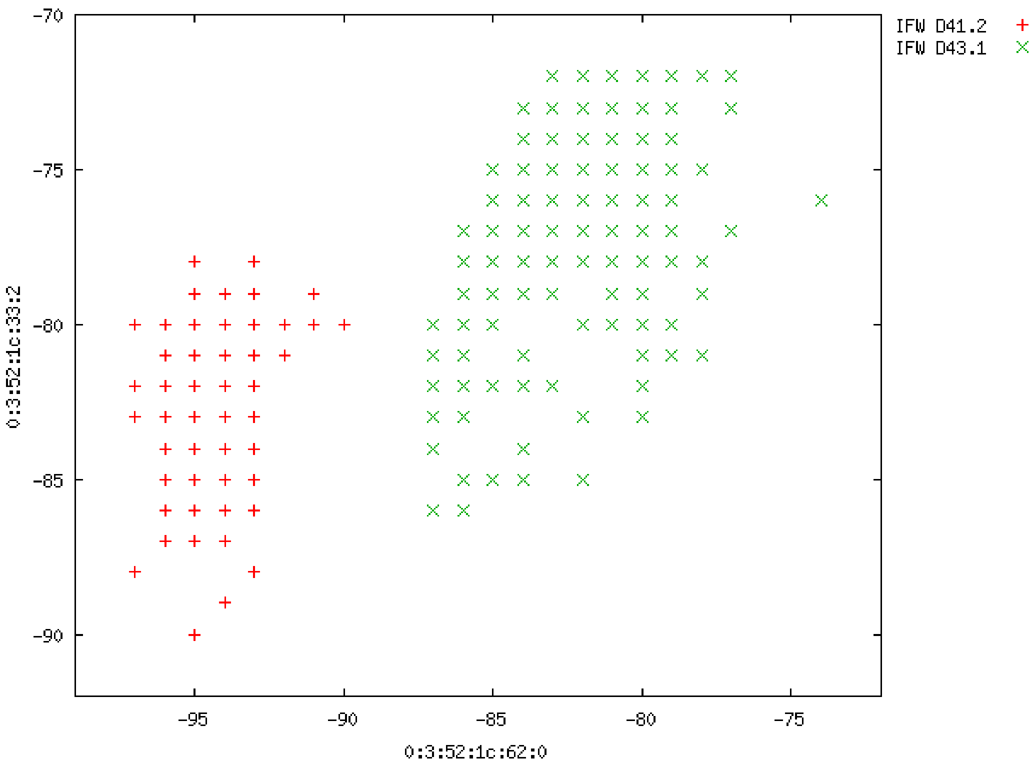
approach. When filtering one must precede with caution. We also found that the visibility of APs can change significantly and, moreover, inconsistently between periods of different length. For example, APs that are only seen rarely when observing for a short time of about 2 hours are visible for over 60% of the time when observing for 5 days. Moreover, we also found changes in the other direction as well, i.e., APs that are seen often in short measurements have low visibility in long measurements. Thus, estimation methods using filtering techniques based on AP visibility might assign erroneous weights to APs when measuring only for a short time.

Separation of Fingerprints

Signal separation denotes the degree to which RSS signals and patterns differ between locations. This attribute is thus essential to indoor positioning systems using location fingerprinting [83]. Graphs of cumulative signal separation, such as Figure 3.11(a) give an illustration of the actual fingerprint of a location. We've already seen in Section 3.1.3 that different APs reveal very different signal separation patterns in different locations and that not every AP contributes to positioning in a positive manner. We found the same to be true in our user-driven study. Figure 3.11 for example shows the signal separation for two adjacent rooms in building "IFW". While we can see clear separation in Figure 3.11(b), the two APs in Figure 3.11(a) do not help to distinguish the two adjacent rooms. We can thus confirm our finding from the first study that usually more than two APs are needed in order to separate between two locations close by. With this second study we got the chance to study the difference of open space offices and offices separated by walls. We found that separation of fingerprints can be problematic in the open space office even for two locations that are relatively far away from each other. Figures 3.12(a) and 3.12(b) illustrate for example fingerprint separations for two locations "R1.2" and "R1.3", locations that are about 9 meters apart. In general, we found that walls significantly influence RSS and thus help to separate fingerprints.

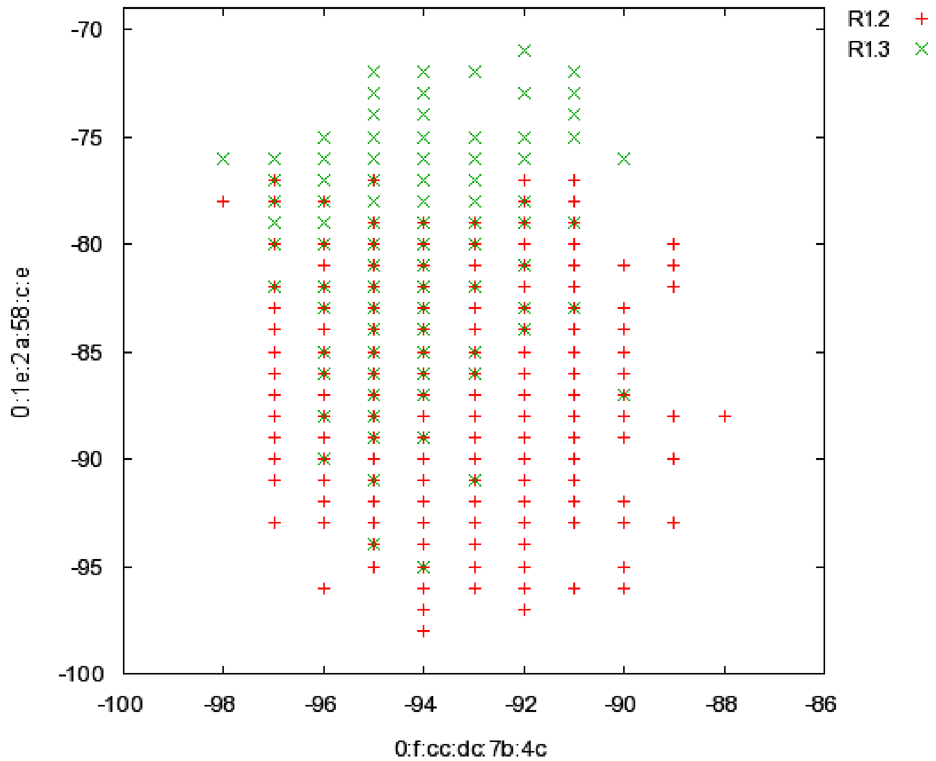


(a) APs 0:3:52:1c:33:1 and 0:3:52:4d:e7:93.

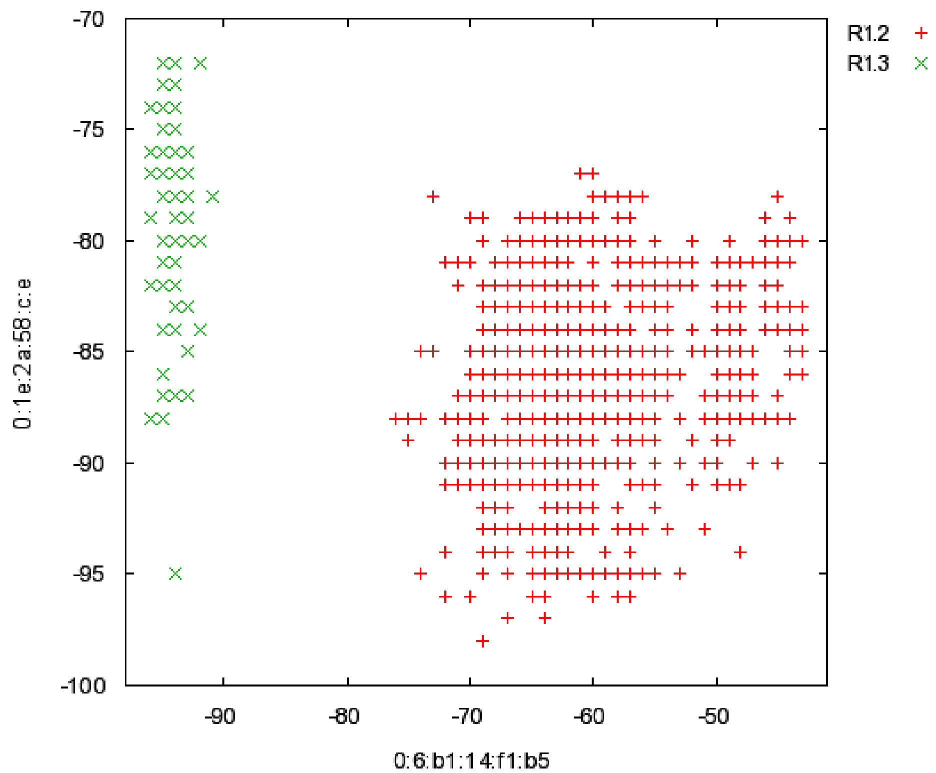


(b) APs 0:3:52:1c:33:2 and 0:3:52:1c:62:0.

Figure 3.11: Separation of fingerprints for two adjacent rooms in building "IFW". Measurements were collected over a 4 day period.



(a) APs 0:1e:2a:58:c:e and 0:f:cc:dc:7b:4c.



(b) APs 0:1e:2a:58:c:e and 0:6:b1:14:f1:b5

Figure 3.12: Separation of fingerprints for two locations in the open space office. Measurements were collected over a 4 day period.

Effect of User Presence

One of the biggest sources of signal variations in office environments are people. The high frequency signal of WiFi is very well absorbed by the human body due to its high water content. As people tend to move around in an office building, they account for most of the of observed short-term signal variation.

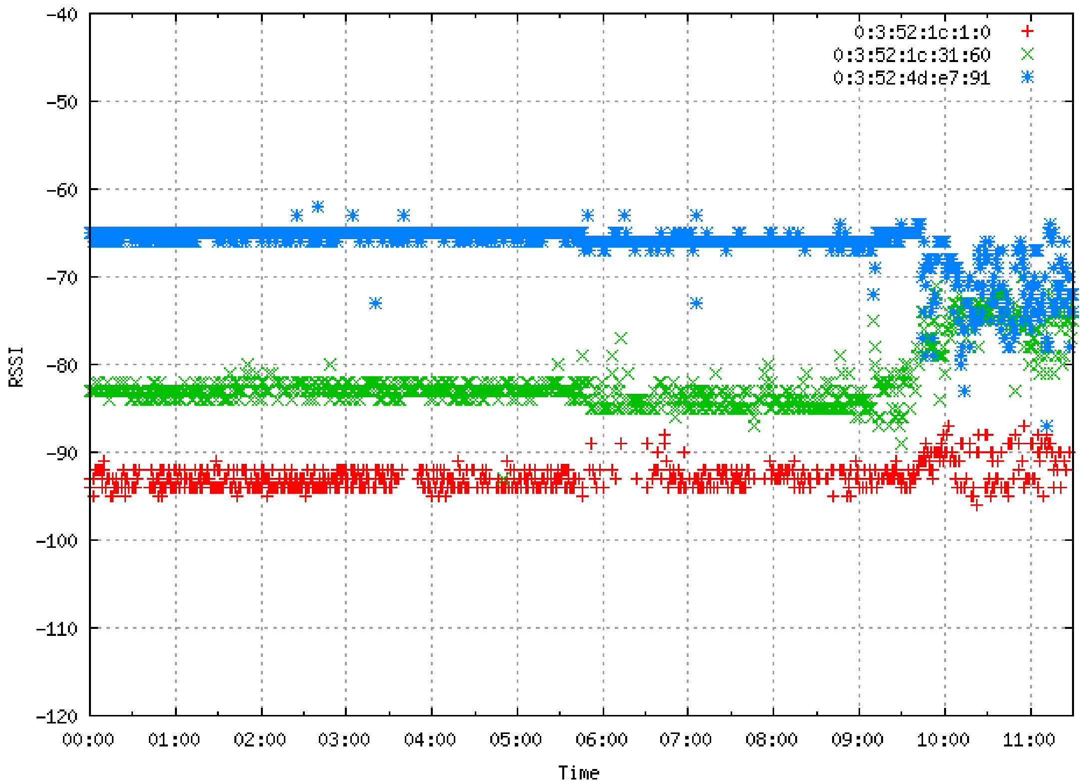


Figure 3.13: Human-caused signal fluctuations, measured by fixed receiver during night, early morning, and morning.

Figure 3.13 illustrates the RSS of three APs observed in one of the rooms in building "IFW" between midnight and 11:30 in the morning. While the signals are relatively stable during night we see little variation in the early morning and heavy variations after 9:30 am. The effect of people's presence on WiFi signals has been studied in the past [83, 102] and our results are similar, confirming their findings.

3.3 Conclusion

Although the findings from the two studies may differ in detail, the general conclusions are the same. First and foremost, we found WiFi RSS to fluctuate substantially, both short and long-term. Besides the fact that there is less signal variation during night when no people are moving around, we could not find patterns of RSS variations. Therefore, we can say that it is not possible to predict RSS variation. Thus, it is necessary to get as many measurements as possible, ideally taken at different times of the day and at different days.

One notable learning from the first study was that there is no significant level off when taking RSS measurements, i.e., when continuously scanning, the RSS does not change in comparison to the RSS value observed during the first scan. In the controlled study we also found that different access points have different variances. As a result of that, APs can appear at different rates independent of the distance to the terminal or the RSS. As expected, we found that receivers close to each other significantly influence each other's RSS. This may be problematic in situations where positioning is required for devices that are mobile but used at the same location for several hours, such as for example laptops. Regarding signal separation, we found that walls greatly help to classify locations with room-level precision. Finally, by analyzing the whole dataset collected in the controlled study, we found that in order to achieve a lower bound accuracy of 95%, at least 5 APs must be observable at any time.

In our second study, where participants used their mobile phones to take measurements in both fixed and moving locations within a room, we found the distribution of RSS to change significantly between different times of the day. The effect was larger with moving than with fixed receivers. This confirms our findings from the first study. We conclude that estimation methods may not depend on a specific RSS distribution. Many proposed localization algorithms use techniques such as filtering to improve the locator accuracy based on specific data samples and show good results. However, the findings of our long-term, user-driven study showed the unfeasibility of data-specific improvements such as filtering. Regarding the difference of fixed to moving receivers, we found that the standard deviation of RSS is generally higher for fixed receivers. The mean RSS on the other hand does not vary much, unless measuring very close to the AP. Generally, we found the deviation in RSS to be smaller the farther away from the AP. However this is not always the case for moving receivers. Confirming the findings from our first study, we found no significant correlation between changes in RSS from different APs. Also in line with the first study is the insight that the presence of people, in particular if they move around, greatly influences RSS.

Summarizing our findings, we conclude that it is not sufficient to measure RSS for a few seconds only. In order to guarantee high accuracy and precision, a WiFi fingerprinting based indoor positioning system must be able to rely on measurements taken for minutes and, once again, repeated over many days. Only then it is possible to effectively cope with both short and long-term signal variations. This finding somewhat invalidates many evaluations of proposed systems where measurements where only taken instantly and in very controlled situations.

The reality for advanced design today is dominated by three ideas: distributed, plural, collaborative.

– Bruce Mau, 2004

4 Collaborative Labeling

It is immanent to the principle of fingerprinting that the result of a look-up is better, i.e. more accurate, the more fingerprint data we have to compare to. In regard to location fingerprinting where the radio map contains measurements of often and fast changing radio signals, it is thus necessary to train the radio map with as many fingerprints as possible in order to get satisfying results. We have seen in Chapter 3 that WiFi RSS fluctuates substantially, both short and long-term, which means that fingerprints have to be maintained over time. Consequently, radio map training has to be a continuous task that is repeated over and over again during the whole lifetime of the positioning system. In particular, the results of Chapter 3 indicate that the best way to cope with long-term variance is to update the radio map frequently by taking measurements at different times of the day and days of the week. We believe this will not only address variations of unknown causes, but also infrastructure changes such as failing or replaced access points.

To overcome all these problems, we propose a new approach to location fingerprinting, which we call *collaborative labeling.* Instead of having trained staff collecting fingerprints during a designated off-line phase, collaborative labeling relies on user contributed labeling. With collaborative labeling, any user is empowered to add new labels to the radio map, update or correct existing labels, or simply add more measurements to an existing fingerprint. Thus, collaborative labeling does not require dedicated training phases but rather allows for continuous updates of the radio map.

Parts of this chapter are based on my paper entitled "Redpin — Adaptive, Zero-Configuration Indoor Localization through User Collaboration", which was published in proceedings of the First ACM International Workshop on Mobile Entity Localization and Tracking in GPS-less Environment Computing and Communication Systems held in San Francisco, USA, in September 2008 [18].

The focus of this chapter is to analyze the feasibility of collaborative labeling and its potential to improve indoor positioning. Using what we have learned so far, we consequently set our focus on *using existing hardware*, building a system that does *not require maps* and, most importantly is *easy and cost-efficient to setup and use* rather than trying to improve accuracy. We start by introducing the concept of collaborative labeling and fingerprinting along with the related terminology. In Section 4.2 we will introduce and analyze different systems and services that allow or require user contribution. In doing so, we will assess the properties and features that are relevant to designing a collaborative indoor positioning system in detail. Subsequently, we show how the problems and challenges stated in Chapter 1 can be solved by collaborative labeling as an approach to indoor positioning. We will discuss the concept of collaborative labeling in location fingerprinting systems and present the design of a reference implementation that was built as proof of concept in order to verify the feasibility of our method in Section 4.3. In Section 4.3.4 we discuss the implementation of this design for different mobile platforms. A discussion and evaluation of our approach concludes this chapter.

4.1 Building Principles

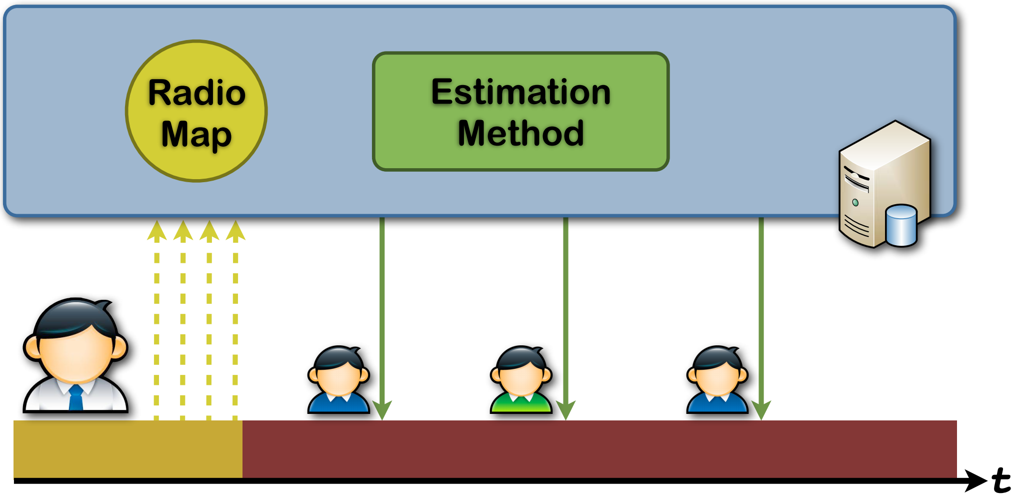


Figure 4.1: The "traditional" approach to train the radio map: before use, an expert adds measurements to the fingerprints in the radio map during a designated, offline training phase.

The basic principle of collaborative labeling is user contribution. Unlike most existing indoor positioning techniques [111] that rely on a designated administrator to collect characteristic radio signal information as illustrated in Figure 4.1, collaborative labeling relies on the system's user to contribute measurements to the radio map. This paradigm shift not only simplifies the setup and the inherently required maintenance of the positioning system but it also allows to benefit from users' knowledge. Especially in open areas, such as the entry hall of a big train station for example, it is not possible to define valid location identifiers as people tend to name places differently.

The key concept of collaborative labeling is to empower the users of the system to create and manage the locations in a collaborative way as illustrated in Figure 4.2. Using collaborative labeling, every user can create, modify and, most importantly, use location information that was created by other users. Moreover, the users may update the radio map at any point in time. We believe that this collaborative approach is feasible as people evidently like to participate and contribute to folksonomy-based or crowd-based systems. The massive success of websites such

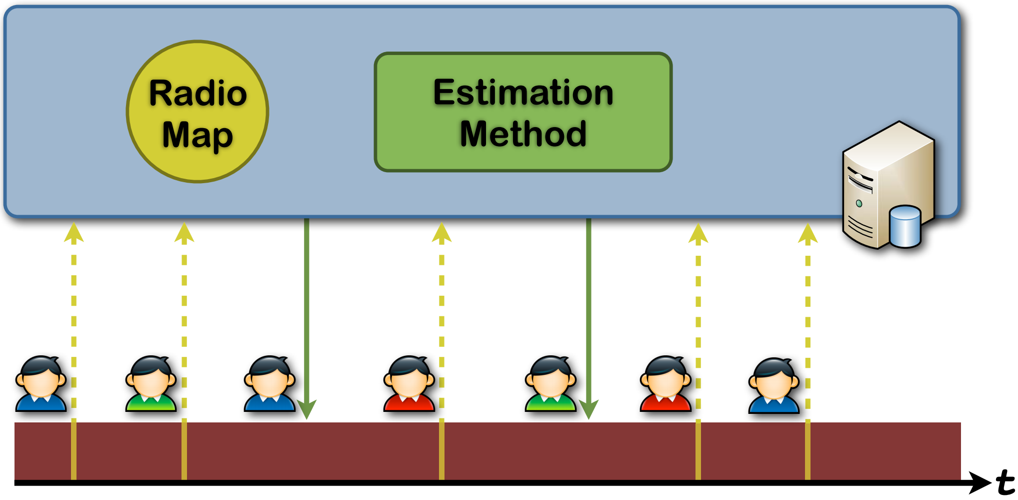


Figure 4.2: Collaborative Labeling: Every user may add new fingerprints or add measurements to existing fingerprints at anytime. There is no designated training phase.

as Wikipedia[1] or OpenStreetMap[2] is just one piece of evidence for this. Recent research in this area has in addition shown that people contribute because of ideological reasons and even more so, because it is fun [31, 122]. This, however, entails that a system that relies on the contribution of its users should provide an appealing user interface. We will discuss these aspects in more detail later on in this chapter.

Bhasker et al. [15] previously explored collecting calibration data during use, rather than in a separate training step. Their localization system employs a two stage process. First it computes geometric location. The result is shown on a map and can be corrected if necessary. Corrections are treated as virtual access points and given higher priority when calculating locations. However, this method requires having a map and interrupting the user's primary activity to collect input. The reported mean error is about seven meters. The system also allows only one correction per location. Unlike Bhasker et al. we collect room labels directly from end-users during their use of the system.

However, employing a potentially large user-base to train the radio map and thus lowering the effort and cost required to setup and maintain

[1] http://www.wikipedia.org

[2] http://www.openstreetmap.org

fingerprints is not sufficient. As we have discussed in Chapter 1, the cost to setup and maintain a positioning system is only one of the problems that hinders broad deployment of indoor positioning systems. Any collaborative system is only as efficient as the number of participating users and the degree of their contribution respectively. This means that we have to lower the barrier to become a user and contributing to the radio map as much as possible. Not only does this imply an easy-to-use interface, but also to support as many hardware terminals as possible. Ideally, the system works on the existing hardware the user already owns and it is thus not necessary to purchase new devices just for the purpose of enabling positioning. Moreover, we found that it is often very complicated and time consuming to get map data or floor plans. As we have shown in Chapter 2, room-level precision is sufficient for most applications in the Ubicomp domain. Thus, following the example of Castro et al. [33] as well as Haeberlen et al. [61], we use an unstructured symbolic location model in order to reduce the effort required to setup the positioning system to an absolute minimum.

We allow for multiple different symbolic identifiers per location. This enables users of the system to actually *label* a location, i.e., to name or rename a location as one thinks best while adding semantic information to the location information. Hence, we do not only get measurements from the system's users but meaningful location labels in addition. This is of particular interest in environments such as office buildings, where locations are often labeled using a logical but otherwise meaningless labeling policy. For example, the room used for coffee breaks in our building is officially labeled "CNB H112", but of course no one calls it this way. Over the first few months of being in this new building, everybody called this room differently until at last, after several weeks, the name "lounge" emerged victorious. Collaborative labeling does not only enable this kind of crowd-based labeling, it can also facilitate this process by assessing the significance of fingerprints based on the number of contributing users.

