UNIVERSITY of CALIFORNIA
Santa Barbara

Improving Location Accuracy And Network Capacity In Mobile Networks

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Computer Science

by

Zengbin Zhang

Committee in Charge:

Professor Haitao Zheng, Chair

Professor Ben Y. Zhao

Professor Amr El Abbadi

December 2014

The Dissertation of Zengbin Zhang is approved.

Professor Ben Y. Zhao

Professor Amr El Abbadi

Professor Haitao Zheng, Committee Chair

August 2014

Improving Location Accuracy And Network Capacity

In Mobile Networks

To all my family and friends.

Acknowledgements

I would like to express my deepest appreciation to my advisor and committee chair, Professor Heather Zheng, who has guided me through every aspect of PhD process and trained me so hard to become a researcher with independent thoughts and critical mind. No one has ever been like Heather, who always is the first one to encourage me when I feel low, and who also is the first one to remind me my weaknesses before I feel self-satisfied. I appreciate all her wise guidance and generous support on my research, my future career and my personal development. I also thank her for her persistency in reminding me to always look at the big picture and build research insights.

I sincerely thank my committee members Professor Ben Y. Zhao and Professor Amr El Abbadi. Ben has witnessed my growth ever since I joined the lab as a PhD student. I thank him for his dedication on training my research skills, critical thinking and academic writing. His insight and enthusiasm on research has significantly affected me. I also deeply appreciate his strong recommendations and wise advices during my job searching, which helps me build a bright path to my future career. I deeply thank Professor Amr El Abbadi for his support on my research, and for his valuable feedback and suggestions on my PhD thesis.

I am very lucky to have two wise mentors for my internships, Dr. Thomas Moscibroda and Dr. David Chu. I especially thank them for their guidance to me on industry research experiences, and the precious opportunities they have introduced me to. I also thank them for helping me build research skills and presentation skills.

I am also grateful to have such a great group of colleagues and friends, Lei Yang, Xia Zhou, Yibo Zhu, Xiaohan Zhao, Gang Wang, Lin Zhou, who have

accompanied me through this journey. I shall always remember all those days of deadlines when we worked together, all those discussions about ideas, all those sunshine when we did experiments outside and all those fun nights and dinners. I am so happy to work with such bright people like you, from whom I learn and grow.

Finally, I would like express my special thanks to my beloved family, who have been giving me their endless love and support ever since I was born, whom I can rely on forever. I thank my mom and dad who have raised me and encouraged me to pursuit my dreams. I thank my sisters who have taken care of our mom and dad while I was far away on the other side of the Pacific ocean. Thank you so much.

Curriculum Vitæ

Zengbin Zhang

Education

2009 — 2014	PhD in Computer Science, University of California, Santa Barbara.
2006 — 2009	Master of Science in Electronic Engineering, University of California, Santa Barbara.
2002 — 2006	Bachelor of Science in Electronic Engineering, Tsinghua University, China.

Experience

2009 — 2014	Graduate Research Assistant, University of California, Santa Barbara.
Summer 2012	Research Intern, Microsoft Research
Summer 2011	Research Intern, Microsoft Research Aisa
Summer 2007	Research Intern, Microsoft Research Asia

Selected Publications

1	Y. Zhu, Z. Zhang, Z. Marzi, C. Nelson, U. Madhow, B. Y. Zhao and H. Zheng: "Demystifying 60GHz Outdoor Picocells", To *Appear in MobiCom*, 2014.
2	Y. Zhu, X. Zhou, Z. Zhang, L. Zhou, A. Vahdat, B. Y. Zhao and H. Zheng: "Cutting the Cord: A Robust Wireless Facilities Network for Data Centers", To *Appear in MobiCom*, 2014.

3 Z. Zhang, L. Zhou, X. Zhao, G. Wang, M. Metzger, H. Zheng and B. Y. Zhao:"On the Validity of Geosocial Mobility Traces", In *Proc. of HotNets*, 2013.

4 X. Zhou, Z. Zhang, G. Wang, X. Yu, B. Y. Zhao and H. Zheng: "Practical Conflict Graphs for Dynamic Spectrum Distribution", In *Proc. of SIGMETRICS*, 2013.

5 X. Zhou, Z. Zhang, Y. Zhu, Y. Li, S. Kumar, A. Vahdat, B. Y. Zhao and H. Zheng: "Mirror Mirror on the Ceiling: Flexible Wireless Links for Data Centers", In *Proc. of SIGCOMM*, 2012.

6 Z. Zhang, D. Chu, X. Chen, T. Moscibroda: "SwordFight: Enabling a New Class of Phone-to-Phone Action Games on Commodity Phones", In *Proc. of MobiSys*, 2012.

7 L. Yang, Z. Zhang, B. Y. Zhao, C. Kruegel, H. Zheng: "Enforcing Dynamic Spectrum Access with Spectrum Permits", In *Proc. of MobiHoc*, 2012.

8 Z. Zhang, X. Zhou, W. Zhang, Y. Zhang, G. Wang, B. Y. Zhao and H. Zheng: "I Am the Antenna: Accurate Outdoor AP Location using Smartphones", In *Proc. of MobiCom*, 2011.

9 W. Zhang, X. Zhou, L. Yang, Z. Zhang, B. Y. Zhao and H. Zheng: "3D Beamforming for Wireless Data Centers", In *Proc. of Hotnets*, 2011.

10 L. Cao, L. Yang, X. Zhou, Z. Zhang and H. Zheng. "Optimus: SINR-driven Spectrum Distribution via Constraint Transformation", In *Proc. of DySPAN*, 2010.

Abstract

Improving Location Accuracy And Network Capacity
In Mobile Networks

by

Zengbin Zhang

Todays mobile computing must support a wide variety of applications such as location-based services, navigation, HD media streaming and augmented reality. Providing such services requires large network bandwidth and precise localization mechanisms, which face significant challenges. First, new (real-time) localization mechanisms are needed to locate neighboring devices/objects with high accuracy under tight environment constraints, e.g. without infrastructure support. Second, mobile networks need to deliver orders of magnitude more bandwidth to support the exponentially increasing traffic demand, and adapt resource usage to user mobility.

In this dissertation, we build effective and practical solutions to address these challenges. Our first research area is to develop new localization mechanisms that utilize the rich set of sensors on smartphones to implement accurate localization systems. We propose two designs. The first system tracks distance to nearby devices with centimeter accuracy by transmitting acoustic signals between the devices. We design robust and efficient signal processing algorithms that measure distances accurately on the fly, thus enabling real-time user motion tracking. Our second system locates a transmitting device in real-time using commodity smartphones. Driving by the insight that rotating a wireless receiver (smartphone)

around a users body can effectively emulate the sensitivity and functionality of a directional antenna, we design a rotation-based measurement algorithm that can accurately predict the direction of the target transmitter and locate the transmitter with a few measurements.

Our second research area is to develop next generation mobile networks to significantly boost network capacity. We propose a drastically new outdoor picocell design that leverages millimeter wave 60GHz transmissions to provide multi-Gbps bandwidth for mobile users. Using extensive measurements on off-the-shelf 60GHz radios, we explore the feasibility of 60GHz picocells by characterizing range, attenuation due to reflections, sensitivity to movement and blockage, and interference in typical urban environments. Our results dispel some common myths on 60GHz, and show that 60GHz outdoor picocells are indeed a feasible approach for delivering orders of magnitude increase in network capacity.

Finally, we seek to capture and understand user mobility patterns which are essential in mobile network design and deployment. While traditional methods of collecting human mobility traces are expensive and not scalable, we explore a new direction that extracts large-scale mobility traces through widely available geosocial datasets, *e.g.* Foursquare "check-in" datasets. By comparing raw GPS traces against Foursquare checkins, we analyze the value of using geosocial datasets as representative traces of human mobility. We then develop techniques to both "sanitize" and "repopulate" geosocial traces, thus producing detailed mobility traces more indicative of actual human movement and suitable for mobile network design.

Professor Haitao Zheng
Dissertation Committee Chair

Contents

List of Figures

List of Tables

Chapter 1

Introduction

We are witnessing an explosive growth of mobile networks. With the penetration of mobile devices, such as smartphones, tablets, smart glasses/watches, smart cars and Internet of things, innovations in mobile applications have taken place to deliver a wide range of services to users on the go.

- *Location based services.* Location based services take a user's current location as input and provide her with information about services or objects in vicinity, *e.g.* restaurants or products. These applications have been growing rapidly in recent years. For instance, Yelp, one of the most popular location based services for browsing local businesses, now has 80 million active business accounts and serves 138 million queries every month [7]. However, limited by the accuracy of localization techniques, existing services mainly target outdoor scenarios driven by GPS-based localization mechanisms. This triggers many research projects on indoor localization [28, 26, 96, 87, 151, 40, 43, 65], *e.g.* to allow a store to recommend products in proximity to a customer based on her physical location.

- *Interactive or collaborative applications.* Various interactive or collaborative applications are expected to appear in the near future. The collaborator can be among humans [33, 63, 121, 139, 142], *e.g.* in a multi-player game mobile devices can sense each other's location or motion, and feed the information into the game. The interaction can also be between a human and a device. For instance, a user can use motions to command a device [113, 108, 72, 44]; a car can locate nearby pedestrians by capturing the signal transmitted from their mobile devices [49]. Similarly, one can find a malfunctioned wireless device or locating a lost item (with a wireless transmitter attached to it).

- *Augmented reality.* Augmented reality applications [27, 146, 18, 11] are expected to recognize the surrounding objects based on the information captured by sensors, and then link digital information to them. These applications rely on sensors to accurately capture the position of these objects, and leverage the information to assist the user, *e.g.* to avoid collisions in a navigation system for the blind. They also need high speed mobile connections to transmit the captured data to the cloud, *e.g.* to enable object recognition, and download the related digital information from the cloud.

- *Real-time video communication & HD media streaming.* These applications are the biggest consumer of mobile traffic. Recent market research predicts that in 2018, 80% of total mobile data traffic will be produced by video/audio applications [9], thanks to the popularity of HD screens on mobile devices. The penetration of smart glasses or smart car systems will further increase the traffic demand in the future.

To support these services, technical advancements are required in many areas. In this dissertation, we focus on two key elements.

Highly Accurate Localization Mechanisms. Most of these services require highly accurate localization systems to locate the device itself as well as other devices or objects. For instance, in the multi-player game scenario, players normally operate in a room-size space (*e.g.* several meters), thus the location accuracy should be in sub-meter level or even centimeter level to accurately capture user motion. Similarly, to smart car systems, the localization accuracy must be high enough to effectively avoid accidents. In addition to accuracy, certain applications have additional constraints. For example, a collision avoidance system for the blind must work anywhere they go, thus the localization mechanism can not assume the existence of external infrastructures. In some applications, the targeted device can only transmit wireless signal, but is unable to actively participate in the localization process, *e.g.* a lost item with a wireless tag.

These requirements lead to significant challenges to existing localization mechanisms, *i.e.* GPS-, Cellular- or WiFi-based systems. First, in these systems the location of a device is measured based on either signal strength or propagation time of wireless signal, which can not reach high accuracy due to the physical nature. Therefore, even the state-of-the-art solution can only reach meters of median error, and tens of meters of maximum error [96, 87]. Second, all these methods require localization infrastructures, like multiple basestations or a group of satellites. For instance, GPS devices are required to be visible to five or more satellites in order to provide an accuracy of 50 meters median error [88].

High-capacity Mobile Networks. Almost all these services require high-bandwidth connections between a device and the cloud, or among devices. Given the exponential growth of mobile devices, our mobile networks need to provide orders of magnitude more bandwidth in the near future. Industry research has predicted that aggregate mobile bandwidth requirements will increase by 1000-fold by 2020 [1].

Our existing networks can not meet this demand as well. First, LTE, WiFi or the new TV White Space only possess hundreds of MHz spectrum. The spectral efficiency is slowly approaching the Shannon limit [46], while acquiring more spectrum facing regulatory hurdles and high monetary costs [2]. Second, the benefit of shrinking cell size is fundamentally limited by interference constraints for the carrier frequencies employed in these systems [115]. Third, the physical size of antennas in these low frequency bands (1 to 5GHz) constrains the maximum number of antennas in mobile devices, thus the benefit provided by MIMO technology is also limited [115]. Therefore, large capacity gains in these networks cannot be seen in the near future.

Finally, user mobility can cause high dynamics in mobile networks. The traffic demand can be significantly different for different locations, or even the same location at different time of a day [14]. To deliver a smooth connectivity, mobile networks need to adapt resource allocation in real time according to user mobility patterns. However, due to privacy and security concerns, existing studies can only hire a limited number of users [157, 156] or rely on coarse-grained registration data from cellular or WiFi networks [144, 32, 152], while large scale, detailed user mobility traces are very hard to acquire.

1.1 Overview

To address the above challenges, we make significant contributions in this dissertation by proposing practical systems and tools in both localization mechanisms and mobile network deployment.

The thesis statement of this dissertation is as follows: **To support a wide variety of mobile applications at scale, we need to develop highly accurate device localization systems, design new mobile networks that achieve orders of magnitude more capacity, and adapt network usage to user mobility. This dissertation addresses these challenges by developing novel and practical localization mechanisms leveraging the rich set of smartphone sensors, by initiating a drastically new cellular design using millimeter wave technologies, and by engineering a data analysis tool that extracts large-scale human mobility traces from geosocial datasets.**

1.1.1 Precise Device Localization

Our first goal is to build localization systems that not only can locate a target device in high accuracy (centimeter-level), but also can work in situations when the target devices don't cooperate in the localization process. We also remove the dependency on localization infrastructures in order to make the proposed systems operate ubiquitously. In particular, we propose the following two systems.

Accurate Mobile Device Localization. This localization system aims to locate another mobile device in centimeter level accuracy, as well as to capture user motion in real time, *i.e.* with measurement delay less than 100ms. To reach such

performance, we use acoustic signals to measure the distance, and design efficient sound processing algorithms to boost the measurement speed. We also design specialized algorithms to handle issues caused by ambient noise, multipath propagation and Doppler effect. Our implementation runs on commodity smartphones and does not require any external infrastructure. Moreover, distance measurement accuracy is comparable to that of Kinect, a fixed-infrastructure motion capture system. This system can enable lots of innovative applications like interactive games, collaborative applications and augmented reality.

Accurate Wireless Transmitter Localization. In many applications, the target devices don't cooperate in the localization process, *e.g.* a misbehaving wireless transmitter. Our second localization system targets to locate such a wireless device using smartphones. This work makes an intuitive observation that the blocking effect of human bodies could significantly degrade the signal strength. Leveraging this effect, we accurately identify the direction of the wireless device by a rotation-based signal strength measurement and a direction analysis algorithm. To make the system fast and energy efficient, we also add a driver level design to boost the measurement speed by 10 times while reducing energy consumption by 10 times. The final system can locate the target device on the fly with a small number of measurements even in rich multipath environments.

1.1.2 Boosting Network Capacity

To supply orders-of-magnitude more network bandwidth, we explore a drastically different alternative in the form of 60GHz millimeter wave picocells with highly directional links. We believe that 60GHz links truly reap the spatial reuse

benefits of small cells while providing high network capacity with 7GHz-wide spectrum band.

Using extensive measurements on off-the-shelf 60GHz radios and system-level simulations, we explore the feasibility of 60GHz picocells by characterizing range, attenuation due to reflections, sensitivity to movement and blockage, and interference in typical urban environments. Our results dispel some common myths, and show that there are no fundamental physical barriers to high-capacity 60GHz outdoor picocells. We also identify open challenges and associated research opportunities to release the full potential of the design.

1.1.3 Capturing User Mobility at Scale

Finally, we answer whether we can acquire representative user mobility traces from existing location-based services, such as Foursquare, to further improve network performance. Our approach is to crawl the ground-truth GPS locations of a group of users along with their "check-in" traces exposed in the social networks. By comparing the traces in parallel, we seek to understand the discrepancies between them. The results are astonishing: we found large numbers of users lying about their own locations, while lots of context-important locations (e.g. Home, Work) were not checked in. A deeper analysis reveals that the faked checkins are due to the rewarding system of this kind of social networks, which were designed to improve user engagement, while the missing checkins are mainly due to personal preferences. Based on these observations, we use machine learning algorithms to filter the faked checkins and design interpolation methods to recover the missed ones.

1.2 Contributions

Our contributions in this dissertation are three-fold. We identify key requirements of mobile networks, and propose novel algorithm designs and data analytic tools to address them. We also verify our designs by solid system implementations while considering practical issues in real world deployments.

Identifying Fundamental Problems. We identify several fundamental problems in mobile networks. In Chapter 2 and 3, our proposed localization systems break some key limitations of previous solutions on localization accuracy, infrastructure support and device cooperation, thus could enable a wide variety of new applications. In Chapter 4, we are the first to propose a outdoor picocell design using 60GHz links, and identify the potentials and challenges of the system using real-world measurements. In Chapter 5, we reveal the limitations of mining massive location data from location-based social networks, and propose sanitization methods to address the limitations.

Novel Algorithm Designs. We leverage key observations about the system or the data, and propose effective designs to solve the problems. In Chapter 2, our observation on acoustic signal processing help us improve the computation efficiency by 10-fold, enabling the system to capture user motion in real time. In Chapter 3, we build an accurate localization algorithm based on the observation of the blocking effect caused by human body. In Chapter 4, we utilize our observations on some distinctive temporal and spatial features of faked Foursquare checkins and apply machine learning algorithms to filter them out.

Solid Implementation. We also validate our designs with real implementations. In Chapter 2 and 3, along with the algorithm design, we propose com-

ponents to improve system robustness and energy efficiency. Both localization systems have been implemented on one or more smartphone platforms and tested by real users. In Chapter 4, we evaluate our picocell design by two types of off-the-shelf 60GHz radios, and perform the experiments under real world environments. In Chapter 5, we verify the effectiveness of our sanitization tools by applying it on a secondary dataset obtained from another location-based social network.

1.3 Thesis Organization

The rest of the dissertation is organized as follows.

Chapter 2 and Chapter 3 present our localization system designs. In particular, Chapter 2 describes the highly accurate device localization method that can reach centimeter level accuracy in real time. Chapter 3 presents our device localization system which does not require the target device to cooperate.

In Chapter 4, we summarize our work on the next generation mobile network design using 60GHz technology. We conduct a series of measurements to validate our design and point out future research directions. In Chapter 5, we describe our work of extracting large-scale human mobility data from location based social networks.

We finally summarize and talk about future directions in Chapter 6.

Chapter 2

Accurate Mobile Device Localization

2.1 Introduction

Mobile gaming constitutes a large and fast growing industry. Estimates on the total worldwide market size differ, but typical numbers are in the order of $10 billion for 2011 [16]. Furthermore, continued growth in the underlying forces – smartphone and tablet sales, mobile Internet subscribers and app downloads – all point to a bright future for the industry.

The landscape of today's mobile games is rich and varied. However, one commonality among existing multiplayer games is that they invariably require the players to be physically passive, look at the screen, and conduct game action via interaction with the screen. The success of Nintendo Wii or Xbox Kinect in console gaming on the other hand, has demonstrated a demand for much more interactive games in which gameplay directly involves the user's physical activity. In this chapter, we report on the development of a novel class of *Phone-to-Phone*

Mobile Motion Games (MMG) that achieve a similar level of physical interactivity. These games are characterized by the fact that the *position, location, orientation, or movement of the phone is an integral part of game play.* In some cases, the phone may even serve as the equivalent of a Wii-stick, and players may not even need to look at the screen while playing. However, in contrast to Wii or Kinect, we do not rely on any external infrastructure such as a microphone array or cameras; MMG games are played purely phone-to-phone.

Consider for example the following *SwordFight* game. The rules are simple: Two players wield their phones, and try to attack each other. A player can attack the opponent by tapping the screen. If player A attacks, and her phone is within 20cm of the opponent's phone, she wins. However, attacking costs energy, and an attack can only be sustained for 4 seconds before the energy has completely depleted. Energy can be regained over time when the player remains in non-attack mode. Thus, one strategy to win is for a player to survive the opponent's attack by quickly moving his phone in such a way that it maintains sufficient distance from the opponent's phone; and then counter-attack while the opponent's energy is depleted.

The key enabling technology for a game like SwordFight is the ability to conduct *very fast*, *accurate*, and *robust distance measurements* between the phones, so that at any moment during play, both phones have precise distance information. Studies have shown that in order to sustain high-speed action games a lag of more than 100ms decreases user satisfaction and degrades player performance, and a lag of 200ms is unacceptable [34]. Therefore, we need to be able to conduct phone-to-phone distance measurements at a rate of at least 10Hz. The measurements also have to be accurate – with no more than a few centimeters of error – and

robust in the face of mobility, noise, and networking issues. In the absence of any external infrastructure, the combination of these three requirements constitutes a significant technical barrier on commodity phones, and no existing solution is able to meet them simultaneously.

It is well-known that acoustic sound can be used for distance measurements. Works such as [107, 114] have demonstrated that under ideal circumstances (e.g., no mobility) and with sufficient computation time, accurate ranging can be achieved even on commodity phones. The problem is that in an motion game scenario, the circumstances are far from ideal and the requirements substantially more challenging. First, existing ranging protocols are based on the assumption that phone positions remain static during the process of taking a measurement. For a game like SwordFight, this is not a valid assumption as human hand speed can be as high as 2m/s. Furthermore, with two phones moving towards or apart from each other at high speed, aspects such as the Doppler effect need to be considered and dealt with. Secondly, acoustic ranging requires the use of expensive cross-correlation algorithms in order to determine the precise time-of-arrival of the sound signal. Cross-correlation algorithms are computationally intensive and cannot be run without modification on phones at sufficient speed to enable a SwordFight game. Third, not only *computation*, but also the *communication* (acoustic tone exchanges, protocol handshakes, etc) incurs a fundamental and significant delay.

In this work, we address these challenges in a systematic manner. Our first contribution is to enable real time distance measurement by replacing the standard computationally intensive cross-correlation algorithm for finding sound peaks with a more sophisticated and efficient multi-stage algorithm. Our algorithm employs

autocorrelation to fundamentally reduce computational complexity while preserving accuracy by targeting cross-correlation to a very narrow search window. Our second contribution is to employ a new pipelined streaming execution strategy that overlaps protocol communication and algorithm computation. While overlapping communication and computation is a well-known technique, in our case, this overlapping comes with a twist – we overlap the *sound waves* of the ranging protocol in addition to the typical networking data packets. It turns out that both pipelining and streaming are critical in order to realize real time measurements. Our third contribution is to understand and tackle the practical sources of measurement error during motion gaming for increased robustness. The first set of robustness optimizations addresses mobility and the effects of Doppler shift on the underlying ranging protocol. The second set of optimizations addresses environmental robustness from effects such as ambient noise, multipath and acoustic tone loss.

Combining these techniques, we develop `FAR` – a Fast, Accurate and Robust localization system that enables two potentially fast moving phones to keep accurate distance estimates to each other. Distance measurements can be taken at a rate of 12Hz with 2cm median error while withstanding up to 0dB noise (e.g., players or spectators talking while playing), multipath (e.g. as encountered in small rooms) and the Doppler effect. To practically demonstrate the system's ability to enable novel gaming concepts, we develop two prototype MMG games: SwordFight and ChaseCat.[1] In both isolated-player and in-situ gameplay experiments, we find that `FAR`'s distance measurements are comparable to Kinect, a

[1]See `http://research.microsoft.com/mobile-motion-gaming` for a video illustrating the gameplay of SwordFight.

dedicated fixed-infrastructure motion capture system. We have publicly tested our games at various locations with real players; and our experience shows that the games are fun and intuitive to play. We anticipate that with the high-speed ranging API we provide in this system, many more MMG games can be developed.

2.2 Assumptions & Requirements

Enabling a game like SwordFight requires a distance ranging subroutine that allows two phones to keep accurate and up-to-date distance information between each other even in the face of noise and high mobility. This poses a unique set of challenges:

- **Phone-to-Phone:** MMG games should be playable everywhere at any time. We do not rely on any external infrastructure beyond the two phones.
- **Commodity phones:** We want MMG games to run on commodity hardware and OSs. This implies handling issues of on-phone sensor sampling rates, computation capacity, or audio playing and recording capabilities, whether they arise from the hardware, (typically closed source) audio driver, or OS.
- **Measurement Frequency:** To create the sensation of continuous real-time distance information, the ranging infrastructure must sample user movement as frequently as possible, at a rate of 10Hz or more [34].
- **High Accuracy:** Gameplay relies on the high accuracy of distance measurements even at a high degree of phone mobility. Specifically, we aim for a target accuracy of within 2-3cm up to a normal range of human social interaction of up to approximately 3m with line of sight.

- **User Mobility:** In a game like SwordFight, players should be able to play without artificial restrictions on their body or hand movement. The underlying ranging infrastructure must be capable of supporting the speed of natural human hand movement (up to 2m/s) [71].
- **Practical in Most Environments:** As it is natural for players and spectators to talk during a game, and because a game may be played indoors or close to walls, the underlying distance measurement framework must be robust against the impact of high ambient noise levels, multi-path effects and tone loss.

2.3 Background

In this section, we review the well-known time-of-flight acoustic ranging principle that we share in common with prior phone-based acoustic localization work [107, 114], and we outline why these existing approaches are unsuited for enabling Mobile Motion Gaming.

2.3.1 Acoustic Distance Measurement

The essential idea is to have two phones A and B play and record known audio tones one after another. These tones are often based on some form of pseudorandom sequences. Each phone records its own emitted tone as well as the tone originating from the remote phone. Suppose A emits a tone first and records this tone's arrival at its microphone at time t_{A1}. B records the arrival of this tone at time t_{B1}. Next, B emits a tone, which is recorded at the microphones of B and

A at times t_{B2} and t_{A2}, respectively. The distance d between the phones can now be calculated as

$$d = \frac{1}{2} \cdot c \cdot [(t_{A2} - t_{A1}) - (t_{B2} - t_{B1})], \tag{2.1}$$

where c is the speed of sound.

Existing schemes [107, 114] follow a traditional execution strategy of *Play/Record then Compute and Exchange* (see Figure 2.2(a)). The figure shows that the two phones start the *Recording* step at the same time, and then send out the tones one after another (*Tone Playing* step). Guard periods are inserted prior to and following each tone to guarantee that the tones do not overlap, and that the *Recording* is complete.

Upon completion of the recording, the next step is for each phone to determine the exact local time when each tone was received. This is done by applying a *cross-correlation* algorithm to each of its recorded sound samples (*Computation* step). Cross-correlation is a standard signal processing technique that searches for the best match between a recorded sample and a reference signal (the best match is indicated by a sharp peak in the cross-correlation). It has been widely used in various ranging systems, e.g. [53, 58, 79, 107, 114, 125, 149] and is computed as

$$CC(t_0) = \frac{\sum_{t \in W}[X(t) - \overline{X(t)}][T(t - t_0) - \overline{T(t - t_0)}]}{\sqrt{\sum_{t \in W}[X(t) - \overline{X(t)}]^2 \sum_{t=1-L}^{0}[T(t) - \overline{T(t)}]^2}}$$

where $X(t)$ is the recorded sound sequence, $T(t), t \in [-L + 1, 0]$ is the reference signal of length L, $\overline{X(t)} = \frac{\sum_{t \in W} X(t)}{L}$, $\overline{T(t)} = \frac{\sum_{t=1-L}^{0} T(t)}{L}$, $W = [t_0 - L + 1, t_0]$. For each t_0, the computation complexity is $O(|W|) = O(L)$, and thus $O(B * L)$ if we have B recorded sound samples. Finally, once the cross-correlation peaks are found at each phone, the corresponding tone arrival time-stamps are exchanged

between the two phones e.g., via 3G or WiFi (*Measurement Exchange* step), and the distance can be computed.

2.3.2 Limitations

As pertains to MMG, existing phone-to-phone acoustic localization schemes all have the same drawbacks: they are too slow to serve the needs of MMG games due to the large measurement delay, and they are not designed for highly-mobile systems.

- The traditional execution strategy of *Play/Record then Compute and Exchange* is unsuited for high frequency measurements, as it takes too much time.
- Cross-correlation is well-known to be computationally expensive, as it grows super-linearly in the size of the reference signal L and the recorded sound sequence T. For example, although [114] applied an energy detection module to reduce the search space in the sound sequence T, the reported time for one measurement is still 800ms.
- Existing protocols are not designed for highly-mobile systems. They assume that during the course of a measurement, the phones remain stationary and if they are not, results will be erroneous.

Finally, in addition to these more fundamental reasons, there are also purely practical system design issues that prevent us from building MMG games on top of existing ranging infrastructure:

- Commodity phones do not offer tight timing guarantees on the operation of the playing and recording controls, and thus systematically incur large and

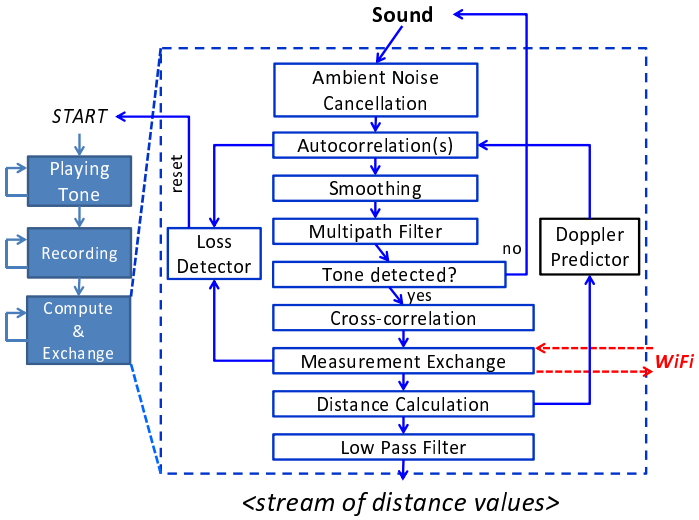


Figure 2.1: `FAR` Architecture

unpredictable delays. Specifically, as we show in §2.8, there is a lag of high magnitude and high variance between the time a playback is initiated and the actual time the tone is played. For example, the lag is 60-90ms for both Nexus One phone running Android 2.3.4 and Samsung Focus phone running Windows Phone 7.5. This caps the frequency at which phones can emit tones. Based on our empirical observations, this lag occurs so commonly among commodity smartphones that we cannot simply discount it as a software- or device-specific phenomenon.

2.4 FAR System Design

In this section, we introduce `FAR`, our fast, accurate, and robust distance measurement system that serves as an API to phone-to-phone MMG games like SwordFight. Meeting the requirements in §2.2 in the face of the limitations in §2.3.2 is

not easy and drives us to a novel system design as well as numerous optimizations at both the algorithm and systems level.

The key goal of the `FAR` system is to systematically improve upon existing phone-to-phone ranging schemes in two directions – i) by making them *faster* and reducing delays (See §2.5); and ii) by making them more *robust* in the face of mobility (See §2.6) – while at the same time keeping the required degree of accuracy.

Figure 3.6 illustrates the `FAR` architecture that achieves these goals. Upon initiation, each of the major stages *Playing, Recording, Computation* and *Exchange* happen continuously. §2.5 discusses the dynamic arrangement of these stages. *Computation* contains the main algorithmic contributions of our work. It is sequenced as follows.

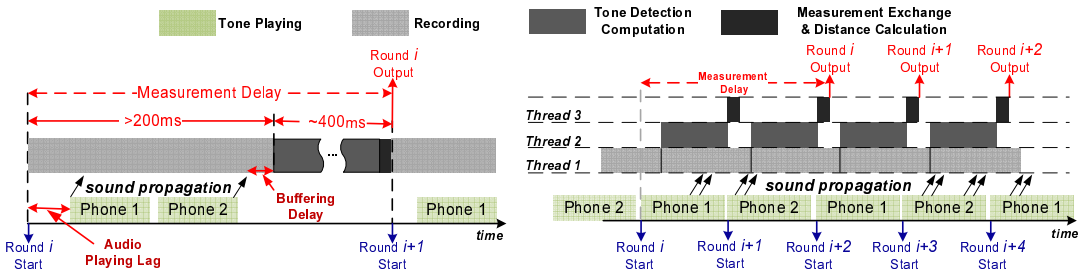
1. An ambient noise filter mitigates environmental noise such as from shouting, talking and crowd noise common during gameplay.

2. One or more lightweight autocorrelators work in tandem to detect the presence of tone signatures.

3. Smoothing reduces the impact of anomalous local minima which are the result of sound distortion played and captured by commodity hardware.

4. Multipath readings (e.g. reflection from nearby objects) are filtered out to identify the actual tone received. If no tones are detected, expensive cross-correlation is not engaged.

5. The cross-correlator identifies the tone reception time stamp for accurate timing information.



(a) Traditional Execution Strategy (b) FAR Execution Strategy

Figure 2.2: Execution Strategies

6. The phones exchange their measured time stamps via WiFi. Failure to receive data here (due to signal jamming or tone misdetection in the autocorrelation step) over prolonged durations results in the loss detector automatically signaling a protocol reset.

7. The exchanged data is used in distance calculation. The distance calculation also informs the Doppler predictor, which adjusts the number of autocorrelators to use in the next round. This enhances measurement accuracy at high mobility.

8. A final low pass filter smooths the calculation before emitting the distance value to the game.

2.5 Fast Distance Measurements

In this section, we present the set of techniques that allows FAR to conduct distance measurements at high frequency and with low lag. To see how our tech-

niques impact lag and frequency, it is useful to again consider the traditional execution strategy as illustrated in Figure 2.2(a). In the traditional execution strategy, the lag of each individual measurement is comprised of the following components:

- **Tone Length:** Time spent on sending out each tone. Since one measurement requires that both phones send out tones, the measurement lag includes two tone lengths.
- **Audio Playing Lag:** Delay between the time of calling the `Play()` API and the time when the tone is actually sent out from the speaker. This delay is observed across platforms on both Android and Windows Phone.
- **Sound Propagation Delay:** Time it takes for the sound to reach the other phone over the air.
- **Buffering Delay:** Time from when the tone has been fully recorded until when the audio driver passes the recorded buffer to the application layer. The reason for the delay is that the audio driver delivers a buffer only when the buffer is full, regardless of whether the tone has arrived.
- **Tone Detection Computation Time:** Time required to compute a recorded buffer and determine the exact time-stamp when the tone arrived.
- **Measurement Exchange & Distance Calculation:** Time it takes for the two phones to exchange their time-stamps, and compute the final distance measurement result using Equation (2.1).

If we want to achieve measurement frequency above the 10Hz required for MMG games, we must address these delays across the board. `FAR` accomplishes this through an efficient system architecture and set of core acoustic signal process-

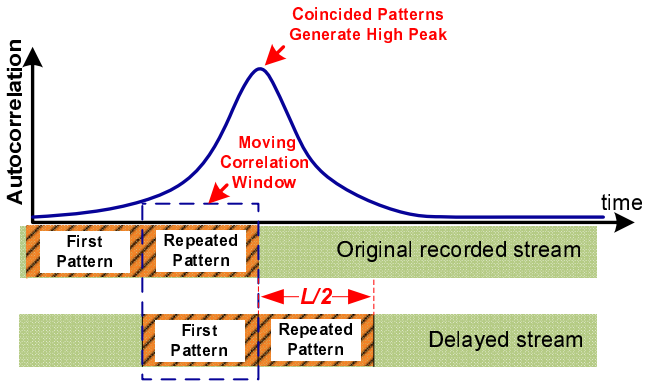
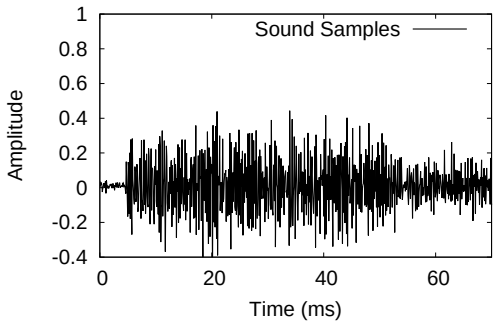


Figure 2.3: Autocorrelation-based Tone Detection. When sliding the correlation window, at a certain time, the repeated pattern in the original stream will coincide with the first pattern of the delayed stream, which generates a high correlation peak.

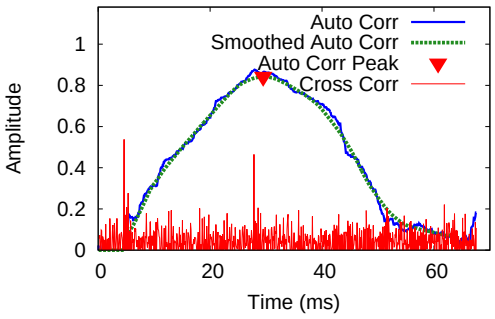
ing algorithms. §2.5.1 discusses how we substantially reduce the tone-detection computation time by a very efficient algorithm. §2.5.2 introduces pipelining to further improve measurement frequency, and streaming to circumvent widespread system lags. Lastly, §2.5.3 summarizes how we reduce other components of delay.

2.5.1 Fast Tone Detection Algorithm

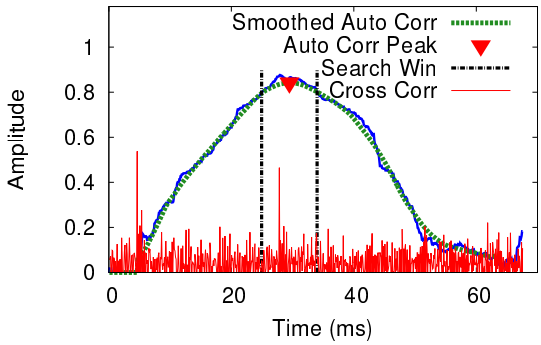
As discussed in §2.2, the problem of relying upon cross-correlation for tone detection is that its computation overhead is high. To find the start time of a tone of length L in a buffer of size B, a total of $O(B * L)$ multiplications are required. Our key observation is that instead of running the cross-correlation directly, we can first use a computationally much more efficient *autocorrelation* primitive. Autocorrelation's strength is that it is much more efficient than cross-correlation at finding *repeating patterns* in data sequences. However, autocorrelation does not give us an exact estimate of the tone's location because of its much flatter peak style. Therefore, after applying autocorrelation, we employ (as a second and third



(a) Captured Sound Samples



(b) Auto- and Cross-correlation Peaks



(c) Peak Search in a Small Window

Figure 2.4: Fast Tone Detection Algorithm. (a) Recording of received tone with repeating patterns. (b) While cross-correlation peaks are sharp, the auto-correlation peak is flat and has an offset to the pattern's actual location. (c) Cross-correlation is used in a small window around the autocorrelation peak to find the precise location of the pattern.

step) a simple smoothing algorithm and then a small-scale cross-correlation in a narrow search window centered around the smoothed autocorrelation peak. We now discuss these steps in detail.

Step 1: Autocorrelation-based Tone Detection. The basic methodology of autocorrelation we use is shown in Figure 2.3. Autocorrelation can only detect correlation of a signal to itself. Therefore, we let each phone send out a tone

consisting of a pseudorandom sequence followed immediately by an exact copy of this sequence. To detect this tone, the receiver phone records the incoming stream $X(t)$, and then generates a second stream $Y(t)$ which is an identical copy of $X(t)$, except that it is delayed by half the tone length L, i.e, $Y(t) = X(t - L/2)$. The receiver now correlates $X(t)$ and $Y(t)$ (i.e., the incoming steam is correlated with a copied, delayed version of itself) in a moving correlation window of length $L/2$. This is done by incrementally computing for each time step t_0 the function

$$R(L/2, t_0) = \frac{\sum_{t \in W}[X(t) - \overline{X(t)}][Y(t) - \overline{Y(t)}]}{\sqrt{\sum_{t \in W}[X(t) - \overline{X(t)}]^2 \sum_{t \in W}[Y(t) - \overline{Y(t)}]^2}}$$

where $\overline{X(t)} = \frac{\sum_{t \in W} X(t)}{L/2}$, $\overline{Y(t)} = \frac{\sum_{t \in W} Y(t)}{L/2}$, and $W = [t_0 - L/2 + 1, t_0]$. Higher autocorrelation values correspond to closer correlations of $X(t)$ and $Y(t)$ at t_0. The idea is that because phones send two identical sequences in each tone, at a certain time slot, one of the repeated patterns in $X(t)$ will match a first pattern in the delayed stream $Y(t)$, thus generating a large autocorrelation peak as shown in Figure 2.3.

The benefit of the above method is that we can compute $R(L/2, t_0)$ in $O(1)$ time. To see this, consider that the above expression can be rewritten as follows.

$$R(L/2, t_0) = \frac{\frac{L}{2}\widehat{XY} - \widehat{X}\widehat{Y}}{\sqrt{[\frac{L}{2}\widehat{XX} - \widehat{X}\widehat{X}][\frac{L}{2}\widehat{YY} - \widehat{Y}\widehat{Y}]}}$$

where

$$\widehat{XY} = \sum_{t \in W}[X(t)Y(t)],\ \widehat{XX} = \sum_{t \in W}[X(t)X(t)]$$
$$\widehat{YY} = \sum_{t \in W}[Y(t)Y(t)],\ \widehat{X} = \sum_{t \in W} X(t),\ \widehat{Y} = \sum_{t \in W} Y(t).$$

Note that all these variables can be computed in $O(1)$ time with a standard moving window method. Taking $\widehat{X}$ as an example, given $\widehat{X}(t_0)$, we can compute $\widehat{X}(t_0{+}1)$

as follows.

$$\widehat{X}(t_0{+}1) = \widehat{X}(t_0) + X(t_0+1) - X(t_0 - L/2 + 1).$$

Since $\widehat{X}(t_0)$, $X(t_0{+}1)$ and $X(t_0{-}L/2{+}1)$ are all known, $\widehat{X}(t_0{+}1)$ is computed in $O(1)$. The other variables can also be updated in the same way. $R(L/2, t_0+1)$ can thus be obtained in $O(1)$ time based on the variables at t_0. For a buffer of size B, this autocorrelation solution has complexity $O(B)$, which is much faster than $O(B * L)$ via traditional cross-correlation.

Inaccuracy of Autocorrelation based Tone Detection. Unfortunately, we cannot rely solely on autocorrelation to find the tone peak as the result will be inaccurate. Figure 2.4(b) shows the output of the autocorrelation step. As can be seen, there is a noticeable offset between the autocorrelation peak and second cross-correlation peak (a sharp line). In an ideal scenario, these two peaks should perfectly align. The reason for this discrepancy is twofold. First, autocorrelation fundamentally has a much flatter peak. Intuitively, if a time slot's autocorrelation value is high, its neighboring time slots' values are also high. Second, the offset can arise due to signal distortion of the tone after propagation. This can result in up to 10cm error in the final result even in quiet environments as we show in the evaluation. For these reasons, we cannot rely exclusively on autocorrelation for tone detection. Rather, we seek to combine the respective benefits of cross-correlation and autocorrelation.

Step 2: Smoothing of Autocorrelation Peak. In order to obtain cleaner peaks, we smooth the autocorrelation curves within a 10ms window. Note that this smoothing process is necessary as the raw autocorrelation output can have many local maxima due to environmental noise.

Step 3: Small Scale Cross-correlation. We interpret the smoothed autocorrelation curve as an indicator for the location of the tone, then we search in a small window S around the autocorrelation peak using standard cross-correlation to detect the precise tone location (see Figure 2.4(c)). The computation overhead of the combined algorithm now is $O(B + |S| * L)$, which is still much smaller than $O(B * L)$, assuming that the autocorrelation indicator allows us to choose sufficiently small $|S|$.

Choosing the Search Window Size $|S|$. The above discussion reveals an interesting design trade-off concerning the selection of the window size $|S|$. A smaller $|S|$ yields faster cross-correlation computation, but risks mis-detecting a tone altogether because it is outside the searching window. This is especially true in noisy environments where the offset between cross-correlation and autocorrelation peaks can be large. Based on our empirical measurements, we set our window size to a very conservative size of 100 samples (offset by -80 to $+20$ around the detected peak), which is sufficient even at high noise. The fact that the search window is not symmetric around the detected peak is probably due to reflected signals, which may generate higher autocorrelation peaks after the direct signal's peak.

2.5.2 Pipelined Streaming Execution Strategy

With the preceding reductions in tone detection computation time, we next introduce `FAR`'s *pipelined streaming execution strategy* for boosting measurement frequency. Its two key improvements upon the traditional execution strategy are that (1) it employs pipelining to increase measurement frequency by overlapping

computation time with tone transmission, and (2) it employs streaming to minimize measurement lag by removing Audio Playing Lag. Figure 2.2(b) illustrates the `FAR` execution strategy.

Pipelined Execution to Mask Computation Time. We devise a pipelined execution strategy that uses separate threads to handle *Playing, Recording, Tone-Detection Computation*, and *Measurement Exchange* (Figure 2.2(b)). In the recording thread, the Audio Recorder periodically fills in a predefined audio buffer. For example, the minimum required buffer size for Nexus One running Android 2.3.4 is 1024 Bytes, which corresponds to 512 sound samples. Once the buffer is filled, the computation thread processes the buffered sound samples, while the recording thread waits for the next buffer. Once computation is finished, the measurement exchange thread exchanges the detected tone arrival times with the other phone using WiFi and outputs the final distance measurement. The net effect is that tone-detection computation is now fully overlapped with the exchange of sound waves (including recording delay, playing delay and signal propagation delay). Given that in our system, these two sets of operations take approximately equal amount of time, the pipelined execution strategy effectively doubles `FAR`'s measurement frequency.

Note that when pipelining, it is of critical importance that the tone-detection computation time is tightly bounded. If the computation on one buffer is not finished before the next buffer arrives, the computation overhead will accumulate over time, resulting in buffer overflows and lost sound samples.

Streaming Execution to Mitigate Audio Playing Lag. Instead of blocking on computation completion as the traditional strategy does, `FAR` implements a

streaming mode to continuously send and record tones. The `Play()` API of the Audio Driver is called at a frequency similar to the tone length, thus one tone will be sent right after another. This helps in eliminating the Audio Playing Lag from the overall measurement time. Similarly, we also implement the Audio Recorder to work in a streaming fashion, continuously recording sound samples for computation. As a result, if there is no lag between tones, measurements can theoretically be conducted at a frequency equal to $1/L$, provided tone detection is fast enough to support such a rate.

2.5.3 Further Optimizations

In addition to the techniques discussed in the previous sections, our system implements several additional optimizations to reduce delay.

Tone Length Minimization. Choosing the right Tone Length is important. Choosing a small value can negatively affect measurement accuracy, but choosing a large value not only adds lag to the overall execution, but also increases the complexity of the cross-correlation computation, thereby reducing the measurement frequency. We conducted extensive experiments over different tone lengths and found that using a tone length of 512 sound samples (corresponding to 46.4ms at a 11.025kHz sampling rate) achieves good accuracy and stability (see Figure 2.5), while still allowing us to keep the measurement frequency above 10Hz. The tone of 46.4ms is similar to that used in [107].

Buffering Delay Minimization. The buffer is pushed to the application layer whenever it is full. The buffering delay is therefore a random period in $(0, B)$. To min-

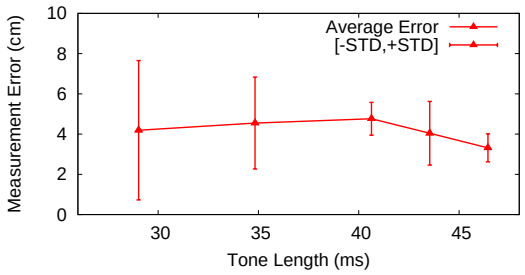


Figure 2.5: Impact of Different Tone Length. A tone length of 46.4ms has good accuracy and low variance.

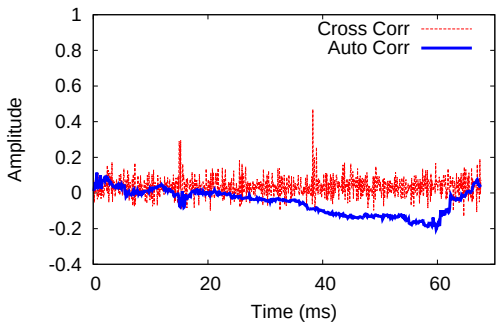
imize this delay in FAR, we use the phone's minimum buffer size (which is 46.4ms (1024Bytes) on Nexus One).

Networking Operation. We use UDP over WiFi, adding a delay of around 2-4ms per exchange.

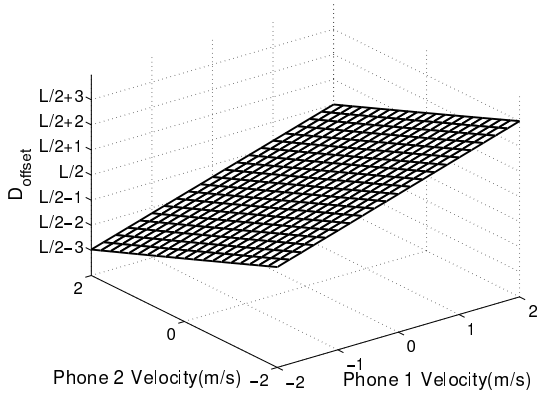
Tone Overlapping. One attempted optimization that we were unable to execute in practice. Instead of spacing the two phone's tones over time, we tried to overlap them in time, but send them on different audio frequencies. Unfortunately, this approach failed because the local speaker's high power masks the remote phone's signal, which generates detection errors.

2.6 System Robustness

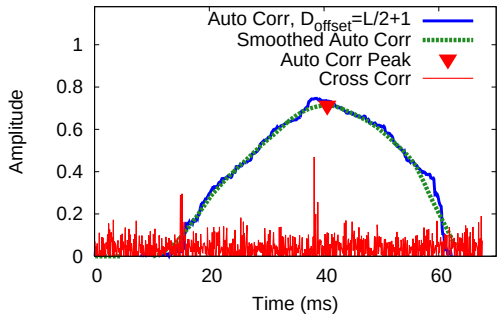
Besides measurement frequency and accuracy, another important challenge we need to tackle is the robustness of the system. A real-time motion game can not be fun if there are frequent measurement errors. In this section, we tackle the possible vulnerabilities a practical system could suffer in real game play environments.



(a) Doppler Effect Example.



(b) Phone Velocity *vs.* D_{offset}.



(c) Peak Recovery by Changing Offset.

Figure 2.6: Doppler Effect and Recovery. (a) The autocorrelation peak is destroyed by the Doppler effect. The interval between the two cross-correlation peaks is $L/2+1$ rather than $L/2$. (b) Given player speed is limited, L_{recv} falls into a limited range. (c) Changing the offset of the autocorrelation to L/2+1 can detect the peak.

2.6.1 Mobility Robustness

One unique challenge to distance estimation of motion gaming is the Doppler effect caused by player movement. Doppler effect happens when there is a relative movement between the sound player and recorder. Intuitively, when two phones are moving towards each other, the sound wave arrives at the recorder earlier than

expected, so it appears *compressed*. However, since the recorder is recording the sound at a constant rate, the recorded sound samples are fewer than expected, which means in the recorded version of the tone, the repeating pattern has a shorter length. Thus, the offset used in the autocorrelation calculation needs to be shortened. Similarly, when the phones are moving further away, the tone is *diluted*, and thus a longer offset is needed.

We define the term *Doppler offset* D_{offset} as the number of samples between the two cross-correlation peaks i.e. the locations of the repeating patterns. In a scenario without Doppler effect, the Doppler offset should be equal to the repeating pattern's length, which is exactly half of the tone length, $D_{offset} = L/2$, and the autocorrelation of $X(t)$ and $X(t - L/2)$ should be calculated to detect the tone. If Doppler shift occurs, the repeating pattern will be separated by a D_{offset} not equal to $L/2$. In these cases, the autocorrelation of $X(t)$ and its delayed stream $X(t - D_{offset})$, which is $R(D_{offset}, t)$, should be calculated to find the tone.

In Figure 2.6(a) we show an example of a dilution, in which the tone does not show an autocorrelation peak. A careful diagnosis uncovers that the offset is $L/2 + 1$ rather than the expected $L/2$. This means that an extra sample has appeared in the tone because of Doppler sound dilution. Thus $D_{offset} = L/2 + 1$ is appropriate in this case.