4.2 Harnessing User Collaboration

Over the past five to seven years, we could witness a literal inversion of the way content is generated and consumed (compare [78]). With the Internet and in particular the web becoming more and more ubiquitous, new services made it ever easier for everybody to add content. This has truly reversed the traditional pattern of content-creation. Instead of very few people creating content for everybody, we nowadays observe millions of people creating content like blogs, news, videos, music, short messages, status updates and many more. This paradigm shift, which goes under many names such as *crowdsourcing* or *collective intelligence*, shows in the success of hugely popular websites like Flickr[3], YouTube[4] or Wikipedia[5]. All these sites rely on their users to add and create new content, mostly without enforcing traditional quality controls. But even simpler services like the social bookmark website delicious[6] [134] rely or enable their users to contribute to the content. In the case of the aforementioned delicious, users can tag bookmarks of websites with freely chosen terms. These tags are subsequently analyzed and grouped by the service to help the users find interesting or relevant content more effectively. This technique of harnessing user knowledge by letting people tag and describe any type of content is commonly known as *folksonomy* [103, 129]. But folksonomies are not just a simplified sub-class of crowdsourcing. As we will discuss in more detail later on, Robu et al. [136] for example showed how folksonomies can be used to extract simple tag vocabularies by analyzing correlations. In the following, we analyze how and why user collaboration works and discuss some systems making use of this.

4.2.1 Crowdsourcing

There are many different definitions of the term crowdsourcing. Brabham for example defines crowdsourcing as *"an online, distributed problem-*

[3]http://www.flickr.com
[4]http://www.youtube.com
[5]http://www.wikipedia.org
[6]http://www.delicious.com

solving and production model" [25]. Generally, Jeff Howe and Mark Robinson are credited with having coined the term crowdsourcing in the June issue of Wired magazine in 2006 [75]. According to Howe and Robinson, crowdsourcing *"... represents the act of a company or institution taking a function once performed by employees and outsourcing it to an undefined (and generally large) network of people in the form of an open call."* [76]. This process of "outsourcing" a potentially business critical task to the general public is something that is difficult and was not thought possible until a few years ago. How can a company or institution trust their users to do the right thing and do it carefully? Surowiecki finds that *"under the right circumstances, groups are remarkably intelligent, and are often smarter than the smartest people in them"* [149]. In every large enough group there is a specialist or an enthusiast that knows of a very good solution. But while this solution might be very good, it almost certainly misses a point. This is where crowdsourcing stems its strengths from. It creates a "wisdom of the crowds" which is derived from aggregating solutions, not from averaging them [24, 26].

The increase in popularity of crowdsourcing mechanisms even at large companies is not only due to the fact that this "wisdom of the crowds" yields results of great quality. Outsourcing tasks to the public allows for potentially huge savings in expense for personnel. Yet, if the users contributing to such a system are not payed a salary, what motivates them to participate? Why do millions of people spend hours and hours working on "other people's problems"? One explanation found by Huberman et al. [78] is that people contributing to crowdsourcing systems perceive their contribution as a private good. Instead of a payment for their efforts, users recognize the attention they receive from others as compensation. As Hubermann et al. showed in 2004 already, attention is a resource so highly valued that *"people are often willing to forsake financial gain to obtain it"* [77]. This is particularly true in the world of academia where attention is usually the only reward. We write papers and try to publish them not only to help solving the world's problems and bringing society forward, but also to get the attention and appreciation of others. As

Franck [50] found, we cite and assess the quality of other researchers' work according to the attention it gathers. Moreover, recognition and in particular status are considered to be the main motivators of users in online communities for contributing [104]. Crowdsourcing is also often compared to open source software development, where a small group of contributors create code that is afterwards used by thousands of passive users who are not actively participating in the process of software development. Moreover, open source models generally emphasize the common good [20, 105]. And yet, studies have shown that the problem commonly known as "free-riding" is not prevalent. Apart from being interested in the engineering challenge, the desire to develop creative skills or being intrigued to find clever solutions, people contributing to open source projects strive for prestige and recognition by the community [117]. This phenomenon was described by Raymond as "gift culture", where participants *"compete for prestige by giving time, energy and creativity away"* [133]. However, it seems that attention and prestige are not the only motivators to contribute to open source projects [64, 146]. Most probably stemming from the fact that software engineers like to solve challenging development tasks and building something new, Ghosh found in his study in 2005 [55] that developers just plain simply take pleasure in writing code. Or to cite Linus Torvalds, the father of the Linux operating system: *"Most of the good programmers do programming not because they expect to get paid or get adulation by the public, but because it is fun to program."* [54].

In the following, we will discuss the issue of what motivates people to contribute to crowdsourcing in more detail by analyzing one of the most successful and most popular service built-on user contribution.

4.2.2 Wikipedia

Wikipedia is a free, web-based encyclopedia that allows anyone to write, correct or update articles. It makes use of the same principles as the open source models discussed above where users contribute their resources such as knowledge and time to create publicly available content by means of a collaborative effort [68]. It is counterintuitive at best that an encyclopedia, written by many and often even anonymously, could be precise. And yet this is exactly what has happened. For this to occur, the participating volunteers' motivation is crucial for sustaining Wikipedia [43]. In an effort to explain volunteering activity, Clary et al. identified six motivational categories [37, 38].

- **Values** Contributing allows volunteers to express values related to altruistic and humanitarian concerns for others.
- **Protective** Addressing volunteers ego, participating might help to reduce negative feelings about oneself or address personal problems.
- **Understanding** Through participation, users may acquire new learning experiences and increase their knowledge or exercise skills.
- **Social** As contribution is a collaborative process, people strengthen social relationships, i.e. they have the chance to find new friends or maintain friendship.
- **Enhancement** Similar to the protective category, this category attends to users' ego, however, in a more positive way. Enhancement describes volunteers possibility to develop and grow psychologically.
- **Career** Lastly, contributing may also help to gain and strengthen skills related to career.

In his study "What motivates Wikipedians" [122], Nov added two more categories, both of which we have already encountered in the previous section: *Ideology* and *Fun*.

- **Ideology** This category includes volunteers opinion and belief that information should be free.
- **Fun** Users contribute to the system quite simply because the activity of doing so is fun.

Working with these eight categories, Nov conducted a study, sending out a questionnaire to 151 Wikipedia users [122]. Motivation was measured using the volunteering motivations scale, introduced by Clary [38], adjusted though to the context of Wikipedia. Participants were asked to state how strongly they agree to a questions representing mentioned categories. Quite surprisingly, Nov found the top motivations to be Fun and Ideology, while others such as Social, Career, and Protective turned out not to be strong motivations. From this, Nov concluded that in order to be successful, a system that relies on user-generated content must focus marketing and retention offers on these motivators to recruit and retain volunteers. In addition, Bryant et al. found in their descriptive study of 2005 [31] that user retention can be leveraged by having expert users setting standard usage patterns. The authors used methods from the field of *social activity* to understand why people become collaborators in Wikipedia. Analyzing the personal user pages and interviewing users by mail and phone, Bryant et al. found that newcomers become active members most easily by observing the practices of experts.

4.2.3 Folksonomy

A folksonomy is a mechanism of classification derived from the methods used in crowdsourcing. Unlike the latter, which can be used to solve more complex problems, folksonomies are the result of *"personal free tagging of information and objects for one's own retrieval"* [153]. The term *folksonomy* was coined by Thomas Vander Val in a discussion on a mailing list and is a combination of the words "folk" and "taxonomy" [154]. Vander Val's definition [58, 153] also clearly states that the act of tagging is performed in a open and shared social environment, done by the person consuming the information in the end. Thus, while conceptional based on crowdsourcing and the methods of collaboratively creating and managing information, folksonomies are used primarily to annotate or categorize content [129]. Hence, it is often refereed to as social indexing and social or collaborative tagging [103]. Another important attribute of folksonomies is that, unlike crowdsourcing, folksonomies are comprised of terms in a flat namespace, i.e., there is no hierarchical relationship between the tags [115].

The concept of folksonomies became famous with the arrival and success of new websites like Flickr or Digg[7] that provided services such as photography annotation and social bookmarking. Many of these sites use so-called tag clouds to visualize different tags and their importance, i.e., popularity. In the process of generating such clouds, some tags were found to be of greater relevance than others. In 2007, Halpin et al. [63] were able to show that consensus around stable distributions and shared vocabularies can emerge despite the lack of a controlled vocabulary.

To understand the impact and benefit of folksonomies, Mathes examined user-generated metadata by means of two web services [115]. In doing so, he particularly studied the difference between metadata created by professional authors and metadata created by the crowd. Thereby, Mathes found the primary problem of the former to be scalability and hence its impracticality for very big amounts of content. Although dedicated professionals working with complex, detailed rule sets and vocabularies are generally believed to produce a more detailed and accurate result, their work is costly in terms of time and effort to produce. A folksonomy on the other hand may be used to cope with even vast amounts of data, as potentially millions of users help creating the desired metadata. In addition, Mathes found the most important strengths of a folksonomy to be its ability to directly reflect the vocabulary of its users. This supports Merholz's finding [116] that a folksonomy reveals the "desired lines", or in his own words: "*A smart landscape designer will let wanderers create paths through use, and then pave the emerging walkways, ensuring optimal utility.*" Without any doubt, applying folksonomy allows for far lower costs in terms of effort and time compared to "traditional" systems using elaborate classification schemes.

[7] http://digg.com/

4.2.4 Games with a Purpose (GWAP)

Building on the idea of a folksonomy, von Ahn created a concept that combined the “human computation” paradigm, extensively used by the Open Mind Initiative[8] (e.g., [147, 148]), with the simple insight that people like playing games. People today spend many hours playing computer games [151] and for the first time in history, hundreds of millions can easily collaborate via the Internet. While computers are great for many purposes, they fail at tasks that are almost trivial for humans to perform, tasks like the ones we discussed in the above sections such as labeling images. But as von Ahn correctly stated, humans, unlike computers, require some kind of incentive, and obviously playing games might just be the seductive method to encourage users to participate. Making the labeling task the challenge of a game allows to take advantage of peoples’ desire to be entertained. Consequently, von Ahn designed the process of tagging and labeling like a game, where two or more “players” are asked to compete in the act of performing the labeling task. The term *game with a purpose*, or GWAP for short, was later coined by Lenore Blum, a research colleague of von Ahn.

In 2004, von Ahn published his first paper on this topic in which he presented a game to created labels for images [152]. By today, many more games with a purpose, i.e., applications of von Ahn’s human computation paradigm such as sound labeling or object tracing in images, followed and von Ahn himself created a website dedicated to the most interesting ones[9]. In the following, we will study the labeling images game in more detail.

The purpose of the labeling images game is to provide labels for an image. If done “correctly”, players receive points. Thus by playing the game, players help determine the contents of the image [152]. The game can be played online by two partners that get randomly paired-up from the large number of people accessing the website. Neither of the players is told with whom they’ve been paired-up and they have no means of

[8]http://openmind.org/

[9]http://www.gwap.com

communicating with each other. The only thing both players have in common is an image both can see. Both players are asked to describe the contents of this image in their own words, providing one or more strings. The more strings match between the partners, the more points both receive. Since they don't know each other and can't communicate, the obvious thing to do in order to get points is to type something related to the image. And as von Ahn found in his study, the string on which the players agree is in most cases a good label [152]. In an evaluative study, von Ahn had 13630 people playing the game for four months, thus generating 1271451 labels for 293760 different images. More than 80% of the players played the game at least twice and, quite astonishingly, 33 players played the game more than a thousand times. From these numbers, von Ahn concluded that the game is fun. Regarding the quality of the labels generated, von Ahn found that all (100%) of the labels made sense with respect to the images retrieved [152].

4.2.5 Collaborative Mapping

Another example, closely related to our problem of collaborative labeling of location fingerprints, that shows just how successful approaches like crowdsourcing and folksonomies are, is the remarkable popularity of websites that allow their users to create maps in a collaborative manner. The most popular of these services is OpenStreetMap[10], a free and open map service that allows its users to view, edit and make use of geographical data of the world. Unlike commercial services like Google Map Maker[11], geographical data is provided under a Creative Commons license [12]. Using any GPS tracking device, users can record their routes and walks while being en route and upload this information later on. In addition, users may edit or annotate geographical data manually and the community powering OpenStreetMap has been given the right to carbon copy aerial photographies from Yahoo and Bing. Counting around 2500

[10]http://www.openstreetmap.org

[11]http://www.google.com/mapmaker

[12]http://creativecommons.org/licenses/

users back in July 2006, the OpenStreetMap website now has more than 350000 registered users. The maps that resulted form years of collaborative mapping are astonishing. A quick, non-scientific comparison of the map quality between OpenStreetMap and Google Maps reveals almost equal level of detail, at least in populated regions of the world such as North America or Europe. The information level in lesser populated areas such as Madagascar is only coarse. Interestingly but not surprisingly, OpenStreetMap provides more accurate information when it comes to naming small things like creeks or forest tracks. This might stem from the type of usage, i.e. that OpenStreetMap users add data to the website that they recorded in their spare time, pursuing a hobby like jogging or ridding a bike. This is also reflected in the fact that OpenStreetMap provides purpose-built maps for a specific use, like for example the OpenCycleMap[13]. Another indicator of OpenStreeMap's success is the interest of CloudMade, a commercial company funding OpenStreetMap to some extent and using the data for their own products. A slightly different service is provided by WikiMapia[14]. Inspired by the success of Wikipedia, two entrepreneurs created a website that allows its users to add notes to any location. Although registration is not required to view, edit, or add notes, WikiMapia has over one million users from around the world who marked and linked over 14 million places.

4.2.6 Location Sharing

In previous sections we have shown that people actually do create and share meta-data. In the case of using these concepts for indoor positioning systems however, we ask people to tag locations. In the process of doing so, it is inevitable that users have to reveal their own location, at least to some level of detail. This might, from a privacy point of view, be problematic. However, we found many very successful services that allow people to share their location. One example of such a location

[13] http://www.opencyclemap.org
[14] http://wikimapia.org

sharing service is Yahoo's Fire Eagle[15]. Fire Eagle is a location broker service that allows its users to share their current location with multiple services in a safe and controlled manner. Hence, users can not only update and access location information on Yahoo's website but using any other, authorized, third party application. While we couldn't find any numbers on how many actual registered users this services has, Yahoo itself lists over 70 applications making use of the service. Another example of a location sharing applications is Google Latitude[16], a location-aware mobile application, which allows its users to track the location of other users. Once a user allows another user to get updates on her location, the two users can locate each other henceforth. While no exact numbers are revealed, Latitude is believed to have more than 10 million active users.

4.2.7 Discussion

Having studied the many different systems and services successfully harnessing user collaboration, we may conclude that if done right, a collaborative system allows to process huge amounts of data while revealing the information that is most important. As collaborative systems are used by humans participating in social interaction, it is possible to obtain information that could not have been gained by traditional expert systems using systematic data gathering. For example, by allowing the users to create place labels in OpenStreetMap, this service provides labels of places that are of most interest to its users. In addition, the labels created by the users reflect their own vocabulary, i.e., the labels correspond to the names people actually use.

A possible disadvantage of collaborative systems is the potential lack of labels for places that are not of interest to the users. This holds particularly for systems that require users to effectively visit a place or location in order to label it. One example of this is the lack of mapping data in uninhabitated or otherwise unpopular places on OpenStreetMap. Hence,

[15] http://fireeagle.yahoo.net/
[16] http://www.google.com/latitude

without additional means of control, a collaborative labeling system will almost never achieve full coverage as it will not contain labels of places people don't go to or don't care about. But, a localization system used to locate people in case of an emergency for example must guarantee 100% coverage at all time. Thus, a collaborative system may require traditional expert editing in addition to user labeling in order to achieve the desired coverage.

Regarding users motivation to contribute to the system, Nov [122] concluded that in order to be successful, a collaborative system must focus marketing and retention offers on these motivators to recruit and retain volunteers. If fun is the most important motivation for users to contribute, the system has to be fun to use. We believe that this includes not only the user interface, but rather the design of the system in general. For example, if new users that are unfamiliar with the purpose and the inner workings of the system struggle to understand how they might contribute, the system is deemed to be a failure. As we have learned from Bryant et al. [31], newcomers become active members most easily by observing the practices of experts. From this finding, we conclude that a collaborative system must be public and contributor's work must be available to anyone.

Finally, by studying systems such as Google Latitude, we have seen that people do share their location, despite privacy concerns. However, people are aware of the potential risk these systems bear and it is necessary to give the users the right amount of control and establish awareness, i.e., users want to know and understand who is given access to their location data and under which circumstances. From this we conclude that if done right, people will share their location, either for their own benefit or for the common good.

4.3 Redpin

Redpin is the name of our reference implementation, an indoor positioning system enabling collaborative labeling of location fingerprints. Redpin was built not only as a proof-of-concept but moreover to provide an indoor positioning system that is very easy to setup and maintain and even easier to use. Given the challenges discussed and motivate in previous chapters, we wanted to achieve four main goals:

- **Hardware**

  Redpin must not require special hardware but work with standard, existing devices.

- **Cost**

  Redpin must be very easy to setup and maintain. Expert know-how about location fingerprinting must not be required.

- **Accuracy**

  Redpin should at least provide room-level precision.

- **Signal Variations**

  Redpin must be capable of coping with both long-term and short-term radio signal variations.

To achieve these goals, Redpin implements the principles of collaborative labeling as presented at the beginning of this chapter. Redpin enables the end-users of the system to create and alter location labels while using the system and provides a very simple, graphical user interface for these actions. Redpin, at its core, is a terminal-assisted positioning system. While the radio map is stored on a central server, which also provides estimation methods for positioning, the terminal, or client, is used to observer and measure RSS. As illustrated in Figure 4.3, we choose smartphones as terminals and implemented Redpin for three platforms, namely Apple's iOS, Google's Android, and Symbian.

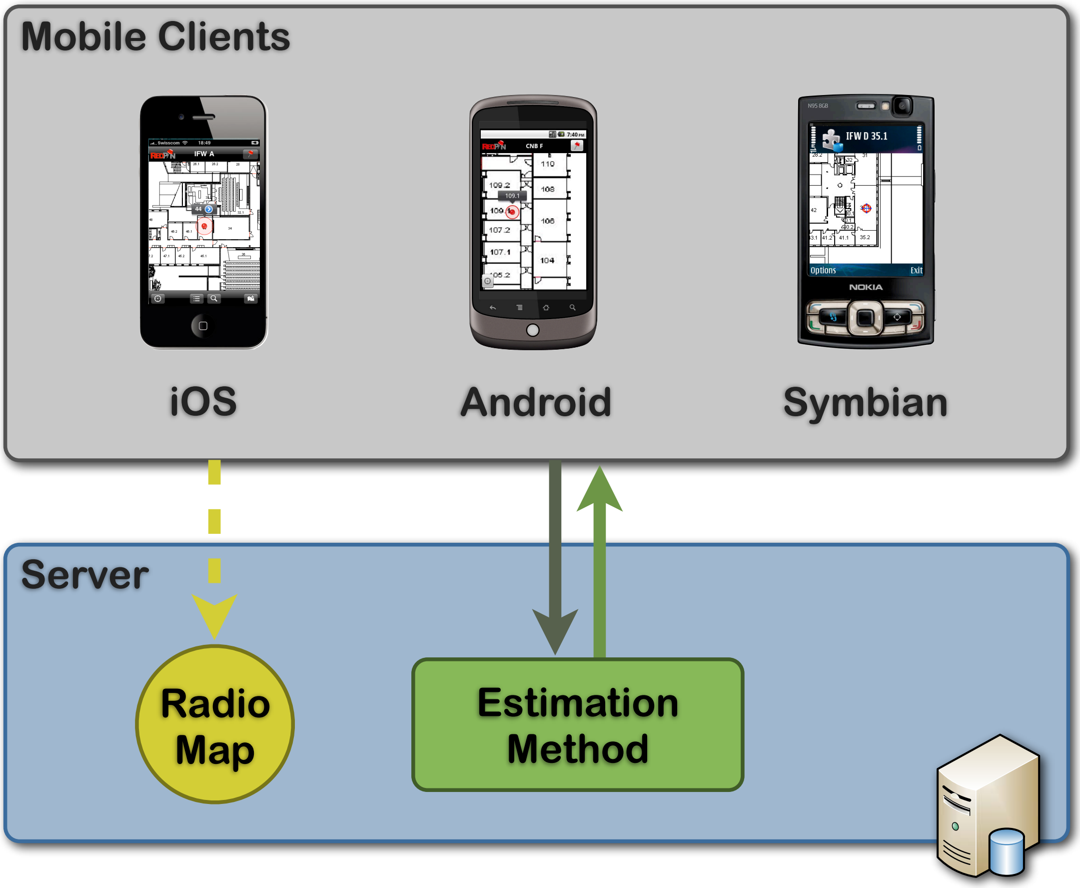
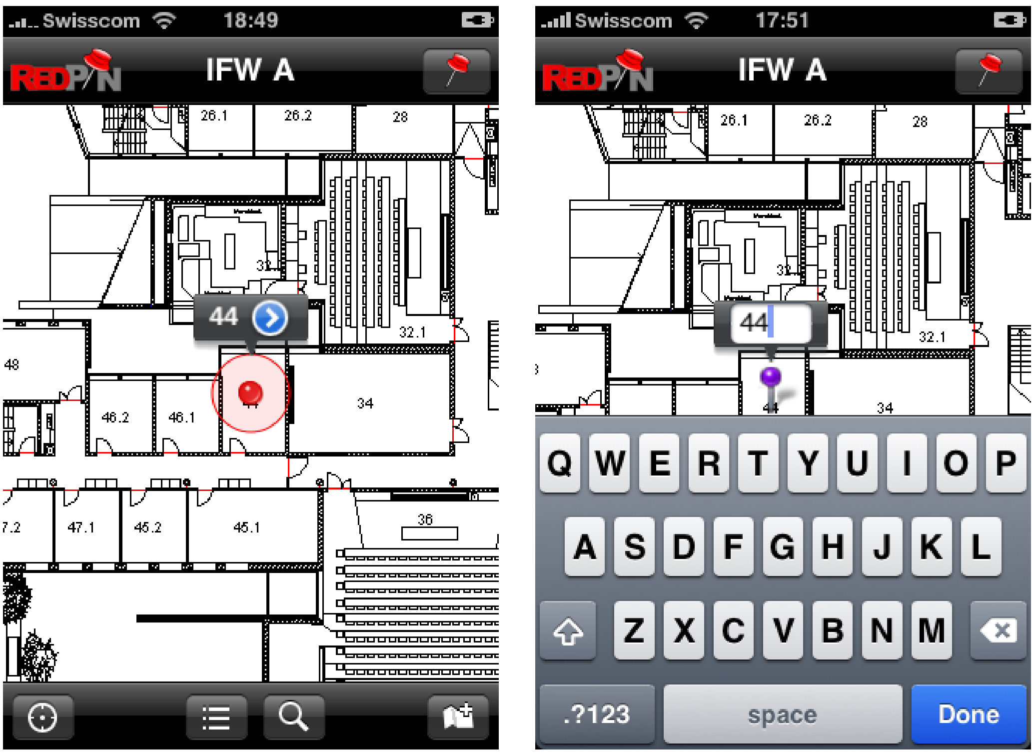


Figure 4.3: Redpin system overview: The backend server provides radio map and estimation method for all three mobile client implementations of Redpin, from iOS to Android and Symbian.

As discussed in Section 4.1, we use unstructured symbolic identifiers to denote locations and places. Hence, from a user's point of view, a location is nothing more than a label. This approach also entails the advantage of being able to forgo a potentially erroneous calculation of exact geographic coordinates. Consequently, localization of a mobile user or device can be reduced to the problem of mapping a set of RSS measurements to a known symbolic identifier, like for example a room number. Note however, that with Redpin it is possible to assign many fingerprints to the same location.

In order to achieve room-level precision, i.e., selecting the correct location given a measurement, Redpin allows to measure the signal strength of the currently active GSM cell, the signal strength of all WiFi access points as well as the Bluetooth identifier of all non-portable Bluetooth devices in range. However, given the API limitations of iOS and Android, we only measure WiFi RSS on these two platforms. Only the Symbian version of Redpin allows to measure all three types of signals. On the latter we could additionally increase the system's accuracy by measuring the signal strength of all GSM cells, and not just the one GSM cell that is currently active, but this is currently not possible with the devices we used.