Assume Phone 1 is sending the tone, Phone 1's velocity is v_{p1} (relative to earth), Phone 2's velocity is v_{p2}, Tone Length is L (so the repeated pattern length is $L/2$), and c is the speed of sound. The delay between the repeating pattern of

received sound sequence D_{offset} can be expressed as

$$D_{offset} = \frac{c + v_{p2}}{c + v_{p1}} * L/2.$$

Figure 2.6(b) shows the relationship between D_{offset} and the relative velocity of the two phones. We assume the maximum speed of a player's hand is 2m/s, as we have observed from our experiments that a higher speed is extremely hard to reach. This is in accord with previous work [71] on human hand functions. Since this speed is not negligible compared to the speed of sound ($c = 343.2m/s$), we can see from the figure the possible range of the Doppler offset as $D_{offset} \in [L/2 - 3, L/2 + 3]$.

Recovery from Doppler Effect with parallel autocorrelators. To recover the correct autocorrelation peaks, a simple method is to calculate the autocorrelation with the appropriate offset. As an example, in Figure 2.6(c), we show that we can recover the peak by calculating $R(L/2 + 1, t)$, rather than $R(L/2, t)$. Given that $D_{offset} \in [L/2 - 3, L/2 + 3]$, we can set up a maximum of 7 parallel autocorrelators to catch any practical case of Doppler shift.

Parallel autocorrelators have both advantages and disadvantages. The advantage is that more autocorrelators increase the detection ratio. The disadvantage is that more autocorrelators lead to longer computation time. Therefore, one problem is to find the optimal number of autocorrelators $\hat{x}$ that maximizes the tones detected per unit time for a given velocity v, i.e.,

$$\hat{x} = \operatorname{argmin} x \left[D_v(x) \max(b, T(x)) \right],$$

where b is the communication bound, $D_v(x)$ is the detection ratio with x autocorrelators at velocity v, and $T(x)$ is the completion time for x parallel autocorrelators.

Unfortunately, each additional autocorrelator causes an additive increase in computation time. In response, we apply a *predictive Doppler estimate procedure*. The idea is to predict the likely Doppler shift based on the recent history of distance measurements, and only execute the autocorrelators likely to match the predicted velocity shift.

Figure 2.8 shows a fast movement case with a maximum player hand speed of 2m/s (and thus phone-to-phone relative speed reaches 4m/s). We show the two phones' distance change over time, and mark the tones affected by the Doppler effect. The Doppler-shifted tones are divided into two groups: diluted cases, and compressed cases. As evident from the figure, compressed cases occur when the distance is decreasing, and diluted cases happen when distance is increasing. Exceptions to this rule happen rarely.

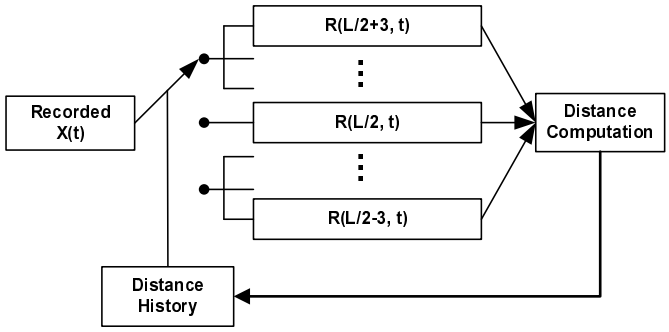


Figure 2.7: Predictive Parallel Autocorrelators. The distance history is leveraged to predict whether sound will be diluted or compressed, and subsequently which group of autocorrelators to use.

Thus, we can use two groups of autocorrelators to deal with the two groups of Doppler effect cases: one group of autocorrelators $R(L/2-1,t), R(L/2-2,t), R(L/2-3,t)$ for compressed cases, and the other group of autocorrelators $R(L/2+1,t), R(L/2+2,t), R(L/2+3,t)$ for diluted cases. For each buffer of

recorded samples, we always run the original autocorrelator $R(L/2, t)$. At the same time we predict the players' relative velocity based on the recent distance change history, then apply the appropriate group of autocorrelators, as shown in Figure 2.7. We can reduce the total overhead by nearly half with this method. In fact, our evaluation suggests that almost all of D_{offset} are within $[L/2+2, L/2-2]$, and thus only 3 parallel autocorrelators are needed to recover from the vast majority of Doppler shifts.

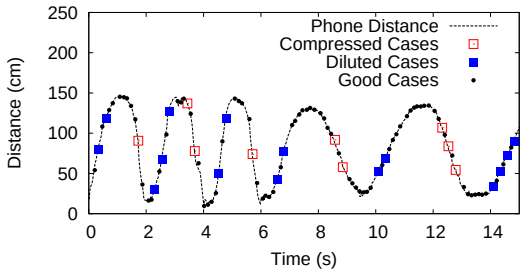


Figure 2.8: Doppler effect causes sound dilution and compression errors. However, we are able to predict these errors based on recent velocity.

The prediction based method might not be correct when the player is changing direction. However, the user's speed naturally slows when changing direction, which will decrease the likelihood of experiencing Doppler shifts. As shown in Figure 2.8, almost all of the tones at the peaks or troughs are good cases.

2.6.2 Environmental Robustness

There are several environmental factors which can impair distance measurements. We address these presently.

Robustness to Ambient Noise. To study the character of ambient noise, we recorded sound from both indoor and outdoor noisy scenarios. One finding is

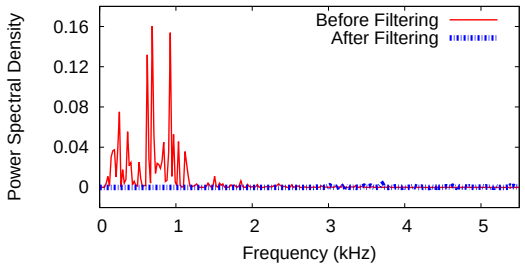


Figure 2.9: Impact of Filtering. The frequency map of a recording of ambient sound from a crowded area before and after filtering.

that the frequency for these sound usually is below 2kHz. Thus, in our system we apply a high pass filter (a 9-th order Butterworth filter) to filter out ambient noise. Figure 2.9 shows the spectrum density map of a 1-minute recording of a crowded place, before and after applying our filter respectively. It is clear that the high pass filter functions well in eliminating ambient noise.

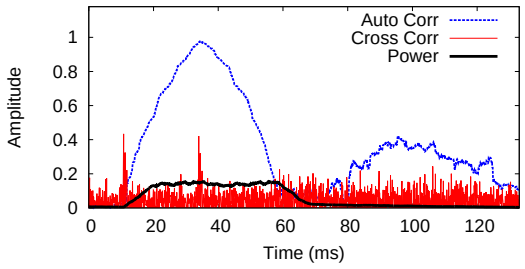


Figure 2.10: Impact of Multipath Components. A multipath component generates a smaller autocorrelation peak (at around 100ms) after the correct peak, but with a much lower power level.

Robustness to Multipath. Multipath effects can also impact the robustness of tone detection if the players are in a relatively small room or hallway. The reflected signal may generate correlation peaks or degrade the main component. Figure 2.10 shows a tone with an observable multipath effect. There is a lower autocorrelation

peak at around 100ms which is caused by the multipath components. We observe that the multipath components usually have low power and signal quality after reflection or scattering. Note that the power threshold might not work well if there is high ambient sound. However, after applying the high pass filter, we can efficiently reduce the ambient sound power level. In our implementation, we empirically set the power threshold to 0.02 and peak level threshold to 0.4, which we show to be sufficient to filter out the multipath components experienced when playing in actual scenarios. In our evaluation section, we conduct experiment in a small room to show the performance.

Robustness to Occasional Tone Loss. `FAR` also provides failsafe methods to deal with occasional tone losses. Since the distance calculation requires the tone measurement from both phones' computation result, it is important to keep the two measurements synchronized. In particular, if a tone is not detected, the phone should be alerted of the situation. We achieve this by a *binary coding* method: we assign a unique pseudorandom sequence for each phone. If detection is flawless, each phone will see alternating codes. Thus if a phone receives any two consecutive tones whose codes are the same, the phone detects a missed tone condition and accounts for this in the distance calculation.

Coding beyond binary (e.g. ternary) is not necessary, because multiple consecutive missing tones are detected by simple threshold on the arrival interval between two consecutive tones. When such cases arise, the system will enter *Failsafe mode*. In Failsafe mode, one phone will send two tones separated by a much larger interval, e.g. 250ms. Both phone will then observe the large interval and reset their state.

2.7 Game Design

In this section, we briefly describe two novel games we have developed on FAR to date.

SwordFight. In SwordFight, the core gameplay works as described in the introduction. In order to ensure that attacks are oriented toward the opponent, the digital compass is used. At the start of a fight round, the players are instructed to point their phones toward one another and the phones' digital compass orientations are recorded. During the fight round, the digital compass is read to ensure that movements are actually oriented toward the opponent.

ChaseCat. ChaseCat is a fast round-based game. In each round, one player is randomly chosen as the Cat and the other player is chosen as the cat chaser, the Dog. The Dog player attempts to get her phone as close to the Cat player's phone as possible, while the Cat player tries to get her phone as far from the Dog player's phone as possible.[2] A random 15-25 seconds after the round starts, players are instructed to Freeze by an audio cue. If the Cat is within 30 centimeters of the Dog during Freeze, then the Dog catches the Cat and scores a point. Otherwise the Cat escapes and scores a point. Movement during Freeze is detected by distance and acceleration measurements.

Note that in the case of both games, distance measurements are expected to be accurate up to several centimeters. Our evaluation results in the next section show that FAR is accurate at this resolution. The maximum distance that can be supported is a function of the maximum volume of the phone speakers and the sensitivity of microphones. The current platforms we use happen to have

[2]We recommend players plant their feet during the game, but this has proven difficult to enforce.

an effective measurement range of 2 meters, so this is the current effective limit of gameplay. Out-of-range occurrences can be detected by sequences of dropped tones and low SNR.

2.8 Implementation

We implement both our games on two phones: Nexus One running Android 2.3.4 and Samsung Focus running Windows Phone 7.5.

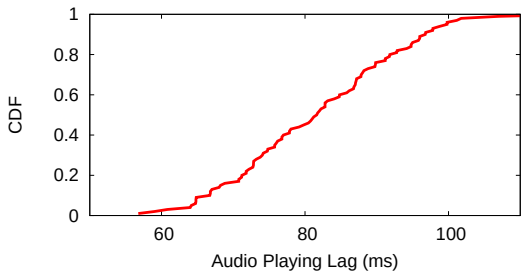


Figure 2.11: Audio Playing Lag Distribution

Randomness of Audio Playing Lag. As mentioned in §2.5.2, the Streaming Execution Strategy mitigates Audio Playing Lag. However, it turns out that the commodity devices we have encountered exhibit an additional random audio playing lag artifact. Figure 2.11 shows that the audio playing lag is randomly distributed between 64ms (5%-tile) and 99ms (95%-tile). Thus, to prevent two consecutive tones from overlapping with each other, we add a 35ms *guard period* when playing each tone. The interval between two consecutive `Play()` calls is then $L+35ms = 46.4ms+35ms = 81.4ms$. In fact, given our reductions in computation and communication overhead which are no longer the bottleneck, it is this ran-

dom system artifact that currently prevents us from increasing the measurement frequency from $1/81.4ms \approx 12Hz$ to a theoretical $1/L = 1/46.4ms = 21.6Hz$.

2.9 Evaluation

We demonstrate through our evaluation that FAR is fast, accurate and robust, and that it is suitable for MMG games. The primary metrics are as follows.

- *Measurement Frequency.* Is FAR efficient enough to support fast measurement? How fast can FAR make measurements?
- *Measurement Accuracy.* How accurate are the measurements?
- *Measurement Robustness.* How well does FAR handle noise and motion?
- *Gameplay.* Do games based on FAR work in practice?

We conducted experiments with two Nexus Ones in a room measuring 14m by 7m by 2.5m that typically serves as a conference room. Static measurements are verified against manual ruler measurements. For dynamic measurements, the main comparable system used is the Xbox Kinect, a dedicated fixed-infrastructure motion tracking system for console gaming. Lastly, as real world validation, we invited public users to play SwordFight and ChaseCat on four separate occasions. The highlights of the evaluation are as follows.

- FAR achieves a measurement frequency of at least 12 Hz. In fact, FAR is fast enough to reach the limits imposed by the phone's audio driver.
- The median measurement error is 2cm. During times of rapid movement, FAR compare favorably to Kinect for both isolated player measurements and in-situ gameplay player measurements.

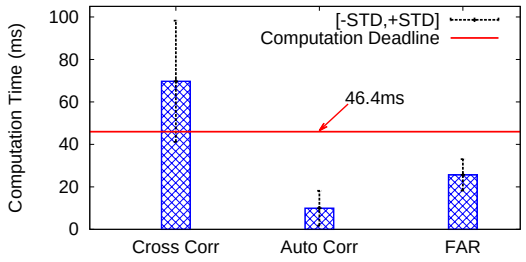


Figure 2.12: Compute time needed to process a 512-sample audio recording.

- Doppler effects, high noise and multipath and are effectively neutralized.
- Experiences of over fifty game players suggest that `FAR` is a good match for motion gaming, and that SwordFight and ChaseCat are fun to play.

2.9.1 Measurement Frequency

We show that `FAR` is efficient enough to take measurements as fast as the phone audio driver can support. We evaluate `FAR`'s feasibility in supporting the pipelined streaming execution strategy, and then report on the achieved frequency.

Feasibility. The fundamental limitation of previous approaches is that the detection of tones requires intensive computation, and thus is not feasible within a pipelined streaming execution strategy bound by tight deadlines. Figure 2.12 illustrates that `FAR` is able to comfortably satisfy the strict deadline of 46.4ms corresponding to a recording of 512 samples. The traditional cross-correlation method is by far the slowest at 70ms.[3] Missing the computation deadline incurs serious buffer overflow problems, leading to loss of sound samples and thus measurements.

[3]Note that this is much smaller than as claimed in [114] because we only process a 512-sample recording, while [114] requires a much larger recording due to its traditional execution strategy.

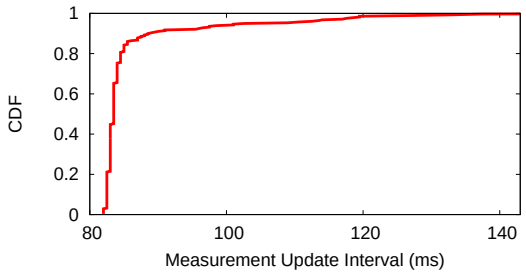


Figure 2.13: Measurement Update Distribution. More than 80% of the time, a new update arrives within 84ms of the previous update.

Measurement Frequency. Figure 2.13 shows the distribution of measurement updates for over 400 individual measurements as generated by `FAR`. The effective measurement frequency is 12Hz over 80% of the time. As noted in §2.8, a random audio playing delay of [64ms,99ms] inhibits us from achieving a theoretical frequency of 21.6Hz.

Measurement Lag. For each distance measurement, the lag is comprised of the following components: *Tone Length, Propagation Delay, Buffering Delay, Computation Time* and *Measurement Exchange Delay*. For motion gaming, the distance between two phone does not exceed 2 meters. Therefore, the *Propagation Delay* is less than 6ms. Figure 2.14 shows the lag of 400 distance measurements when the phones are placed 50cm apart. The measurements are sorted by their total delay. With the exception of the last 5% of cases with large network delays, the majority of measurements exhibit a total lag of around 100ms. As expected, the delay due to the length of the tone is constant, and the network delay and propagation delay are very small. The buffering delay is uniformly distributed in (5ms, 46ms).

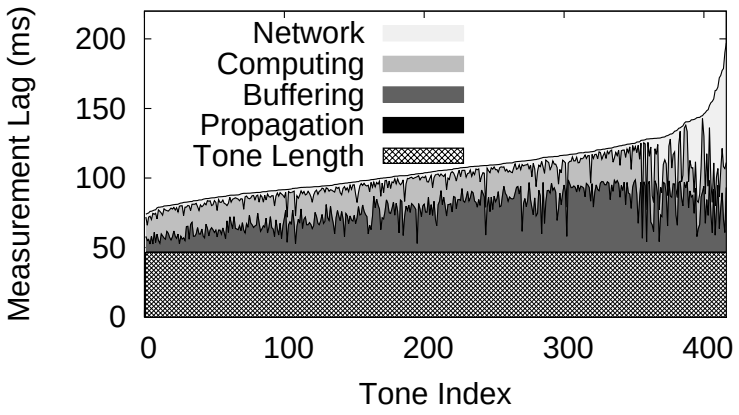


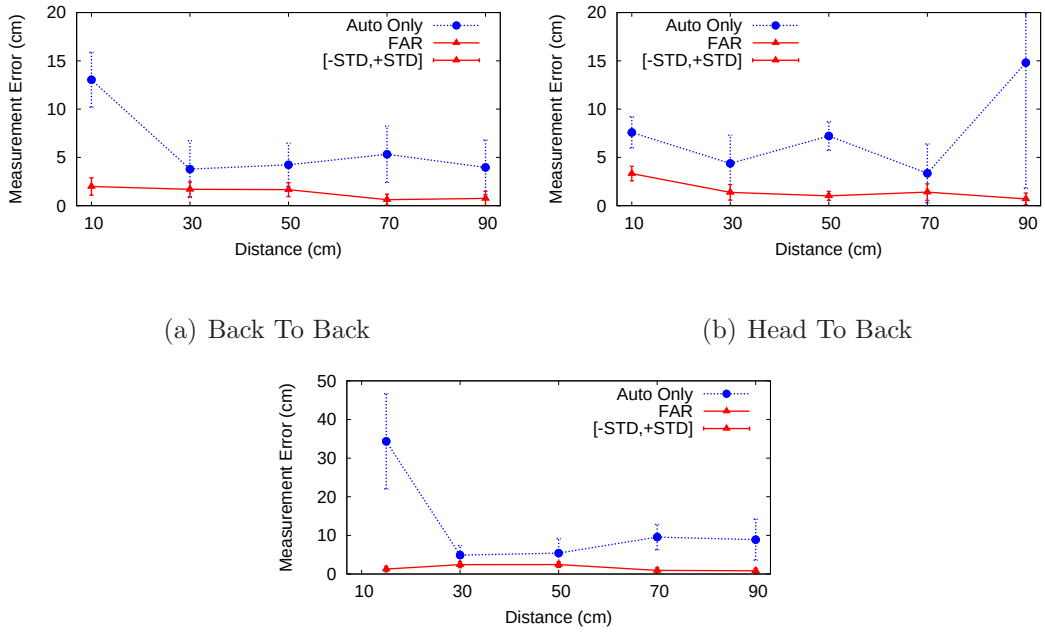
Figure 2.14: Measurement Lag Composition. Several individual delays contribute to the time between when a tone is played and when a distance measurement is completed.

2.9.2 Measurement Accuracy

Next we juxtapose the measurement accuracy of autocorrelation (Auto Only) and `FAR`. We statically position the two phones in three typical phone-to-phone orientations at different distances. Figure 2.15 illustrates the measurement accuracy of the two methods, each computed from over 400 experimental cases. It is clear that autocorrelation exhibits measurement error that is severe, whereas `FAR`'s error is very low with a median of 2cm and standard deviation of 1cm. The figure also highlights that `FAR` is robust across varying phone orientations and distances.

2.9.3 Impact of Player Movement

We next investigate whether our system is capable of accurate measurements while players are moving. After exploring several possibilities including a studio motion capture system, we settled upon Kinect, an off-the-shelf motion capture

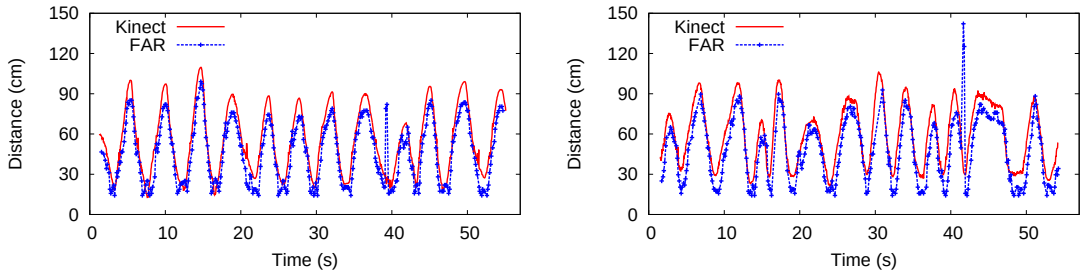


(a) Back To Back

(b) Head To Back

(c) Head To Head

Figure 2.15: Measurement Accuracy in a Large Room.

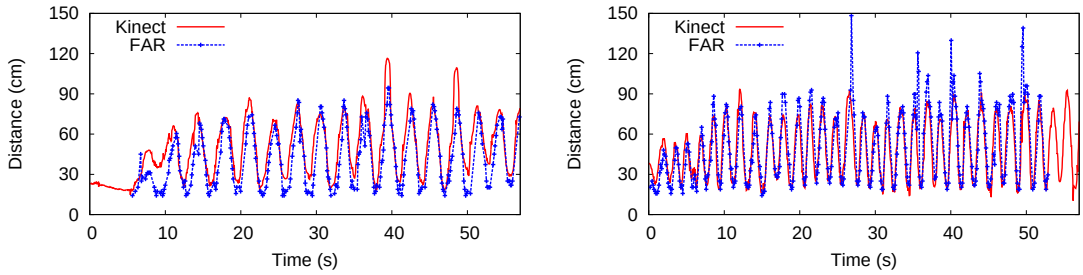


(a) Experiment Case 1

(b) Experiment Case 2

Figure 2.16: Isolated Player Measurements. Simultaneous distance measurements of `FAR` and Kinect when a single player is moving her hands freely with a phone in each hand.

system for the Xbox console gaming system, as the candidate comparable system.



(a) Experiment Case 1

(b) Experiment Case 2

Figure 2.17: In-Situ Gameplay Measurements. Simultaneous distance measurements with FAR and Kinect of players engaged in SwordFight.

The reason Kinect is appealing is because it is commercially available, practical to use, and has already proven successful for infrastructure-based console gaming.

Kinect measurements were obtained with the Kinect SDK which is a Kinect-to-computer interface that allows programmatic collection of the human skeleton coordinates as seen by the Kinect camera. We employed both isolated player experiments and in-situ real player experiments.

Isolated Player Movement. In isolated player experiments, we positioned the Kinect camera statically, and placed a subject in front of the camera with a phone running FAR in each hand. The subject was then asked to vary her hand position at her discretion. FAR performed phone-to-phone distance measurements while Kinect independently tracked the hands.

Figure 2.16 illustrates the distance measurements simultaneously measured by Kinect and FAR for two representative traces. For the most part, FAR matches the Kinect motion path closely. In both traces, there are only 5 measurements out of 700 that deviate significantly from the Kinect estimate. In some instances, Kinect and FAR measurements can be offset from one another by up to 12cm.

The reason for this is actually a reflection of the Kinect SDK, which can exhibit 5-15cm variance of the reported hand coordinate because it is ambiguous as to whether it measures distance from the base or tip of the hand. While an ideal ground truth would not have such issues, the results are encouraging enough to suggest that `FAR` compares favorably to a commercially successful motion gaming system.

In-Situ Gameplay Player Movement. In the in-situ gameplay experiments, six players were invited from the UC Santa Barbara Link Lab to conduct nine in-situ rounds of measurement. Players were each given a phone and asked to try out the SwordFight MMG game. During the game, `FAR` provided the in-game distance measurements while Kinect independently recorded their skeletal hand positions.

Figure 2.17 compares the corresponding `FAR` and Kinect measurements for two representative traces. There are occasionally slightly more mismatches between `FAR` and Kinect during in-situ gameplay than during isolated player experimentation. This is due to the fact that players – in the heat of competition – tend to grip or orient the phones such that mics or speakers may be temporarily blocked.

Interestingly, we have found that our system can in fact be more accurate than Kinect because Kinect cannot accurately track the hand when the player's hand is obstructed from the camera's view (e.g. by her own body or the opponent's body). These situations happen frequently in SwordFight and similar games. For example, two right-handed players facing off will often mean one player is poorly oriented toward Kinect. In these situations, `FAR` measures distance better because there are typically no obstructions between the two phones in the players' hands for games in which opponents are facing each other. In the in-situ experiments

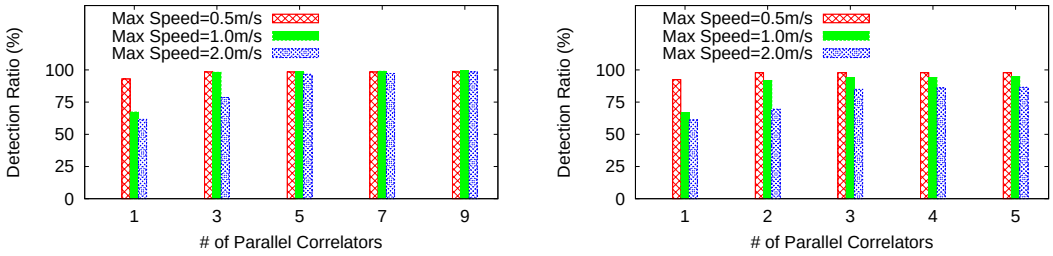
presented in Figure 2.17, we asked that the players compensate for this Kinect limitation for the sake of experimentation: one player was instructed to play with the left hand, the other player was instructed to play with the right hand, and both players were instructed to not obstruct the line-of-sight between the Kinect and their phones. These patently artificial constraints on gameplay highlight why FAR may be potentially even more suitable than fixed infrastructure measurement systems for certain styles of motion gaming, regardless of infrastructure availability.

2.9.4 Measurement Robustness

We quantify FAR's effectiveness at handling three sources of measurement error: Doppler effect, multipath and noise.

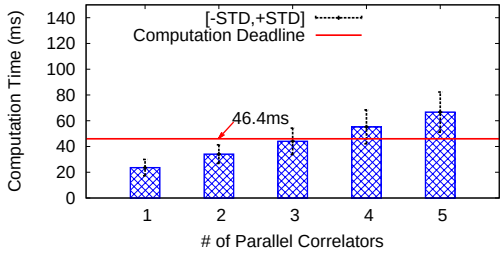
Robustness to Doppler Effect. FAR detects Doppler shifts with the use of parallel autocorrelators. Our experimental setup consisted of recording FAR and Kinect measurements simultaneously while instructing the players to swing hands at varying rates, including as fast as possible. We collected fifteen motion traces, each consisting of over 400 distance measurements. We calculated a phone-to-phone relative velocity from the Kinect data (which is not affected by Doppler shift). We group motion traces according to the maximum velocity reported by Kinect.

Figure 2.18(a) and Figure 2.18(b) show the detection performance of parallel autocorrelators (without prediction and with prediction respectively) for representative traces with motion speeds of 0.5m/s, 1m/s and 2m/s where the Doppler shift does occur. As expected, even one autocorrelator performs reasonably for



(a) Detection Ratio w/o Prediction

(b) Detection Ratio w/ Prediction

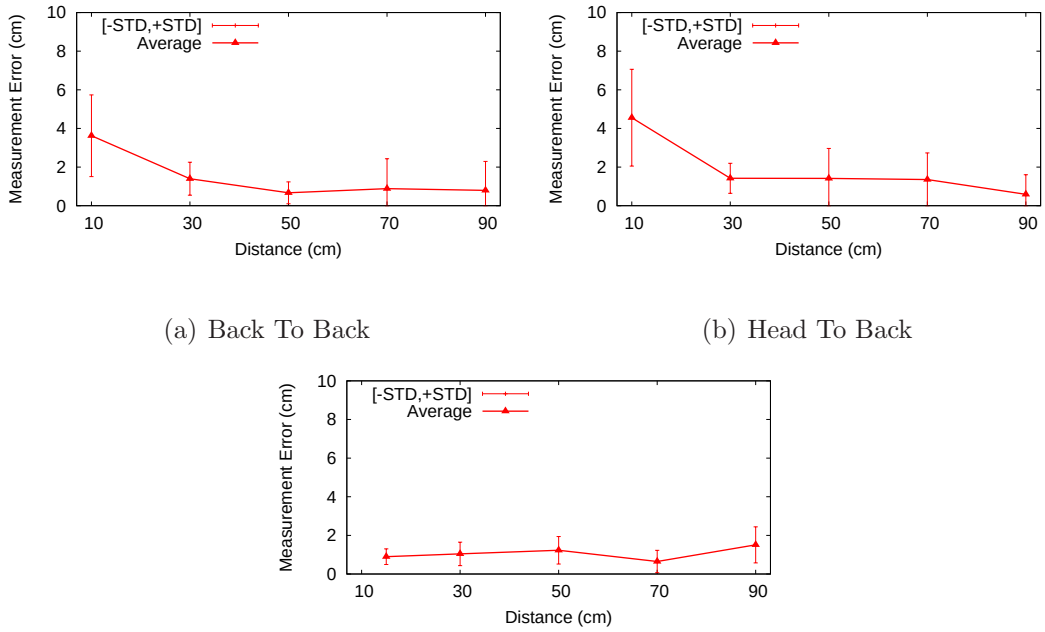


(c) Computation Time of Autocorrelators

Figure 2.18: Parallel Autocorrelators. (a) 90+% of the tones can be detected by five parallel autocorrelators, which corresponds to offset range $[L/2-2, L/2+2]$ (b) With predictive parallel autocorrelators, only three autocorelators are needed. 85+% of the tones can be detected even at 2.0m/s hand movement speed. (c) Three parallel autocorrelators can still meet the computation deadline.

0.5m/s but more autocorrelators are needed as the velocity increases. Most of the tones affected by Doppler shift can be recovered by five parallel autocorrelators as shown in Figure 2.18(a). This means the Doppler offset is usually within $[L/2-2, L/2+2]$.

Figure 2.18(b) shows that after applying prediction, only three parallel autocorrelators can recover most of the errors, and achieve 85+% detection ratio even in a high speed scenario of 2m/s. The detection ratio gain for more than three autocorrelators is marginal. Figure 2.18(c) shows the computation time required



(a) Back To Back

(b) Head To Back

(c) Head To Head

Figure 2.19: Accuracy In A Small Room. The impact of the reflection is not noticeable.

for increasing numbers of parallel autocorrelators. Three correlators can be run simultaneously while maintaining or just barely exceeding the computation deadline required for the fastest measurement frequency. Our empirical observations indicate that sustained velocities of 2m/s or even 1m/s are very hard for players to maintain due to physical human limits; there is a natural opportunity to detect and complete measurements after sudden bursts of fast movement. Therefore, `FAR` uses three parallel autocorrelators which work well in practice.

Robustness to Multipath and Ambient Noise. To test robustness to reflected sounds, we ran `FAR` in the smallest contained space we could find, a private room measuring 3m by 2m. Figure 2.19 illustrates the performance in the small

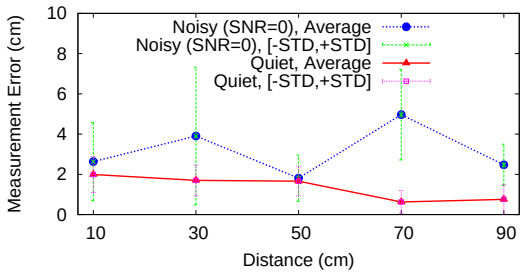


Figure 2.20: Measurement accuracy when injecting heavy noise (SNR=0). The accuracy drops a little but can still achieve an accuracy of 5cm in average.

room with various phone orientations and distances. While the variance of measurement error is 5cm and larger than in the original room, the measurement error is again very small, with a median error of 2cm.

We injected an ambient sound to evaluate the anti-noise property of `FAR`. In order to ensure repeatability, the ambient sound is a 1-minute recording from a very crowded area, with people talking and laughing loudly. We define the SNR of the signal as the energy of the sound recorded by the remote phone over the energy of the ambient sound. While we tested over different scenarios, we report only the performance of the back to back position result with SNR=0 here in Figure 2.20 due to the similarity of the results. As expected, the ambient sound degrades the accuracy, but the impact is manageable. Recall that SNR=0 means a noise energy level equal to the tones', which is high.

2.9.5 Deployment Experiences

We have prepared SwordFight and ChaseCat for public play on multiple occasions: at the SenSys 2011 conference for any conference participant [154]; in Beijing for interns working at the Microsoft Research Asia building late at night;

in Redmond for office colleagues taking a work break, and; at UC Santa Barbara for officemates, friends and family. Qualitatively, the public reception has been positive. Just as the Wii and Kinect ushered in a new class of console gameplay, SwordFight and ChaseCat introduce an element of physical action to mobile gaming that players seem to enjoy.

2.10 Related Work

Motion Gaming. Recently, non-mobile console gaming systems have embraced player motion-based gameplay, developing a variety of schemes for player localization. The Wii remote [100] uses an infrared camera to track the relative position of the TV-mounted sensor bar which emits two infrared light sources. The PS3 Move controller [138] has a light-emitting ball affixed which is tracked by a TV-mounted video camera. The TV-mounted Kinect [95] projects an infrared mesh and tracks its time-of-flight back to the Kinect camera. These console gaming systems demonstrate the compelling nature of motion gaming, but are unfortunately tethered to fixed console-based infrastructure.

The idea of phone-to-phone motion gaming was first mentioned in the recent work of Qiu et al. [114], described in that paper's introduction as high-speed, locational, phone-to-phone (HLPP)-gaming. [114] was the first to propose phone-to-phone localization as a primitive for mobile gaming. The authors developed a 3D acoustic localization system that utilizes each device's two microphones, one speaker, 3-axis accelerometers and 3-axis magnetometers to calculate coordinates of one phone in three dimensions relative to the other. However, the scheme proposed in [114] is not actually applicable to high-speed motion games such as

SwordFight since the reported update interval is over 800ms. Nevertheless, their 3D positioning algorithm may in the future serve as a complement to the high-speed ranging techniques presented in this work.

Acoustic Ranging. The authors of [107] developed an earlier method for acoustic ranging between two phones but its update rate can be arbitrarily slow since the two devices are not synchronized, and the applications are unclear.

Much research exists on infrastructure-based localization algorithms (see [82] for a survey). ActiveBat [57] and Cricket [112] are two representative infrastructure-based systems. Each is capable of achieving centimeter resolution, but dense ultrasonic infrastructure requirements are not compatible with ad hoc mobile gaming. Infrastructure-based acoustic localization includes [35, 89, 133]. These systems achieve resolution in meters, which is insufficient for action gaming. Several systems have reported achieving centimeter resolution [53, 59, 78] with custom-built hardwares which makes them less appealing in light of the pervasiveness of commodity mobile phones.

Components of our algorithm's signal processing techniques have been used previously in the communication systems but not in the context of localization. [132] proposed using autocorrelation to reduce the complexity of frame and carrier synchronization. [93] uses a technique similar to running multiple parallel detectors for handling Doppler effects underwater, but does not consider predicting Doppler shifts.

2.11 Discussion and Conclusion

In this chapter, we report on the design and implementation of a new class of phone-to-phone games: Mobile Motion Games (MMG). The key underlying technology is a new highly-accurate and real-time phone-to-phone distance measurement substrate, which provides the API on top of which games such as SwordFight and ChaseCat are built. Our experience with having people play these games is positive. They are intuitive and fun to play. However, our work here is a starting point. There remain open challenges that could affect the game experience. One issue is that continuous acoustic tones are noticeable, since mics and speakers on commodity phones only support the audible frequency range. A possible approach is to embed these tones in game music. A second issue is that line-of-sight blockage between the phones degrades the measurement accuracy, and players can accidentally or purposefully block the mics and speakers. Future work includes creation of a blockage warning or cheat detection protocol. In addition, we are interested in extending the API to permit simultaneous ranging between more than two phones. Lastly, we are working on designing and prototyping additional MMG games on top of our phone-to-phone gaming API.

Chapter 3

Accurate Wireless Transmitter Localization

3.1 Introduction

WiFi networks today are ubiquitous in our daily lives. WiFi access points (APs) extend the reach of wired networks in indoor environments such as homes and offices, and enable mobile connectivity in outdoor environments such as sports stadiums, parks, schools and shopping centers [12, 19]. Even cellular service providers are relying on outdoor WiFi APs to offload their 3G traffic [19, 20]. As Internet users become reliant on these APs to connect their smartphones, tablets and laptops, the availability and performance of tomorrow's networks will depend on well tuned and managed access points.

A critical part of managing access points is the ability to locate individual access points based on their signal [81, 104, 123]. Doing so allows network administrators to identify APs causing excessive interference to others, or unauthorized APs that may provide easy entry for malicious attackers [111]. For individual users, it allows them to locate and get closer to neighborhood WiFi hotspots,

build war-driving databases, or pinpoint neighboring APs so they can better position their own basestations.

Unfortunately, much of this is not possible today, because current techniques to locate outdoor WiFi access points require either extensive wardriving measurements, followed by significant offline computation [55], or complex hardware components such as steerable directional phased array antennas [140]. For the wardriving scenario, locating each AP requires measurements from a large number (35+) of locations, making the process extremely time and energy intensive. For the directional antenna solution, the specialized hardware components cost several thousand dollars each, which clearly limits their availability to only a small portion of system administrators for large enterprises.

In this work, we ask the question: *"is there a cost- and time-efficient alternative to perform accurate outdoor location of WiFi access points?"* A potential solution using common-off-the-shelf (COTS) hardware would break access point location out of a small niche market of enterprise administrators, and make it available to home users and small businesses managing their own local hotspots.

Our solution is derived from a key insight: *"by rotating a standard wireless receiver around a blocking object, we can effectively emulate the sensitivity and functionality of a directional antenna."* We exploit the property that the signal strength observed by a wireless receiver drops most significantly when there is a large obstacle directly between it and the transmitter. Such a drop in signal strength is strongest when the receiver is directly adjacent to the obstacle, and should be observable as long as the obstacle is large enough to block a significant portion of the reception angle. Therefore, by "rotating" the receiver's position with respect to the obstacle (and the signal "void" it produces), and observing

the received signal strength, we can determine the approximate direction of the transmitter. We refer to this process as *directional analysis*.

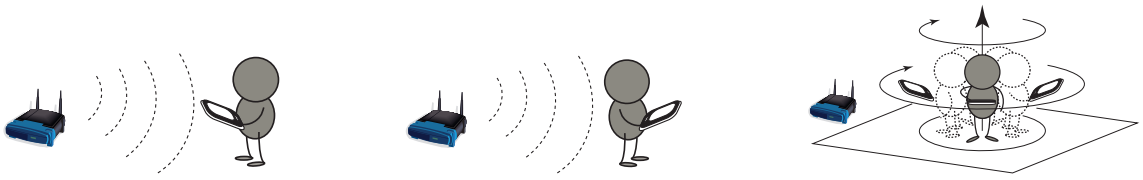
While this is a general observation potentially applicable to a variety of wireless transmissions on different frequencies and hardware, we focus our attention in this work on a single use case: outdoor location of WiFi access points using smartphones. Applying our insight, we hypothesize that a user can accurately locate WiFi APs using common-off-the-shelf smartphones as receivers, and her own body as the blocking obstacle. To perform a directional analysis operation, she slowly rotates her body around 360 degrees, while keeping the smartphone in front of her and performing periodic received signal strength (RSS) measurements. The observed RSS should be at its lowest point when the user's body is directly between the smartphone and the wireless AP. By walking towards the predicted direction of the AP, and periodically repeating the directional analysis, a user should be able to zone in and locate any specific AP.

To validate our hypothesis, we first perform detailed experiments to see if such dips in signal strength can be observed using today's smartphones and standard WiFi access points. We run tests on several different environments on the UC Santa Barbara campus, target both 802.11b/g and 802.11n (MIMO) APs, and use three different smartphone platforms, Android, WindowsMobile, and Apple iPhone4. Using a large number of real measurements, we do in fact observe the expected dip in signal strength when the tester's body blocks the smartphone from the access point. We consistently observe the artifact across both platforms, environments with different levels of obstructions, and for users of different heights and weights.

Borealis. These measurement results lead us to develop *Borealis*, a system for locating outdoor WiFi access points using software on commodity smartphones. Borealis users can perform robust directional analysis by turning their bodies on a 360° axis, and use this technique to locate a transmitting access point. We address several challenges in the process of building Borealis, most significant of which is that errors in directional analysis are impacted by environmental conditions, and particularly exacerbated by multipath propagation around areas with multiple buildings. Our solution is to build a model that predicts the impact of blocking obstacles on signal strength at the receiver, use it to identify large blocking effects, and in doing so, produce more accurate predictions of the AP's direction. Combined with prior techniques on direction-guided user navigation [54, 75, 76], we produce a system that efficiently guides users to an AP. In scenarios where the AP is housed indoors, our system guides the user to an outdoor location closest to the AP.

We implement and deploy Borealis as a user application and a set of kernel modifications on the Android platform. We modify the WiFi driver to focus scans on specific channels, thus allowing more frequent RSS measurements while reducing energy usage. Our user application assists the user in directional analysis by logging each RSS measurement along with data from the orientation sensor, and using it to compute the most likely direction of the access point along with a prediction confidence. The user can repeat the process with success until she is within 2 meters of the AP.

We perform detailed experimental evaluation on Motorola Droid and HTC G1 phones. Our results show that Borealis produces significantly more accurate AP direction values than GUIDE [54], the recent system using triangle gradients to



(a) Facing AP. (b) Back facing AP. (c) The user rotating in place.

Figure 3.1: Illustrations of users facing the AP, with back to the AP, and rotating while holding a smartphone.

compute AP direction. We also find that running Borealis in real-time produces similar accuracy compared to an offline version using learning techniques on a large number of RSS measurements. Finally, our results show that using our direction predictions, Borealis leads the user on a path that reasonably approximates the shortest path to the AP.

Our work shows that with small software modifications, today's smartphones can effectively replace directional antennas to locate outdoor WiFi access points. Borealis is the first example of a potential class of systems that approximate directional antenna systems using rotation around signal-blocking obstacles, and its underlying principle can potentially be applied to build location systems for other wireless transmissions.

3.2 Preliminaries

In this section, we briefly describe the problem scenario and assumptions. We then summarize existing work that is most relevant to our target scenario.

Our focus is to accurately locate outdoor WiFi AP using common off-the-shelf smartphones as receivers. This functionality is a critical part of managing WiFi access networks for both small business and home users. The problem scenario is simple – a user, holding a smartphone, seeks to find the physical location of a WiFi AP from its BSSID. Note that our AP location problem is different from the general wireless localization problem [29, 37, 60]. The user seeks to locate a transmitting AP rather than determining her own location.

3.2.1 Related Work

In general, a receiver locates a transmitter by examining received signal strength (RSS), time of arrival (TOA) [39], time difference of arrival (TDOA) [62], or angle of arrival (AOA) [105]. The latter three methods all require simultaneous measurements at multiple receivers or antenna array. They are commonly used in cellular networks where neighboring basestations collaborate to locate a mobile device. They are, however, not feasible in our target scenario. Thus we focus our discussion on the RSS based method.

We categorize the RSS based solutions into three groups.

Model-based. Solutions in this category consider absolute values of RSS. Using RSS values measured at one [128] or multiple [48] locations, existing designs seek to derive the physical distance between the tester and the transmitter or the exact transmitter location, based on a radio propagation model. This solution, however, is fundamentally limited by the inaccuracy of propagation models for practical environments [60].

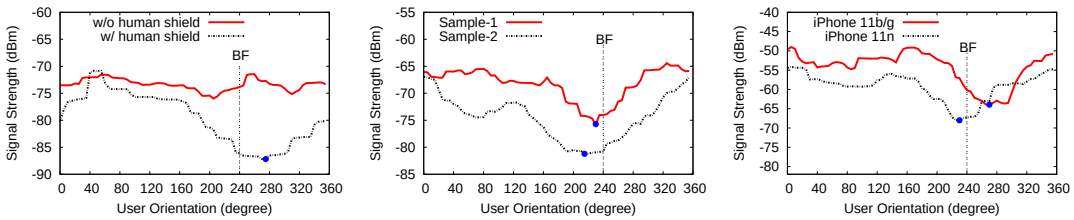
Gradient-based. This type of solution compares the RSS values obtained from different locations. Based on the assumption that a location closer to the AP will have a higher RSS value, existing designs estimate the AP direction by computing the gradient of the RSS value across different locations. This is done either online [54, 141] using a small set of local measurements, or offline [55] by integrating the results of a large number of measurements. The problem with this line of solution is its idealized assumption that RSS values increase as the receiver moves closer to the transmitter. Our own measurements show that this assumption often breaks in practice, leading to large errors.

Directional Antennas. The use of directional antenna, either at transmitter [97, 134] or receiver [75, 76, 99, 140], can significantly boost localization performance. For example, by rotating the beam of its antenna, a receiver can pinpoint the direction of the AP as the direction that provides the highest receive signal strength [140]. The drawback is that it requires specialized hardware.

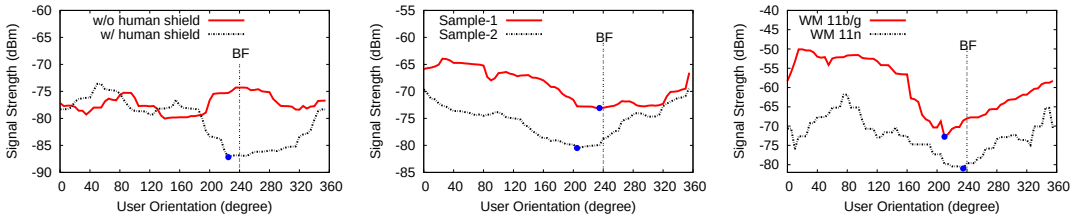
Among the above solutions, the directional antenna based solution achieves most of our system requirements: it is accurate, uses a single radio, and operates online with a small measurement overhead. However, directional antennas are prohibitively expensive for home users and not portable enough for personal use.

3.3 The Blocking Obstacle Effect

The key insight that enables our approach to locating access points is surprisingly simple. We hypothesize that when placed next to a large obstacle, the signal strength perceived by a wireless receiver will be highly dependent on where the obstacle is relative to the position of the receiver. The closer the obstacle is to



(a) Droid, Simple LOS (c) Droid, Complex LOS (e) iPhone, Simple LOS



(b) G1, Simple LOS (d) Droid, NLOS (f) Win Mobile, Simple LOS

Figure 3.2: Observed signal strength profiles with user rotation. We mark the user direction facing the AP as 60°, the direction with user's back to the AP as 240° (marked by BF). (a)-(b) When a user holds the phone with Droid and G1 phones in a simple LOS environment, the signal profile displays a clear low signal artifact. (c)-(d) The artifact is consistently visible in the complex LOS and NLOS environments. (e)-(f) The artifact is easily visible using iPhone 4 and Windows Mobile phones with both 802.11b/g and 802.11n APs.

blocking line-of-sight between the AP and the receiver, the more of the signal is blocked, and the weaker the signal strength seen by the receiver.

We apply this hypothesis to our context of smartphone based AP location. The body of a user holding a smartphone will block a portion of the incoming WiFi signal. The closer the user is to being on the straight line between the smartphone and the AP, the weaker the signal perceived by the phone. This blocking effect of the human body has been observed on a variety of frequencies and radio hardware [52, 124, 148], even in indoor environments [29]. Consider figure 3.1(a)-(b). When her back is towards the AP, the user's body becomes an

obstacle and blocks the signals from reaching the smartphone. When facing the AP, the body is no longer an obstacle. Therefore, we expect that as a user rotates herself in place, as in figure 3.1(c), the phone's received signal strength will display an interesting artifact: a peak when she faces the AP and a dip when her back faces the AP. Thus by measuring signal strength at different rotational angles, a user can gain a hint of which direction points towards the AP.

In this section, we verify our hypothesis using detailed smartphone experiments. Our experiments seek to study the impact the body as an obstacle has on signal strength as the user rotates herself. We seek to understand the factors that cause this artifact, and the impact on this artifact by factors such as propagation environment, phone hardware and WiFi standards.

3.3.1 Smartphone Experiments

Our experiments use four popular smartphones: Motorola Droid, HTC Dream/G1, Apple iPhone 4, and LG Fathom (WindowsMobile 6.5). The first two phones support 802.11b/g and the last two support 802.11b/g/n. We program each smartphone to poll its RSS reports, and record the received WiFi signal strength from a given AP. Using its built-in compass, each phone records its relative orientation[1] as the user rotates in a clockwise direction. To diminish the impact of fast fading, we smooth the RSS values using a moving average window of 20° along the measured angle.

Our experiments will answer five key questions:

a) *Is there a clear observable artifact*?

b) *What is the source of this artifact*?

[1]Relative to the initial direction the user was facing.

c) *Is it a function of the propagation environment*?

d) *Is it sensitive to different phone hardware or AP configurations,* i.e. *802.11b/g vs. 802.11n*?

e) *Is it sensitive to the height/weight of users, or how they position the phone with respect to their body*?

Experiment 1: Signal Patterns during Rotation. Our first experiments examine the signal strength artifacts during user rotation. At each test location, a user holds a smartphone in the hand and rotates her body by 360° at a speed of 6° per second. Each experiment records the instantaneous RSS values and the phone/user orientations. We set up the AP in an open space area where a direct line-of-sight path exists between the AP and the phone. In the rest of the chapter, we refer to this environment as Simple LOS. This basic configuration allows us to study the impact of body position with minimum impact from other factors.

The results confirm our hypothesis. figure 3.2(a) and (b) plot instances of the signal strength distribution as a function of the user orientation, for both Droid and G1 phones. For easy illustration, we mark the direction of user facing the AP as 60° and the direction with back to the AP as 240°. The results, marked as "w/ human shield", show a clear dip pattern. When the user has her back facing (BF) the AP, the received signal strength is about 10-15dB lower than that when the user faces the AP. We observed similar patterns as we varied the distance between the phone and the AP. We observed up to 18dB difference in RSS values even when the user was only ≈2 meters away from the AP. We also varied the height of the AP between 2 and 18 meters, and observed no difference in results. Finally, in each graph, we mark the angle representing back-facing the AP with a vertical line (BF), and mark the point of lowest signal strength with a blue dot.

Experiment 2: Source of the Artifact. Our next question is to identify whether the signal strength artifacts are caused by the position of the user's body, or the orientation of the phone antenna [126]. First, we repeat Experiment 1 multiple times, each time changing the orientation of the phone antenna, and observe no change in the results. Next, we repeat the same experiment, but remove the human body as an obstacle by placing the phone flat on top of a small chest-high table. We shift the orientation of the phone in 20° increments, and measure the signal strength for 30 seconds with the phone in each orientation. The results are plotted in figure 3.2(a) and (b) as "w/o human shield," and show that the low-signal artifact is clearly gone. Thus we conclude that the low signal artifact is mainly caused by position of the human body, and its impact in blocking the AP's signal.

Experiment 3: Dependency on Environments. We also repeat the above experiments under different propagation environments. In addition to Simple LOS, we include an open area with surrounding buildings and moving objects (Complex LOS), and an environment with multiple buildings and objects and thus no line-of-sight paths between the phone and the AP (NLOS). We show illustrations of these three measurement areas in borealis/figure 3.3. We perform multiple measurements at different positions in each environment, and plot two sample results for Complex LOS and NLOS in figure 3.2(c) and (d). We see that the low signal artifact still exists, while the magnitude of the dip may vary between different positions. This is expected, since reflections from surrounding objects can create multipath signals that reduce the impact of the user body blocking signals.

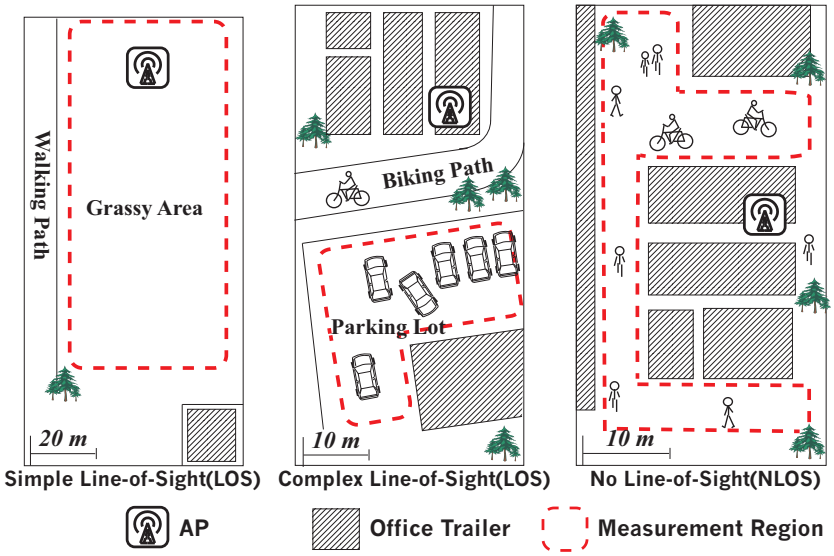


Figure 3.3: Graphical illustration of the three propagation environments used. all located on the UC Santa Barbara campus. For both Complex LOS and NLOS, the AP was mounted right below the roof of an office trailer, and there are many static and moving obstacles nearby, including trees, cars, bikers and pedestrians. For NLOS, the path between the AP and the measured region is blocked by office trailers.