In the remainder of this chapter, we will present how Redpin works, discuss its design and implementation and discuss its performance by means of an experiment. We will start by explaining Redpin on iOS from a user's point of view.

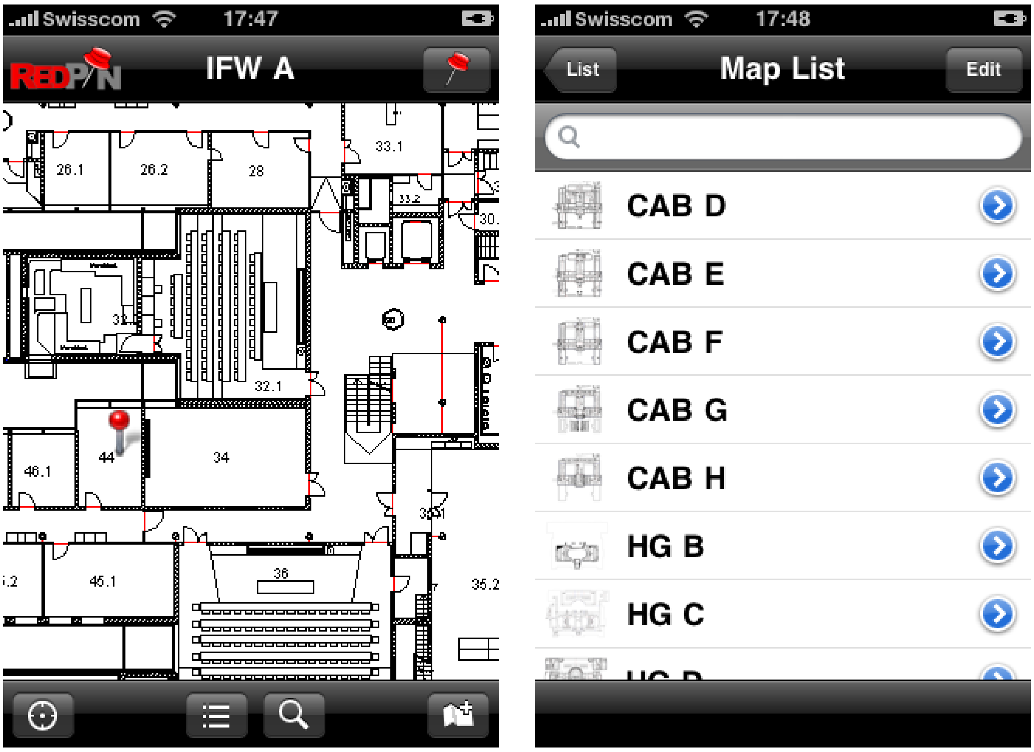


(a) (b)

Figure 4.4: Using Redpin on iOS, the user is shown the current position as a red circle along with the according label. The user can correct this or enter new location labels by tapping the label.

4.3.1 Redpin in Action

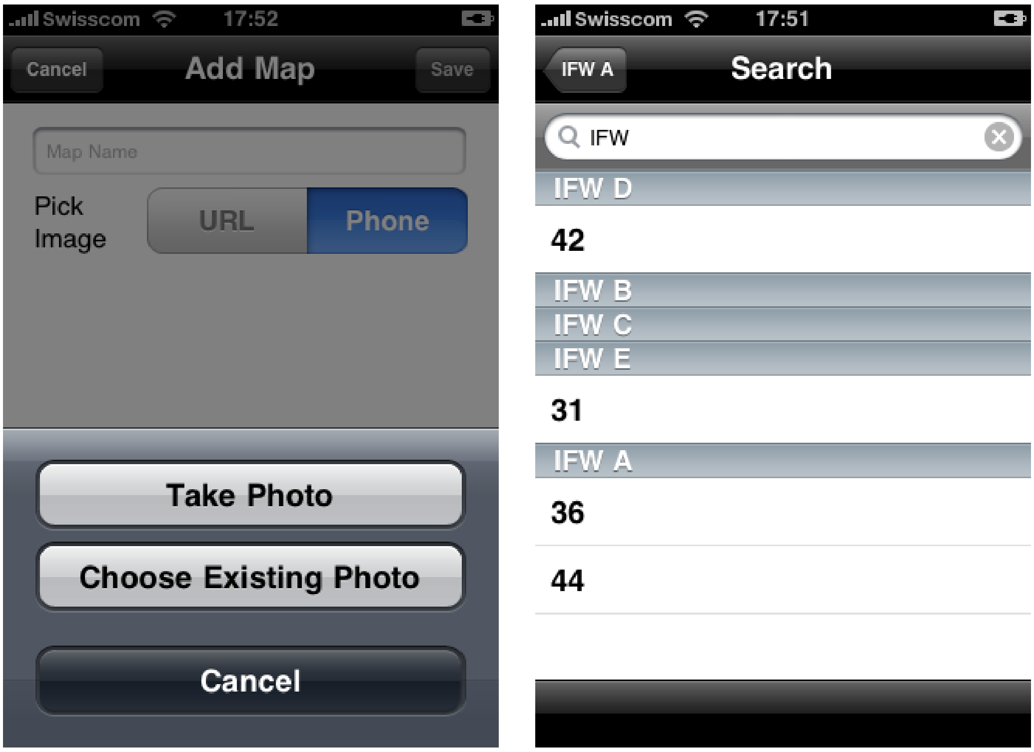
After installing Redpin on iOS, the user can start-up the application right away. Already during initialization, Redpin is scanning for WiFi access points and measures the RSS of all WiFi access points in range. This measurement is then sent to the Redpin server which will subsequently try to locate the mobile device given all known fingerprints in the radio map. If the system can locate the mobile device, the user is presented with the map of the current floor and the current location, which is indicated by a red circle, as illustrated in Figure 4.4(a). The user can change the map section by dragging the map as well as zoom in and out using the “‘pinch” gesture. If the system can not locate the mobile device,



(a) (b)

Figure 4.5: Known locations are shown as red pins in Redpin. By tapping the list button, the user can access the list of all available maps. To switch maps, the user just has to select it from the list.

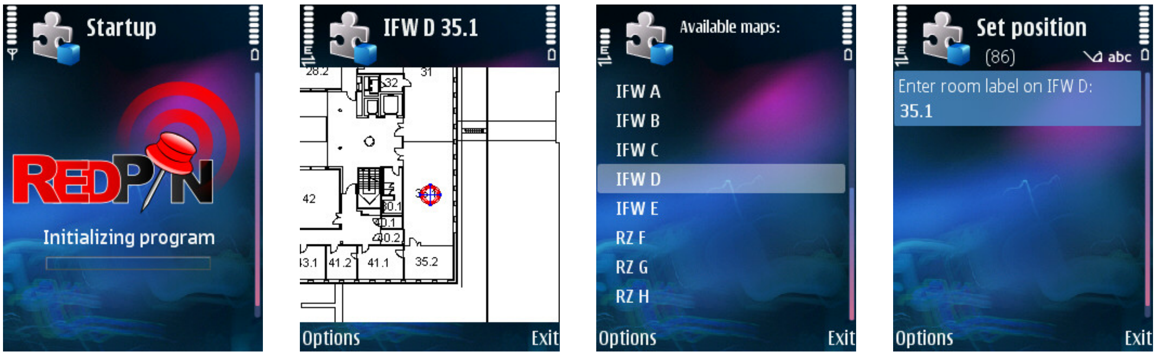
for example because the location is yet unknown, the user is informed accordingly and Redpin will display the last known location. In the background, the system is continuously taking measurements, comparing the last three measurements, thereby trying to detect a stable state. Upon detecting a stable state, the system will again try to locate the device. If the device can still not be located, the user will be prompted to name the place of the current location and indicate the appropriate position on the map. Thus, the user can choose from a list of known floor plans (see Figure 4.5(b)), set the marker (purple pin) to its current position, and enter the name of the current location, for example the room number as illustrated in Figure 4.4(b). In addition, a user can always correct the location in case Redpin provided the wrong identifier. This way several fingerprints may be stored for the same identifier with a different timestamp. In order to display not only the name of the current



(a) (b)

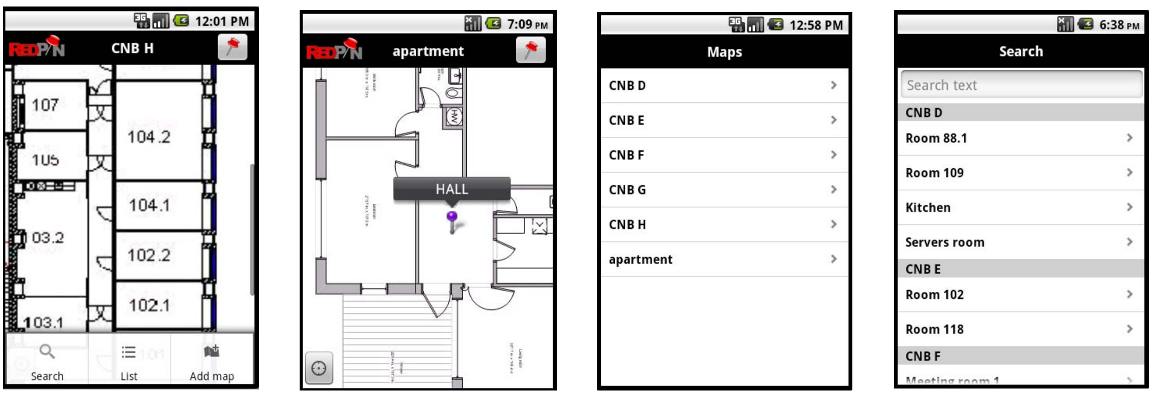
Figure 4.6: Advanced features on iOS: Adding new maps and searching the list of maps and locations.

location but also show the position on the map, the system must be given images or map renderings. These images can be uploaded to the server at any time. However, the system does not require floor plan images since a location is defined solely by its symbolic identifier in Redpin. As illustrated in Figure 4.6(a), the user can indicate the URL to an existing image, choose to upload an existing image from his phone, or take a photograph in order to create a new map. In addition to browsing the list of location labels, the user can also search it by entering any part of a label. The result list, as illustrated in Figure 4.6(b), is updated while the user is typing.



(a) (b) (c) (d)

Figure 4.7: Using Redpin on a Nokia N95: The user interface is similar to iOS. Instead of entering and correcting labels directly on the map view, the user is presented a “Set position” dialog (d).



(a) (b) (c) (d)

Figure 4.8: Using Redpin on Android: the interface is identical to the iOS version of Redpin.

4.3.2 Architecture

Being a terminal-assisted system, Redpin consists of two basic components: the server, which holds the radio map of stored fingerprints and executes the estimation method to determine the current position, and the client, which gathers and collects radio signals from different wireless devices in range to create a *measurement* and provides the user interface.

While the component to collect radio signals has to run on the mobile device for obvious reasons, the estimation method could be run either on a central server or on each mobile device separately. As discussed before, while running the estimation method (and hence storing the radio map with the fingerprints) locally would be beneficial considering the user's privacy, we need to store this data on a central server in order to simplify user collaboration. This way a user can immediately make use of any changes made to the radio map by every other user.

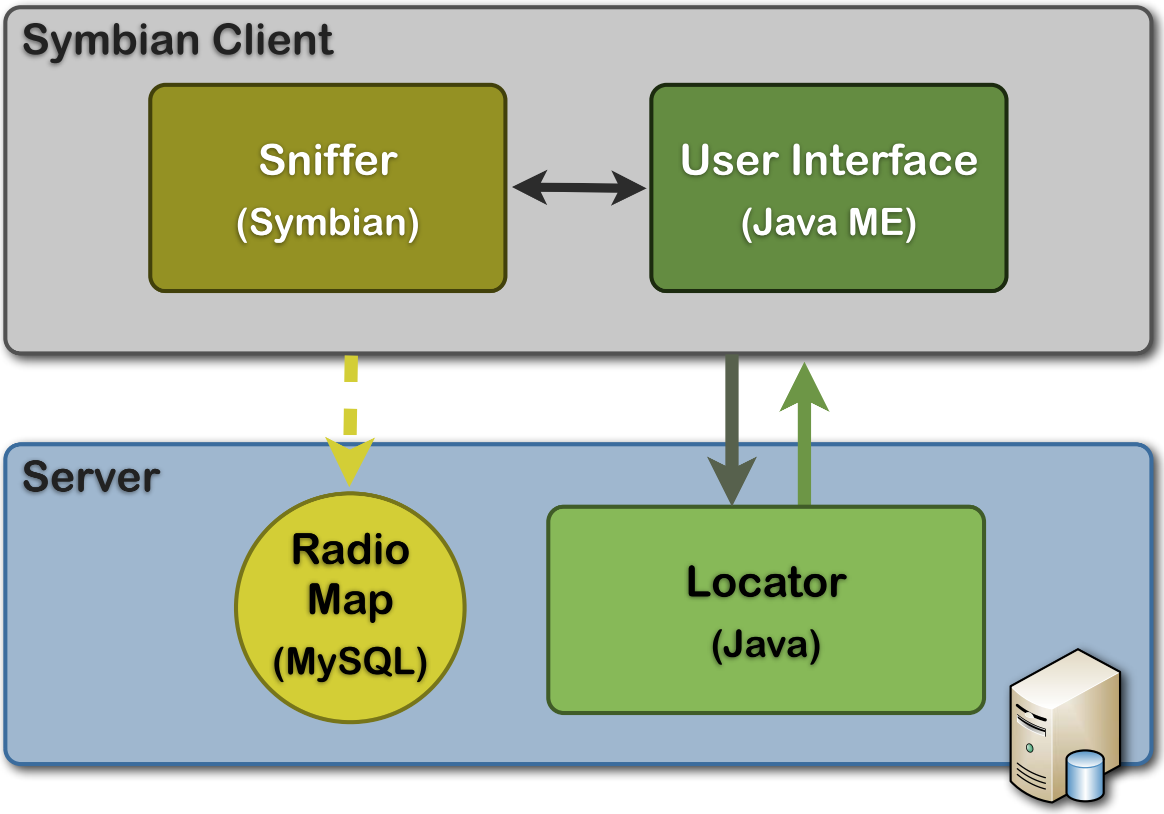


Figure 4.9: System Architecture Overview of the Redpin Implementation for Symbian.

Hence, Redpin implements the radio map and the estimation method as a server service, using Java and MySQL as illustrated in Figure 4.9. For the communication with the client, Redpin provides a well-defined interface and communication protocol. We will discuss the two in more detail later on.

The client's main job is to measure and collect radio signals of all devices in range. Obviously, having many readings from many different devices is favorable as this additionally helps to set the fingerprints apart. Unfortunately, not all platforms on which we implemented Redpin so far allows access to WiFi, GSM, and Bluetooth. On iOS, the API forbids access to GSM and Bluetooth, only WiFi is accessible[17]. On Symbian, we use Java Micro Edition for the GUI and all communication aspects, and Symbian Series 60 to collect the measurements. As illustrated in Figure 4.9, we refer to this special application as *Sniffer*. This separation was necessary, as only the Symbian API would allow us to get the information we wanted to collect.

4.3.3 Redpin Server

The Redpin server, hosting the radio map and the estimation method, provides several services for mobile clients. First and foremost, it provides a service that allows to store and update the fingerprints in the radio map. This service is called whenever a mobile client creates, corrects or redefines a location label. Another service allows mobile clients to create and retrieve maps, i.e., images of the floor plan that are associated with a certain location. Most importantly, the server provides a service to determine the position of a mobile device, i.e., to compare a measurement to all known fingerprints and selecting the location that matches best. In the following, we will discuss these services and the necessary concepts in detail.

[17]While it is possible to access this information on iOS, it is not allowed by Apple's App Store Guidelines, which prevents publication of Redpin in said store.

Communication Protocol

All services provided by the server are made available to the clients by means of HTTP GET and POST calls, using JSON[18] to encode the payload. As there are many JSON libraries for all supported platforms, from iOS to Android and Symbian, implementing the data communication between the server and the client is straightforward. In the following, we present the definition of the request as well as the response using EBNF-notation.

```
request = '{' '"action":' action [',' '"data":' object ' ] }'

action = '"setFingerprint"' | '"getLocation"' | '"getMapList"'
       | '"setMap"' | '"removeMap"' | '"getLocationList"'
       | '"updateLocation"' | '"removeLocation"'

object = fingerprint | location | map | measurement

id = ['"id":' Integer ',']

fingerprint = '{' id '"location":' location ','
       '"measurement":' measurement '}'

location = '{' id '"symbolicID":' String ',' '"map":' map ','
       '"mapXcord":' Integer ',' '"mapYcord":' Integer ','
       '"accuracy":' Integer '}'

map = '{' id '"mapName":' String ',' '"mapURL":' String '}'

measurement = '{' id ['"timestamp":' timestamp ',']
       ['"gsmReadings":' gsm ','] ['"bluetoothReadings":' bluetooth ',']
       '"wifiReadings":' wifi '}'

wifi = '[' [ wifireading {',' wifireading } ] ']'
wifireading = '{' id '"bssid":' String ',' '"ssid":' String ','
       '"rssi":' Integer ',' '"wepEnabled":' bool ','
       '"isInfrastructure":' bool '}'

bool = 'false' | 'true'
timestamp = Long (* unix time stamp *)
String = '"' { Char } '"'
```

Listing 4.1: Shortened Definition of Redpin Request

[18] JSON - the JavaScript Object Notation is a well-defined, lightweight data-interchange format, which is easy for humans to read and write (http://www.json.org).

Request A request to the Redpin server must always contain an *action*, i.e., an identifier of what the client wants from the server (e.g. *getLocation*). The action is followed by an *object*, which is either a *fingerprint*, a *location*, a *map*, or a *measurement*. All objects are again well defined which allows for easy and fast parsing methods on both the client and the server. A fingerprint for example has an id, a location and a measurement. A measurement may contain any number of GSM, Bluetooth or WiFi readings.

Response A response from the Redpin server is equally simple in structure. Every response contains a status message, indicating whether the request could be processed successfully or whether the call prompted problems or even failed entirely. In case the request could be processed successfully, the response contains a response object or a list of response objects (for example if asked to send the list of available maps). The definition of response objects is the same as for the request (see listing above).

```
response = '{' '"status":' status [ ',' '"message":' message ]
               [ ',' '"data":' data ' ] }'

status = '"ok"' | '"failed' | '"warning"' | '"jsonError"'
data = list | object
list = '[' [ object {',' object } ] ']'
```

Listing 4.2: Shortened Definition of Redpin Response

Example In the following listing, we present a simple example of a Redpin request-response call from a client to the server. Using the *setMap* action, the client tells the server to create a new map object for the floor "IFW A", using the map image given by the mapURL. After successfully creating a new map object, the server responds with an "ok" status message containing the unique id used to identify the map object throughout the system, i.e. on the server as well as on all client devices.

```
request
       {"action":"setMap","data":{
              "mapName":"IFW A",
              "mapURL":"http://www.redpin.org/maps/ifw_a.gif"
              }
       }
response
       {"status":"ok","data":{
              "id":57,
              "mapName":"IFW A",
              "mapURL":"http://www.redpin.org/maps/ifw_a.gif"
              }
       }
```

Listing 4.3: Example of a simple Request-Response communication with the Server.

Data Model

The data model used to represent and store required data on the server is given in Figure 4.10. As discussed before, a *location* is defined only by its symbolic identifier. As Redpin uses unstructured symbolic identifiers, there are no further associations between locations. However, to visualize the location in a way that is both appealing and easy to understand, a location may, but is not required to, be associated with a *map*. The map entity is a named proxy for an image file, providing a name and a URL, which can be used to download the actual image data. Every location is associated with exactly one *fingerprint*. The fingerprint represents the radio signal characteristics of a location. Building on the basic data concept of terminal-assisted location fingerprinting that we discussed in Section 2.4.1, every fingerprint may have a (theoretically) unlimited number of *measurements* associated with it. Consequently, a measurement is a collection of radio beacons or *readings* observed at a certain point in time. As Redpin supports WiFi, GSM, and Bluetooth, a measurement may be associated with any number of readings of any type. A reading represents the radio signal transmitted by a wireless device along with available meta-data such as a unique device identifier. To process the readings later on, i.e., when executing the estimation method to determ-

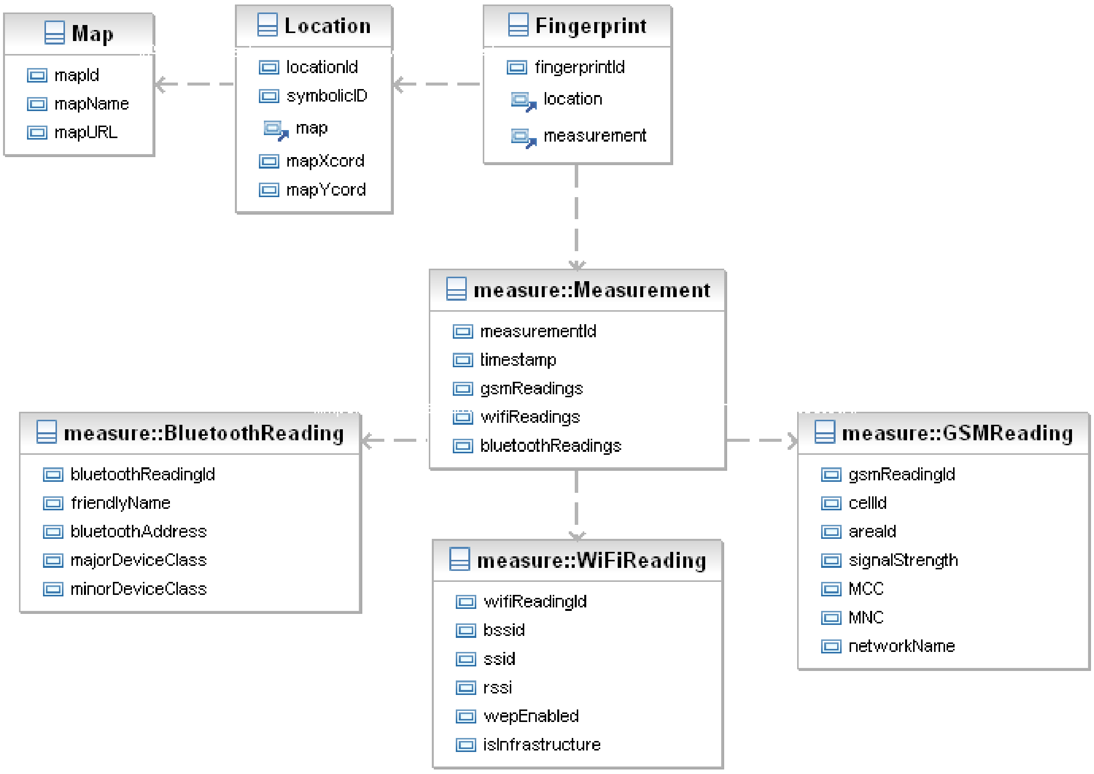


Figure 4.10: The Redpin Data Model

ine a position, every reading must be uniquely associated with the actual devices transmitting the beacon.

To create an internationally unique GSM identifier, we readout the cell identifier (CI), the mobile country code (MCC), the mobile network code (MNC), as well as the location area code (LAC). In the case of WiFi it is sufficient to get the basic service set identification (BSSID) as this value is unique by definition. Bluetooth devices can be uniquely identified by the Bluetooth device address (BD_ADDR), similar to the MAC addresses of a network card. In addition to these unique identifiers, a reading also represents the received signal strength (RSS) as an absolute value. However, due do technical limitations this is only possible with WiFi and GSM beacons.

Estimation Method

Because a location is simply expressed by a symbolic identifier in Redpin, the problem of calculating the current position is reduced to the problem of finding the one fingerprint that best matches the given measurement. For this purpose, Redpin implements a very simple variant of the well-known and often used k-nearest-neighbor (k-NN) algorithm using our own distance metric for comparison. While being ranked among the simplest machine learning algorithms, k-NN entails the big advantage of being "lazy", i.e., all computation may be deferred until classification. With respect to Redpin, this allows to add measurements any time during use and still being able to guarantee that the estimation method will also consider the newest measurements.

To compare different measurements, we defined a simple distance metric that allows to check the level of equality. As the estimation method makes heavy use of this method, the quality of this metric greatly accounts for the accuracy of the positioning. Note that for the reference implementation being discussed in this chapter, we did not focus on perfecting positioning accuracy. However, we also developed more sophisticated methods, which are presented in Chapter 5.

$$d_W(M_x, M_y) = \sum_{i=0}^{\#SSID_{match}} (B_W * ||RSSI_{M_x} - RSSI_{M_y}||) + \#SSID_{nonmatch} * M_W$$

$$d_G(M_x, M_y) = \sum_{i=0}^{\#CID_{match}} (B_G * ||RSSI_{M_x} - RSSI_{M_y}||) + \#CID_{nonmatch} * M_G$$

$$d_B(M_x, M_y) = \#BTID_{nonmatch} * B_B + \#BTID_{nonmatch} * M_B$$

The distance between two measurements, $d(M_x, M_y)$ is computed using a straightforward model. For every type of measurement, Redpin calculates a specific distance. The smaller this distance the more likely we found the fingerprint corresponding the the user's current location. In the case of WiFi for example, the distance $d_W(M_x, M_y)$ is given by the sum of all matching identifiers, i.e., matchings in which the WiFi BSSID occurs in both measurements, multiplied with an additional contribution that is calculated based on the difference of observed RSSI ($||RSSI_{M_x} - RSSI_{M_y}||$). Differing identifiers, i.e., in case the BSSID does not match ($\#SSID_{nonmatch}$), cause a diminution. While matching pairs are rewarded a bonus (B_W ¡ 1.0), non-matching pairs are given a penalty (M_W ¿ 1.0). The calculation works similarly for GSM readings ($d_G(M_x, M_y)$). In case of Bluetooth readings ($d_B(M_x, M_y)$), only the number of matching and non-matching BTIDs are compared while the RSSI is not considered. The overall distance between two measurements M_x and M_y is thus given by:

$$d(M_x, M_y) = d_W + d_G + d_B$$

To determine the position of a mobile device, the estimation method compares the current measurement, as given by the mobile device, with all known fingerprints in the database by calculating the distance metric as described above. If a fingerprint can be found whose distance to the current measurement is smaller than a predefined threshold, i.e., the decision boundary, the associated location will be returned to the mobile device. If multiple fingerprints are found, the system will return the best match. To be able to implement estimation methods other than our

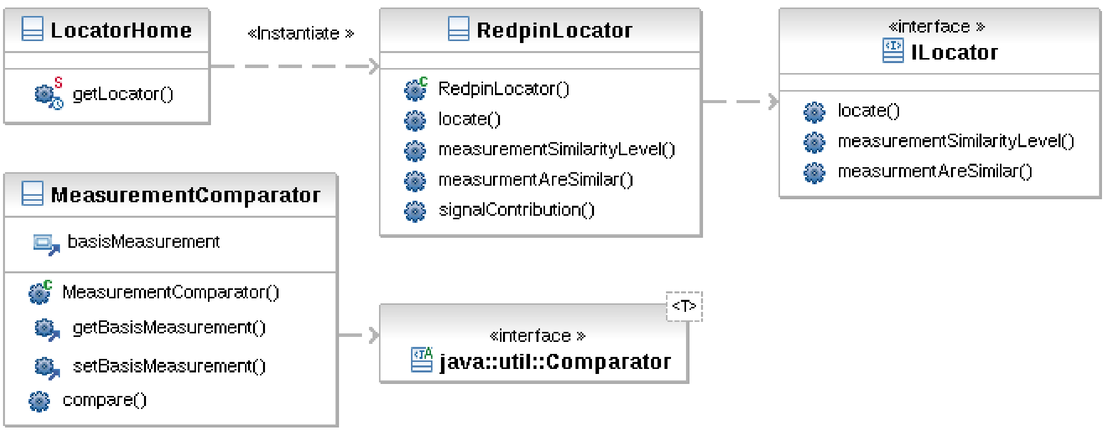


Figure 4.11: Data Model and Interface Design for the Locator Component

k-NN variant, we defined an abstract *locator* interface as depicted in Figure 4.11. This way Redpin can use different estimation methods and is even capable of mixing the results of different algorithms run in parallel. Every locator must implement methods that allow for comparison of similarity of two measurements as well as providing the actual positioning (by means of the *locate* method). Illustrated in Figure 4.11, the estimation method discussed above is implemented as *RedpinLocator*. The distance metric is provided by our own implementation of Java's comparator interface. This abstraction allows to exchange distance metrics easily.

4.3.4 Mobile Clients

To meet our goal of making Redpin as easy-to-use as possible while using existing hardware, we implemented the Redpin client software for the three popular smartphone platforms iOS, Android and Symbian. While designing the UI, we tried to incorporate best-practices and features as used in Google's own mobile map application. Hence, the user-interface itself focuses on presenting locations on a map.

The main feature of all Redpin mobile clients is of course the ability to locate the device. Also, the user can browse and search locations and maps. In addition, and for the purpose of Redpin most importantly, by means of the mobile client the user is capable of adding new map images, creating new locations (i.e. add new labels), as well as correcting existing locations (i.e. adding more measurements to existing locations). A rundown of Redpin in action is given in Section 4.3.1.

While implementing Redpin for iOS and Android was straightforward, Symbian caused many non-trivial problems. In particular, we wanted to use Java for the UI on Symbian in order to create code that would be reusable on other platforms. However, as it is not possible to get radio signal measurements using only the Java API, we had to implement a Symbian application just for the purpose of collecting radio signals. Whereas corresponding libraries on iOS and Android are restricted when it comes to commercial distribution, the required API calls are integrated and easy to use from a software developers point of view. In the following, we will hence discuss the implementation for the Symbian operating system in detail and only present the biggest challenges we faced when implementing Redpin for iOS.

Symbian

Our decision to implement Redpin for Symbian made it necessary to have two applications on the mobile device as illustrated in Figure 4.12. As we wanted our source code to be as easy and portable as possible, we decided to implement the client software in Java ME. But as the limited API of Java ME would not allow access to the current RSS of neither the GSM nor WiFi, we had to implement the Sniffer component in Symbian. Hence, the Sniffer maintains a separate, asynchronous thread for each signal type (GSM, WiFi, and Bluetooth) that collects the appropriate information and stores it in a common buffer. This is necessary, as scanning GSM and WiFi signals is usually a matter of seconds whereas scanning for Bluetooth devices can take up to two minutes, depending on how many devices currently are in the vicinity. To alleviate this problem, we additionally limit the Bluetooth scanner to ten seconds. After this timeout, the Bluetooth scanner will automatically stop scanning and report the devices found so far. Eventually, the Sniffer communicates its current measurement to the Java MIDlet via a local TCP socket. The Java MIDlet on the other hand provides the user with the graphical user interface and handles all the communication with the server. To increase the overall localization accuracy, in our case the success rate of calculating the correct location identifier, we measure three different signal sources, namely GSM, WiFi, and Bluetooth. In addition, we try to read the RSS of as many different sources as possible. While both GSM and WiFi signals may fluctuate, Bluetooth devices are not always detected in the very short time range during which we scan for devices. As a result, measurements may differ considerably, even when taken at the same place and in short succession. Hence, the biggest advantage of having combined fingerprints of GSM, WiFi, and Bluetooth signals is that the estimation method may adapt depending on the actual measurement at hand (see Section 4.3.3 for details).

Unfortunately, Symbian's Telephony API[19] only provides information about the currently active GSM cell. Thus, a GSM reading only contains

[19] Symbian Version 9.2

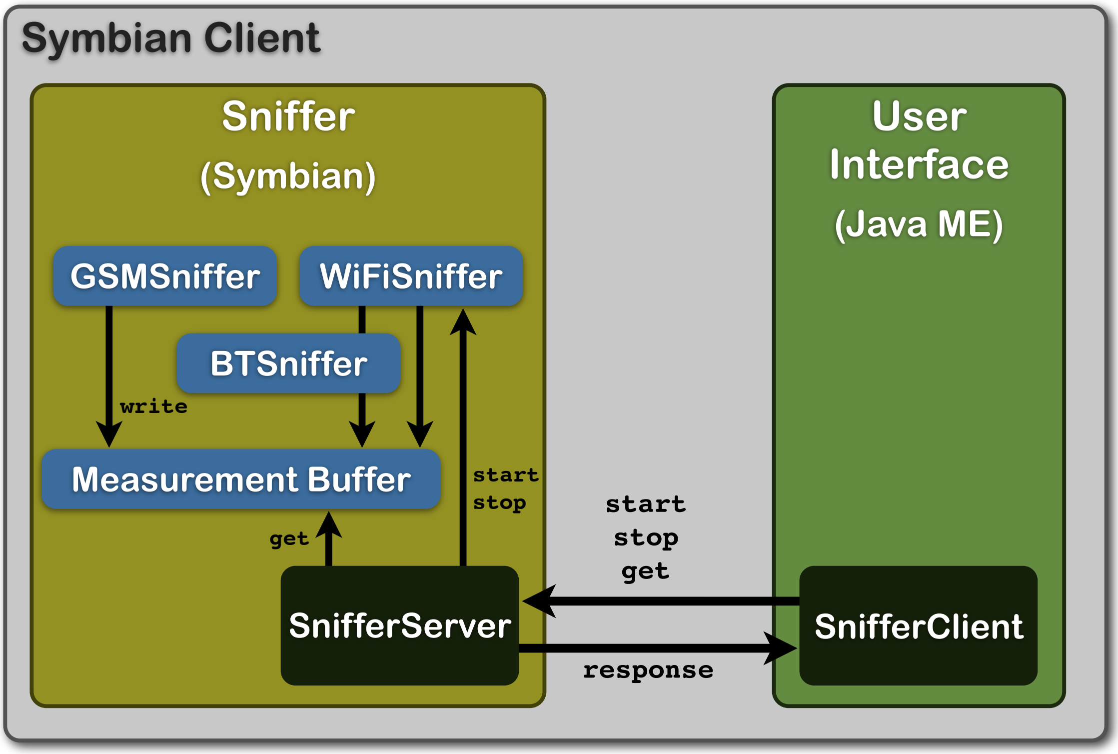


Figure 4.12: Sniffer architecture on Symbian: the Sniffer application is written in Symbian and provides the measurements to the Java ME user interface as a local server service.

one entry instead of possibly up to 15, which would obviously contribute to even better positioning accuracy. Unlike with GSM, we are able to collect this information about WiFi access points. Even when using active scanning, a WiFi measurement usually contains information about all access points in range, including the BSSID and the RSSI. Regarding Bluetooth, we have to retrieve the major and the minor device class during inquiry as we only want to consider non-portable devices. This way we can ignore mobile devices like mobile phones or portable audio devices that would distort the result otherwise. The RSS, although available on the Bluetooth host controller interface (HCI), is not exposed in the Symbian API.

Stable Detector As discussed before, we need to detect quasi-stable states in order to detect whether the device is stationary or in motion. This is necessary as Redpin only considers measurements taken while being in stable state in order to further improve accuracy. In its simplest form, a *stable state* can be detected by comparing the distance measure of at least three successive measurements as illustrated in Figure 4.13. If the distance between all measurements is lower than the threshold, we assume that the mobile device has not been moved.

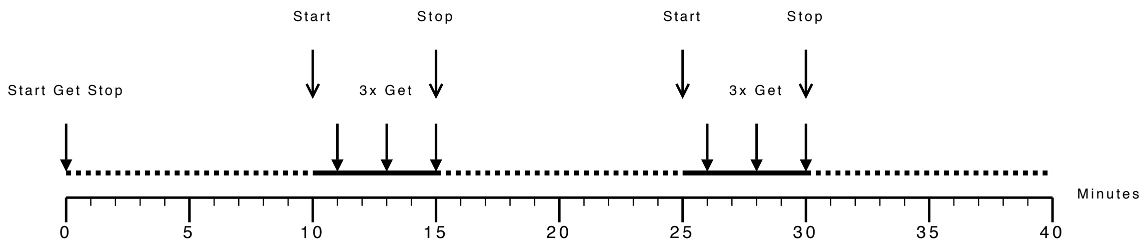


Figure 4.13: Detect stable states by comparing three consecutive measurements to a threshold.

Note that detecting stable state by comparing measurements has the advantage of working on all platforms and the results of this method are sufficient for collaborative labeling. However, when recording measurements over long time periods for interval labeling purposes, we need a more reliable method of detecting stable state. Hence, we developed a more sophisticated method, which we will discuss in Chapter 5.

iOS

Unlike with the Symbian version of Redpin, we are not able to get GSM or Bluetooth readings from iOS due to API restrictions. Even getting WiFi measurements, although technically possible, requires API calls that are marked *private* by Apple. Hence, although we are able to realize a Redpin iOS app using WiFi, the resulting app may never be published in Apples official App Store. On the upside, iOS features very rich data persistency and communication layers. Thus, our primary focus with the iOS version of Redpin was to support online as well as offline operation along with a data synchronization framework that is aware of the current connection state.

The implementation of this synchronization framework was the biggest challenge. On one hand, data must be stored locally on the iOS device.

On the other hand data must be sent to and received from the Redpin server and aligned with local data. To guarantee data integrity, the iOS app must therefore distinguish between online and offline mode. In order to detect the connection mode, we try to connect to the Redpin server using the *InternetConnectionManager*. If we can successfully connect to the server, the app goes into online mode while switching to offline mode in case of failure. Stable state detection in Redpin for iOS is basically implemented as described above (Section 4.3.4). However, as most iOS devices feature an integrated accelerometer, we are able to detect device movement more accurately by also considering acceleration force. In our first implementation of the accelerometer-based stable detector, we simply compare the current mean acceleration of all three axis to a given threshold. If the reading is bigger than an empirically determined threshold, we assume the device is being moved by the user. As mentioned above, we will discuss a more elaborate version of this algorithm in Section 5.

4.3.5 Preliminary Evaluation

In this section we present a short and preliminary evaluation of the Redpin system as the focus of this chapter is on presenting the reference implementation only. Because we will introduce improvements and enhancements to our indoor positioning system in the next chapter, a more detailed analysis and evaluation will follow in Chapters 5 and 6. The four main goals of Redpin are, as given in the beginning Section 4.3:

- **Hardware**

  Redpin must not require special hardware but work with standard, existing devices.

- **Cost**

  Redpin must be very easy to setup and maintain. Expert know-how about location fingerprinting must not be required.

- **Accuracy**

Redpin should at least provide room-level precision.

- **Signal Variations**

  Redpin must be capable of coping with both long-term and short-term radio signal variations.

The first goal, namely not to require special hardware, has clearly been achieved. By strictly using standard programming languages such as Java for the server and implementing Redpin mobile client software for the most popular smartphone platforms, Redpin may be adopted in any office or home environment without the need to purchase additional hardware. Having the user in mind in every design decision we made, Redpin is very easy to both setup and use. For example, it only requires one Java command to setup and start the Redpin server on any computer on which Java is installed. As Redpin enables collaborative labeling, the usually time-consuming and costly offline training phase can be omitted entirely. In addition, as Redpin's concept of location is very simple, anyone is able to create, change or correct location labels. This is further simplified by the graphical, map-based user interface on the mobile clients that is intuitive and familiar to anyone who has been using Google Maps or a similar application before. Because Redpin allows for multiple measurements per location, taken at different times, a first mechanism to cope with signal variation is in place. However, as we will introduce a more sophisticated method of coping with signal variations in Chapter 5, we will not evaluate the quality and performance of this mechanism in this chapter but refer to the next chapter instead.

Evaluating the accuracy of an indoor positioning system is not straightforward as it depends on the output of the system as well as on the definition of location. As Redpin uses unstructured symbolic identifiers to denote location, we can evaluate the systems performance by answering two questions. First, how good is the positioning, i.e., in how many cases is the room correctly determined? And second, how long does it take until a device can be located in every room, i.e., until the map for a building is complete? The latter question should be a good indication

Figure 4.14: Points where measurements were taken. The labels A to W indicate on-floor measurements while X, Y, and Z indicate measurements that were taken on the stairs between the floors.

of whether collaborative labeling of location fingerprints can actually replace expert training in an offline-phase.

To get answers to these questions, we installed the client software on multiple mobile phones (two Nokia N95 and one Nokia N95 8GB) and conducted several experiments in our office building. In order to investigate the accuracy, we added fingerprints of randomly chosen rooms of one floor to the radio map as illustrated in Figure 4.14. Note that some rooms in this building are smaller than 5 by 3 meters. Subsequently, we used another mobile device to determine the current location. We repeated the verification several times and over several days, during work hours as well as during the night. Overall, the system located the device in the correct room in 9 out of 10 cases. The cases where the algorithm returned the wrong identifier could be explained by our threshold settings used in the estimation method, which were set to very strict values in order for the system to work in buildings with small rooms. In this case, the estimation method would return the identifier of a room next to the one sought-after. Note that we never added additional fingerprints during the experiment to adapt to changes in the environment.

Given these results, the time it takes to get at least one fingerprint for every room depends only on how active users are in contributing to the system and on their mobility. A very short survey showed that when only 10 (out of 50 people working on this floor) contribute to the system, the map is complete after just one day.

4.4 Conclusion

With the main goal to make indoor positioning systems easier to setup, use, and maintain while saving costs in comparison to existing systems, we presented Redpin, our reference implementation of a location fingerprinting system using collaborative labeling. The system relies on the users to contribute measurements to the radio map as opposed to existing indoor techniques that rely on a designated administrator to collect characteristic radio signal information. Using collaborative labeling, every user can create, modify and, most importantly, use location information that was created by other users. We have shown that harnessing user collaboration is a concept that has been used with great success in crowdsourcing or folksonomy based systems. Moreover, by analyzing correlations folksonomies can even be used to to extract simple tag vocabularies. Hence, if applied to the problem of labeling places and locations, our approach to indoor positioning can help to solve to problem of finding the "correct" label for a location.

However, in order for a collaborative system to be successful, the barriers get users involved must be low. Only when as many users as possible are capable and willing to contribute to the system, the effect of "wisdom of the crowds" can arise. Thus, it is not only necessary to give users access to the system by supporting different devices and platforms, but also the knowledge of how to participate. These requirements could be fulfilled with Redpin. As discussed in Section 4.3.5, Redpin supports existing hardware while being very intuitive to use. As a result, Redpin succeeds in providing room-level indoor positioning while being cost effective.

In our first implementation, we did not actually capture the concept of a user, i.e., every mobile device that contributes to the system uses the same radio map. This allows to easily share knowledge about locations and enables a quick mapping of a building. On the other hand, this aspect entails security and privacy implications which are not yet addressed.

Lastly, we are happy to report that the resulting Redpin source code was released under an open-source license. The resulting project can be found at http://www.redpin.org. To this day, Redpin was downloaded over 1000 times and has an active user community.

Prediction is very difficult, especially if it's about the future.

– Niels Bohr

5 Interval Labeling

While the Redpin system discussed in the previous chapter proved to be a good solution to the problem of training the radio map and coping with long-term signal variations, we have shown that short-term signal variations, which may occur over the course of minutes and hours, are still an issue. Given the characteristics of location fingerprinting systems in general, we know that the estimation method's accuracy and performance is better the more measurements the radio map contains. Ideally, measurements are taken at different times of the day and at different days of a week. The signal traces we have seen during our WiFi signal study (see Chapter 3) clearly showed that the best way to reduce the error caused by short-term signal variance is to average a large number of measurements taken during a short time. To cope with short-term signal variations, the radio map must contain different measurements that have been taken in short succession. Thus, the obvious solution to this problem is to record measurements over an interval of many minutes instead

Part of this chapter is based on joint work with Kurt Partridge, Maurice Chu, Marc Langheinrich [19].

of just one discrete instant. We believe that by extending user-provided labels from an instant to an interval, i.e., a period of time over which the device is stationary, can greatly improve positioning accuracy.

Parts of this chapter, in particular Section 5.3, are based on joint work with Kurt Partridge, Maurice Chu, Marc Langheinrich. While I was the main researcher on this topic, Kurt, Maurice and Marc supported my analysis and initial investigation into interval labeling. Together we published the results in our paper entitled "Improving Location Fingerprinting through Motion Detection and Asynchronous Interval Labeling", which was published in proceedings of the Fourth International Symposium on Location- and Context-Awareness (LoCA) held in Tokyo, Japan, in May 2009 [19]. I was the main author of this paper and wrote the main parts, this chapter is based on, myself. I worked on this paper while I was visiting researcher at PARC. Maurice and Kurt advised me in my research and, together with Marc, helped me to improve the quality of the paper by giving it more structure and polishing my english.

In this chapter, we will present our asynchronous interval labeling method. We start by introducing the technique and main building blocks. In Section 5.2 we will elaborate on the problem of detecting stationary state. In doing so, we will also compare our solution to current state-of-the art methods used in the field. As with the approach of collaborative labeling presented in the previous chapter, we will then present our reference implementation that was built as proof of concept in order to verify the feasibility of interval labeling. A discussion and evaluation of our approach in Section 5.3.3 concludes this chapter.

5.1 Building Principles

Collecting measurements may be tedious and is not something an end-user is very eager to do, especially if this needs to be done several times a day. Two challenges are: How can a system get users to contribute many labeled measurements to the system even over the course of one day *without interrupting their work routine*? And how can a system continue

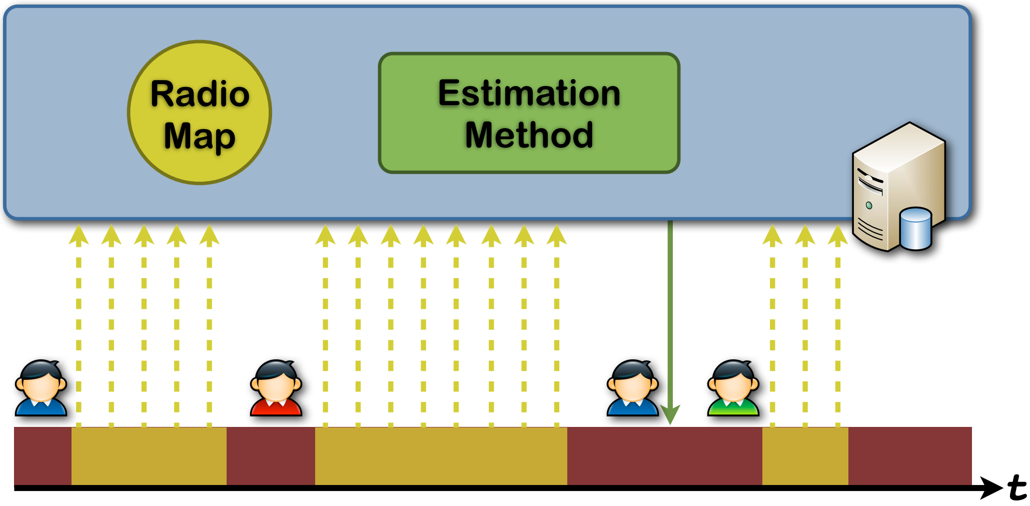


Figure 5.1: In contrast to the collaborative labeling method (see Figure 4.2), interval labeling allows to record many measurements in short succession without the need of user input.

to update the radio map over days and weeks, again *unobtrusively*?

Our method of interval labeling addresses these two challenges. Labels provided by end users are applied not only to the immediate signal strength measurement, but to all measurements taken during the interval while the device was stationary at the same place. Figure 5.2 gives an example of the process of interval labeling. Using data from the accelerometer, we partition time into alternating periods of "moving" and "stationary" as indicated in the second row of the figure. (The implementation of the motion detection process is described in Section 5.2.1.) Whenever the system is stationary, it continuously adds measurements to the interval. When it detects movement, it stops taking measurements until the device rests again, at which time a new interval begins.

In addition to increasing the number of WiFi measurements that can be associated with a location label, intervals can improve the user experience of labeling. Because intervals are known to be periods of immobility, they can be more easily labeled asynchronously. Users are more likely to remember their location during the entire interval (knowing its starting time and duration) than they are likely to remember their location at a specific instant. In consequence, we enable the user to label a location at any time. Our system does not have to prompt the user as soon as a new

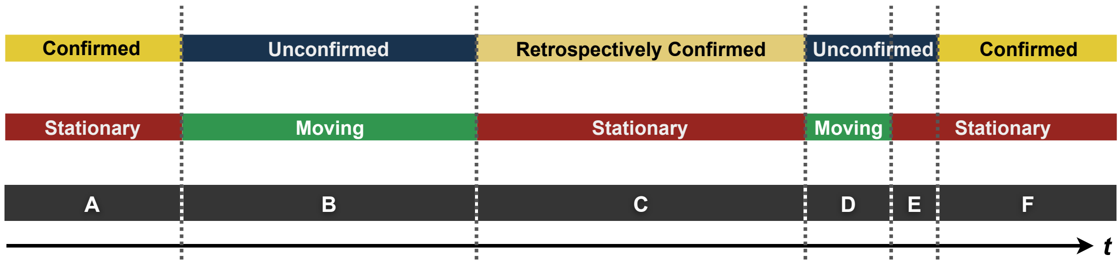


Figure 5.2: Interval labeling allows the user to update the radio map with all data taken while the device is stationary. Because intervals provide more cues to users (starting time, ending time, and duration of interval), users are more likely to remember where they were during an interval than at an instant.

location is entered but supports *asynchronous* labeling. This gives the system the freedom to postpone labeling until a more convenient time such as the start of the next stationary period, or when users return to their desk. This can further help the system reduce the obtrusiveness of any explicit location prompts.