Experiment 4: Impact of Phone and AP Configuration. These experiments measure the low signal artifact under different phone hardware and AP configurations. figure 3.2(e) and (f) show the standard experiment results using both iPhone4 and WindowMobile phones in a Simple LOS environment. We see that both phones exhibit the same artifacts for both 802.11b/g and 802.11n APs. The AP hardware results are notable, since it means that even connecting to 802.11n APs who use multiple antennas, the low signal artifact still comes across clearly in measurement results.

Experiment 5: Impact of User Posture and Body Shape. We repeat the above experiments with 6 additional users, with varying heights and

weights, ranging from 5'4", 100lbs to 6', 160lbs. Each user holds the phone using a slightly different posture. The same artifact consistently appears in all results, thus demonstrating that the low signal effect is prevalent across users.

3.3.2 Key Observations

Overall, our experiments lead to two key findings. First, our experiments confirm that the position of the user's body can significantly affect the smartphone's received WiFi signal strength. When the user has her back facing the AP, her body becomes an obstacle and significantly degrades the received signal strength.

Second, for each of our result graphs, we use BF (for back facing AP) to mark the opposite AP direction where the blocking effect should be at its strongest. We also mark the point of lowest signal strength with a large blue dot. It is clear that across all of our experiments with different phones and different environments, the angle with the lowest signal strength does not capture the predicted direction away from the AP. The "error" between the angle showing the lowest signal and the ideal angle can be as low as near zero (WindowsMobile, 11n), or as high as $\approx 40^\circ$ (Sample 2, Droid NLOS). Clearly, an accurate AP location system cannot simply rely on finding the angle with lowest signal strength, and must use more sophisticated techniques to determine the AP direction.

3.4 Accurate Access Point Location

Motivated by our findings, we propose *Borealis*, a new AP localization system for smartphones that leverages signal strength artifacts to compute the direction of an AP. Unlike conventional solutions that either require sophisticated radio hard-

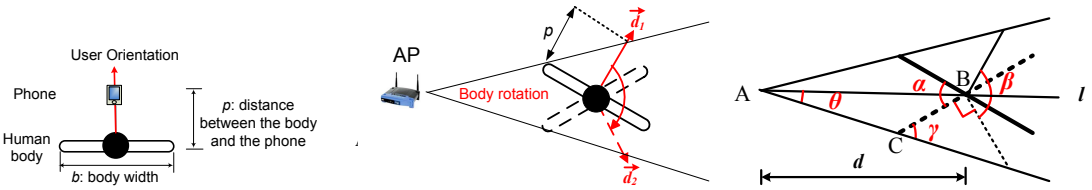
ware (*i.e.* directional antennas) [140], or extensive war-driving measurements [55], our solution uses off-the-shelf smartphones and produces real-time results with a small number of measurements.

3.4.1 Borealis Overview and Challenges

The concept behind our design is simple. At a fixed location, a user rotates herself and records the signal strength profile of the target AP. Based on the observed signal artifact, she estimates the direction of the AP from her current location. She moves towards the AP, occasionally repeating the measurement step, until she arrives at the AP.

Borealis has two key requirements. First, given the limited resources on smartphones, Borealis must use minimal energy and computational resource in its directional analysis. Second, Borealis must produce results in real-time, and its directional measurements should be minimized to conserve user effort. We can meet both of these goals by designing a system that minimizes computation while producing accurate results. The more accurate the result, the fewer number of direction analysis measurements are necessary, thereby saving both device battery power and user effort.

Our biggest challenge is how to reliably determine the AP direction based on the measured signal strength profile. Our initial experiments in Section 3.3 show that simply using the angle with lowest signal strength to estimate AP direction can generate large estimation errors (as indicated in Figure 3.2 by the distance on the x-axis between BF and the dot representing minimum signal strength).



(a) (b) (c)

Figure 3.4: An abstract model of the human shield effect. (a) A simple model of a user holding a smartphone, with body width b and phone/body distance p. (b) The signal propagation condition as the user rotates. When the user orientation is between $\vec{d_1}$ and $\vec{d_2}$, the LOS path will be blocked by the human body. This range of orientation is referred to as the blocking sector. (c) A geometric representation of the blocking sector.

Across roughly 40% of all our experiments, using minimum RSS to estimate the AP direction leads to an error between 40° and 120°.

This error comes from two factors. First, as the user rotates, we measure the signal strength of various directions sequentially, not simultaneously. Thus time dynamics in signal propagation produces variance in signal strength values across different directions. Second, the measurement time duration at each angle is limited, which leads to measurement noise. Extending the per-angle measurement time can help reduce noise, but also increases user effort, power consumption, and gap between sequential measurements.

3.4.2 Modeling the Body as a Signal Obstacle

Borealis addresses the significant error introduced by time dynamics and propagation effects using a model-driven approach. We first study the smartphone measurement traces to understand the relationships between signal strength profile, user orientation, and AP direction. We build a model to capture these rela-

tionships and predict the impact of blocking obstacles on signal strength. We use this model to develop an accurate AP direction prediction mechanism, along with a way to estimate the confidence of each estimate.

Our first observation from our experiments is that as the user rotates, not only does her body block the wireless signal and reduce its strength, but also that this signal degradation occurs *at a range of angles*, not just when the user is facing away from the AP.

To better understand this phenomenon, we introduce a simplified object-blocking model. Consider a simple propagation environment without any obstacles blocking the signal; a line-of-sight (LOS) path is the dominant signal path between the AP and the phone. We model the human body and the phone in a diagram in Figure 3.4(a), where the human body has width b and the distance between the phone and the body is p. Next, Figure 3.4(b) illustrates the signal propagation condition as the user rotates herself. We see that when the orientation of the user and phone is within an angular sector between $\vec{d_1}$ and $\vec{d_2}$, no LOS path will reach the phone. We refer to this angular sector as the *blocking sector*. Because the dominating path is blocked, the signal strength observed in this sector will be significantly lower than that of the other orientations.

Modeling the Blocking Sector. Using simple geometry, we can further characterize the blocking sector, particularly its size in degrees.

Theorem 1. *When line-of-sight is the dominant propagation path between the AP and the smartphone, the angular size β of the blocking sector is defined by*

$$\beta = 180^\circ - 2(\arctan\frac{2p}{b} - \arcsin\frac{bp}{d\sqrt{4p^2+b^2}}). \tag{3.1}$$

where p is the distance between the human body and the phone, b is the body width, and d is the distance between the AP and the user.

Proof. As shown in Figure 3.4(c), line l passes the AP (denoted by A) and the center of human body (denoted by B). Assume the rightmost point of human body is C and consider the triangle $\triangle$ABC. By the sine law, we have $\frac{b}{2\sin\theta} = \frac{d}{\sin\gamma}$, where θ and γ are marked in Figure 3.4(c). Similarly, we have $\gamma = \theta + \frac{\alpha}{2}$. Given that $\gamma = \arctan\frac{2p}{b}$, we arrive at $\alpha = 2(\arctan\frac{2p}{b} - \arcsin\frac{bp}{d\sqrt{4p^2+b^2}})$. Since $\beta = 180^\circ - \alpha$, we prove that Equation (3.1) holds. □

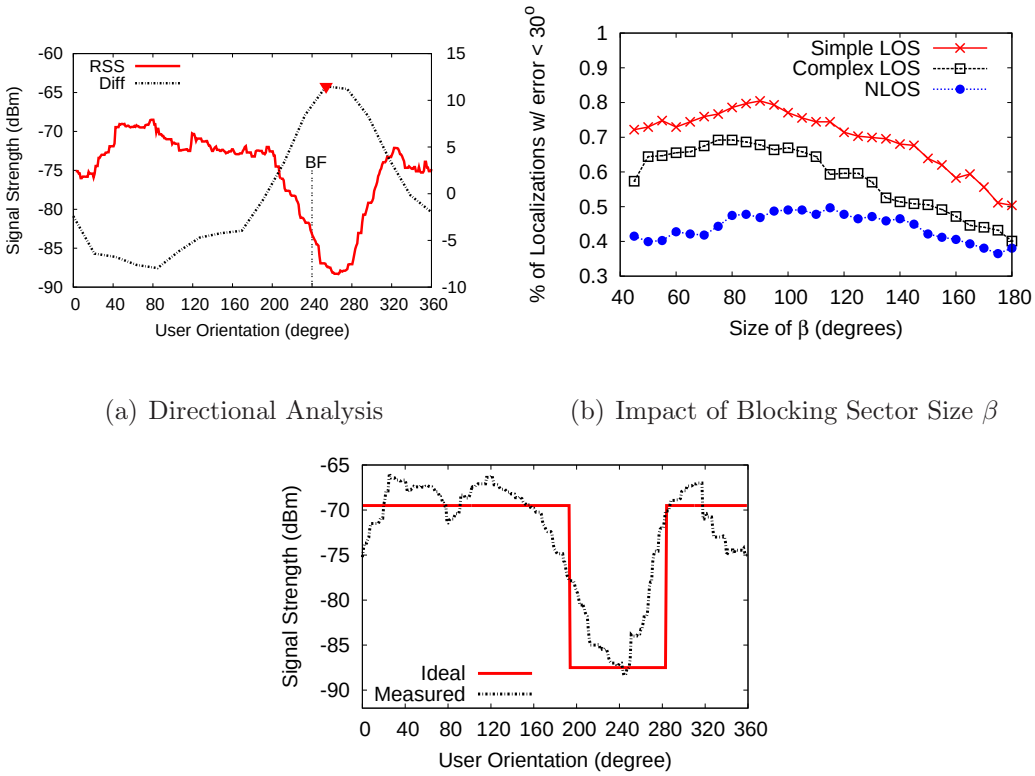
In practice, the distance between the AP and the user is much longer than the human body width ($\approx$ 0.4m), namely $d >> b, d >> p$. Furthermore, when a user holds the smartphone in a natural posture, the smartphone is roughly half of the body width away, namely $b \approx 2p$. Then we can reduce (3.1) to

$$\beta \approx 180^\circ - 2\arctan\frac{2p}{b} = 90^\circ. \tag{3.2}$$

Summary of Findings. The objective of our analysis is not to model the exact impact of the human shield effect, but to capture the large-scale trend of the signal strength profile, and its relationship with the user orientation and the AP direction. Along these lines, our analysis leads to two key insights:

• During user rotation, the signal strength degrades heavily when the user orientation is within a range, defined by the blocking sector. The angular size of the sector is roughly 90° under general configurations.

• We can derive an estimate of the direction to the AP, by taking the opposite direction of the center angle of the blocking sector β, as in Figure 3.4(c).



(a) Directional Analysis

(b) Impact of Blocking Sector Size β

(c) Predicting Confidence Value

Figure 3.5: Borealis' directional analysis. (a) Deriving the AP direction based on the blocking sector.(b) Choosing the blocking sector size. (c) Predicting the confidence of our direction analysis by computing the cross-correlation between the measured and ideal signal strength profiles.

3.4.3 Directional Analysis via Blocking Sector

Motivated by the insights from our model, we propose to estimate AP direction by locating the blocking sector within the signal strength profile. In essence, we organize the observed signal strength profile into overlapping sectors of size β, and locate the candidate sector that displays the largest relative signal degradation. The opposite direction from the center of the blocking sector is then the AP direction.

We now describe the detailed procedure of our proposed directional analysis. The input to our analysis is the *raw* RSS value (in dBm) corresponding to each measured user orientation. Here we do not apply any data smoothing, unlike the results shown in Section 3.3. We represent the measured RSS profile during a user rotation by

$$R = \{(RSS(\theta_i), \theta_i) | i \in [1, N]\} \tag{3.3}$$

where N is the number of measurement points, $RSS(\theta_i)$ represents the raw RSS reading (dBm) at point i, and θ_i is the user orientation at point i (the clockwise angle from the North orientation).

Organizing Candidate Sectors. We first form candidate sectors of width β. This is done by applying a sliding window of width β on the cyclic version of the signal strength profile, and shifting the window by Δ after forming a sector. The j^{th} sector contains measurement points whose θ satisfies: $(j-1)\Delta \leq \theta < (j-1)\Delta + \beta$. As a result, we create a total of $K = 360/\Delta$ overlapping candidate sectors. The value of Δ directly affects the accuracy and computation overhead. From our experiments, we did not observe noticeable gain of using $\Delta < 20°$. Thus we chose $\Delta = 20°$ in this work.

Locating the Blocking Sector. For each candidate sector S_j, we compute the *relative signal degradation* by subtracting the average signal strength of the sector from the average signal strength outside the sector:

$$\text{Diff}(j) = \frac{\sum_{\theta \notin S_j} RSS(\theta)}{N - |S_j|} - \frac{\sum_{\theta \in S_j} RSS(\theta)}{|S_j|} \tag{3.4}$$

where $|S_j|$ is the number of measurement points within the sector S_j and N is the total measurement points in the signal profile. By considering the signal strength distribution within and outside the sector, this relative degradation metric helps to

mitigate the negative impact of non-uniform distribution of measurement points within the signal profile. While (3.4) may not be an optimal function for identifying the blocking sector, it works sufficiently well for our purposes.

Using the Diff function, the blocking sector S^* is then

$$S^* = \underset{S_j}{\operatorname{argmax}} \operatorname{Diff}(j), \tag{3.5}$$

the sector that suffers the heaviest signal strength degradation. After locating S^*, we mark the opposite direction of the center orientation of S^* as the AP direction.

Figure 3.5(a) demonstrates our proposed direction analysis in terms of the RSS signal profile, the function Diff, the center angle of the detected blocking sector (marked by the red triangle), and the actual direction opposite to the AP direction (marked by BF). In this example figure, we smooth the RSS profile using a 20° sliding window average to show the general trend. We see that Borealis obtains an estimate within 10° from the actual direction. Yet if we use the MinR method, the estimated direction will be 30° away from the ideal result. The gain of our solution comes from the fact that we examine the signal strength distribution at the sector level, which not only captures the key effect of body blocking, but is also robust against low levels of statistical variance in the measured data.

Impact of β. The above discussion also leads to another question: *How important is the chosen value of β?* While this is difficult to study analytically, we performed experiments to verify the impact of β on the direction estimation. Figure 3.5(b) plots the percentage of measurement locations whose estimation angular error is less than 30°, as a function of β, for all three environmental settings in Figure 3.3. We see that the value of β that produces the least amount

of error is between $[80°, 100°]$ for all three settings. For simplicity, we use $\beta = 90°$ in subsequent experiments.

3.4.4 Confidence of Directional Analysis

Borealis' directional analysis provides another result, by providing a confidence level associated with each estimate of AP direction. Since no direction estimate is perfectly accurate, an associated confidence value gives the user additional useful information. As we show in Section 3.4.5, Borealis uses this confidence value of each estimate to control how often a user needs to repeat the direction estimate. Therefore, Borealis can bound the impact of directional analysis errors during user navigation.

We compute the confidence by comparing the signal strength profile to an idealized profile derived from our abstract model on body blocking. That is, ideally, the measured signal strength profile should display a dip pattern, where the signal strength is significantly lower for a range of angles. Based on this insight, we build an ideal profile using a square-waveform-like curve, shown in Figure 3.5(c). The width of the dip is β ($\beta = 90°$ in our system), and the center of the dip is set to the opposite of the (estimated) AP direction. The amplitude of the square is A_p for the peak and A_d for the dip, $A_p > A_d$.

We use the confidence to capture the similarity between the measured signal strength profile and the ideal profile. It can be computed as the cross-correlation coefficient of the two profiles [106], a widely used similarity metric for any two waveforms [36]. The correlation coefficient seeks to capture the similarity between the measured and ideal dip patterns, assuming both patterns produce the same

AP direction estimation. Hence before computing the correlation, we first align the center of the dip in the ideal profile to the opposite direction of the estimated AP direction, as shown in Figure 3.5(c). Next, let $\mathbf{T} = [t_0, ..., t_{m-1}]$ and $\mathbf{R} = [r_0, ..., r_{m-1}]$ denote the vectors of RSS values for the idealized (and aligned) profile and the measured profile, respectively. The confidence value ρ is:

$$\rho = \frac{1}{m} \cdot \frac{\sum_{i \in [0,m-1]} (t_i - \overline{t})(r_i - \overline{r})}{\sigma_{\mathbf{T}} \cdot \sigma_{\mathbf{R}}}, \tag{3.6}$$

where $\overline{t}, \overline{r}$ are the mean values of $\mathbf{T}, \mathbf{R}$ respectively, and $\sigma_{\mathbf{T}}, \sigma_{\mathbf{R}}$ are the standard deviations of $\mathbf{T}, \mathbf{R}$. The larger ρ is, the more confident Borealis is about its estimate. It is easy to prove that the values of A_p and A_d do not affect ρ, as long as $A_p > A_d$. Thus we set A_p=1 and A_d=0.

3.4.5 Direction-Guided User Navigation

While direction analysis provides a good estimate of the AP's direction relative to a location, the goal of Borealis is to allow users to determine the AP's physical location. We choose to adopt methodology similar to prior work [54, 75, 76], where a user moves towards the AP and performs periodic direction estimates to tune its direction. Navigation ends when she reaches the AP.

We modify prior approaches by leveraging our prediction confidence results to determine how frequently a user should update her direction of movement. Doing so less frequently, *i.e.* longer distances between measurements, reduces the number of measurements required. But this means error from a single measurement will have a greater impact on the efficiency of the path taken. We dynamically select the step size based on the confidence value predicted for each direction estimate. A high confidence implies a reliable estimate and means the user can travel a longer

distance before repeating the estimate. In contrast, a direction estimate with low confidence means the user is likely in a location with complex propagation conditions, and should repeat the estimate after moving a short distance along the predicted direction. We use detailed experiments to evaluate this in Section 3.6.

3.5 A Borealis Prototype

We implemented a prototype of Borealis on both Motorola Droid and HTC Dream/G1 phones, running the Android 2.2 Froyo OS. In the application layer, we implemented Borealis' directional analysis (with confidence prediction) and user navigation using Java and Android SDK. In the OS layer, we modified the WiFi driver to perform faster and more efficient RSS measurements. The architecture of our prototype is shown in Figure 3.6.

Application Layer. Our program allows the user to identify a target AP based on its SSID, and directs the user to start the rotation. During each rotation, our program continues to collect user orientation and RSS values by polling the compass sensor and WiFi RSS reports. Because the smartphones report orientation data at a finer granularity than RSS, we pair each RSS reading with an orientation reading based on their time stamps. Our program then analyzes the measured signal profile, and computes the AP direction and the confidence value of the current estimation. The direction is then displayed on the phone to guide the user navigation. During navigation, our program also determines online whether the next measurement point is reached, and informs the user accordingly.

OS Layer. While our main development effort lies in the application layer, in the OS layer we modified the Android WiFi driver to boost the speed and

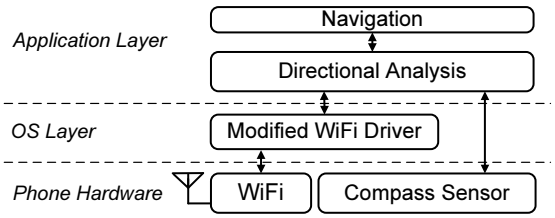


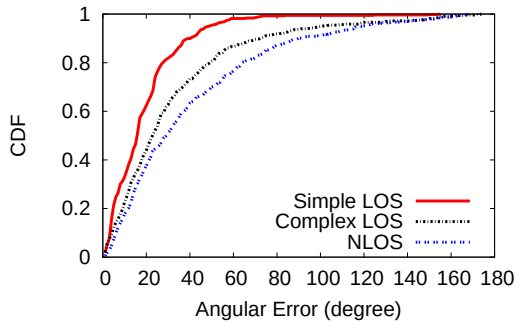
Figure 3.6: Borealis architecture overview.

efficiency of WiFi signal measurements. Android offers a native "Scan" function to collect RSS reports of a target AP. However, this operation scans all 13 WiFi channels sequentially, wasting both time and energy. To fix this, we modified the WiFi driver to only scan and report RSS on requested channels. This change reduces the RSS report time by a factor of 10.

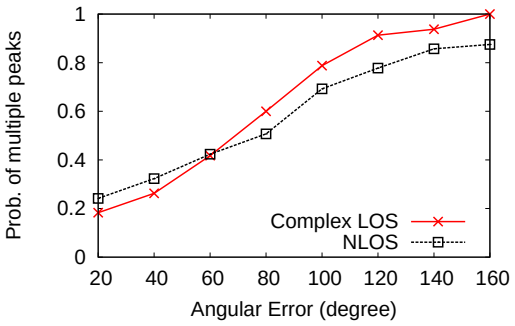
For our current prototype, each user rotation takes 1 minute to obtain a smooth RSS profile. We are currently working on understanding the minimum duration that produces an accurate result. Our preliminary results show that rotation duration of 10 seconds produces similar results to those of 1 minute durations. In addition, we are also investigating techniques that would allow us to measure RSS signal profiles opportunistically using users' natural movements, rather than forcing them to perform in-place rotation.

3.6 Evaluation

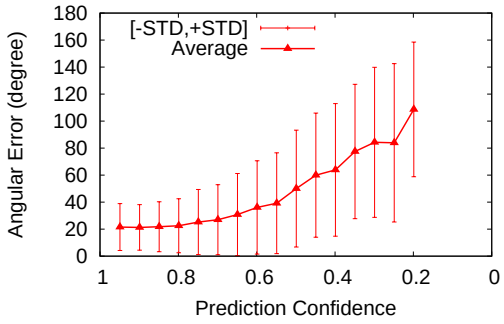
In this section, we evaluate our Borealis prototype using experiments on five Motorola Droid and HTC G1 phones. We use a Linksys WRT54GL 802.11b/g router as the WiFi access point, with a transmit power of 200mW. Our experiments were conducted over multiple days by seven users with different body shapes



(a) Accuracy of Direction Estimation



(b) Probability of Multiple Peaks in Diff.



(c) Accuracy of Confidence Prediction

Figure 3.7: The performance of Borealis directional analysis. (a) The CDF of the angular error for the three propagation environments. Borealis is fairly accurate for most of the test locations. (b) When Borealis produces larger errors, the signal profile often displays multiple dips, which creates multiple peaks in the Diff function. In this case, Borealis chooses the highest peak to estimate AP direction. (c) We observe a general trend where the confidence value of a direction estimation scales inversely with the angular error.

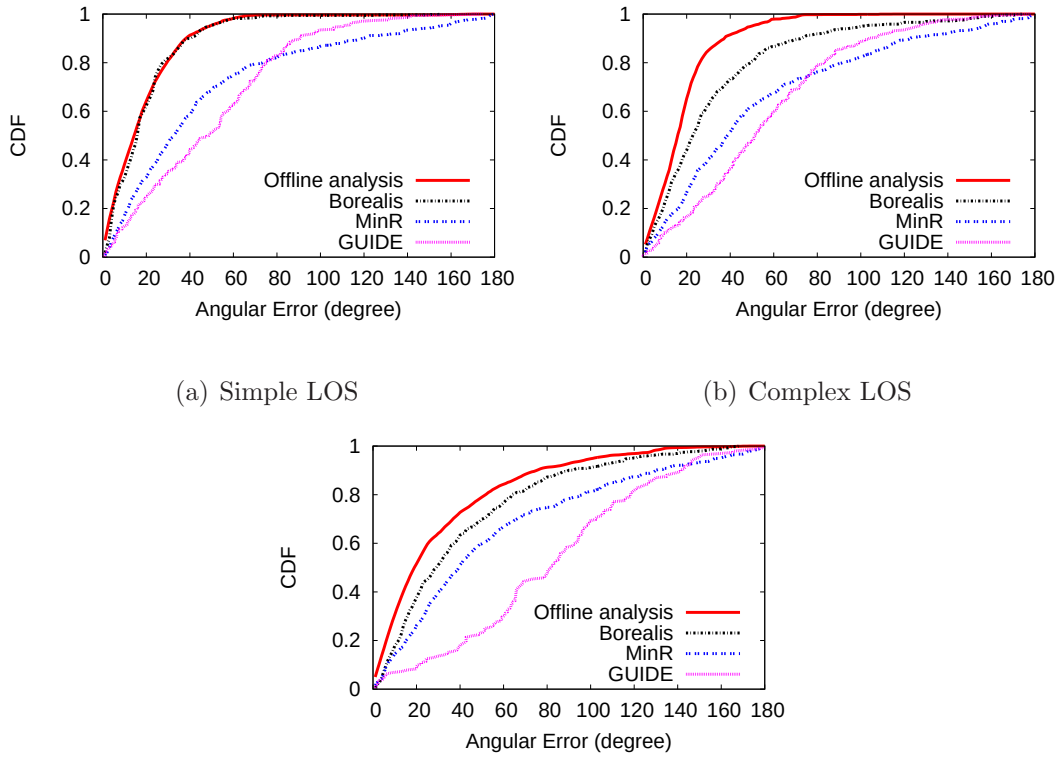
and ways of holding phones. Because G1 and Droid display similar results, we only show the Droid results for brevity.

We evaluate the impact of radio propagation by experimenting in three representative environments, as illustrated by Figure 3.3. In *Simple LOS*, we placed the AP on top of a shelf 2 meters in height, on a 100m×200m lawn. All the ex-

periments were on the lawn and away from large buildings. In *Complex LOS*, we placed the AP on top of a trailer building (5-meter in height) and experimented in the nearby parking lot surrounded by trailer buildings of similar height. In these locations, we could still see the AP. For *NLOS*, we used the same configuration of Complex LOS but experimented along the hidden walking paths where the AP was no longer in sight. For each environment, we experimented with at least 400 locations for each phone. There were random human movements throughout our experiments, *e.g.* people walking or biking.

To evaluate Borealis, we compare four systems for deriving AP direction via signal measurements.