Asynchronous labeling can also make sure that only *important* labels are solicited, i.e., of places that the user stays in for a long time or visits repeatedly. If the user has been at an unknown place for only a few minutes, the system can decide not to prompt the user altogether. Once the user enters a label, however, the system can label a particular signal fingerprint retroactively, thus incorporating all measurements taken at that particular location into the radio map.

For the purpose of putting asynchronous labeling to the test, we use a quite simple heuristic for deciding when to prompt the user for an asynchronous label: the system detects the change from battery power to AC power and interprets this as the user returning to office or home, which it believes to imply a task closure after a previous, battery-powered outing (e.g., a meeting). Consequently, we then prompt the user to confirm the current as well as the previous label. If the system is unsure about the current position, it might also ask to confirm/enter this information as well. For the choice of which previous interval to label or confirm, we again rely on simple heuristics. Our system prefers longer intervals over shorter ones, and more recent ones over older ones. However, we noticed in our initial experiments (see section 5.3.3) that users were quite comfortable identifying locations even if prompted several hours later, as long as they did spend a sufficiently long time there.

5.2 Detecting Stationary State

Obviously, recording measurements over an interval of time is only possible if the system is certain the recording device is still at the same location. Hence, we need an additional system component that is able to detect whether a device, and thus a user, is actually stationary or moving. A user's physical activity is considered a major aspect of his context. Thus, many systems have been developed to infer and classify human activities such as standing, walking, or running. Most of these systems make use of several accelerometers that are distributed over a user's body. Bao and Intille for example describe a system [8] that uses 5 accelerometers and allows to recognize everyday activites such as folding laundry or brushing teeth with an accuracy rate of 84%. A system proposed by Lester et al. [107] showed that comparable accuracy rates can be achieved even when only using a single accelerometer. While being able to infer very complex activities, Kern et al. showed in [85] that using an accelerometer to distinguish "moving" (be this walking, running, or jumping) from "still" (be this stand or sit) is straightforward. Some positioning systems also perform motion detection. For example, Krumm and Horovitz's LOCADIO [98] uses WiFi signal strength to both localize a device and infer whether it is moving. However, due to the natural fluctuation of signal strength readings even when standing still, this motion detection's error rate is 12.6%, which results in a high number of false state transitions (e.g., from "stationary" to "moving") during experimental use (24 reported when only 14 happened).

King and Kjærgaard [86] also use WiFi to detect device movement, reporting results similar to Krumm and Horovitz's on a wider variety of hardware. They use motion data to minimize the impact of location scanning on concurrent communications: If the device is stationary, the system does not need to re-compute its position (which might interfere with regular communications as both activities share the same WiFi card). In contrast, we use motion information not only for positioning, but also to aid the training: If the device is stationary, the system can collect stable WiFi measurements. In addition, instead of using WiFi to infer both location and movement, we detect the latter using accelerometer data.

5.2.1 Motion Detector

The motion detector we propose is a discrete classifier that reads the accelerometer to determine whether the device is being moved or whether it is stationary. Classification needs to be somewhat forgiving, so minor movements and vibrations caused by readjusting the screen or resting the computer on one's lap are still classified as "stationary". Only significant motion such as walking or running should be classified as "moving."

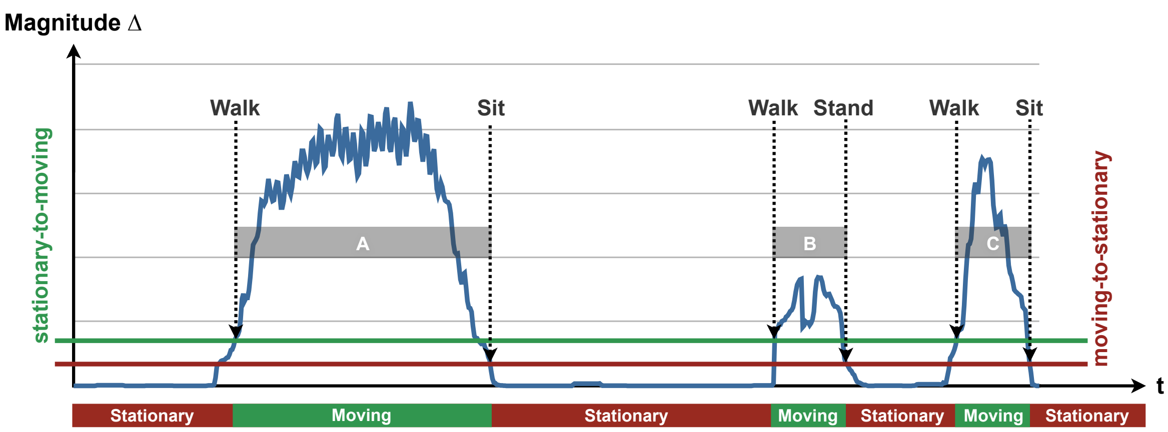


Figure 5.3: Example data from the motion detector. As soon as the magnitude delta exceeds the stationary-to-moving threshold, the device is considered to be moving. This holds as long as the magnitude delta does not fall below the moving-to-stationary threshold.

To classify the device's motion state, the motion detector samples all three accelerometer axes at 5 Hz. It then calculates the acceleration magnitude and subtracts it from the previously sampled magnitude. To prevent misclassification of small movements as "moving," the signal is smoothed into a moving average of the last 20 values. Figure 5.3 shows that this method yields a sharp amplitude increase in the magnitude delta whenever the user is walking. The classifier includes hysteresis with different threshold values when switching between the moving and stationary states. The threshold values were established through a series of informal experiments. Figure 5.3 shows the motion magnitude trace of a user going from his office to a colleague's office (A) and back (B and C), with two stationary phases in between: a longer discussion at the colleague's office and a brief chat in the hallway. The sequence bar at the bottom of the figure shows the motion detector's output. Due to the use of a moving average, the system imposes a small delay of 2-4 seconds before values fall below the threshold for the stationary state.

5.3 The PILS System

As with Redpin for collaborative labeling we built a reference implementation to showcase and test the concept of asynchronous interval labeling. Building on lessons learned from Redpin but using different hardware, we built PILS, an adaPtive Indoor Localization System.

Our main goal and reason of building PILS was to show the feasibility of interval labeling in location fingerprinting systems. In particular with focus on the concept of asynchronous labeling, which requires the system to prompt the user for an asynchronous label, we decided to implement PILS for laptop devices. This way, we were able to determine when to prompt the user simply by detecting the change from battery power to AC power, a simple but accurate idea that would not have worked with smartphones. While reusing parts of the source code of Redpin for iPhone, we decided to re-design and re-implement some components for PILS in order to adhere to the new concepts. The most defining difference between Redpin and PILS lies in the system architecture. While Redpin is a pure terminal-assisted system using a central radio map and estimation method, PILS is designed as a hybrid solution. In its basic setup, PILS uses a local radio map, i.e. location fingerprints are not shared between devices. However, PILS may also use any Redpin server to store and exchange location fingerprints. This ability was devised to integrate low-power devices that require a central server to execute the computationally heavy estimation method.

Figure 5.4 gives an overview of the three main system components of PILS: a *scanner* to listen for announce beacons, a *locator* to compare current measurements with the assembled radio map from a fingerprint database, and a *motion detector* to inform the locator about interval boundaries (i.e., changes between the moving state and stationary state).

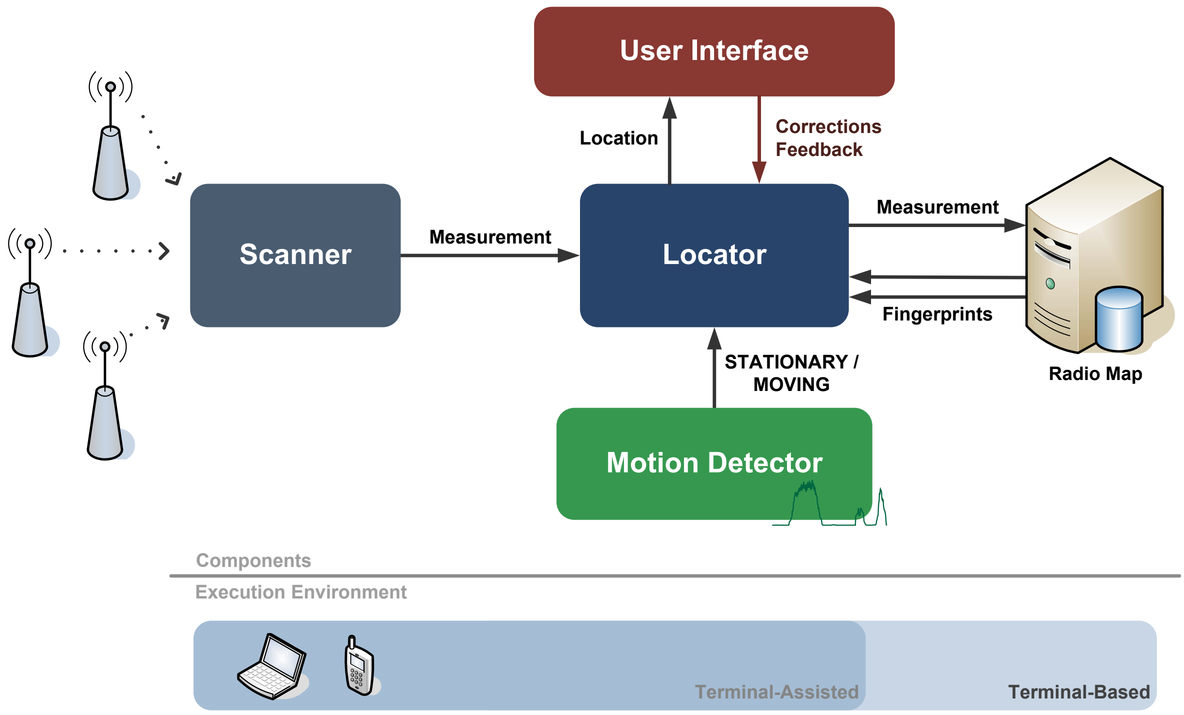


Figure 5.4: Our terminal-based system has four components. The signals observed by the scanner are sent to the locator, which estimates the location using the fingerprints stored in the radio map. The motion detector informs the locator whether the device is stationary or moving, and the user interface collects the labels.

In the following we give a description of the platforms and the hardware we used to implement PILS and explain the estimation method used for the locator in more detail.

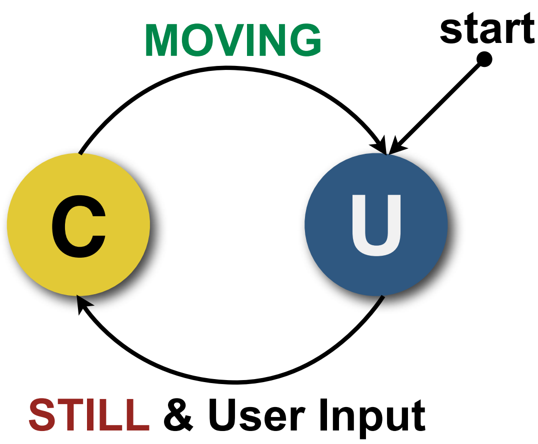


Figure 5.5: Asynchronous interval labeling is based on a motion detector triggered state machine that captures whether the user confirmed the location (C). The system may continue collecting labeled measurements as long as the system has confirmed location and the device is stationary.

5.3.1 Hardware and Setup

PILS requires a WiFi communications module and an accelerometer in the terminal—two components that are often available in today's laptops and high-end mobile phones. We implemented our initial version of PILS on Mac OS X 10.5 using MacBook Pros (revision B and C), making sure PILS would be easily portable to the iPhone platform due to their large architectural overlap. The 15-inch machines that we used have a WiFi network card from Atheros or Broadcom. In addition, these laptops possess an accelerometer, which is used for their motion-based hardware and data-protection system. This system detects sudden acceleration, for example when dropping the computer, and prepares the hard disk for impact by disengaging its heads. In the 15-inch machines we used, the accelerometer is a Kionix KXM52-1050, a three-axis accelerometer chip, with a dynamic range of $+/-$ 2g and a bandwidth up to 1.5kHz.

From the WiFi measurement data described in Chapter 3, we estimated that at every location within the building at least five access points (AP) would be visible. Given the characteristics of the 2.4 GHz radio signal used in IEEE 802.11, this is usually the case in an office building where a wireless LAN has been installed to be used for business crucial purposes. However, for security reasons the WiFi network might be

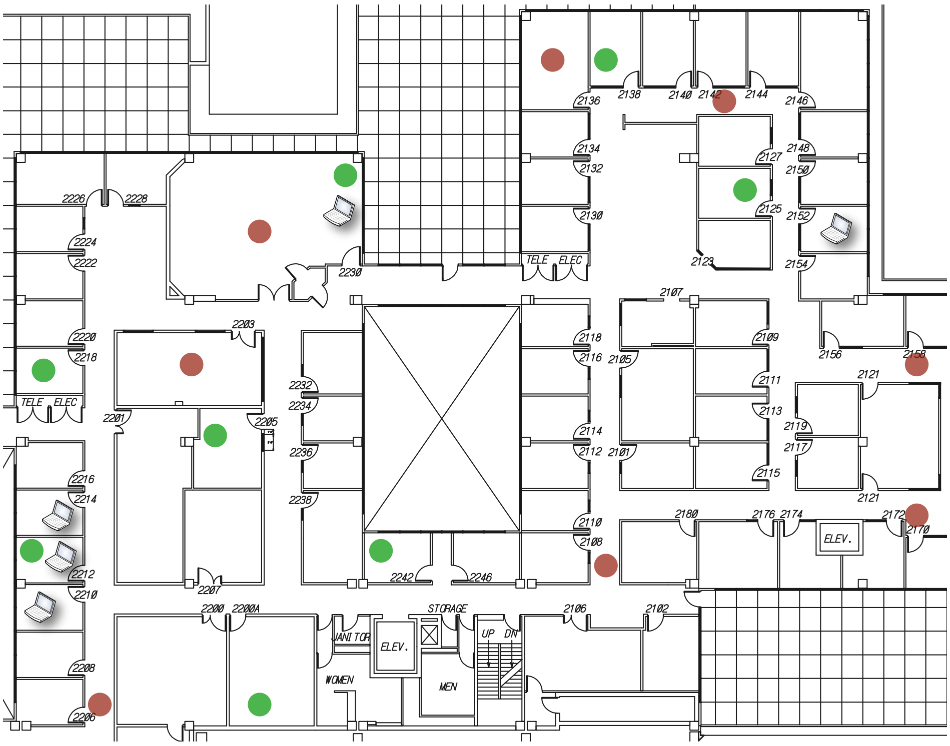


Figure 5.6: Overview of the office environment at PARC. The red circles indicate the APs of the public WiFi network. The green circles indicate additionally APs we added later on.

configured such as that access points do not send out announce beacon frames, i.e., the SSID of the network becomes invisible. As this was the case in our office building, we extensively examined possible solutions like passive scanning or actively opening data connections. Eventually, we found that the simplest and cheapest solution to this problem is to just add more open access points. A simple WiFi access point today costs no more than \$30.-. As it does not even have to be connected to the corporate network, but only needs to send beacon frames, it is also not a security problem.

In our office environment, as illustrated in Figure 5.6, we found eight public access points of which we could get beacon frames and thus the SSID and the received signal strength (RSS). To guarantee that at least 5 access points were visible in every location within the testbed, we bought and installed 8 additional access points, thus reaching a total of 16 access

Note that the exact effect depends on the network card. While some cards allow to at least get SSID readings, network cards from other manufacturers completely hide such networks.

points to cover 70 rooms with a combined area of about 1000m^2. This comes out to a density of 0.23 access points per room, or 1 access point per 62.5m^2 of office area.

5.3.2 Probabilistic Estimation Method

As we described in Section 2.4.2 of this thesis, probabilistic positioning methods make use of a large number of measurements per fingerprint. As probabilistic estimation algorithms apply statistical methods on fingerprint data in the radio map, the performance and thus the accuracy can obviously be improved by adding more measurements. Given the fact that the radio map contains much more measurements when using interval labeling, we expected a probabilistic estimation method to provide better accuracy than k-nearest neighbor. Hence, we did not use the Redpin estimation method discussed in Chapter 4. Our approach for PILS to location fingerprinting is to learn a probabilistic model of the likely readings of received signal strength (RSS) of WiFi beacons for each location we are interested in. With these learned models, we estimate the device's location by choosing the model that gives the maximum likelihood.

Our probabilistic model is similar to the approach taken by Chai and Yang [34], except that we use normal distributions for RSSI rather than quantizing RSSI values and using histograms. As long as the RSSI values are not multi-modal, such a unimodal approach still offers good performance while being computationally much simpler. By keeping only the mean and variance, updates are very fast and do not use much memory. In addition, the larger number of free parameters in a histogram approach is more susceptible to over-fitting when there is not much data.

Each received signal strength reading is stored as a pair consisting of the access point's BSSID and the measured indicator of its signal strength, i.e., $b_t = (BSSID_t, RSSI_t)$, with $RSSI_t$ being the received signal strength from the WiFi access point with unique identifier $BSSID_t$ at time t.

For each location l we learn a model of the readings received by a device in location l. For a set of n readings $\{b_1, ..., b_n\}$ in location l, we adopt the following model for the likelihood of the set of readings:

$$P_l(b_1, ..., b_n) = \prod_{i=1}^{n} p_l(BSSID_i) \cdot N(RSSI_i; \mu_l(BSSID_i), \sigma_l^2(BSSID_i)) \tag{5.1}$$

where N is the normal distribution and $p_l(BSSID)$ is the probability that the reading in location l comes from WiFi access point BSSID. We model each reading to be independently generated from a normal distribution with mean $\mu_l(BSSID_i)$ and variance $\sigma_l^2(BSSID_i)$, which can be different for each access point.

Given a set of n readings $\{b_1, ..., b_n\}$ in location l, the model parameters which maximize the likelihood of the readings are given by:

$$p_l(bssid) = \frac{R_{bssid}}{n}$$

$$\mu_l(bssid) = \frac{1}{R_{bssid}} \sum_{i:BSSID_i=bssid} RSSI_i$$

$$\sigma_l^2(bssid) = \frac{1}{R_{bssid} - 1} \sum_{i:BSSID_i=bssid} (RSSI_i - \mu_l(bssid))^2$$

where $R_{bssid} = |\{b_i | BSSID_i = bssid\}|$ is the number of readings that came from WiFi access point *bssid*. Note that a location l will not get readings from all access points. For those access points which were not part of the readings for learning the model, we set $p_l(bssid)$ to a very small value, e.g., 10^{-15}. The parameters $\mu_l(bssid)$ and $\sigma_l^2(bssid)$ can be chosen in any way as long as the product of p_l and the normal distribution is small. To estimate the most likely location $\hat{l}$ from a set of readings $\{b_1, ..., b_n\}$, we can compute Eq. 5.1 and find the maximum likelihood location as follows:

$$\hat{l} = \arg_l \max P_l(b_1, ..., b_n) \ .$$

We compute $\log P_l(b_1, ..., b_n)$ as it is numerically stable and the monotonic property of the logarithm guarantees the same answer for $\hat{l}$.

5.3.3 Evaluation

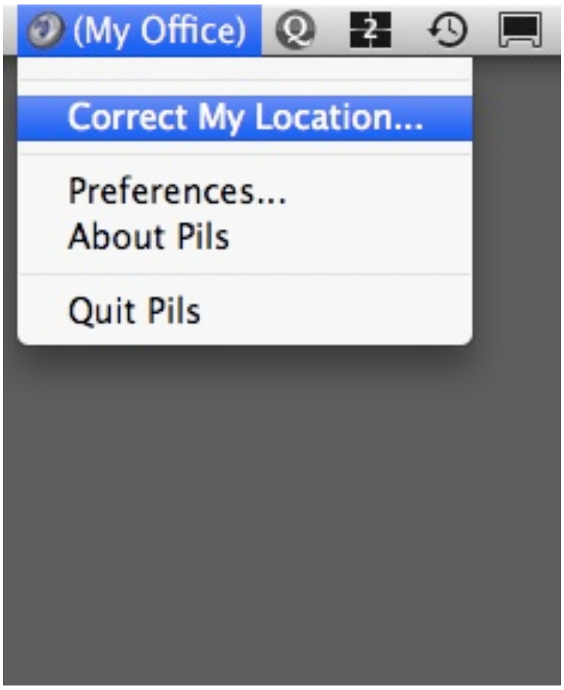
To understand whether interval labeling would work well in practice, we conducted a user study. The study examined whether users would voluntarily correct incorrect location predictions, what the characteristics of the labeled intervals were, and whether labeling increased the system's confidence in the user's location.

Experimental Setup

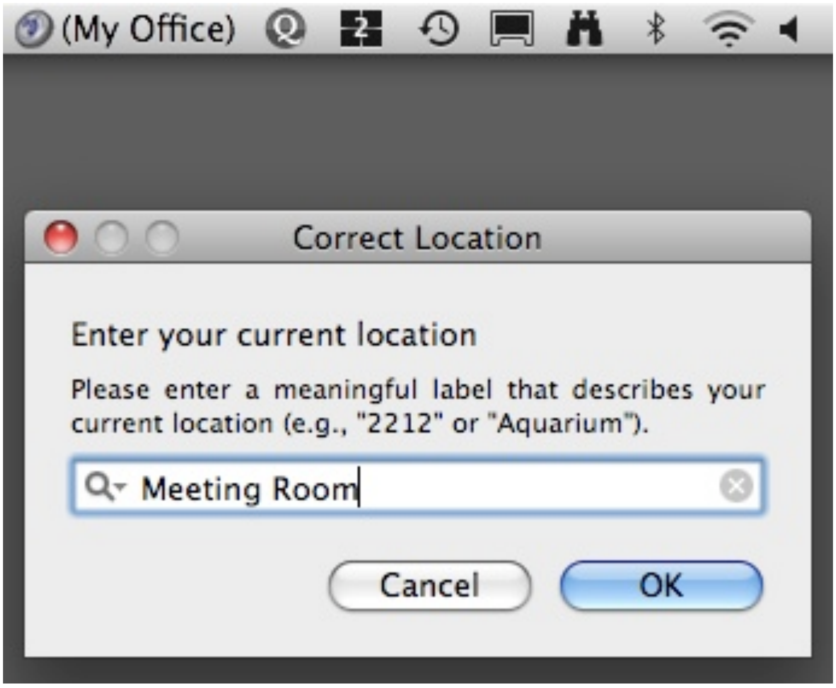
We recruited 14 participants who installed a custom application on their MacBooks. The software placed an extra menu in the right side of the menu bar, as shown in Figure 5.7. Users were instructed to correct the system if they saw that it incorrectly guessed the location. This was also the mechanism for adding new labels to the system. The users gained no benefit from the application other than the satisfaction of making the correction. The study ran for five weeks, which included the winter holiday period.

To remind users about the study and to provide additional feedback to the user about the system's inferences, the user could optionally enable a voice announcement of "moving" and "stationary" when the device transitioned between moving and stationary states. Music could also optionally be played while the device was in the moving state. However, as the laptops went to sleep when their lids were closed, the music typically did not continue for the entire moving duration.

Location inferences were made on the users' laptops, however all WiFi measurements and labeled data were uploaded to a server for later analysis.



(a) The user corrects an erroneous inference through the "Correct My Location..." menu option.



(b) The user can enter any label for the current location by a simple dialog.

Figure 5.7: User interface for collecting label corrections: The system's prediction of the room is placed in the menu bar to provide ambient awareness.