- **MinR** – The baseline algorithm for our proposed directional analysis. It treats the opposite direction of the weakest signal as the AP direction.
- **Borealis** – Our proposed directional analysis.
- **Offline Analysis** – An offline version of our directional analysis using a clustering-based learning algorithm. It first collects the signal strength profile from roughly 360 locations in each environment, along with the actual direction of the AP in each case, and uses clustering techniques to build an optimized model, which is then applied to the remaining 60 locations for each environment to generate accuracy results.
- **GUIDE** [54] – A prior work that measures signal strength at three locations (forming a triangle) and computes the signal strength gradient to determine the AP direction. This method is the most comparable to Borealis since it is online and requires a very small set of measurements.



(a) Simple LOS

(b) Complex LOS

(c) NLOS

Figure 3.8: Comparing Borealis to Offline Analysis, MinR, and GUIDE in the three environments. Borealis significantly outperforms MinR and GUIDE, and is within a small distance from its offline version (Offline Analysis).

In the following, we evaluate Borealis' accuracy in AP direction estimates using both per-location measurements and user navigation experiments. We also examine the energy consumption of Borealis on both Droid and G1 phones.

3.6.1 Accuracy of Borealis Direction Estimation

We start from examining the accuracy of Borealis' direction analysis. Figure 3.7(a) shows the cumulative distribution of the angular error in AP direction

estimation. The angular error is the absolute difference in angular degree, between the estimated AP direction and its true direction. The results show that Borealis is fairly accurate in the Simple LOS environment – in 80% of locations, Borealis produces no more than 30° angular error. For the Complex LOS and NLOS environments where multipath propagation becomes dominating, Borealis can still maintain an error of no more than 50° and 65° for 80% of locations, respectively.

Sources of Large Errors. The above results show that occasionally, Borealis does produce large errors in direction estimation, particular for the Complex LOS and NLOS environments. To understand the cause of such errors, we studied the signal strength profile of locations with error higher than 60°. We observed that in most cases, the signal profile displays multiple dips, creating multiple peaks in the Diff function used in the directional analysis. In this case, Borealis' direction analysis uses the highest peak to estimate AP direction (based on eq. (3.5)). Figure 3.7(b) plots the probability of having multiple peaks in Diff, as a function of the angular error. Clearly, the larger the angular error, the more likely that it is caused by multiple dips in the measured signal profile. We also examined the spatial measurement locations with large estimation errors ($> 80^o$), and did not find any strong correlation between the two.

Confidence Prediction. Different from conventional solutions, Borealis also measures the confidence of its direction estimation. Figure 3.7(c) plots the relationship between the angular error of the direction estimation and the confidence value, using measurement data from all three environments. We observe a general trend that the angular error scales inversely with the confidence value.

An interesting question is whether the pattern observed in Figure 3.7(b), *i.e.* multiple dips in the signal profile or multiple peaks in Diff, can be used to refine the confidence estimation. The answer is no. This is because having large errors always maps to having multiple peaks in Diff, but not vice versa. From Figure 3.7(b) we see that the pattern appears even for cases with small angular errors.

3.6.2 Comparison to MinR & Offline Analysis

To examine the optimality of Borealis within the proposed directional analysis, we now compare Borealis' sector-based estimation to MinR, a simple estimation method, and Offline Analysis, the offline version of Borealis that uses training data to optimize direction decision. The comparison to MinR allows us to understand the gain of sector-based analysis, while the comparison to Offline Analysis allows us to understand the distance between Borealis and the "upper bound" performance of our directional analysis. The comparison is shown in Figure 3.8 in terms of the statistical distribution of the angular error.

We make two key observations. First, there is a large performance gap between MinR and Borealis. For example, for 80% of locations, the bound on angular error of MinR is 60° for Simple LOS, 120° for Complex LOS, and 135° for NLOS. This is roughly 2 times the Borealis' estimation error. As we have discussed earlier, such large error is because inherent signal variations create random ripples in signal strength profile. Thus a single point based direction analysis is highly sensitive to such random variations, leading to large errors. These errors are further exacerbated by multipath propagation in complex environments.

Second, the gap between Borealis and its offline-trained version is rather small, and even negligible in Simple LOS. We note that the offline version has the advantage of optimizing the direction estimation mechanism based on training data, so that it is able to recognize certain patterns in complex environments and produce a more accurate decision. On the other hand, the cost of such small improvement is the large measurement overhead and the fact of being an offline learning solution requires actual knowledge of the AP direction. Overall, because the gap between the two methods is small, we conclude that Borealis is a practical and effective solution for determining AP direction in real time.

3.6.3 Comparison to GUIDE

We now turn our attention to GUIDE [54] , the most relevant work in AP direction prediction using signal measurements. Similar to Borealis, GUIDE operates online, uses only a single receiver and requires a small set of measurements (at three locations). Different from Borealis, GUIDE applies a triangle-gradient based solution to estimate the AP direction.

We implemented GUIDE on our smartphones and experimented it in the three environments. Figure 3.8 compares the performance of GUIDE to that of Borealis. We see that for all three environments, Borealis significantly outperforms GUIDE. The key reason behind such large performance gap is that GUIDE (and other gradient based solutions) assumes that received signal strength degrades with the distance between the transmitter and receiver. In practice, this does not always hold, even in the simple LOS environment. Our own experiments show that the above assumption breaks in roughly 30% of the measurement cases.

For fairness, we did not compare Borealis to other AP localization methods, such as [140] and [55] . This is because these designs either require sophisticated directional antenna [140] or large measurements [55].

3.6.4 Locating Indoor APs

We also examined the scenario where the AP is placed indoors and an outdoor user collects signal measurements to estimate AP direction. Specifically, we consider the complex LOS setting in Figure 3.3 but place the AP inside the office trailer so that the wall of the trailer blocks all paths from the AP to the measured locations. The AP is located in the center of the trailer, away from windows and doors. We repeat the experiments at the same measurement locations used in our previous experiments.

In Figure 3.9, we compare the performance of both Borealis and GUIDE to prior results when the AP is placed outdoors. We see that multipath propagation in the indoor AP scenario does degrade the accuracy of direction estimation. However, the degradation is nearly negligible for Borealis. More importantly, Borealis still significantly outperforms GUIDE when locating indoor APs. These results show that Borealis is effective, and more accurate for locating indoor APs using outdoor signal measurements than GUIDE.

3.6.5 Borealis Navigation Efficiency

We also evaluate Borealis' end-to-end performance in terms of user navigation. We randomly selected 40 starting points in the three environments. They are within 60-140 meters from the AP and have RSS values around -90dBm. Further

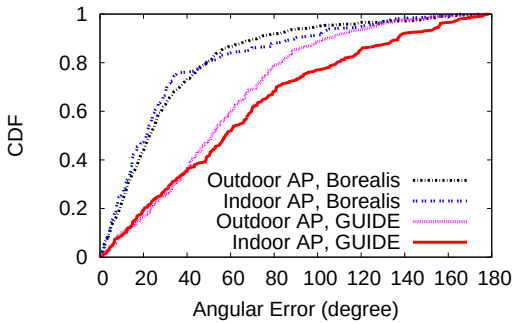


Figure 3.9: The performance of directional analysis when outdoor users locate an indoor AP using Borealis or GUIDE. The accuracy is comparable to the cases where the same AP is placed outdoors.

locations are not considered because no AP is detected. This range is similar to those used by prior work on outdoor AP location [54, 55].

We compare two Borealis navigation designs: i) navigation with periodic directional analysis (*e.g.* the tester rotates every 20 meters), and ii) confidence-guided adaptive navigation which determines the next location of directional analysis using the confidence of the current estimation. Specifically, if the confidence is above 0.8, the user rotates again after walking 30 meters. If the confidence is less than 0.3, she rotates after 10 meters.

We use two performance metrics: *navigation overhead* and *measurement frequency*. The navigation overhead is the normalized extra distance traveled to locate the AP: $\frac{\text{navigation distance}}{\text{shortest distance}} - 1$. We compute the navigation distance using GPS records of the navigation, and the shortest distance from GPS coordinates of the AP and the starting point. Because the shortest path might be blocked by obstacles, this metric serves as the upper bound on the extra distance traveled to locate an AP. On the other hand, the measurement frequency defines average

distance between two consecutive user rotations. It should be close to 20 meters for the periodic method, but larger for the adaptive design.

Table 3.1 lists both metrics averaged from the 40 experiments, using the periodic and adaptive methods. We see that in both Simple and Complex LOS environments, the adaptive design not only reduces the navigation overhead but also the measurement frequency. In particular, even in Complex LOS, it reduces the navigation overhead by half, and only invokes user rotation every 30+ meters. The gain is smaller in NLOS due to lower confidence in our directional analysis. Note that navigation overheads are very high in the NLOS case because we compared real physical walking paths taken by Borealis against shortest possible point-to-point distance to the AP. The high navigation overhead is from the user walking around obstacles such as buildings, trees and cars.

Figure 3.10 plots three sample navigation paths for the NLOS environment. We see that the navigation paths generally follow feasible walk paths, which demonstrates the efficiency of our proposed solution. Buildings and trailers do affect the accuracy of Borealis' direction estimates. But such errors are easily avoided by the user moving to a different location. An interesting observation is that multipath propagation is not always harmful. Often the strongest signal component comes from a path circumventing the obstacle, and the measured signal profile will indicate a strong dip near the direction of the open path. While it might not point to the exact AP direction, it certainly helps the user to identify a feasible path to the AP.

Borealis vs. GUIDE. We repeat the above experiments using GUIDE and its navigation procedure presented in [54]. For practical reasons, we stop GUIDE's navigation process when its navigation overhead is greater than 200% for Simple

		Navigation Overhead	Measurement Frequency
Simple LOS	Periodic	48%	22.59m
	Adaptive	15%	31.26m
Complex LOS	Periodic	74%	21.63m
	Adaptive	37%	30.62m
NLOS	Periodic	134%	20.45m
	Adaptive	107%	21.47m

Table 3.1: Performance of Borealis' navigation in the three propagation environments. Adaptive navigation guided by the confidence value not only reduces measurement frequency, but also shortens navigation distance.

LOS and 500% for Complex LOS and NLOS. Note that under this constraint, Borealis, using both periodic and adaptive navigation, can always reach the AP within a distance of 2m. But GUIDE's navigation rarely finds the AP. Only for 15% of all cases is GUIDE able to approach the AP within 11m. For all other cases, the user is still 35-184m away from the AP and does not have an accurate angular direction to the AP location. This result is consistent with the results offered by [54]. It is not surprising, since Borealis significantly outperforms GUIDE in direction analysis (Figure 3.8).

3.6.6 Energy Consumption

We now evaluate Borealis' battery consumption using both Droid and G1 phones. Our analysis leverages the battery usage summary tool offered by Android. In addition to identifying the total battery usage of each Borealis' directional analysis operation, we also study the distribution of energy costs across the different components involved.

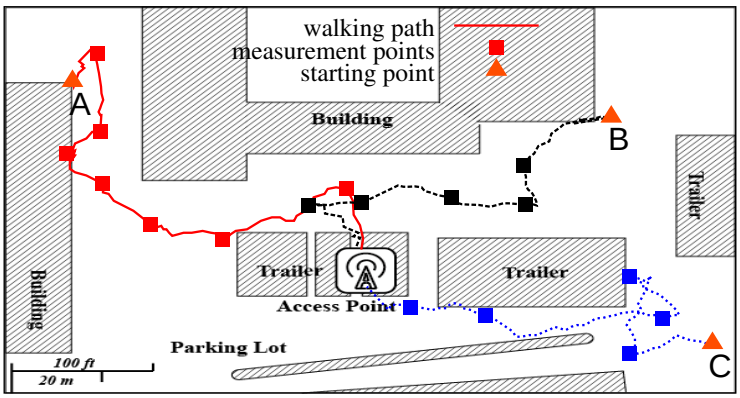


Figure 3.10: Sample navigation paths of Borealis in the NLOS environment. Points A, B, C mark the three starting points, and squares mark the locations of user rotation.

We configure our energy experiments as follows. Because the Android battery report has a coarse granularity (per 1% battery usage for G1 and 5% for Droid), we use a brand new, fully charged battery for each phone and run Borealis repeatedly to drain the battery. For each experiment, we also call the Android PowerManager API to log the phone battery level every 10 minutes. We verified offline that this API has negligible impact on battery usage. Using the battery log and the Borealis trace, we compute the energy consumption of each Borealis operation. We use two Droid and two G1 phones in our experiments, and show the average results for each phone category.

Table 3.2 summarizes the results of our energy experiments, including a detailed breakdown to five major components (as reported by the Android battery usage summary). The total energy consumption of a single Borealis directional analysis takes 0.36% of the total battery for Droid phones and 0.78% for G1 phones. The majority of energy cost comes from Display (for both phones) and

	Droid	**G1**
% of battery consumed per Borealis operation	0.36%	0.78%
% of battery consumed ignoring Display and Standby	0.15%	0.29%

Distribution of Energy Usage across Components		
Display	54%	32%
Cell Standby	3%	31%
WiFi Radio	5%	12%
OS	21%	11%
Other Borealis Activity	17%	13%

Table 3.2: Energy consumption analysis of Borealis' directional analysis on Droid and G1 phones.

Cell Standby (for G1). If we remove these factors, the battery use of a Borealis direction analysis operation reduces to 0.15% for Droid and 0.29% for G1. The OS component of Borealis consumes a substantial bit of energy. This is not due to computation, but the fact that each Android app must run inside its own lightweight virtual machine.

Finally, we note that WiFi uses much less energy compared to other components, because Borealis' signal measurements are passive and do not involve packet transmission. More importantly, Borealis only requires signal measurements during user rotation, and each navigation only requires a user rotation every 20-30m. Therefore, we conclude that normal usage of Borealis will not significantly impact the battery life of a smartphone.

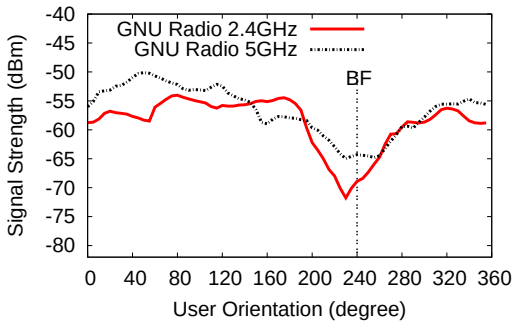


Figure 3.11: Sample signal strength profiles as a user rotates, measured using USRP2 GNU radios operating on 2.4GHz and 5GHz. We observe the same signal-blocking artifact. As before, BF marks the direction where the user has her back facing the AP.

3.7 Conclusion

In this chapter, we described Borealis, a smartphone-based system for locating WiFi access points in real time. While our tests show Borealis to be effective on Android phones in different environments, earlier measurements suggest that the same techniques would be effective on other smartphone platforms as well.

More importantly, the underlying principle behind Borealis, using signal dips from blocking obstacles to locate wireless transmitters, is general and could be applied to locate other types of transmitters. For example, Figure 3.11 plots two sample signal strength profiles obtained from a rotating user holding a USRP2 GNU radio operating on 2.4GHz and 5GHz. It is clear that the same signal-blocking artifact exists for these frequencies as well. For transmitters on lower frequencies that penetrate deeper and exhibit more multipath propagation behavior, users can potentially "rotate" or move around larger obstacles such as trees, vehicles, or buildings. Development of these systems could address significant

network management issues in the future, such as the enforcement of authorized transmissions in dynamic spectrum networks.

Chapter 4

Boosting Network Capacity via 60GHz Picocells

4.1 Introduction

We are all witness today to an explosion in mobile broadband usage. Thanks to bandwidth-hungry HD/3D video applications, industry research predicts that aggregate bandwidth requirements will increase by 1000-fold by 2020 [1]. This staggering growth is taxing mobile cellular networks past their limits and creating a capacity crisis [2]. Existing methods to increase capacity have reached their limits: spectral efficiency is slowly approaching the Shannon limit [46]; and acquiring more spectrum provides limited capacity gains, while facing regulatory hurdles and high monetary costs [2].

To deliver the three orders of magnitude bandwidth growth required by mobile applications, we need to consider dramatically different approaches for delivering wireless content. A viable approach must meet several key requirements. *First*, it must provide total capacity that scales with the number of base stations in a given region. This means base stations must support dense deployments with

minimal mutual interference. *Second*, it must increase capacity per base station by increasing amount of spectrum used and minimizing intra-cell interference between nearby mobile users. *Finally*, per-user data rates must be at least an order of magnitude higher than currently available rates, while providing robustness and reliability that match or exceed current standards in mobile and static scenarios.

Going Small. To minimize incremental deployment costs, cellular operators are exploring techniques to deploy smaller, denser cells for higher capacity, *e.g.* LTE picocells on lampposts or WiFi hotspots [3, 31], which augment existing LTE macrocells. Shrinking cell radii from several kilometers down to 100s (or even 10s) of meters is necessary to improve link quality and increase cell capacity. However, this approach is *fundamentally limited* by interference constraints for the carrier frequencies employed in today's cellular systems [115]. Specifically, at carrier frequencies of 1-5 GHz, physical form factor constraints put a hard limit on the number of antennas that can be accommodated on a picocellular basestation and a mobile device. Thus, even MIMO technologies can only mitigate interference to a limited degree, and picocell systems based on low carrier frequencies (LTE, WiFi, TV whitespaces) are unlikely to ever deliver the capacity increase we need [115].

The Promise of mmWave. But there is hope. Higher carrier frequencies, particularly millimeter-wave (mmWave) bands (30-300GHz), offer an attractive alternative. Using highly directional beams that drastically reduce interference, they enable small cells to truly deliver on their promise. Take 60GHz for example. First, the 60GHz band provides 7GHz of unlicensed spectrum for bandwidth hungry mobile applications, and is supported by the IEEE 802.11ad standard [13] (currently targeting indoor multiGigabit wireless networks) that defines multiple

data rates between 385Mbps and 6.76Gbps. Second, since 60GHz has a much shorter carrier wavelength, *i.e.* 5mm, it is possible to pack large antenna arrays into relatively small form factors. A 100-element 60GHz array easily fits in no more than a square inch[1], which can be easily integrated into today's mobile devices. Third, the high density of elements allows very high aggregate bandwidth, while the narrow directional beams limit interference and provide potentially "unlimited" levels of spatial reuse. Finally, low-cost 60GHz radio chips are already available on the mass market [4, 5], *e.g.* Dell Latitude 6430u laptops can be ordered with multi-Gbps WiloCity 60GHz chips for an additional $37.5.

Clearly, 60GHz links offer an attractive and practical solution that satisfies many of the requirements we mentioned. The goal of this work is to show that, while most of the attention in academia and industry focuses on 60 GHz for *indoor* applications, it is also a strong candidate for delivering the 1000-fold capacity increase demanded for outdoor cellular networks in dense urban environments.

A 60GHz Picocell Architecture. We propose a picocell architecture using 60GHz links, as depicted in Figure 4.1. We envision picocellular base stations densely deployed on lampposts or building ledges, with each station mounting multiple antenna arrays on each of its "faces." Properly tuned, each array can transmit independently to a different mobile user. To see the potential of this approach, consider a face which is 6x6 square-inches to a side. This can accommodate up to 36 60GHz 100-element arrays. If each array transmits at 2Gbps, the maximum data rate per face is 72 Gbps. Thus a single base station with 4 faces could support up to 288 Gbps downlink! Augmenting existing LTE cells

[1]In theory, this is hundreds of times smaller than arrays with the same number of elements for frequencies at or below 5GHz.

with such a high-capacity picocell deployment provides a path to future-proofing our cellular networks.

Despite the immense potential of 60GHz outdoor picocells, so far, both industry and academia have *explicitly* precluded 60GHz from outdoor cellular designs [109, 120]. For example, Samsung's recent outdoor mmWave attempts demonstrated a device prototype for transmission at 28GHz (a licensed band) [109]. Key concerns include propagation loss due to oxygen absorption, which peaks at 60GHz, and increase in propagation loss at higher carrier frequencies (unless highly directional antennas are correctly aligned at both endpoints). Another concern is blockage (diffraction around obstacles is more difficult at higher frequencies). These worries led to the limited use of outdoor 60GHz links in static backhauls, where manually adjusted high-gain antennas are positioned high above to avoid potential blockages.

Our Contributions. Our first contribution is to dispel common myths about 60GHz via extensive measurements and simulations, showing that the potential for 60GHz outdoor picocells can be realized without bending the laws of physics. Unlike prior efforts that focus on propagation models [90, 45, 94, 119], we experiment with off-the-shelf 60GHz radios, and study key questions such as transmission range, robustness to blockage and user motion, and spatial reuse. We explore how these issues are affected by future hardware upgrades (especially in terms of increasing the number of antenna elements). Unlike proposals for cellular systems in other mmWave bands [109], design and evaluation of our *outdoor* network can leverage extensive efforts already put into 60GHz PHY hardware and standards for *indoor* networks. Our second contribution is to point out the cross-layer archi-

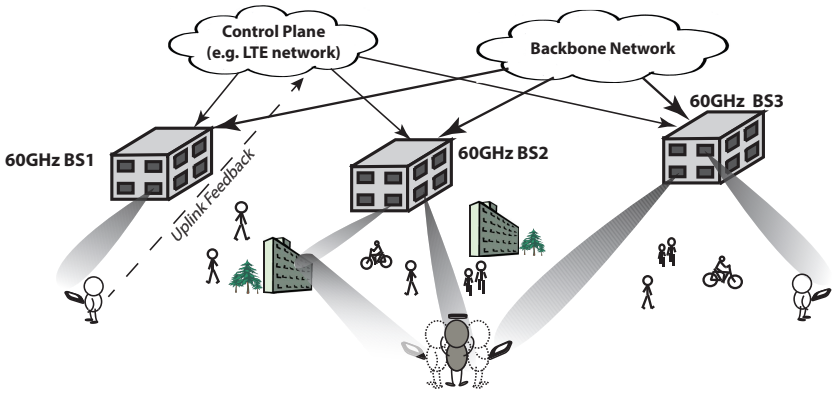


Figure 4.1: A potential 60GHz picocell architecture for outdoor mobile broadband augmenting existing LTE networks for significant increase in capacity. Each basestation contains many antenna arrays, each containing a large number of elements. Leveraging an LTE uplink as the feedback loop, each basestation can track and form highly directional beams simultaneously to multiple users. The beams follow users through direct or reflected paths, creating high-bandwidth downlink transmissions. Users can form beams in the reverse direction to build 60GHz uplinks, or use existing LTE uplinks.

tectural innovations required to truly deliver on this potential, and identifying an interdisciplinary research area spanning the PHY, MAC and networking layers. We now provide a quick preview of the myths we dispel.

- **Myth #1: 60GHz's range is too small for outdoor picocells.** Within FCC's power regulations, and *taking into account oxygen absorption and heavy rain,* 60GHz offers coverage area that exceeds 130m for 385Mbps and 79m for 2+Gbps. A propagation loss of 36 dB/km due to oxygen absorption and heavy rain makes it difficult to achieve kilometer-long links. But these factors only produce a 3.6 dB loss in signal strength at 100 m.
- **Myth #2: 60GHz is highly sensitive to blockage, hence robust outdoor connectivity is not feasible.** Indeed, 60 GHz links can be blocked

easily by a human body (20-50 dB of loss), and we do not have enough transmit power to burn through obstacles. However, an electronically steerable antenna array can point its beam towards one of several reflected paths available off of outdoor surfaces. The additional reflection loss of 5-7 dB can be handled within our desired link budget. We show that this, together with the concept of "picoclouds" (multiple base stations serving a given mobile user), can provide robust connectivity even in obstacle-rich urban environments.

- **Myth #3: User motion will easily break 60GHz links.** Detailed measurements show that electronically steerable antennas in off-the-shelf radios can easily keep up with outdoor pedestrian motion, and only need to realign once every few seconds to maintain high rate connections. In essence, the time constant of electronic adaptation is small compared to the time constants of human mobility.
- **Myth #4: Interference will significantly limit network capacity for small cells.** While this is indeed true at lower carrier frequencies [115], 60GHz's small interference footprint means that 60GHz picocells can be densely deployed with significant overlap but minimal interference, leading to a significant boost in spatial reuse. The reduced interference results from two factors: highly directional links that reduce interference between nearby users, and oxygen absorption and blockage from buildings that severely attenuate interference from distant cells.

In the rest of the chapter, we elaborate on these results, and show that 60GHz outdoor picocells are indeed a feasible approach for delivering orders of magnitude increase in network capacity.

Limitations and Future Work. As an exploratory study, our work has several limitations that need to be addressed in future work. *First*, the preceding myths correspond to *physical* properties of 60 GHz communication. Thus, dispelling them amounts to saying that, while the characteristics of 60 GHz are very different from those at lower carrier frequencies, there are no fundamental barriers in realizing its immense potential. However, as we discuss in §4.6, there are significant design challenges (and hence research opportunities) across a range of research disciplines, including hardware, signal processing and cross-layer protocol design. *Second*, our measurement experiments are constrained by limitations of available 60 GHz hardware, hence these results are supplemented by projections and simulation-based predictions (see §4.6 for a discussion on the role of such techniques for future research). *Third*, while picocellular deployments may be used to add capacity in a number of environments, our focus here for 60 GHz picocells is restricted to dense urban environments with high availability of reflecting surfaces such as walls. These are also locations likely to first experience full utilization of LTE capacity. Some of our conclusions may not apply, for example, to picocells covering large open spaces, where lower carrier frequencies may be a better choice.

4.2 Background and Methodology

In this section we introduce background on 60GHz communications and describe our experimental methodology.

4.2.1 60GHz Background

Directional Transmissions. To overcome attenuation and reduce interference, 60GHz links use beamforming to concentrate transmission energy in a desired direction. Today, 60GHz beamforming radios are readily available and affordable, either using horn antennas or antenna arrays. Radios with horn antennas use a flaring metal horn to direct radio waves, but require mechanical rotators to achieve fine-grained directional control [158]. On the other hand, radios with antenna arrays implement directional transmissions using electronic beamsteering [98, 4], employing signal processing at RF and baseband to adjust beam directions with minimal delay (*e.g.* a few ns). Such instantaneous beam adaptation makes antenna arrays an ideal fit for picocells.