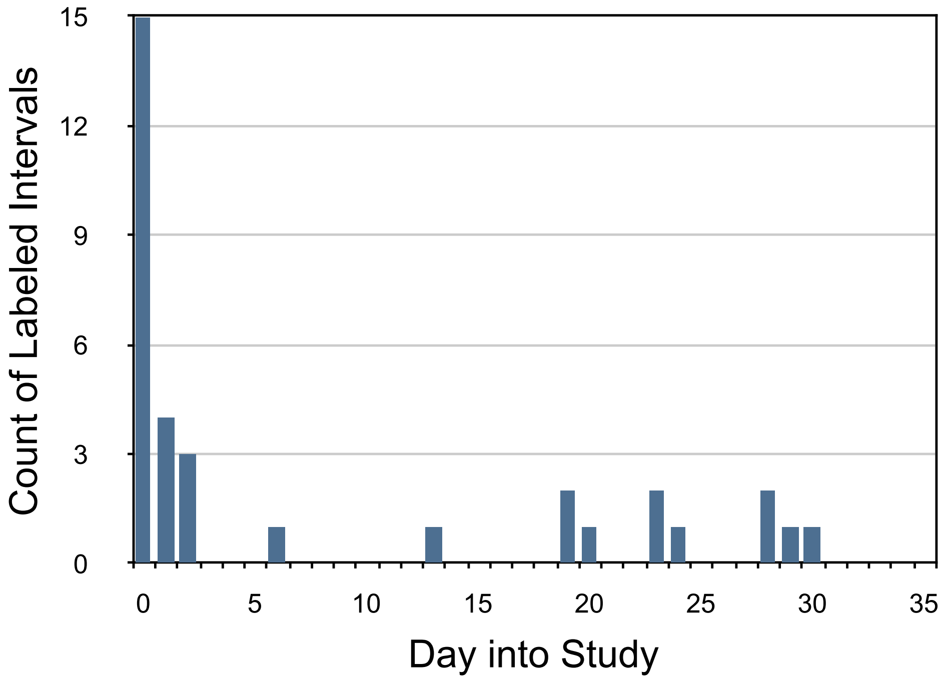
Results

WiFi Scans and Label Frequency When running, the program conducted an active WiFi scan once every five seconds. A total of 322,089 WiFi measurements were taken. Each scan contained on average 6.6 beacons, with a standard deviation of 4.4.

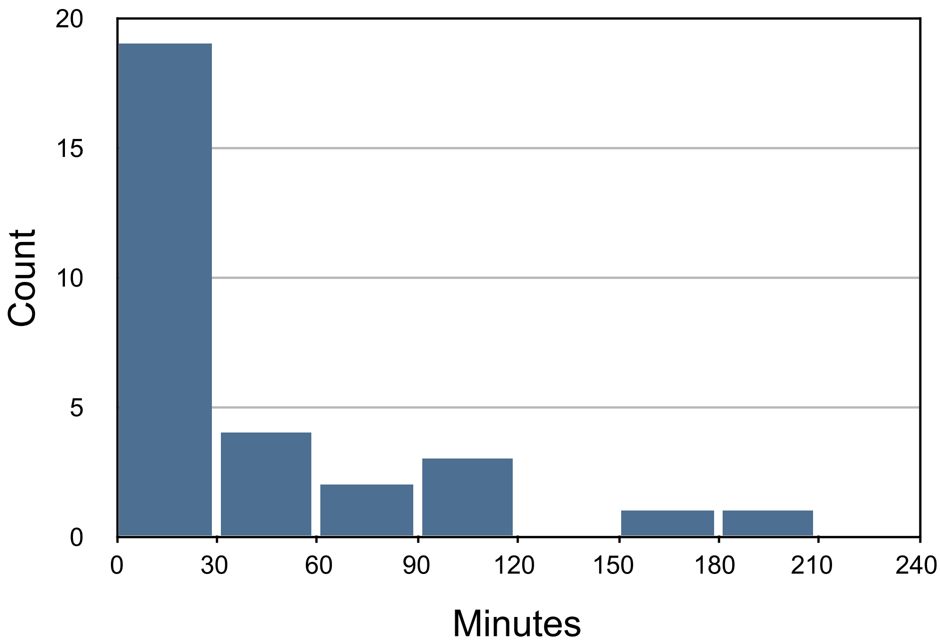
Users labeled 31 intervals, with a surge on the first day, and declining frequency afterward (see Figure 5.8(a)). However, users continued to correct the system at a roughly constant rate until the end of the experiment, despite not receiving any reminders about the study other than the ambient awareness in the menu bar. Furthermore, continued labeling was not concentrated in a couple individuals—the contributions after the tenth day came from five different participants. All these results suggest that providing corrections is a low-overhead activity that can be sustained for at least a month.

Interval Characteristics Figure 5.8(b) shows a histogram of interval durations. Most intervals were only a few minutes long. Of those under a half hour, five lasted less than a minute, and sixteen less than ten minutes.

Generally, users provided labels at the beginning of an interval. 28 intervals were labeled within the first two minutes. Of the remaining three intervals, one was labeled at the end of a half-hour interval, and two others were labeled in the middle of multi-hour intervals. From these observations we conclude that since users chose to enter corrections when arriving at a new place, this is the best opportunity for a more proactive system to query users for location data.



(a) Number of new labels added per day. Around a third of the labels were added on the first day. The decline and later uptake in labeling likely resulted from the holiday schedule.



(b) Histogram of labeled interval durations. Most intervals lasted less than a half hour. Note that there is an outlier not shown on the graph at 21.3 hours.

Figure 5.8: Label Frequency and Interval Durations

Benefits of Labeling Intervals To understand how much the system benefitted from interval labeling, we examined the recorded data more closely. A sample of 1,000 WiFi measurements was drawn. Each scan was classified according to its most likely location, given the labels that the system knew about at the time the scan was taken. Two classifiers were compared, one that learned from all WiFi scans in previously labeled intervals, and another that learned only from the WiFi scan at the instant a label was assigned.

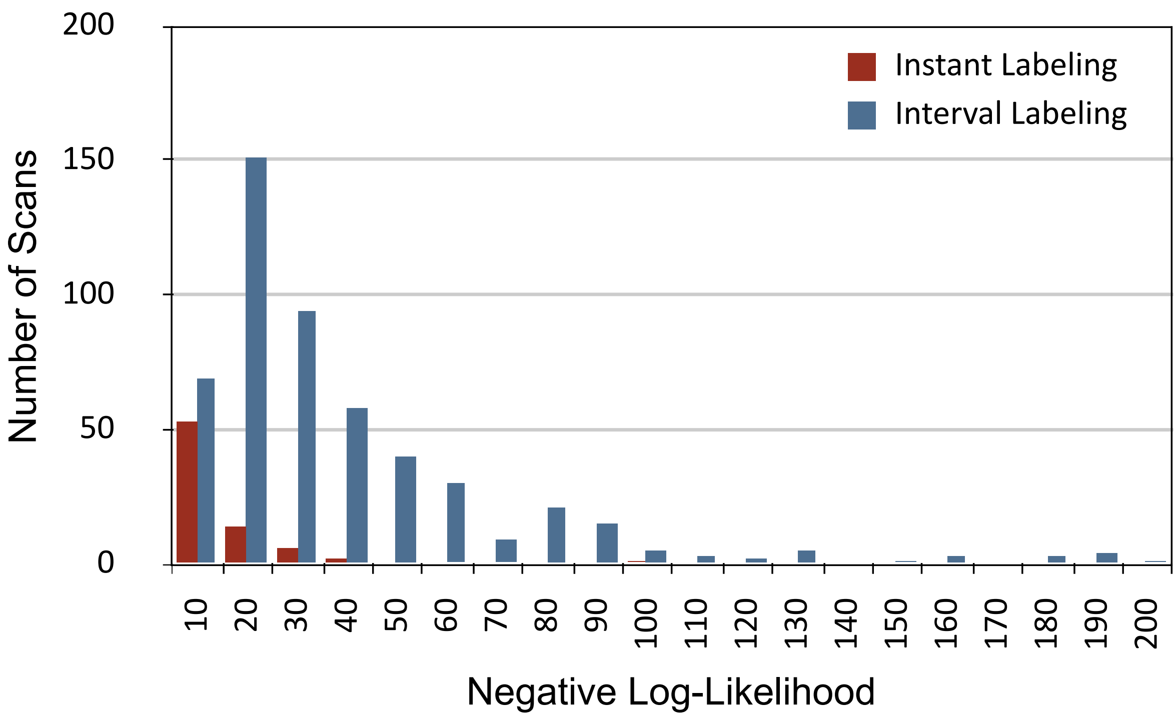


Figure 5.9: Distribution of the log-likelihoods of 1,000 random WiFi scans, excluding those with zero likelihood (which include 484 for Interval Labeling, and 924 for Instant Labeling). The proportionally higher likelihood scores indicates that WiFi scans are much more likely to find labels when using Interval Labeling than when using Instant Labeling.

Figure 5.9 compares the distribution of maximum log-likelihoods for the class returned by each classifiers. The graph does not include the scans whose WiFi likelihood scores were zero, as explained in the caption. For the over 92% of scans in the instant labeling condition, the likelihood value gives no information about which label is best. Likelihood values can be computed, however, for over half of the scans in the interval labeling condition. Furthermore, even when a likelihood value is computed, the values are, in general, relatively higher in the interval labeling condition, which indicates greater certainty of the answers.

Survey

Following the user study, we surveyed participants to better understand the user experience. We felt that it was important to get users' perspective on both the accuracy of the system as well as the overhead involved in collecting the labels. At the end of the five week study period, we sent out a questionnaire to all 14 participants asking to give a qualitative assessment by answering the 6 questions as listed in Figure 5.10. Eleven of the participants responded to the survey.

Questions	Answer Choices
1. The labeling prompts in PILS were intrusive.	scale from 1-7; 1=strongly disagree, 7=strongly agree
2. I was prompted very often by PILS	scale from 1-7; 1=strongly disagree, 7=strongly agree
3. The prompts after connecting to AC power did not interrupt my workflow	scale from 1-7; 1=strongly disagree, 7=strongly agree
4. The accuracy got better over time	scale from 1-7; 1=strongly disagree, 7=strongly agree
5. PILS often showed a wrong or missing label	Number
6. How many times per day do you reconnect your laptop to AC power (on average)?	Free Text

Figure 5.10: Questionnaire sent out to all participants at the end of the user study.

Participants' perceptions about the system accuracy were mixed. On a Likert scale from 1–7, where 1 stands for "strongly disagree," responses to "PILS often showed a wrong or missing label" had a mean of 3.0 and standard deviation of 1.9. But in response to "the accuracy got better over time," responses averaged 4.3 with a standard deviation of 0.8.

In free responses, participants offered several improvement suggestions, such as reducing the latency to make an estimate and improving the autocompletion of labels. Two participants appreciated the music that played when the laptop was moving. One found it to be not only a useful form of feedback about the system's operation, but also an interesting prompt for social engagement. The other wanted to be able to choose the music from their iTunes library. One participant particularly appreciated the audio feedback that indicated when the device was moving. He found it to be not only a useful form of feedback about the system's operation, but also an interesting prompt for social engagement.

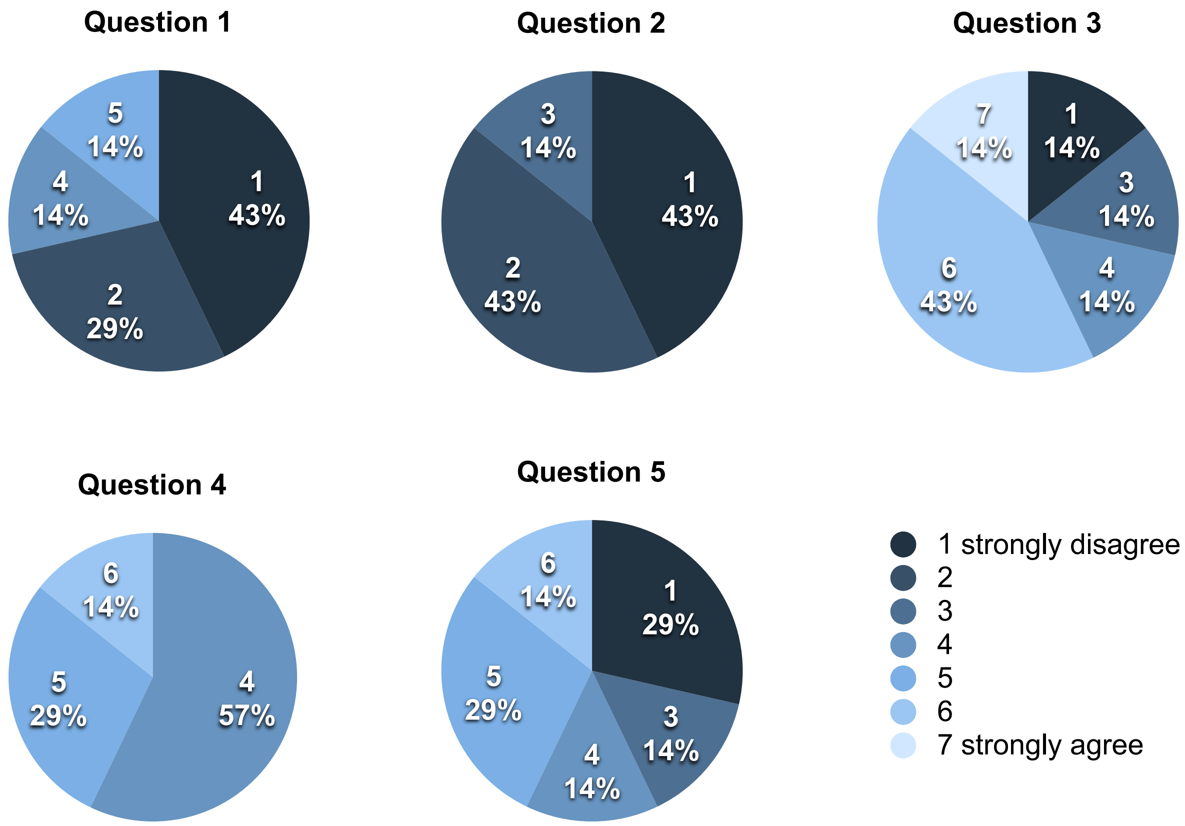


Figure 5.11: Results from the participants survey.

5.4 Optimizing Location Estimation

From our experiences with PILS we learned that interval labeling, in particular asynchronous interval labeling, will greatly increase the number of measurements in the radio map. However, the evaluation has shown that the used probabilistic estimation method did not perform to the level we expected. While the accuracy of PILS was significantly higher than with Redpin, our survey revealed that in some cases and for some users the accuracy actually decreased over time (see previous section).

Moreover, having thousands of measurements per fingerprint in the radio map posed a new problem: the query time, i.e. the time required for a location lookup was growing. Eventually, our estimation method took several seconds for a single lookup.

To analyze the effect of growing numbers of measurements in the radio map and to subsequently optimize the estimation method, we integrated the radio map and estimation method implementation of Redpin and PILS. In addition, we included a new, kernel-based estimation method based using the principle of a "support vector machine" (SVM) as described in [121]. To implement a simple kernel-based estimating method, we used LIBSVM. This way, we were able to compare three different estimation methods: the k-nearest neighbor method used in Redpin (see Section 4.3.3), the bayesian method used in PILS (see Section 5.3.2) and an SVM based method as described above.

This section is based on joint work with Luba Rogoleva that was first presented in her master thesis on "Crowdsourcing Location Information to Improve Indoor Localization" [138]. Luba collected data and implemented combined estimation method algorithms under my supervision. Together we analyzed her data and created a new toolset to evaluate estimation methods using very large datasets. With Luba's consent, in this section we present figures that were created using data and tools first used for her thesis.

5.4.1 Method Comparison

Having these three estimation methods at hand, we wanted to further analyze the effect of interval labeling and the improvements over instant labeling. To evaluate the performance, we re-used the data set of our user-driven WiFi study (see Section 3.2). Thereby, we made use of the most popular approach for estimating the accuracy of a given classifier, namely running it through cross validation [139]. This technique of performance estimation involves repeatedly partitioning a given dataset into non-overlapping *training set* and *testing set*. The training set is being

See http://www.csie.ntu.edu.tw/∼cjlin/libsvm/.

used to induce the classifier, which is then validated using the unseen instances in the testing set. To simulate instant labeling, we we randomly selected measurements of non-consecutive readings.

First, we compared the accuracy of interval labeling using datasets of different size. The base datasets were created by choosing a single interval of measurements per location, varying the length of the selected interval between 5 and 100 minutes. In this scenario, an interval of 5 minutes contains 10 measurements as the mobile devices used to collect the data scanned the WiFi environment every 30 seconds.

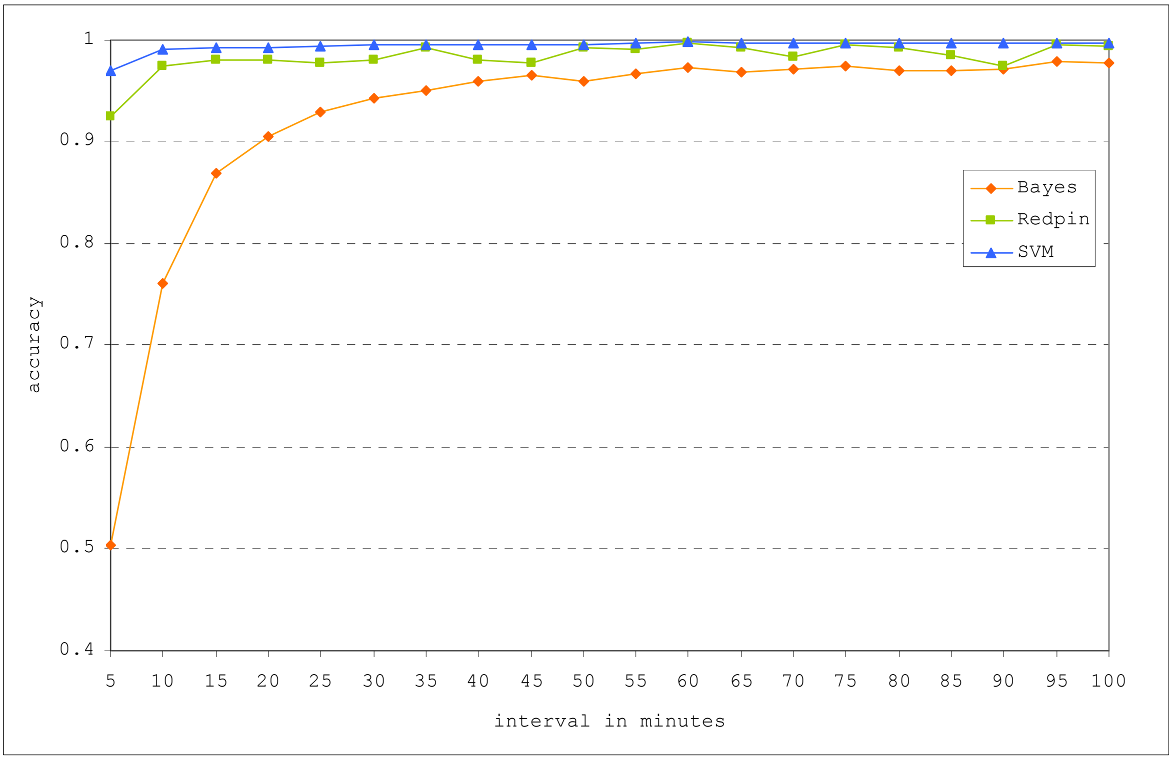


Figure 5.12: Accuracy comparison of different estimation methods using interval labeling technique.

As we can see in Figure 5.12, the bayesian method yields the lowest accuracy, in particular below an interval length of 45 minutes. Both, the SVM and the Redpin estimation methods improve accuracy up to an interval length of 15 minutes and yield consistently good results thereafter. SVM in particular shows very good results, reaching almost 100% accuracy with intervals of 25 minutes length or more.

Second, we compared the accuracy of instant and interval labeling. We know from previous evaluations (see Section 5.3.3) that interval labeling will outperform instant labeling. With this experiment, we wanted to analyze this effect in more detail. For this experiment, we selected N random intervals of different length and compared the results with instant labeling using N random measurements.

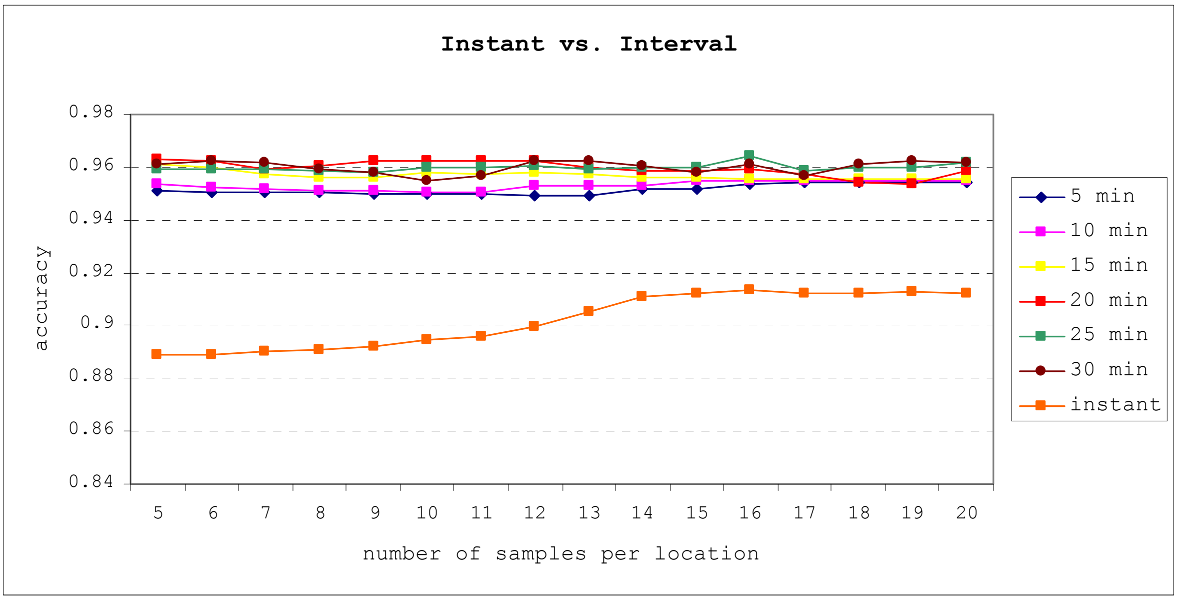


Figure 5.13: Accuracy comparison of instant labeling and interval labeling using the Redpin estimation method.

Although the results using instant labeling are satisfactory (even a small number of measurements offers accuracy of about 0.89), the results presented in Figure 5.13 clearly show the advantage of using interval labeling. Using even short intervals of 5 minutes yields accuracy improvements of 6%. While using longer intervals allows for improved accuracy in general, the improvements of using intervals longer than 15 minutes are negligible. This confirms our observation that using overlong intervals does not improve accuracy per se. In some cases, using overlong intervals might even diminish accuracy, as we can see in Figure 5.13 by comparing the results of using intervals of 20 and 30 minutes.

Finally, we wanted to study the effect of using different estimation methods towards time required for query and training. While probabilistic algorithms like Redpin's k-nearest neighbor or bayesian methods don't require training, i.e., the insertion of a measurement into the data structure used as radio map does not yield computational overhead, using kernel-based methods such as SVM require explicit training. As SVM lookups operate on a deduced data set of classifications, the classifier has to recompute this regression every time a new measurement is added to the radio map. In the following, we analyze the query and training time of Redpin and SVM in both the C and the Java version.