The performance of antenna arrays scales with the number of antenna elements. The more the elements, the stronger the beam and the less the signal leakage. Today's low-cost 60GHz consumer products use 2×8 arrays (16 elements), while researchers have demonstrated 60GHz 16×16 array prototypes (total of 256 elements) [101]. Furthermore, recent breakthroughs make it computationally feasible to adapt even larger arrays (1000+ elements, but still fitting in palm-sized form factors!) [116], which puts us in a position to take advantage of future hardware advances to synthesize even narrower beams to further increase spatial reuse.

Oxygen Absorption and Rain Loss. At longer ranges, the advantages of directionality are offset by signal strength losses from oxygen absorption, which peaks at 60GHz, causing a loss of about 16 dB/km [6, 150]. While this makes realizing long-range (kilometers) 60 GHz link very difficult, it is actually an advantage for picocellular networks because it significantly enhances spatial reuse while

having relatively small impact on the link budget at the link ranges of interest to us, as also confirmed by our testbed experiments in §4.3. For example, the net loss due to oxygen absorption and very heavy rain of 2 inches/hour is 36 dB/km, which works out to a modest 3.6 dB for a typical link range of 100m.

Impact of FCC Regulations. The emitted power at 60GHz is constrained by the Federal Communications Commission (FCC) as follows: the transmit power plus beamforming gain, termed Effective Isotropic Radiated Power (EIRP), is limited to 40 dBm. For antennas that are extremely directional (e.g., using parabolic dishes), the FCC has relaxed its specifications to permit further increase of EIRP [10], but we are unable to take advantage of this in the operating regime of interest to us: our architecture is based on highly directional, electronically steerable beams, but the beamwidth is still too large for the new FCC regulations to permit an EIRP higher than 40 dBm. Thus, increasing the number of elements in the transmit array does not improve link range if we already at an EIRP of 40 dBm, hence enhancements in link range must come from the receiver, by decreasing the receiver's noise figure or by increasing its antenna directivity. Of course, increasing the number of transmit elements has a big impact on *network* performance, narrower transmit beams reduce interference and enhance spatial reuse.

Leveraging IEEE 802.11ad. The IEEE 802.11ad standard for 60GHz [13] defines a number of modulation and coding schemes to adapt to different range-rate tradeoffs. In our envisioned picocellular architecture, we leverage this 802.11ad *physical layer* on the downlink, and show that link ranges of the order of 100m are attainable (see Table 4.1). Of course, *our MAC and network layers will be*

different from that of 802.11ad, given the need to adapt to a far more dynamic outdoor environment than the indoor networks the 802.11ad standard was currently intended for.

The link budget calculations resulting in these ranges use conservative assumptions (measurements from our testbed lead to more optimistic predictions). The receiver sensitivity for each data rate (*i.e.*, minimum received power required to sustain that data rate), obtained from the 802.11ad standard, already accounts for large values of implementation loss (5 dB) and noise figure (10 dB). The link budget accounts for the FCC EIRP restriction of 40 dBm and loss due to free space propagation and oxygen absorption. On top of that, we add a link margin of 15 dB, which is more than enough to account for reflection loss (in case we are using a bounce rather than a LoS path, which typically leads to at most 5-7 dB loss) and for for rain (even for heavy rain, the additional loss on top of oxygen absorption is about 20 dB/km, hence the rain loss over 100m is 2 dB). That is, the receiver sensitivity as a function of range R (in meters) is computed as

$$P_{RX}(dBm) = EIRP(dBm) + G_{RX}(dBi) - L_{ploss}(R) - L_{margin} \quad (4.1)$$

where $G_{RX}(dBi) = 10\log_{10} N_{RX}$ is the receiver antenna directivity as a function of the number of receive elements N_{RX} (we consider $N_{RX} = 16, 64, 100$ for the 4×4, 8×8 and 10×10 arrays). The path loss L_{ploss} in dB is

$$L_{ploss}(R) = 10\log_{10} \frac{16\pi^2 R^2}{\lambda^2} + \alpha R \quad (4.2)$$

with $\alpha = 0.016$ dB/m (16 dB/km) accounting for oxygen absorption, and $\lambda = 0.005$m at 60 GHz.

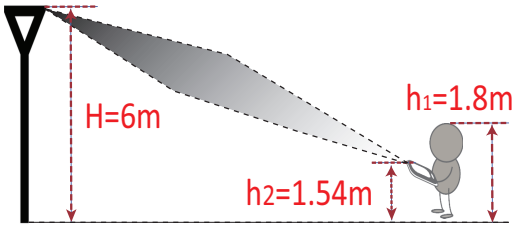


Figure 4.2: Controlled environment: a picocell system with 60GHz radios at both basestations and mobiles.

Figure 4.3: Real-life environment: a downtown street area with random pedestrians.

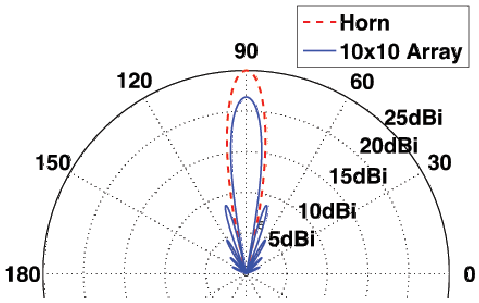


Figure 4.4: Radiation patterns of our testbed's horn antenna and a 10x10 rectangular array.

4.2.2 Measurement Methodology

Unlike prior efforts that focus on propagation models using custom equipment [45, 90, 94, 119], we experiment with commercial off-the-shelf 60GHz radios. We study key issues regarding the feasibility of 60GHz picocells, including range, sensitivity to motion and blockage, and interference footprint.

Measurement Environments. We perform experiments in two types of environments.

- *Controlled Environments.* These are environments without any uncontrolled pedestrian. We use these measurements to evaluate the impact of individual factors in isolation. To emulate a picocell, we place a radio on a second floor balcony (6m high), representing a basestation fixed on a lamppost (Figure 4.2). A second radio represents the end user.
- *Real-Life Environments.* We also performed experiments in a downtown street walkway (Figure 4.3) and a campus plaza during busy hours with uncontrolled pedestrian. The basestation is placed on a second floor balcony (6m high).

Data rate	RX sensitivity	4×4 RX	8×8 RX	10×10 RX
385 Mbps	-68 dBm	63m	114m	138m
1.155 Gbps	-64 dBm	41m	77m	94m
2.31 Gbps	-61 dBm	30m	57m	70m

Table 4.1: Outdoor range/rate tradeoffs with 802.11ad PHY

60GHz Radio Platforms. We consider two types of 60GHz radios with antenna arrays, representing today's off-the-shelf hardware and high-performance radios for future deployments. By comparing their results, we also seek to identify issues that can be addressed by reasonable hardware upgrades, and those that are fundamental to 60GHz picocells and hence require careful network design. For both platforms, each radio operates on a single 60GHz channel of width 2.16GHz.

- *Wilocity 2×8 Antenna Arrays.* We use pairs of Dell 6430u laptops (as users) and D5000 docks (as basestations), designed for indoor usage. Each has a Wilocity 60GHz radio chipset with a 2×8 rectangular antenna array, operating according to IEEE 802.11ad. Because Wilocity does not provide any information on the beamwidth, we apply the antenna array theory [56] to

compute the lower bound on the beamwidth: the horizonal 3dB beamwidth is 13^o and the vertical 3dB beamwidth is 60^o. Each radio adapts link rate between 385Mbps and 3.85Gbps, and maintains directional connectivity over time (e.g., if the LoS path is blocked, it seeks a reflected path and beamforms along it). Currently, the radio does not provide external control of link directions, hence we rely on its built-in algorithm to manage link directionality. The radio does not report signal strength values but link rates, so we monitor link performance using fine-grained reports on TCP/UDP throughput and link rate.

- *Emulating 10×10 Antenna Arrays.* There are no suitable 60GHz radios with large antenna arrays on the market. Instead, we emulate the performance of a large antenna array using a horn antenna. For experiments that do not consider interference, this can provide accurate results. This is because 60GHz link performance is largely determined by directionality and signal patterns of the main beam lobe[2]. In this regard, large antenna arrays and horn antennas are very close except for differences in antenna gain.

  Specifically, we use two HXI Gigalink 6451 radios with 0dBm transmit power and 1Gbps link rate, operating on a proprietary (non-802.11ad) configuration. Each radio has a horn antenna of 10^o 3dB beamwidth and 25dBi gain. Its main lobe pattern closely align with that of a 10x10 array with 1dBi elements with 21dBi gain (Figure 4.4). We set each radio on a mechanical rotator and adjust its beam direction in units of 0.5^o. Each measurement records the received

[2]Antenna arrays produce side lobes (*i.e.* unwanted transmissions in undirected directions) aside from the main lobe. The side lobes are at least 13.26dB weaker than the main lobe and thus not responsible for data communication [73].

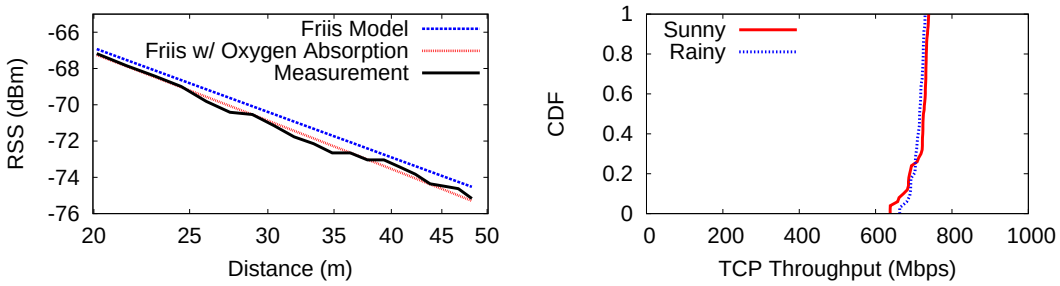
signal power level (RSS), repeated once every 50ms. The measured noise level is -80dBm.

Since HXI radios use a proprietary physical layer and only support a single data rate, they cannot fully capture the performance of a picocell operating based on 802.11ad. However, we can extrapolate the attainable range to 802.11ad settings: increasing the EIRP to 40 dBm from the 25 dBm in our testbed leads to a 15-4=11dB increase in RSS (counting the 4dB gain difference of horn antenna and 10×10 arrays). This provides us with a practical means of using our testbed to verify link budget calculations as in eq. (4.1), and also to predict the supported data rate for each RSS measurement following the 802.11ad receiver sensitivity table. Specifically, we use the 802.11ad receiver sensitivity table to derive the required SNR for each data rate, and use it to derive the data rate supported by each RSS measured by our testbed. A similar methodology has been used in prior work on 60GHz systems [158].

4.3 Single-User Feasibility Study

Using testbed measurements, we perform initial evaluation on the feasibility of 60GHz picocell systems, focusing on dispelling the four myths discussed in §5.1.

Assumptions. Due to hardware limitations, our measurement study makes a few assumptions. *First*, our experiments emulate a single array forming a beam to serve a single user, using either 2×8 WiloCity arrays at both ends or 10×10 arrays at both the base station and the user. Our envisioned picocellular base stations would have multiple such arrays, one for each user, as discussed in §5.1. *Second*, we assume the target user is stationary (sitting or standing), although her body



(a) 60GHz Outdoor Propagation

(b) TCP Throughput over Time

Radio Type	Weather	Minimum Link Rate (Mbps)		
		385	1155	2310
Wilocity 2x8 EIRP=23dBm	clear	23m	15m	10m
	heavy rain	22m	-	-
HXI 10x10 EIRP=40dBm	clear	178m	124m	93m
	heavy rain	139m	102m	79m

(c) Ranges of 60GHz Arrays

Figure 4.5: Measurements on 60GHz range and throughput. (a) The RSS results collected by HXI radios show that 60GHz outdoor propagation follows the Free-Space Friis model with 16dB/km oxygen absorption loss. (c) CDF of per-minute TCP throughput measured over 10 days using the Wilocity radios. (b) Downlink LoS ranges when 60GHz radios operate in the 802.11ad single carrier modulation (SC) setting, with 2×8 and 10×10 antenna arrays. The Wilocity 2×8 results are based on actual measurements, while the HXI 10×10 results are projected by increasing EIRP to 40dBm and mapping a data rate to each measured RSS value based on the 802.11ad specification.

may rotate or move a short distance due to natural posture changes. *Finally*, basestations do not coordinate in these measurements. We study the benefits of coordination via simulations in §4.4.

4.3.1 Measurements in Controlled Environments

We begin by verifying the framework for the link budget calculations in §4.2.1. Our measurements use the HXI radios since they report detailed receiver RSS values. Figure 4.5(a) plots the RSS values at different link distances under clear weather, along with the well-known Friis model [50] on 60GHz free space propagation with and without oxygen absorption (16dB loss per km). The results show that 60GHz outdoor LoS propagation does follow the classical Friis model with oxygen absorption loss, *i.e.* eq. (4.2).

Range & Throughput

Our first task is to evaluate the range and throughput of 60GHz transmissions in the context of outdoor picocells.

Picocell LoS Range. Figure 4.5(b) lists the measured LoS range of the Wilocity 2×8 array under different link rate targets. On a clear day, it can reach 23m at 385Mbps, and 9m at 2.31Gbps. While this is clearly insufficient for 60GHz outdoor picocells, we can easily extrapolate beyond the hardware limitations of the Wilocity platform, which is designed for indoor short-range usage and has an EIRP of 23dBm. When we increase the EIRP from 23dBm to 40dBm, the path loss formula in (4.2) can be used to translate the 17 dB increase in signal strength to a corresponding increase in range. Because of the mixture of free space loss of $1/R^2$ and exponential decay of $e^{-\alpha R}$, the actual range increase depends on the initial value of range. While this is easy enough to calculate numerically, it is instructive to separately address these two effects via an example. Specifically, can we get a data rate of 385 Mbps at 100 m using 2×8 arrays as in the Wilocity

radios in heavy rain (e.g., 2 in/hr, where the loss due to rain and oxygen absorption is about 36 dB/km)? The exponential loss is 3.6 dB at 100 m, and ignoring the baseline loss under the clear conditions for our measurements at the starting range of 23 m, we still have 17-3.6=13.4 dB signal strength increase to compensate for free space loss, leading to 13.4/2=6.7 dB, or 4.67X, increase in range relative to the baseline of 23 m. This yields a range of 100+ m. The point of this calculation is to show that, for the moderate ranges of interest, oxygen absorption and rain have a relatively small effect on the link budget.

In order to quantify the effect of using larger arrays, we project the measurements using our HXI radios, which we using as a proxy for 10×10 arrays, by adding 15 dB to the EIRP (going from 25 dBm to 40 dBm), and subtracting 4 dB in receiver directivity (25dBi horn replaced by 10×10 receive array with 1 dBi elements). If we keep the EIRP fixed, then increasing the number of transmit elements does not help the link budget (it only decreases the emitted power per element), hence the source of performance gain is the increase in receiver directivity. Based on this projection, the range in clear weather is as high as 178 m at 385 Mbps.

Figure 4.5(b) also lists transmission ranges under heavy rain (0.38in/h). The measured range of Wilocity radios remains the same while that of 10×10 arrays, projected from our HXI measurements, drops to 139m at 385Mbps and 102m at 1.15Gbps. This is as expected, since the additional loss due to rain impacts longer links. However, the range (100+m) is still sufficient to support picocell operations.

Link Throughput over Time. We also measured 60GHz link throughput over 10 days (February 2014) to examine its long-term performance in outdoor

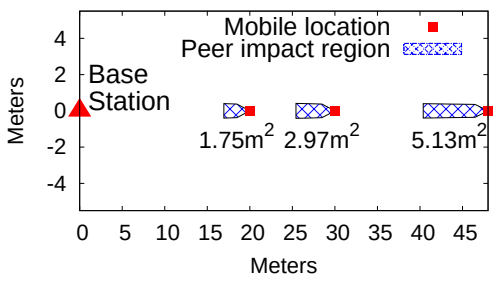


Figure 4.6: Impact of human blockage in terms of peer impact region for HXI radio. The base station is positioned at (0,0), and receiver is at 20m, 30m and 48m away.

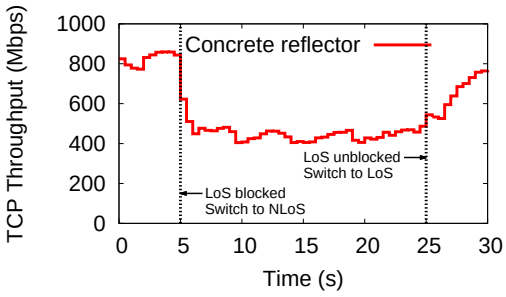


Figure 4.7: Measured TCP throughput as a Wilocity 60GHz link switches between different paths.

Radio	Path Length	Flat steel, Aluminum, Plastic board	Exterior Building Walls		
			Brick, Paint, Whitewash	Coarse paint	Coarse concrete
Wilocity (2x8)	10m	0%	0%	16%	16%
	15m	0%	17%	33%	33%
HXI (10x10)	30m	0%	17%	33%	46%
	50m	0%	19%	25%	25%
RSS Loss		0dB	3dB	4.5dB	5dB

Table 4.2: The reflection loss of eight different materials measured using both Wilocity and HXI radios.

environments. The weather condition for the first 9 days was clear and the last day we had heavy rain (0.38in/h). We used the Wilocity radios, set the link distance to 20m, close to its maximum range of 23m, and recorded per-minute TCP throughput using Iperf. Figure 4.5(c) plots the CDF of the per-minute TCP throughput of the first 9 days and that of the last 1 day. We see that the Wilocity 60GHz link offers 638+Mbps TCP throughput continuously over time (and 700

Mbps 80% of the time). The minor rate fluctuations (638-700 Mbps) are randomly distributed across the measurement period, and may be the result of occasional structure vibrations caused by wind and/or pedestrians walking on the balcony. We see no difference in throughput between clear weather and heavy rain, which confirms that, at short distances, 60GHz is highly robust to weather conditions. We also confirm this observation using the HXI radio measurements. At $50m$, heavy rain (0.38in/h) only introduces 0.86dB RSS loss, which did not introduce any degradation to the link.

Robustness to Blockage

Outdoor 60GHz transmissions experience blockage from various obstacles including buildings, trees, pedestrians and vehicles. Our experiments focus on pedestrians, because they are moving obstacles commonly seen in outdoor settings. We also examine the effectiveness of switching to reflected paths on addressing blockage.

Impact of Pedestrian Blockage. Our experiments show that when holding a 60GHz radio and facing away from a basestation, the user herself will block the LoS signal. This observation triggers another important question: "will the connection to a user, either LoS or via reflection, be constantly blocked by nearby pedestrians?" To answer this question, we measure for both radios the LoS peer impact region, *i.e.* the area around a user where a peer of similar height blocks the LoS path between the basestation and the user. For Wilocity radios, we placed RF absorber around the radio to prevent it from switching path. We also derive the impact region of reflection paths following the same method. Figure 4.6 shows

the LoS impact regions for the HXI radio at different link distances (the Wilocity radio result is similar and thus omitted for brevity).

We see that the observed impact regions are all small areas with a narrow cone shape, and the width (y-axis) is significantly smaller than its length (x-axis). The region also enlarges linearly with the link distance. This is because the basestation is placed in a high location (6m in our experiments, representing a lamppost placement) and the peer must be quite close to the user to block the signal.

Overall, our results imply that in outdoor environments, surrounding static peers can potentially block paths to a user. The impact, however, is limited because the peer has to be in a small region near the user. On the other hand, our measurements in busy walkways (§4.3.2) show that groups of moving pedestrians have a heavier impact on connectivity. However, such blockage can be effectively addressed by either switching to another reflection path (discussed next) or connecting to another basestation (see §4.3.2).

Handling Blockage via Reflection. 60GHz signals are reflected by most surfaces with losses of the order of 5-7 dB [83]. Thus, when the LoS path (or an existing path) to a user is blocked, the basestation can adjust link directionality to create another reflection path. In fact, our measurements show that Wilocity radios can already switch to reflected paths in real time upon LoS blockage (see Figure 4.7).

Next, we study the impact of reflection on link performance. We experimented with eight surfaces, commonly seen in outdoor street environments. These include three flat surfaces and five exterior building surfaces. We repeated our experiments with different incident angles (10^o-80^o), and found that the angle has a very minor

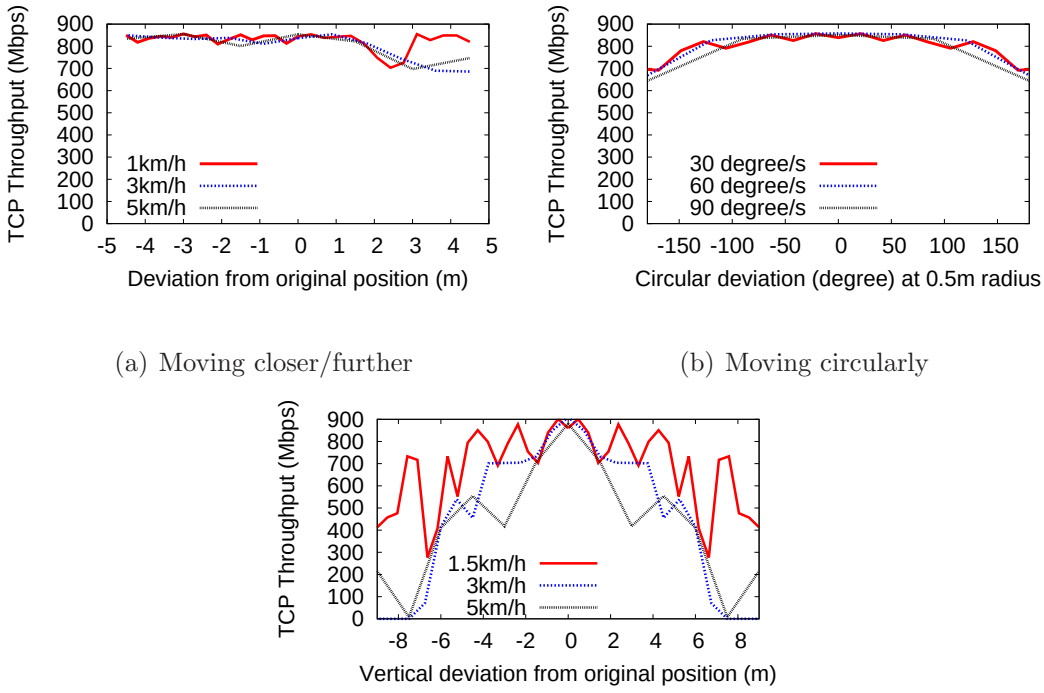
impact on the result[3]. For brevity we only show the results of 45°. Table 4.2 shows the normalized throughput loss (compared to a LoS path of the same length) for both 2×8 and 10×10 arrays at different path lengths. The HXI radio experiments also report the amount of RSS loss due to reflection. For both radios, the first three flat (and smooth) materials are perfect reflectors with negligible signal loss. The five building surfaces introduce 3-5dB signal loss and up to 46% throughput loss. The throughput loss varies with the link distance while the signal loss (in dB) remains constant. We noticed that in our results with the HXI radios the measured reflection losses were slightly smaller (*i.e.* better) that those reported in [83]. This might be due to the subtle difference in the reflection materials. Overall, these results show that reflection paths are good alternatives to LoS paths, offering ample opportunities for picocell coverage.

Robustness to User Motion

In practice even a "stationary" user will naturally make local movements, such as change her posture, or move a small distance (*e.g.* up to a few meters). These minor movements could change 60GHz link directionality and thus performance. We now present some initial results on their impact on link performance and implication on picocell design.

Specifically, we seek to examine whether 60GHz radios can re-align their arrays to maintain a LoS connection. To do so, we place absorber around the transmitter radio to prevent the use of any reflection path, and measure TCP (or UDP) throughput continuously during the motion to study its impact. We consider four

[3]Our results align with existing results: for thick materials (≥3cm), the reflection loss decreases as the incident angle increases [83].



(a) Moving closer/further

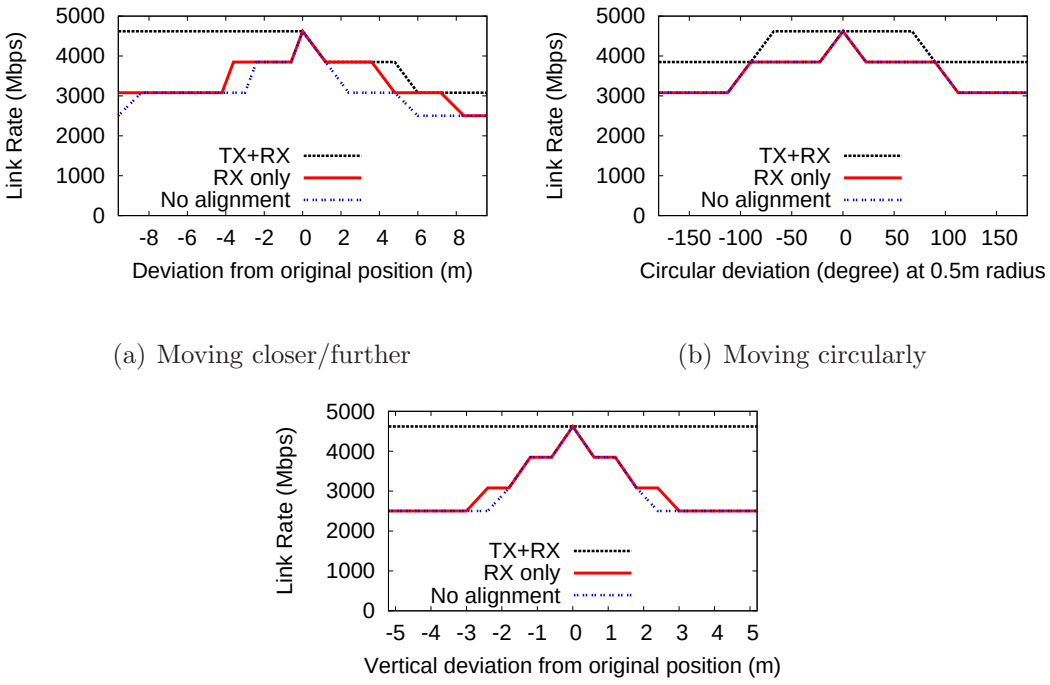
(b) Moving circularly

(c) Moving perpendicularly

Figure 4.8: Measured throughput using our Wilocity testbed (2x8) when the receiver moves in three different motion patterns.

types of user motion: rotating the body, moving closer/further to the basestation[4], moving perpendicularly to the basestation, and moving in a circular motion. Each experiment starts with the basestation and user radio directly facing each other and separated by $10m$ (Wilocity) or $40m$ (HXI). While the absolute results depend on factors like link distance and user body shape, our goal is to understand the general trend and its implication, as well as how the two 60GHz radios react to these motions.