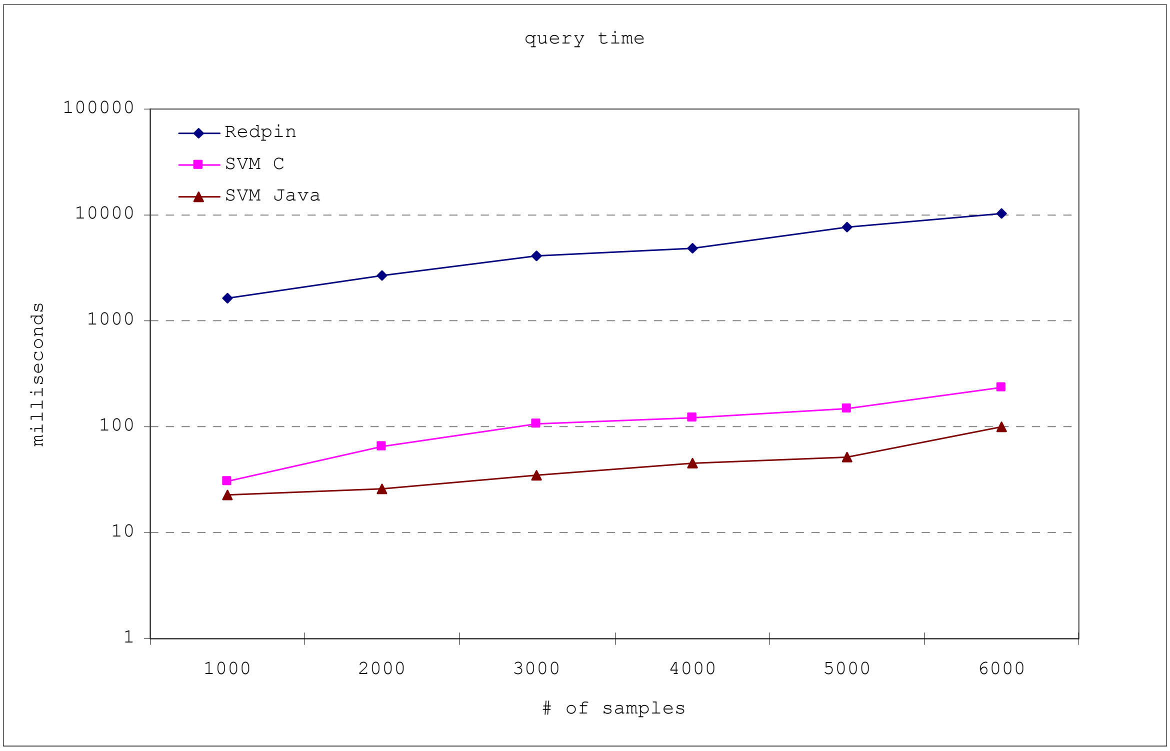


Figure 5.14: Comparison of average query time using Redpin, SVM C, and SVM Java for datasets of increasing size.

While Redpin requires significantly more time to perform a query, SVM shows much better performance for average query time. SVM C takes slightly more time than SVM Java. We believe this is due to the additional overhead incurred by the interface between the C code of LIBSVM and the Java code of the Redpin system into which SVM Java was integrated by implementation. As the estimation method used in Redpin has to compare a measurement with every entry in its radio map to lookup a location, the query time of this method obviously increases with the size of the radio map. We expect the same effect for SVM, but given the optimized structure of the dataset created by the classifier, we expect this effect to be much smaller. Surprisingly, the increase in query time due to bigger datasets is almost similar to the effect seen when using Redpin. Thus, using SVM does not solve to problem of increasing query time on its own but requires additional improvements when using very large datasets.

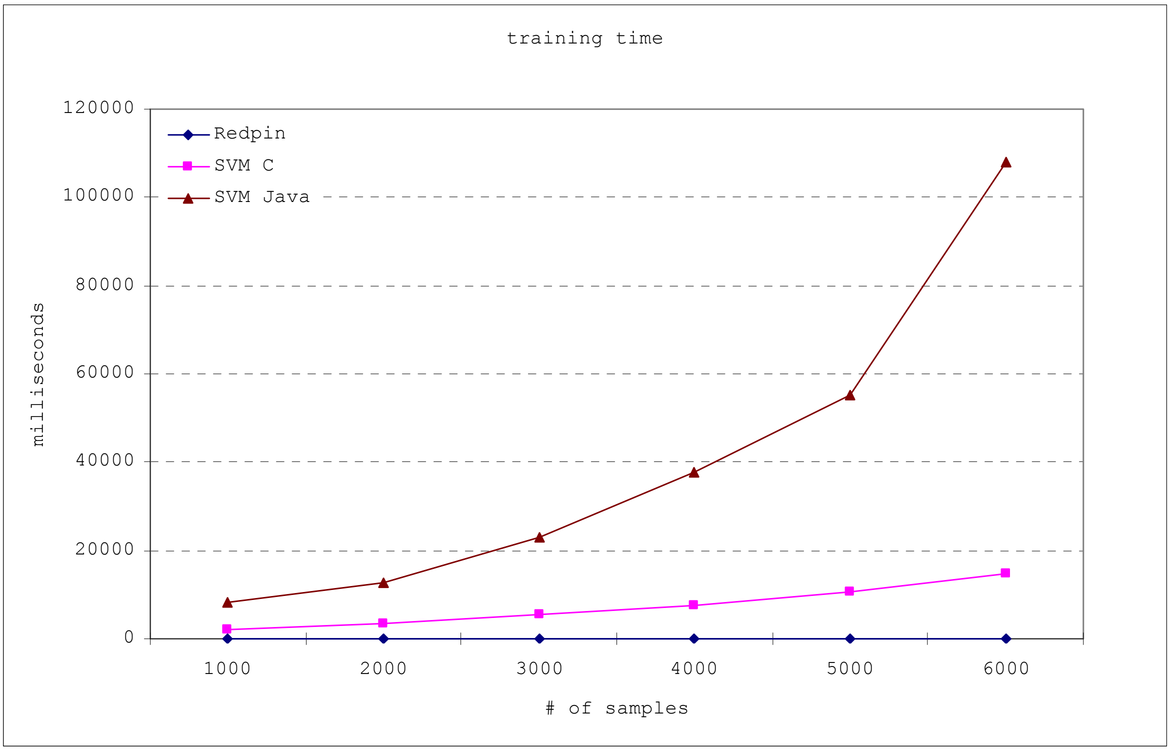


Figure 5.15: Comparison of average training time using Redpin, SVM C, and SVM Java for datasets of increasing size.

As already mentioned, Redpin does not require training phase as results in Figure 5.15 clearly show. On the other hand, the average time required to train SVM growths exponentially. In our case, the Java implementation of SVM in particular required very long time to run the kernel-based regression.

From the analysis of query and performance time, we learned that the performance of SVM in regard to accuracy is generally higher than the other estimation methods. In regard to query time, Redpin in its current implementation requires very long time for a lookup. Even for small radio map datasets with less than 2000 measurements, the average query time is more than 2.5 seconds. A kernel-based method like SVM outperforms probabilistic methods in this regard. But SVM based methods require to update the regression classification every time a new measurement is added to the radio map. Thus, while providing great accuracy and fast query time, using SVM yields the disadvantage of computationally expensive classification to update the radio map.

5.5 Conclusion

Building on the principle of collaborative location fingerprinting introduced in the previous chapter, we proposed and described new mechanisms that allow to label measurements taken during intervals, i.e. longer periods of time, instead of just instants. By using the built-in accelerometer, we were able to create a motion detector, which is able to determine whether a device is stationary or moving with high accuracy. In only very few cases did the motion detector report a false stationary state, while false reports of moving states never occurred. This enables the WiFi scanner component to continue recording measurements as long as the device is stationary. Consequently, the radio map contains more measurements by order of magnitudes. In addition, as these measurements where taken over periods of many minutes, our system is able to cope with the problem of short-term signal variations we observed in our WiFi studies (compare Chapter 3).

To further reduce obtrusiveness of the system, we introduced the concept of asynchronous labeling. With asynchronous labeling, the system continuously records intervals of measurements in the background. Using simple heuristics, the system determines the optimal point in time to prompt the user for a label of previously recorded intervals. In addition, by making intervals the unit of labeling, the labeling process can be performed at a less obtrusive time, since users are more likely to recognize intervals of stability than they are to recall their locations at instants. By means of this concept, we expect users to label even more measurements over a longer period of time.

While we saw a noticeable drop after the first day in the number of labels entered into the system, this is most likely due to the fact that by then most important places had been labeled, rather than assuming that users grew tired of the system. This is supported by the fact that users continued to correct the system at a roughly constant rate during the whole period of the experiment. Asynchronous labeling can also ensure that only “important” labels are solicited, such as the places that the user stays for long time periods or visits repeatedly. If the user stays at an unknown place for only a few minutes, our system can omit the prompt, thus further reducing the intrusiveness of the system.

Our initial results from both the experimental study and the survey give a strong indication that the accuracy of location fingerprinting can be improved by interval labeling. However, about one third of the survey participants reported that accuracy seemed to decline over time, which could have arisen from long-term signal fluctuations or over-fitting effects in the radio map. Moreover, the user study shows that labels can be collected without greatly burdening users, and that when such labels are applied to intervals, the maximum-likelihood of a new WiFi measurement is much higher than it would be if only instants were labeled.

In regard of estimation method optimization when using interval labeling, we found that probabilistic algorithms like k-nearest neighbor used in Redpin or the bayesian method used in PILS have disadvantages when the number of measurements is high. The biggest issue of probabilistic algorithms is the time required for location query. Once the radio map contains several thousand measurements, probabilistic algorithms like k-nearest neighbor require several seconds to find a matching fingerprint. In addition, the probabilistic algorithms we tested showed unstable accuracy when used with large datasets. Both of these problems can be diminished using kernel-based approaches like SVM. However, while our SVM based estimation method showed greatly reduced query time and very high, stable accuracy, this approach has the disadvantage of requiring computational overhead to train the classification. But as adding one or two measurements to a radio map that already contains tens of thousands of readings does not make a difference on average, we conclude that the classification can be run in the background at times when CPU time is available, for example overnight.

I may not have gone where I intended to go, but I think I have ended up where I intended to be.

– Douglas Adams

6 Conclusion

In this thesis we have presented several new methods to improve indoor positioning. From our analysis and evaluation of existing methods to indoor positioning using radio signal fingerprinting, we learned that this method shows good results in terms of accuracy and precision. However, in order to achieve sufficient accuracy, the radio map used for position estimation must contain ample number of measurements. Traditionally, these measurements are taken during an offline training phase, often by specially trained personnel. This method of training the radio map is time consuming and costly. Moreover, as radio signals fluctuate substantially both short- and long-term, the radio map needs to be updated periodically, thus aggravating the issue. We propose a new approach to location fingerprinting that omits this offline training phase. With our approach, any user is empowered to add new labels to the radio map, update or correct existing labels or simply add more measurements to an existing fingerprint. By outsourcing the task of radio map training to the users of the system in a collaborative manner, we were able to reduce the effort of setting-up and maintaing the radio map. Based on the findings of our studies on WiFi signal characteristics, we proposed an addition to collaborative labeling called “asynchronous interval labeling”. As we presented in this thesis, repeatedly collecting measurements over an interval of several minutes instead of just instants can greatly improve accuracy and helps mitigating the problems of short- and long-term signal variations.

In the first part of this thesis we presented an overview of existing location models and positioning technologies. Thereby, we have shown that hybrid location models are best suited to realize complex scenarios and applications in the field of Ubicomp. But as most existing location models are tightly integrated to other components, it is not possible to easily exchange location information between different systems. Corresponding abstractions that would allow for location oriented programming have been proposed but no model has prevailed to date. Recent work in the field of location models and efforts to build a common abstraction of positioning systems have certainly simplified the usage of such system for application programmers. However, a lot of work needs to be done as most concepts have drawbacks in either usability or their potential field of application. Of particular importance is work on location models with a focus on distributed applications. Especially issues like reliability, privacy, or authenticity are not satisfyingly covered by current work.

We thus chose to use a simple model to represent location information in order for our system to be easily integrated with other solutions. As we have shown, room level accuracy is generally sufficient for applications in the Ubicomp domain. Hence, we use an unstructured symbolic location model representing location simply by an identifier. However, as we exclusively rely on user input to label different locations, any user can (re)-define a location’s label at any time. Consequently, users will most probably use different labels for the same location. People don’t always use the “right” labels or same tag for different locations. For example, while some users prefer the term “bathroom”, others might use the label “toilet” to denote the same location. As we have shown in this thesis, the problem of ambiguous labels can emerge as users contributing to an otherwise uncontrolled collaborative system apply different labels in different ways. In this thesis we did not look into this problem in particular. But as crowd-sourcing becomes ever more common, we are confident that solutions to this problem will eventually emerge.

By means of two WiFi studies, we investigated WiFi signal characteristics. We found Wifi radio signals to fluctuate substantially, both long-term and short-term. Affected by absorption, reflection, refraction, multi-path, humidity, temperature, and many other factors, observed WiFi signals change over time without identifiable patterns. In particular the presence of humans changes the received signal strength considerably. This becomes evident when looking at RSS measurements taken during the night, which show significantly less signal variation compared to measurements taken during the day when people are moving around. We also found that different access points show different variances. Hence, access points may appear at different rates irrespective of their distance to the measuring terminal. In consequence, it is not possible to predict signal variation. It is thus necessary to take as many measurements as possible and update the radio map periodically. Moreover, we found that it is not sufficient to measure RSS for a few seconds only. To achieve high accuracy and precision, a fingerprinting based positioning system must be able to rely on measurements taken for minutes and, once again, repeated over many days. Only then is it possible to effectively cope with both short- and long-term signal variations. On the other hand, we could show that walls greatly improve signal separation. From this we conclude that achieving room-level precision is most feasible.

In the main part of this thesis, we presented novel approaches to location fingerprinting that minimize the effort of setting up and maintaining the radio map while coping with signal variations at the same time. With Redpin, we presented our implementation of a location fingerprinting system based on collaborative labeling. Using our method, every user may contribute measurements to the radio map at any time. This is opposed to existing fingerprinting methods that rely on specially trained personnel to collect RSS measurements in an offline-phase. Our initial experiments showed that even with a very simple distance measure and locator algorithm, the system achieves accuracy of about 90%. In addition, we could show that to get a complete map of an office building, only a few users must actually contribute to the system. Also, as Redpin allows multiple measurements per fingerprints for the same location, it is able to adapt to changes in the environment since users can always add new measurements by correcting their location. With Redpin we managed to create an easy-to-use and cost effective indoor positioning system that provides room-level precision with high accuracy.

While we have shown the feasibility of our approach by comparing it to existing systems making use of crowdsourcing and folksonomies, the barrier to participate must be low in every collaborative system. It does not suffice to simply give users access to the radio map by supporting different platforms. Users must be educated about the workings of the system and how they can partake and, more importantly, benefit by participating. We believe that the best way to achieve long-term user contribution is to design the user interface and the application per se such that location labeling is an integral part of the user experience instead of an additional task. For example, indoor positioning might be used for a location-aware chat application. In this case, the user is willing to enter a location label when "checking-in" to a location in order to start communicating. This procedure is used by very successful mobile apps such as Foursquare and shows that users are willing to enter a location label if they get a benefit from the action.

To maximize the leverage of user contribution, all fingerprints are shared between all users of the system. We employed this simple approach in both Redpin and PILS. This method allows to easily share location information between users and enables fast setup of radio maps in new buildings. On the other hand, this mechanism entails security and privacy implications. As we have shown in this thesis, the information about a user's location can be used to deduce important information about friends, activities or even political preferences. Because location information is so useful, it is most interesting to malicious attackers. In consequence, the user must be given complete control of her location information. Ideally, the positioning system provides different methods to anonymize, hide or mask a user's location. Different approaches that allow for these measures have been discussed in the introduction of this thesis. However, with the current version of both Redpin and PILS we have not yet addressed this issue. Another issue that we left for future work is the problem of coping with sensor variations. Using different hardware, i.e. different antennas and different network cards with different operating systems yield completely different RSS readings. Methods to eliminate these differences such as normalization are known but have not been implemented in our work.

With our method of "asynchronous interval labeling", we presented a method that can help to cope with both short- and long-term signal variations. Instead of taking single instances of measurements, we created a method that allows to label measurements taken over longer periods of time. By making use of the accelerometer we designed a motion detector, which can determine stationary state with high accuracy. This allowed us to continue taking measurements as long as the device is stationary. Hence, we were able to collect more measurements by order of magnitude compared to previous solutions were measurements were only take during instants. First, this allowed to average readings taken during intervals and thus eliminating the problem of short-term signal variations. Second, as the radio map contains much more readings, we could substantially increase the system's accuracy. However, we found that probabilistic algorithms such as k-nearest neighbor previously used as estimation method have major drawbacks when used on very large radio maps. The biggest issue we found was that query time of these estimation methods is very high. Even for relatively small datasets, containing only a few thousand readings, it took several seconds to execute a single lookup query. To alleviate this problem, we implemented a kernel-based estimation method using support vector machines. As we could show in this thesis, using an SVM-based estimation method allows for much shorter query time and provides slightly better and more stable accuracy. However, for this method to work, the system must update the regression mapping to account for newly added measurements, a process which is computationally expensive. But as adding a few measurements to a fingerprint that already contains hundreds of readings rarely makes a difference, we can afford to run the classifier in the background at times when CPU time is cheap.

Building on our learnings from the evaluation of Redpin, we introduced the concept of asynchronous labeling to further reduce obtrusiveness. While continuously recording measurements in the background, our system employs simple heuristics to determine the optimal point in time to prompt the user for a label later on. This way, the labeling process can be deferred to a point in time at which prompting the user for feedback does most likely not interrupt her workflow. However, with our first version of PILS we did use the change from battery power to AC power on a laptop to determine when a user is returning to her office or home. Obviously, different heuristics are required to apply this method to mobile phones. Since users are more likely to remember intervals of time than instants, we expected the rate of contribution to be stable. In addition, asynchronous labeling can also ensure that the user is only prompted for relevant labels, as the user has to be stationary at the same location for a longer period of time before being prompted. Our evaluation of asynchronous labeling has shown that although the number of newly entered labels dropped after the first day, users label more measurements over a longer period of time when being prompted for important labels only.

One possible drawback of interval labeling is the fact that continuously scanning the WiFi environment might degrade the bandwidth of this communication channel. Most network cards do not allow concurrent data transmission and network scanning. Therefore, data transmission has to be interrupted every time the positioning system wants to record a new measurement, effectively degrading the WiFi channels original purpose of transferring data. This issue is obviously bigger with interval labeling. However, as we have shown in our study on WiFi characteristics, it usually takes minutes for the WiFi signal to change significantly. We thus suggest to adapt the frequency of recording measurements given the variance of the WiFi signal observed over the last minutes and the amount of bandwidth used by other applications. While we did not evaluate the effect of interval labeling to bandwidth degradation, we believe that by simply lowering the scan frequency given simple heuristics, this problem can be resolved.

Bibliography

[1] G. D. Abowd, C. G. Atkeson, J. Hong, S. Long, R. Kooper, and M. Pinkerton. Cyberguide: A mobile contextaware tour guide. *Wireless Networks*, Jan 1997.

[2] U. Ahmad, Y.-K. Lee, and S. Lee. A distributed and parallel sampling system for efficient development of radio map. *International Conference on Information and Knowledge Engineering*, 2007.

[3] I. Anderson. Towards qualitative positioning for pervasive environments. *EUROCON*, Jan 2005.

[4] D. Ashbrook and T. Starner. Using GPS to learn significant locations and predict movement across multiple users. *Personal and Ubiquitous Computing*, Jan 2003.

[5] P. Bahl and V. Padmanabhan. Radar: an in-building RF-based user location and tracking system. *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE, Tel Aviv, Israel*, Jan 2000.

[6] M. Baldauf, S. Dustdar, and F. Rosenberg. A survey on context-aware systems. *International Journal of Ad Hoc and Ubiquitous Computing*, 2(4):263–277, 2007.

[7] U. Bandara, M. Hasegawa, M. Inoue, and H. Morikawa. Design

and implementation of a bluetooth signal strength based location sensing system. *Radio and Wireless Conference*, Jan 2004.

[8] L. Bao and S. S. Intille. Activity recognition from user-annotated acceleration data. *Pervasive Computing*, pages 1–17, Jan 2004.

[9] M. Bauer, C. Becker, and K. Rothermel. Location models from the perspective of context-aware applications and mobile ad hoc networks. *Personal and Ubiquitous Computing*, 6:322–328, Dec 2002.

[10] C. Becker and F. Durr. On location models for ubiquitous computing. *Ubiquitous Computing*, 9:20–31, Dec 2005.

[11] M. Beigl, T. Zimmer, and C. Decker. A location model for communicating and processing of context. *Personal and Ubiquitous Computing*, Dec 2002.

[12] V. Bellotti, B. Begole, E. H. Chi, N. Ducheneaut, J. Fang, E. Isaacs, T. King, M. W. Newman, K. Partridge, B. Price, P. Rasmussen, M. Roberts, D. J. Schiano, and A. Walendowski. Activity-based serendipitous recommendations with the magitti mobile leisure guide. In *Proceeding of the twenty-sixth annual SIGCHI conference on Human factors in computing systems*, CHI '08, pages 1157–1166, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-011-1.

[13] A. Beresford and F. Stajano. Location privacy in pervasive computing. *Pervasive Computing*, Jan 2003.

[14] A. Beresford and F. Stajano. Mix zones: User privacy in location-aware services. *Proceedings of the First IEEE International Workshop on Pervasive Computing and Communication Security (PerSec'04)*, Jan 2004.

[15] E. Bhasker, S. Brown, and W. Griswold. Employing user feedback for fast, accurate, low-maintenance geolocationing. *Pervasive Computing and Communications (PerCom)*, Jan 2004.

[16] R. W. Bill N. Schilit, Norman Adams. Context-aware computing applications. *Proceedings of the workshop on mobile computing systems and applications*, pages 85–90, Jan 2008.

[17] J. Bohn. Prototypical implementation of location-aware services based on a middleware architecture for super-distributed RFID tag infrastructures. *Personal and Ubiquitous Computing*, Jan 2008.

[18] P. Bolliger. Redpin - adaptive, zero-configuration indoor localization through user collaboration. *Workshop on Mobile Entity Localization and Tracking in GPS-less Environment Computing and Communication Systems (MELT), San Francisco*, 2008.

[19] P. Bolliger, K. Partridge, M. Chu, M. Langheinrich, A. Quigley, and T. Choudhury. Improving location fingerprinting through motion detection and asynchronous interval labeling. *Location and Context Awareness: 4th International Symposium, LoCA 2009 Tokyo, Japan, May 7-8, 2009 Proceedings*, page 37, 2009.

[20] A. Bonaccorsi and C. Rossi. Why open source software can succeed. *Research Policy*, Jan 2003.

[21] J. Borenstein and L. Feng. Umbmark: a method for measuring, comparing, and correcting dead-reckoning errors in mobile robots. *Technical Report #94-22, University of Michigan*, Jan 1994.

[22] J. Borenstein, H. R. Everett, L. Feng, and D. Wehe. Mobile robot positioning: Sensors and techniques. *Journal of Robotic Systems*, Jan 1997.

[23] G. Borriello. Methods and challenges in location systems. *Pervasive Computing Tutorial. Fifth International Conference, Pervasive 2007. Toronto, Canada*, May 2007.

[24] D. Brabham. Moving the crowd at istockphoto: The composition of the crowd and motivations for participation in a crowdsourcing application. *First Monday*, Jan 2008.

[25] D. Brabham. Crowdsourcing as a model for problem solving: An introduction and cases. *Convergence*, 14(1):75, 2008.

[26] D. Brabham. Moving the crowd at threadless: Motivations for participation in a crowdsourcing application. *Association for Education in Journalism and Mass Communication*, 2009.

[27] B. Brandherm and T. Schwartz. Geo referenced dynamic bayesian networks for user positioning on mobile systems. *Location- and Context-Awareness*, Jan 2005.

[28] B. Brumitt, B. Meyers, J. Krumm, A. Kern, and S. Shafer. Easyliving: Technologies for intelligent environments. *Handheld and Ubiquitous Computing*, pages 12–29, 2000.

[29] M. Brunato and R. Battiti. Statistical learning theory for location fingerprinting in wireless lans. *Computer Networks*, Jan 2005.

[30] M. Brunato and C. Kalló. Transparent location fingerprinting for wireless services. *Proc. Med-Hoc-Net*, 2002.

[31] S. Bryant, A. Forte, and A. Bruckman. Becoming wikipedian: transformation of participation in a collaborative online encyclopedia. *Proceedings of the 2005 international ACM SIGGROUP conference on Supporting group work*, Jan 2005.

[32] V. Bychkovsky, B. Hull, A. Miu, H. Balakrishnan, and S. Madden. A measurement study of vehicular internet access using in situ Wi-Fi networks. *Proceedings of the 12th International Conference on Mobile Computing and Networking*, Jan 2006.

[33] P. Castro, P. Chiu, T. Kremenek, and R. Muntz. A probabilistic room location service for wireless networked environments. *Ubiquitous Computing (Ubicomp)*, pages 18–34, Jan 2001.

[34] X. Chai and Q. Yang. Reducing the calibration effort for location estimation using unlabeled samples. *Pervasive Computing and Communications (PerCom)*, Jan 2005.

[35] G. Chen and D. Kotz. A survey of context-aware mobile computing research. *Technical Report: TR2000-381*, 2000.

[36] Y.-C. Cheng, Y. Chawathe, A. LaMarca, and J. Krumm. Accuracy characterization for metropolitan-scale Wi-Fi localization. *Proceedings of the Third ACM International Conference on Mobile Systems, Applications, and Services*, Jan 2005.

[37] E. Clary and M. Snyder. The motivations to volunteer: Theoretical and practical considerations. *Current Directions in Psychological Science*, 8(5), Dec 1999.

[38] E. Clary, M. Snyder, and R. Ridge. Understanding and assessing the motivations of volunteers: A functional approach. *Journal of Personality and Social Psychology*, 74(6):1516–1530, Jun 1998.

[39] L. Cong. Hybrid TDOA/AOA mobile user location for wideband cdma cellular systems. *Wireless Communications*, Jan 2002.

[40] B. N. D Niculescu. Ad hoc positioning system (APS) using AOA. *Proceedings of Twenty-Second Annual Joint Conference of the IEEE Computer and Communications*, pages 1734–1743, Jan 2003.

[41] B. N. D Niculescu. VOR base stations for indoor 802.11 positioning. *Proceedings of the 10th annual international conference on mobile computing and networking*, Jan 2004.

[42] C. Delakouridis, L. Kazatzopoulos, G. F. Marias, and P. Georgiadis. Share the secret: enabling location privacy in ubiquitous environments. *Location- and context-awareness (Springer)*, Jan 2005.

[43] P. Denning, J. Horning, D. Parnas, and L. Weinstein. Wikipedia risks. *Communications of the ACM*, 48(12):152, Dec 2005.

[44] A. Dey. Understanding and using context. *Personal and Ubiquitous Computing*, Jan 2001.

[45] A. Dey and G. Abowd. Towards a better understanding of context and context-awareness. *CHI 2000 Workshop on the What*, Jan 2000.

[46] S. Domnitcheva. Location modeling: State of the art and challenges. *Proceedings of the Workshop on Location Modeling for Ubiquitous Computing, September 30, Atlanta, Georgia, 2001*, Sep 2001.

[47] F. Durr and K. Rothermel. On a location model for fine-grained geocast. *Proceedings of the Fifth International Conference on Ubiquitous Computing 2003*, Jan 2003.

[48] F. Durr and K. Rothermel. An overlay network for forwarding symbolically addressed geocast messages. pages 427 –434, Jan 2007.

[49] B. Ferris, D. Hahnel, and D. Fox. Gaussian processes for signal strength-based location estimation. *Procedures of Robotics Science and Systems*, Jan 2006.

[50] G. Franck. Essays on science and society: Scientific communication–a vanity fair? *Science*, 286(5437), Jan 1999.

[51] J. Froehlich, M. Chen, I. Smith, and F. Potter. Voting with your feet: An investigative study of the relationship between place visit behavior and preference. *LECTURE NOTES IN COMPUTER SCIENCE*, Jan 2006.

[52] H. Funk and C. Miller. Location modeling for ubiquitous computing: Is this any better? *Location modeling for ubiquitous computing*, page 29, 2001.

[53] D. Garlan, D. P. Siewiorek, A. Smailagic, and P. Steenkiste. Project aura: Toward distraction-free pervasive computing. *Pervasive Computing*, 21(2):22–31, May 2002.

[54] R. Ghosh. Interviews with Linus Torvalds: What motivates software developers. *First Monday*, 3(3), Dec 1998.

[55] R. Ghosh. Understanding free software developers: Findings from the floss study. *Perspectives on free and open source software*, pages 23–46, Jan 2005.

[56] D. Graumann, W. Lara, J. Hightower, and G. Borriello. Real-world implementation of the location stack: the universal location framework. pages 122–128, oct. 2003. doi: 10.1109/MCSA.2003.1240773.

[57] E. D. M. Gregory D. Abowd. Charting past, present, and future research in ubiquitous computing. *ACM Transactions on Computer-Human Interaction*, Jan 2000.

[58] T. Gruber. Ontology of folksonomy: A mash-up of apples and oranges. *International Journal on Semantic Web & Information Systems*, 3(2):1–11, Jan 2007.

[59] M. Gruteser and D. Grunwald. Anonymous usage of location-based services through spatial and temporal cloaking. *Proceedings of the 1st international conference on mobile systems, applications and services*, pages 31–42, 2003.

[60] Y. Gu and A. Lo. A survey of indoor positioning systems for wireless personal networks. *IEEE communications surveys & tutorials*, 11, Jan 2009.

[61] A. Haeberlen, E. Flannery, A. Ladd, and A. Rudys. Practical robust localization over large-scale 802.11 wireless networks. *International Conference on Mobile Computing and Networking (MobiCom)*, pages 70–84, Jan 2004.

[62] D. Hahnel, W. Burgard, D. Fox, K. Fishkin, and M. Philipose. Mapping and localization with RFID technology. *Robotics and Automation*, 2004.

[63] H. Halpin, V. Robu, and H. Sheperd. The complex dynamics of

collaborative tagging. *Proceedings of the 16th International Conference on the World Wide Web, Banff, Canada*, Jan 2007.

[64] A. Hars and S. Ou. Working for free? motivations for participating in open-source projects. *International Journal of Electronic Commerce*, 6(3):25–39, Jan 2002.

[65] A. Harter, A. Hopper, P. Steggles, A. Ward, and P. Webster. The anatomy of a context-aware application. *Wireless Networks*, 8(2-3): 187–197, Jan 2002.

[66] C. Hauser and M. Kabatnik. Towards privacy support in a global location service. *Proceedings of the IFIP Workshop on IP and ATM Traffic Management*, Jan 2001.

[67] M. Hazas and A. Hopper. Broadband ultrasonic location systems for improved indoor positioning. *Mobile Computing, IEEE Transactions on*, 5(5):536–547, 2006.

[68] D. Hendry, J. Jenkins, and J. McCarthy. Collaborative bibliography. *Information Processing & Management*, 42(3):805–825, Jan 2006.

[69] J. Hightower. The location stack. *PhD Thesis*, Oct 2004.

[70] J. Hightower and G. Borriello. Location systems for ubiquitous computing. *Computer*, 34(8):57–66, 2001.

[71] J. Hightower and G. Borriello. A survey and taxonomy of location systems for ubiquitous computing. *IEEE Computer*, Jan 2001.

[72] J. Hightower, R. Want, and G. Borriello. Spoton: An indoor 3d location sensing technology based on rf signal strength. *UW CSE 00-02-02, University of Washington, Department of Computer Science and Engineering*, Jan 2000.

[73] J. Hightower, D. Fox, and G. Borriello. The location stack. *IRS Technical Report UW CSE 03-07-01*, 8, 2003.

[74] A. Hossain, H. Van, Y. Jin, and W. Soh. Indoor localization using multiple wireless technologies. *Mobile Adhoc and Sensor Systems (MASS)*, Jan 2007.

[75] J. Howe. The rise of crowdsourcing. *Wired Magazine*, Jan 2006.

[76] J. Howe. Crowdsourcing – a definition. *Crowdsourcing (Blog: http://crowdsourcing.typepad.com)*, Jun 2006.

[77] B. Huberman and C. Loch. Status as a valued resource. *Social Psychology Quarterly*, Jan 2004.

[78] B. Huberman, D. Romero, and F. Wu. Crowdsourcing, attention and productivity. *Journal of Information Science*, Jan 2009.

[79] IEEE. Ieee standard for information technology— telecommunications and information exchange between systems— local and metropolitan area networks— specific requirements part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications. *IEEE Std 802.11-1997*, Jan 1997.

[80] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal. Ariadne: A dynamic indoor signal map construction and localization system. *International Conference On Mobile Systems, Applications And Services (MobiSys)*, pages 151–164, Apr 2006.

[81] C. Jiang and P. Steenkiste. A hybrid location model with a computable location identifier for ubiquitous computing. *UbicOmp*, Aug 2002.

[82] E. Kaasinen. User needs for location-aware mobile services. *Personal and Ubiquitous Computing*, Jan 2003.

[83] K. Kaemarungsi. Design of indoor positioning systems based on location fingerprinting technique. *Dissertation, School of Information Sciences, University of Pittsburgh*, Jan 2005.

[84] J. I. Karen Henricksen and A. Rakotonirainy. Modeling context information in pervasive computing systems. *Pervasive Computing*, Jan 2002.

[85] N. Kern, B. Schiele, H. Junker, P. Lukowicz, and G. Tröster. Wearable sensing to annotate meeting recordings. *Personal Ubiquitous Computing*, 7:263–274, October 2003. ISSN 1617-4909.

[86] T. King and M. B. Kjaergaard. Composcan: adaptive scanning for efficient concurrent communications and positioning with 802.11. *International Conference On Mobile Systems, Applications And Services (MobiSys)*, pages 67–80, Jan 2008.

[87] T. King, S. Kopf, T. Haenselmann, C. Lubberger, and W. Effelsberg. Compass: A probabilistic indoor positioning system based on 802.11 and digital compasses. *Proceedings of the First ACM International Workshop on Wireless Network Testbeds, Experimental evaluation and CHaracterization (WiNTECH)*, Aug 2006.

[88] T. King, T. Haenselmann, and W. Effelsberg. Deployment, calibration, and measurement factors for position errors in 802.11-based indoor positioning systems. *Proceedings of the Third International Symposium on Location- and Context-Awareness (LoCA)*, 4718:17–34, 2007.

[89] M. B. Kjaergaard. Automatic mitigation of sensor variations for signal strength based location systems. *Proc. of the Second Int. Workshop on Location and Context Awareness (LOCA)*, Jan 2006.

[90] M. B. Kjaergaard. Cleaning and processing RSS measurements for location fingerprinting. In *Proceedings of the Third International Conference on Autonomic and Autonomous Systems*, page 12, Washington, DC, USA, 2007. IEEE Computer Society. ISBN 0-7695-2859-5.

[91] M. B. Kjaergaard. A taxonomy for radio location fingerprinting. *Location- and Context-Awareness (LoCA)*, pages 139–156, Jan 2007.

[92] M. B. Kjaergaard. Indoor positioning with radio location fingerprinting. *PhD Dissertation, Department of Computer Science, University of Aarhus, Denmark*, Jan 2008.

[93] M. B. Kjaergaard. Indoor positioning with radio location fingerprinting. 2008.

[94] M. B. Kjaergaard and C. Munk. Hyperbolic location fingerprinting: A calibration-free solution for handling differences in signal strength. *Proc. of the Sixth Annual IEEE International Conference on Pervasive Computing and Communications*, pages 110–116, Jan 2008.

[95] M. B. Kjaergaard, G. Treu, and C. Linnhoff-Popien. Zone-based RSS reporting for location fingerprinting. *Proceedings of the Fifth International Conference on Pervasive Computing (Pervasive)*, 2007.

[96] D. Kotz, C. Newport, and C. Elliott. The mistaken axioms of wireless-network research. *Technical Report TR2003-467, Dartmouth College*, Jan 2003.

[97] J. Krumm. A survey of computational location privacy. *Adjunct Proceedings of the Ninth International Conference on Ubiquitous Computing*, 13:391–399, August 2009. ISSN 1617-4909.

[98] J. Krumm and E. Horvitz. Locadio: Inferring motion and location from Wi-Fi signal strengths. *Mobile and Ubiquitous Systems: Networking and Services (MOBIQUITOUS)*, 2004.

[99] A. Küpper. Location-based services. *John Wiley & Sons*, Jan 2005.

[100] A. Küpper, G. Treu, and C. Linnhoff-Popien. Trax: A device-centric middleware framework for location-based services. *IEEE Communications*, Jan 2006.

[101] N. Kuster and Q. Balzano. Energy absorption mechanism by biological bodies in the near field of dipole antennas above 300 MHz. *Vehicular Technology*, Jan 1992.

[102] D. Lambeth. Design considerations for an indoor location service using 802.11 wireless signal strength. *Master Thesis*, Jan 2009.

[103] R. Lambiotte and M. Ausloos. Collaborative tagging as a tripartite network. *Computational Science*, pages 1114–1117, Jan 2006.

[104] J. Lampel and A. Bhalla. The role of status seeking in online communities: Giving the gift of experience. *Journal of Computer-Mediated Communication*, Jan 2007.

[105] D. Lancashire. The fading altruism of open source development. *First Monday*, 6(12), Jan 2001.

[106] M. Langheinrich. Personal privacy in ubiquitous computing. *Dissertationsschrift, ETH Zurich*, 2005.

[107] J. Lester, T. Choudhury, and G. Borriello. A practical approach to recognizing physical activities. *Pervasive Computing (PERVASIVE)*, pages 1–16, Jan 2006.

[108] Z. Li, W. Trappe, Y. Zhang, and B. Nath. Robust statistical methods for securing wireless localization in sensor networks. *Proceedings of the Fourth International Symposium on Information Processing in Sensor Networks*, Jan 2005.

[109] H. Lim, L. Kung, J. Hou, and H. Luo. Zero-configuration, robust indoor localization: Theory and experimentation. *INFOCOM, Barcelona, Spain*, 2006.

[110] H. Lin, Y. Zhang, M. Griss, and I. Landa. Wasp: An enhanced indoor locationing algorithm for a congested Wi-Fi environment. *Mobile Entity Localization and Tracking in GPS-less Environnments (MELT)*, Jan 2009.

[111] H. Liu, H. Darabi, P. Banerjee, and J. Liu. Survey of wireless indoor positioning techniques and systems. *Systems*, Jan 2007.

[112] K. Lorincz and M. Welsh. Motetrack: a robust, decentralized approach to RF-based location tracking. *Personal and Ubiquitous Computing*, Jan 2007.

[113] M. Macomber. World geodetic system 1984. *Defense mapping agency, Washington DC*, Jan 1984.

[114] D. Madigan, E. Elnahrawy, R. Martin, and W. Ju. Bayesian indoor positioning systems. *IEEE INFOCOM*, Jan 2005.

[115] A. Mathes. Folksonomies - cooperative classification and communication through shared metadata. *Computer Mediated Communication*, Jan 2004.

[116] P. Merholz. Metadata for the masses. *Adaptive Path Technical Report*, Jan 2004.

[117] A. Mockus, R. Fielding, and J. Herbsleb. Two case studies of open source software development: Apache and mozilla. *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Jan 2002.

[118] C. Morelli, M. Nicoli, V. Rampa, and U. Spagnolini. Hidden markov models for radio localization in mixed los/nlos conditions. *IEEE transactions on signal processing*, 55(4):1525, 2007.

[119] T. Mundt. Two methods of authenticated positioning. *Proceedings of the 2nd ACM International Workshop on QoS and Security for Wireless and Mobile Networks*, Jan 2006.

[120] F. Naya, H. Noma, R. Ohmura, and K. Kogure. Bluetooth-based indoor proximity sensing for nursing context awareness. pages 212–213, 2005.

[121] X. Nguyen, M. I. Jordan, and B. Sinopoli. A kernel-based learning approach to ad hoc sensor network localization. volume 1, pages 134–152, New York, NY, USA, August 2005. ACM.

[122] O. Nov. What motivates wikipedians. *ACM Communications*, 50: 60–64, 2007.

[123] L. Ojeda and J. Borenstein. Personal dead-reckoning system for GPS-denied environments. *Proceedings of IEEE International Workshop on Safety, Security and Rescue Robotics*, pages 1 – 6, 2007.

[124] V. A. Otsason, Veljo, E. de Lara, and A. LaMarca. Accurate GSM indoor localization. *Proceedings of the Seventh International Conference on Ubiquitous Computing*, Jan 2005.

[125] X. C. P. J. Brown, J. D. Bovey. Context-aware applications: from the laboratory to the marketplace. *Personal Communications*, Jan 2002.

[126] P. K. P. Prasithsangaree and P. K. Chrysanthis. On indoor position location with wireless LANs. *Personal, Indoor and Mobile Radio Communications*, pages 720–724, 2002.

[127] J. Pan, J. Kwok, Q. Yang, and Y. Chen. Multidimensional vector regression for accurate and low-cost location estimation in pervasive computing. *IEEE transactions on knowledge and data engineering*, pages 1181–1193, 2006.

[128] S. J. Pan, V. W. Zheng, Q. Yang, and D. H. Hu. Transfer learning for WiFi-based indoor localization. *Association for the Advancement of Artificial Intelligence (AAAI) Workshop*, page 6, May 2008.

[129] I. Peters. Folksonomies: Indexing and retrieval in web 2.0. *De Gruyter Saur (Berlin)*, Jan 2009.

[130] N. Priyantha, A. Chakraborty, and H. Balakrishnan. The cricket location-support system. *Proceedings of the Sixth Annual ACM International Conference on Mobile Computing and Networking*, pages 32–43, Jan 2000.

[131] C. Randell, C. Djiallis, and H. Muller. Personal position measurement using dead reckoning. *Wearable Computers, 2003. Proceedings. Seventh IEEE International Symposium on*, pages 166 – 173, 2003.

[132] A. Ranganathan, J. Al-Muhtadi, S. Chetan, R. Campbell, and M. D. Mickunas. Middlewhere: a middleware for location awareness in ubiquitous computing applications. *Proceedings of the 5th ACM/IFIP/USENIX international conference on Middleware*, pages 397–416, Jan 2004.

[133] E. Raymond. Homesteading the noosphere. *First Monday*, Jan 1998.

[134] M. Rethlefsen. Tags help make libraries del. icio. us: Social bookmarking and tagging boost participation. *Library Journal*, Jan 2007.

[135] A. C. Rice, R. K. Harle, and A. R. Beresford. Analysing fundamental properties of marker-based vision system designs. *Pervasive and Mobile Computing*, Jan 2006.

[136] V. Robu, H. Halpin, and H. Sheperd. Emergence of consensus and shared vocabularies in collaborative tagging systems. *ACM Transactions on the Web*, pages 14:1–14:34, Jan 2009.

[137] T. Rodden, A. Friday, H. Muller, and A. Dix. A lightweight approach to managing privacy in location-based services. *Proceedings of Equator's Annual Meeting*, 02, Jan 2002.

[138] L. Rogoleva. Crowdsourcing location information to improve indoor localization. *Master Thesis (ETH)*, Jan 2010.

[139] S. Salzberg. On comparing classifiers: Pitfalls to avoid and a recommended approach. *Data Mining and Knowledge Discovery*, Jan 1997.

[140] M. Satyanarayanan. Pervasive computing: vision and challenges. *IEEE Personal Communications*, Jan 2001.

[141] B. Schilit, N. Adams, R. Gold, M. Tso, and R. Want. The parctab mobile computing system. *Workstation Operating Systems*, Jan 1993.

[142] A. Smailagic and D. Kogan. Location sensing and privacy in a context-aware computing environment. *IEEE Wireless Communications*, 9:10–17, 2001.

[143] I. Smith, J. Tabert, T. Wild, A. Lamarca, A. Lamarca, Y. Chawathe, Y. Chawathe, S. Consolvo, S. Consolvo, J. Hightower, J. Hightower, J. Scott, J. Scott, T. Sohn, T. Sohn, J. Howard, J. Howard, J. Hughes, J. Hughes, F. Potter, F. Potter, P. Powledge, P. Powledge, G. Borriello, G. Borriello, B. Schilit, and B. Schilit. Place lab: Device positioning using radio beacons in the wild. In *In Proceedings of the Third International Conference on Pervasive Computing*, pages 116–133. Springer, 2005.

[144] T. Sohn, W. G. Griswold, J. Scott, A. LaMarca, Y. Chawathe, I. Smith, and M. Chen. Experiences with place lab: an open source toolkit for location-aware computing. In *Proceedings of the 28th international conference on Software engineering*, ICSE ’06, pages 462–471, New York, NY, USA, 2006. ACM. ISBN 1-59593-375-1.

[145] P. Steggles and S. Gschwind. The ubisense smart space platform. *Adjunct Proceedings of the 3rd International Conference on Pervasive Computing*, pages 73–76, Jan 2005.

[146] K. J. Stewart and S. Gosain. The impact of ideology on effectiveness in open source software development teams. *MIS Quarterly*, 30(2), Aug 2006.

[147] D. Stork. The open mind initiative. *IEEE Expert Systems and Their Applications*, 14, Jan 2000.

[148] D. Stork and C. Lam. Open mind animals: Ensuring the quality of data openly contributed over the world wide web. *AAAI Workshop on Learning with Imbalanced Data Sets*, Jan 2000.

[149] J. Surowiecki and M. Silverman. The wisdom of crowds. *American Journal of Physics*, page 190, Jan 2007.

[150] J. C. Tang, N. Yankelovich, J. Begole, M. Van Kleek, F. Li, and J. Bhalodia. ConNexus to awarenex: extending awareness to mobile users. pages 221–228, 2001.

[151] L. von Ahn. Games with a purpose. *IEEE Computer Magazine*, pages 96–98, Jan 2006.

[152] L. von Ahn and L. Dabbish. Labeling images with a computer game. *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 319–326, Jan 2004.

[153] T. V. Wal. Folksonomy. *Information Architecture Institute Members Mailing List*, Jan 2004.

[154] T. V. Wal. Folksonomy. *Online Information*, Jan 2005.

[155] R. Want, A. Hopper, V. Falcão, and J. Gibbons. The active badge location system. *ACM Transactions on Information Systems (TOIS)*, 10(1):91–102, Jan 1992.

[156] R. Want, B. N. Schilit, N. I. Adams, R. Gold, K. Petersen, D. Goldberg, J. R. Ellis, and M. Weiser. The parctab ubiquitous computing experiment. *IEEE PERSONAL COMMUNICATIONS*, 2:28–43, 1995.

[157] A. Ward, A. Jones, and A. Hopper. A new location technique for the active office. *IEEE Personal Communications*, 4:42–47, Oct 1997.

[158] M. Weiser. Hot topics: ubiquitous computing. *IEEE Computer*, 10:71–72, 1993.

[159] M. Weiser. The computer for the 21 stcentury. *ACM SIGMOBILE Mobile Computing and Communications Review*, Jan 1999.

[160] M. Weiser. Some computer science issues in ubiquitous computing. *ACM SIGMOBILE Mobile Computing and Communications Review*, 3(3), 1999.

[161] M. Weiser and J. Brown. Designing calm technology. *PowerGrid Journal*, 1:1–5, 1996.

[162] M. Weiser and J. S. Brown. *The coming age of calm technolgy*, pages 75–85. Copernicus, New York, NY, USA, 1997. ISBN 0-38794932-1.

[163] M. Weiser, R. Gold, and J. S. Brown. The origins of ubiquitous computing research at parc in the late 1980s. *IBM Syst. J.*, 38: 693–696, December 1999. ISSN 0018-8670.

[164] R. Yamasaki, A. Ogino, T. Tamaki, T. Uta, N. Matsuzawa, and T. Kato. TDOA location system for IEEE 802.11 b WLAN. *Wireless Communications and Networking Conference*, Jan 2005.

[165] M. Youssef and A. Agrawala. The horus WLAN location determination system. pages 205–218, 2005.

[166] M. Youssef, J. Krumm, E. Miller, G. Cermak, and E. Horvitz. Computing location from ambient FM radio signals. *Wireless Communications and Networking Conference*, 2:824–829, Mar 2005.

[167] S. A. Zekavat, H. Tong, and J. Tan. A novel wireless local positioning system for airport (indoor) security. *Proc. SPIE*, 2004.

Curriculum Vitae

Particulars

Name	Philipp Lukas Bolliger
Date of Birth	March 27, 1978
Birthplace	Frauenfeld, TG, Switzerland
Citizenship	Küttigen, AG, Switzerland

Education

2006–2010	Research and teaching assistant supervised by Prof. Dr. Friedemann Mattern in the Distributed System research group at ETH Zurich
1999–2006	Study of Computer Science at ETH Zurich
1998–1999	Basic Military Training and Officer School, Swiss Army
1993–1998	Matura Type E, Kantonsschule Büelrain, Winterthur
1991–1993	Secondary School, Weisslingen
1985–1991	Primary School, Weisslingen

Work Experience

2009–today	Founder and CEO, Koubachi AG, Zurich, Switzerland
2004–2005	Software development at AlmafinJaeger, SunGard, St. Gallen, Switzerland
2001	Internship as software developer at Winterthur Insurances, Winterthur, Switzerland