[4]Since the basestation is placed higher than the user radio, moving further/closer to the basestation does change link directionality.



(a) Moving closer/further

(b) Moving circularly

(c) Moving perpendicularly

Figure 4.9: Measured RSS values using our HXI testbed (10x10) when the receiver moves in three different motion patterns. We consider three types of alignment methods: no alignment, partial alignment (RX only) and full alignment involving both TX and RX (TX+RX).

Wilocity 2×8 Array Results. Since 802.11ad already enables radio realignment via beam tracking, we expect that the Wilocity radios can realign with each other to a certain degree. Thus the goal of our measurements is to understand the speed of realignment and the robustness against user motion.

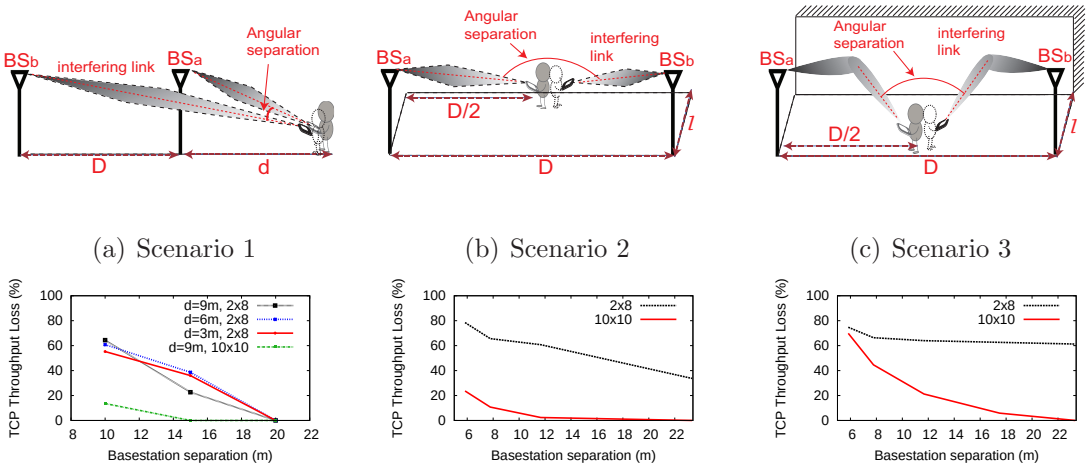
Because Wilocity radios do not expose any information on beam directionality, we first performed a set of UDP experiments to understand their array alignment behaviors. In each experiment, we first let the basestation and user radios face

each other and record UDP throughput, and then move/rotate the user radio locally for 1-3s and stop for 15s (to allow re-alignment).

From both UDP and link rate reports, we see that the low-cost Wilocity radios already adapt beam direction on-the-fly to cope with changes in link directionality. But the adaptation has two limitations. *First*, the adaptation is not instantaneous: in our experiments, it takes 2-7s for UDP/link rate to fully recover. Such delay can come from the time for adjusting the array and that for coordinating both ends. *Second*, the amount of beam adjustment is limited, *i.e.* the array cannot tune to all the directions, possibly due to the hardware enclosure and/or low-cost array design.

With these observations in mind, we now study the impact of user motion on TCP performance. Each experiment starts from the two radios being perfectly aligned with TCP throughput at 900Mbps. *First*, we experimented on user rotation at different rotation speeds (40^o/s - 100^o/s). We found that until the rotation angle reaches 85^o, the TCP throughput remains at 600Mbps. After that, the link breaks down.

Next, for the other three patterns, Figure 4.8 shows the TCP throughput during motion, as a function of the receiver's deviation distance from the original position. We make two observations. *First*, a user moving further/closer or in a small circle at moderate speeds (Figure 4.8(a)-(b)) lead to a minor degradation in TCP throughput. *Second*, moving perpendicular to the LoS has more impact since it introduces larger/faster directionality changes. During the movement, TCP throughput fluctuates and gradually degrades to 400Mbps as the deviation reaches $8m$. Specifically, when the user moves at 1.5m/h, the throughput first drops due to array misalignment, then rises as both radios align their arrays, and



(a) Scenario 1 (b) Scenario 2 (c) Scenario 3

(d) Scenario 1: TCP thpt loss (e) Scenario 2: TCP thpt loss (f) Scenario 3: TCP thpt loss

Figure 4.10: Three sample "worst-case" interference scenarios for a sidewalk environment and the TCP throughput loss due to interference. Two 60GHz basestations are deployed on lampposts (6m in height) on the edge of the side walk, each serving a user on the sidewalk.

repeats the same pattern. By examining the patterns of peak/valley across all the speeds tested (1.5-5mph), we see that the radio alignment frequency aligns with the above UDP results, *e.g.* once every 2-7s.

In summary, our experiments on Wilocity radios show that today's low-cost 60GHz radios already offer online array alignment to handle moderate user motions. However, these radios are designed for quasi-static indoor environments, and the speed and amount of their beam adjustments are not enough to handle certain motion patterns that introduce fast/large changes in link directionality.

Alignment Requirements for 10×10 Arrays. The above discussion triggers another question: "*As the link becomes more directional by using larger arrays, how fast should array alignment take place to handle user motion*?" As already discussed, the horn antennas on the HXI radios serve as a good proxy for 10×10

arrays. Hence, to answer this question, we mount the HXI radios on mechanical rotators (with a precision of 0.5^o) and manually adjust beam directions as the user moves. Each experiment starts with the two radios perfectly aligned. As the user moves, we measure the RSS values of three scenarios: adjusting both TX and RX arrays (full alignment), only adjusting RX array (partial alignment), and no alignment. From these results we estimate the alignment speed required to support typical user motion. Figure 4.9 plots the link rate projected from the RSS values measured during motion. We make two key observations.

- *Full alignment is required to handle user motion*, which is particularly true for movements towards and perpendicularly to the basestation. It involves both ends and requires a coordination channel between the basestation and the user.
- *Re-alignment once every second is sufficient for local movements.* Again perpendicular movements lead to the fastest signal degradation. Without alignment, the transmitter and receiver beams deviate from each other gradually, *e.g.* 3.6^o when the user moves 2.5m. The RSS drops by 2.47dB and the link throughput to half. At a moderate motion speed of 3-5mph, moving 2.5m takes 1.8s-3s. Thus at a link distance of 40m, re-aligning once every second should maintain quality connections during local movements. We also study the re-alignment frequency at other link distance values, and obtain similar findings.

Interference and Spatial Reuse

We now study the spatial reuse provided by 60GHz links. While averaged performance over large-scale deployments is required to accurately quantify spatial

reuse, we use testbed measurements to explore several worst-case interference scenarios when 60GHz picocells are deployed in a street environment. We also examine how the choice of arrays affects the degree of spatial reuse.

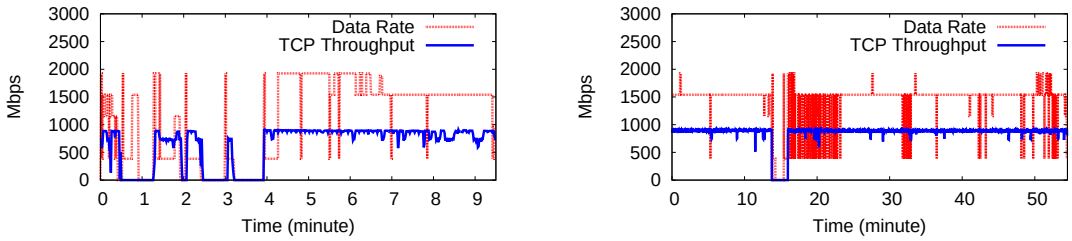
Interference Scenarios. Intuitively, two directional links interfere when their directions are too similar to each other. Hence, in the context of 60GHz picocells, we consider in our experiments three example "worst-case" downlink interference scenarios, shown in Figure 4.10. All three scenarios involve 2 basestations (BS_a, BS_b) deployed on lampposts at the edge of the side walk next to the street, each serving a user (a and b) on the sidewalk, using the same 60GHz channel. In each scenario, the interference condition depends heavily on the angular separation between the two links, which is defined by the distance between the two basestations (D), user a's horizontal distance to BS_a (d) in scenario 1, and the sidewalk depth (l) in scenario 2 and 3. In our experiments, we consider $l = 7m$, modeling major urban streets like those in Manhattan.

Since these interference measurements require at least two links, we used Wilocity radios in our measurements (we only have two HXI radios or one link due to cost constraints). To estimate the performance of larger arrays (thus narrower beams), we mount a metal horn-shaped case on the Wilocity device to reduce its beamwidth to roughly approximate a 10×10 array. However this also eliminates side-beams so our larger-array results are rather optimistic. Later in §4.4 we use network simulations to further study this issue. Finally, for all the experiments, we place radio absorbers to ensure that both links do not change their path directions.

Findings. We plot in Figure 4.10(d-f) the normalized TCP throughput loss due to interference. We make two key observations. *First*, the low-cost 2×8 arrays have wide beams that result in large interference. The two basestations must be widely separated to avoid inter-basestation interference: 20-22m for scenario 1 and 2, and >>22m for scenario 3. But upgrading to a larger array can potentially reduce basestation separation to 12-14m for scenario 1 and 2, and 22m for scenario 3, enabling dense basestation deployment. *Second*, the use of reflected paths (for overcoming blockage) creates diverse interference conditions beyond those of LoS. In certain scenarios, even using a narrow beam array may not eliminate interference. This calls for *coordinated scheduling*, where basestations coordinate user scheduling to proactively avoid heavy interference.

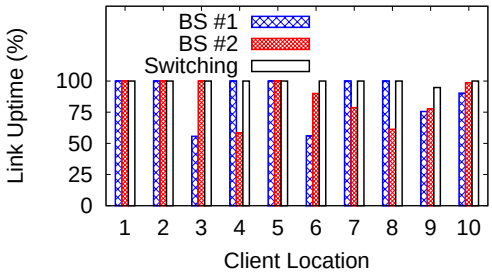
4.3.2 Measurements in Real-life Environments

Our experiments took place in two sites: a downtown street walkway area and a campus plaza area, both during busy hours with many moving pedestrians. These "in the wild" experiments differ from the "controlled" experiments as follows. *First*, the user is surrounded by *random moving pedestrians* that create unpredictable blockage. *Second*, we use Wilocity radios and leverage its built-in array adaptation mechanisms to maintain link connectivity. *Third*, we deploy multiple basestations around the area to study whether switching between basestations can effectively improve connectivity. At both sites, we placed basestations on a 2nd floor balcony of 6 meters in height, and experimented at 10 user locations (6 at downtown, 4 at campus) and 2 basestation locations. At each location, we let the user face the basestation(s), thus her own body did not block the LoS



(a) TCP/link trace, a downtown sidewalk point

(b) TCP/link trace, a campus plaza point



(c) Link Availability

Figure 4.11: The impact of pedestrian blockage in real-life environments. (a) A 10-minute segment of TCP/ link rate traces at a downtown sidewalk location. (b) A full 1-hour trace at a campus plaza location. (c) Link availability at 10 locations w/ and w/o basestation switching.

path. We then recorded TCP throughput and link layer rate (reported by the radio driver) for 1 hour per location.

Single user, single basestation. We first consider the scenario where a single user is being served by a single basestation. To illustrate the impact of pedestrian blockage on link connectivity, we plot in Figure 4.11(a)-(b) the TCP and link rate traces at two sample user locations at the edge of the walkway, with a roughly 12m link distance. In Figure 4.11(a), the connection experienced outage (both link rate and TCP rate drop to 0) several times during the first 250s.

During this time period, we observed a continuous crowd of pedestrians passing by, creating prolonged blockage to the user. After that, the link rate fluctuates (due to transient pedestrian blockage and subsequent rate adaptation), but the TCP throughput remains stable[5]. In fact, when a small group of 2-3 pedestrians pass by the user, they only create transient blockage. The link rate drops largely but quickly recovers.

In Figure 4.11(b), between minute 13 and 23, there is a 2-minute outage followed by a period of rate fluctuations. This 10-minute period was in between two class sessions, thus large groups of students passed by our user location. Different from the last example, the link was able to recover and sustain high TCP throughput after the two-minute outage, despite the prolonged blockage. This is because the radios switched to a reflected path (via a nearby building wall) to avoid blockage, and the frequent fluctuation was the result of rate adaptation to mitigate reflection loss. As a result, the link didn't fail repeatedly like in the previous example. In the rest of the hour, the pedestrian density was moderate and only caused transient blockage thus zero outage.

Figure 4.11(c) shows the link availability, *i.e.* (1-outage), over the busiest 300s across all the measured user locations (thus the period with the lowest availability), for the two basestation locations. The availability varies largely across user locations, and can be as low as 50%. The low availability scenarios are those where the user was on a crowded walkway and blocked by many moving pedestrians.

Single user, multiple basestations. We then examine scenarios where two basestations coordinate to serve a single user. To do so, we place the second

[5]The Wilocity TCP throughput is capped to 1Gbps, due to the 1Gbps Ethernet interface that connects the radio to the laptop.

basestation 10m-20m away from the first basestation[6]. Because Wilocity radios do not allow clients to switch between basestations on-the-fly, we emulate real-time switching by connecting a second receiver (co-located with the first receiver) to the second basestation on a different 60GHz channel. This allows us to simultaneously measure for each location its link availability to each basestation, and calculate the link availability when enabling real-time switching. Results in Figure 4.11(c) confirm that serving a user with a picocloud consisting of multiple basestation can effectively boost link availability.

4.3.3 Summary of Observations

Our detailed measurements dispel the four common myths about 60GHz outdoor picocells listed in §5.1. We now summarize our findings and their implications on picocell design.

- **Range:** While today's low-cost radios have limited outdoor range due to its low EIRP and small array size (designed for indoor scenarios), larger antenna arrays, *e.g.* 10×10, can support a range up to 100+m at 385Mbps even in heavy rain conditions.
- **Blockage:** The user body, other pedestrians, and other obstacles can all block connections to a user. Hence it is critical to maintain a rich path inventory for each user by continuously tracking both LoS and various reflection paths. When necessary, users can be served using a picocloud comprising of multiple basestations (supported by results in §4.4).

[6]We did not observe any notable impact of the separation distance.

- **User Motion:** Local user motion changes 60GHz link directionality and degrades performance, unless both endpoints realign their arrays in real time. Since they were designed for quasi-static indoor settings, realignment on Wilocity radios can take up to 7 seconds, making them less robust against certain motion patterns. For larger arrays (10×10) with narrower bandwidth, realignment frequency needs to be at least once per second.
- **Spatial Reuse:** Today's low-cost 2×8 arrays have wide beams that limits the degree of spatial reuse. Using larger arrays up to 10×10 can significantly reduce interference footprints, but the large variety of path directions can still lead to considerable interference in certain scenarios. Handling these requires intelligent user scheduling and basestation coordination. While we will provide initial insights on these issues in §4.4, far more research is needed in this area.

4.4 Simulation Results

In this section, we perform detailed simulations to further examine 60GHz picocells in terms of blockage and interference.

Simulation Setup. We consider a street in a typical urban environment: two basestations are deployed on opposite sides of the street (at lamppost height of $6m$), forming a picocloud to serve users on both sidewalks. Figure 4.12 plots the geometry of a typical Manhattan street observed from the Google map. Our simulation considers pedestrians, cars and trees as obstacles that can potentially block or attenuate 60GHz links. We model each pedestrian as a cylinder of radius $0.3m$ with height uniformly distributed between $1.5m$ and $2m$, and trees as cyclinders with radius $0.4m$ and height $3m$. We model cars as rectangular

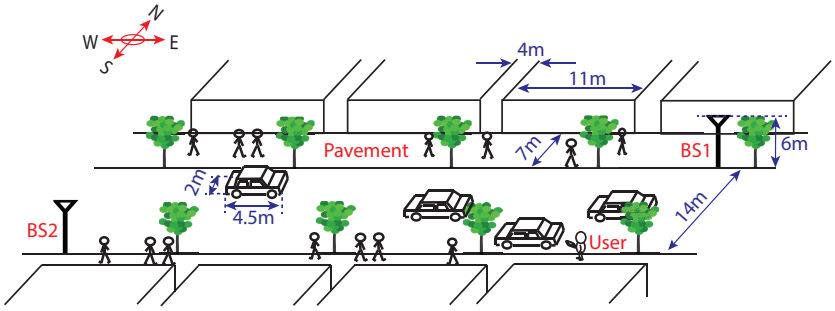


Figure 4.12: Simulation environment: a typical street in New York City (observed from the Google Map), where two basestations serve users on both sidewalks along the street.

cuboids of size $2m \times 4.5m \times 1.5m$. We consider a moderate pedestrian density of $0.08 person/m^2$ [91].

The link budget in our simulations is as in eq. (4.1)-(4.2) defined in §4.2.1, also consistent with the measurement-based feasibility analysis in §4.3. We consider regular square arrays with standard radiation patterns [56], varying the size of the transmit and receive arrays around default settings of 10×10 at each end. We fix the EIRP to 40dBm. Thus, the size of the transmit array does not impact the link budget, but can have a significant effect on interference. The receive array size impacts both link budget and interference. The 60 GHz channel is well characterized by a small number of dominant rays (see [153] and references therein), with path strengths decaying rapidly with multiple bounces. Hence, in our model, we consider the LoS ray and one-bounce paths.

4.4.1 Link Availability

We consider several factors that may affect link availability. The first is user orientation, *e.g.* when facing basestation 1, the user's body blocks the LoS path

from basestation 2. We study its impact by comparing link availability among four possible orientations: north (facing far wall), south (facing near wall), east (facing basestation 1), and west (facing basestation 2). In general, the wall-facing orientations do not have LoS blockage, and the LoS availability is impaired mainly for the east-west orientations. We omit orientation-specific results for brevity; our later results are averaged over these four orientations. The second factor affecting link availability is blockage by other obstacles, *e.g.* other pedestrians, trees, and cars. We consider a moderate pedestrian density of 0.08 per square meter[7]. We find that, in general, cars are not tall enough to block LoS and wall reflections to users on sidewalks, but may disrupt ground reflections. Hence our simulations do not utilize any ground reflections from the street or sidewalk. Finally, the third factor is the presence of gaps between buildings (see Figure 4.12), which may eliminate possible reflected paths.

Taking into account these factors, we run simulations to estimate the distribution of RSS at different receiver locations, accounting for different receiver orientations and locations of other pedestrians and cars. We use this to compute the *availability rate* for three 802.11ad data rates listed in Table 4.1 (a data rate is "available" if the RSS exceeds its required receiver sensitivity).

Benefits of Picoclouds and Reflected Paths. Table 4.3 summarizes the availability rates for four scenarios: a mobile is served by (i) only the LoS path from basestation 1, (ii) a path among the LoS and 2 wall-reflected paths (if feasible) from basestation 1, (iii) only the LoS paths from basestation 1 and 2, and (iv) a path among the total 6 possible paths from basestation 1 and 2. This result as-

[7]We obtain similar results for much higher densities since it turns out that blockage of user body is the key source of link impairment.

Data rate (Mbps)	**385**	**1155**	**2310**
LoS (BS1)	66.0%	66.0%	66.0%
LoS + 2 NLoS (BS1)	94.2%	93.8%	93.5%
LoS (BS1& BS2)	94.5%	94.5%	94.5%
6 paths (BS1&BS2)	100%	99.95%	99.95%

Table 4.3: Effect of picocloud and reflection paths on availability, 10×10 RX array, 100m separation between the two basestations.

RX array	BS spacing	Clear			Rainy		
		Data rate (Mbps)					
		385	1155	2310	385	1155	2310
4×4	100m	99.9%	99.5%	99%	99.8%	99.4%	98%
	200m	99.4%	94.6%	85%	98.3%	89.2%	75.7%
8×8	100m	99.9%	99.75%	99.6%	100%	99.75%	99.7%
	200m	99.5%	99.45%	98.9%	99.5%	99%	97.2%
10×10	100m	100%	99.95%	99.95%	99.9%	99.9%	99.8%
	200m	99.5%	99.5%	99.3%	99.5%	99.5%	98.8%

Table 4.4: Availability rates (blockage scenario).

sumes a basestation spacing of 100m, but we observe similar conclusions for 150m and 200m. This result shows that using LoS path from a single basestation cannot provide robust connectivity. But when both basestations are involved, the link availability jumps from 66% to 94.5% at 385Mbps. Adding just two wall-reflected paths further increases the availability rate to 99.95-100%.

Basestation Spacing and RX Array Size. Assuming a picocloud of two basestations and the use of both LoS and reflected paths, Table 4.4 reports the availability rate for different basestation spacing and RX array size. We see that even heavy rain (50mm/hr) barely has any impact. For the 10×10 RX array and a BS spacing of 200m, we have high availability even at 2 Gbps (99.3% in clear,

and 98.8% in rainy conditions). This implies that even with a sparse deployment, the network offers robust connectivity and coverage.

4.4.2 Interference and Spatial Reuse

We report on some preliminary results that indicate the massive spatial reuse achievable in our architecture. To this end, we lower the basestation spacing to 20m, and consider mutual interference between two users, each served by one of two basestations forming a picocloud. In addition to using different receive arrays, we also consider 10×10 and 32×32 transmit arrays. The EIRP is fixed, hence the transmit array size does not impact the link budget, but can have a significant effect on interference.

We first consider a scenario with "multiuser diversity": there are five users, out of which two are picked at any time. We consider two strategies (neither of these is optimal, but they suffice to illustrate the impact of interference). In the first, termed without coordination (WoC), the basestations do not account for interference in scheduling: the first basestation serves the user to which it has the best path, and the second chooses the user to which it has the best path among the remaining 4 users. In the second strategy, termed With Coordination (WC), the first basestation chooses the user for which it has the best path (as before), but the second basestation chooses a user to maximize the minimum SINR among the two chosen users. Our simulations of availability are as before, except that we now also account for the presence of interference in determining receiver sensitivities. We find that the availability is 100% for all antenna combinations, with or without coordination! Basically, noise is not a limiting factor at this shorter range, and

RX array	Mode	TX array 10×10			TX array 32×32		
		Data rate (Mbps)					
		385	1155	2310	385	1155	2310
4×4	WoC	70%	60%	55%	89.5%	79.5%	72%
	WC	75%	68%	62%	89.8%	82.6%	76%
10×10	WoC	94.3%	91%	81%	98%	96.8%	93%
	WC	94.7%	91%	84%	98%	96.8%	94.8%

Table 4.5: Availability rates with blockage and interference, where the user pair are co-located in between the two basestations.

RX array	Mode	TX array 10×10			TX array 32×32		
		Data rate (Mbps)					
		385	1155	2310	385	1155	2310
4×4	WoC	59%	48%	46%	73%	58%	52%
	WC	66%	59%	56%	77%	66%	61%
10×10	WoC	90%	81%	75%	92.5%	90%	85%
	WC	91%	85%	81%	93%	91%	88%

Table 4.6: Availability rates in the offset geometry.

a user configuration that is "bad enough" (to produce large interference despite beamforming at both ends) is a rare event. We plan to develop an analytical framework for understanding the probabilities of such rare events in future work.

Next, we consider a "worst-case" scenario where two users next to each other are being served by the two base stations. Table 4.5 shows availability rates, averaging over the location of the user pair between the basestations, and their individual orientations. Here the transmit array size has a large effect. Using the larger array drops the outage rate from 30% to 10.5% for the smallest 4×4 receive array. For the 10×10 receive array, the main effect of using a larger transmit array is to increase the availability rate for the 2 Gbps rate.

The performance gets slightly worse in the "offset" geometry depicted in Figure 4.10(a), where the user pair is no longer between the two basestations. The

availability in Table 4.6 drops to 93% even for the largest transmit and receive arrays considered. Of course, these worst-case settings are unlikely for randomly placed users, and can be avoided if we have multiuser diversity (so the picocloud does not have to simultaneously transmit to pairs of nearby users).

These results, despite their preliminary nature, indicate the promise of massive spatial reuse provided by the use of narrow beams at both transmit and receiver, and motivate more detailed analysis and simulations for characterizing the nature of interference in both average and worst cases.

4.5 Related Work

mmWave Cellular Networks. Picocellular networks in the mm wave bands have attracted significant attention recently. Much recent effort has focused on trying to attain ranges of the order of 200m or even more, thus considered bands such as 28GHz, 38GHz and 73GHz to avoid oxygen absorption that peaks at 60GHz (see [110] and a recent survey [117]). Recent work also proposed to increase range by combining signals from many NLoS paths incoherently [24].

Our proposed 60GHz picocell design differs from existing efforts by taking a different perspective. Aiming for short links (50-100m)[8], the oxygen absorption at 60GHz barely dents the link budget while still cutting down on interference from far-away basestations and hence significantly increasing spatial reuse. Together, the large per-user data rate and large spatial reuse can potentially enable the 1000-fold increase in network capacity. Furthermore, our proposal leverages current and

[8]Mobility management becomes difficult at short ranges. This can be potentially addressed by the picocloud architecture, but still remains an open challenge that demands further research study.

ongoing 60GHz hardware technologies and the IEEE 802.11ad standard developed for indoor applications, while the unlicensed nature of 60GHz bands lowers the entry barrier and promotes innovation.

Outdoor 60GHz Systems. The outdoor use of 60GHz is not new: there are point-to-point Gigabit links on the market today (e.g., [8]), attaining link ranges of kilometers using carefully installed parabolic dishes. These fixed beam antennas have significantly higher directivities than the electronically steerable antenna arrays envisioned in our architecture. Recent work has examined 60GHz mesh networks with electronically steerable links for backhaul [135, 136]. Finally, the impact of blockage on short-range outdoor 60 GHz links has been explored independently in [21, 22] using ray-tracing simulations. Unlike our work, this study is not grounded in measurements, and does not consider the detailed tradeoffs in the picocloud architecture.

60GHz Measurements. There have been significant efforts in 60 GHz channel measurement and modeling both for indoor [92, 51] and outdoor settings [137, 145, 83, 119], confirming the accuracy of quasi-optical propagation models on 60GHz propagation. Most measurements used channel sounding and probing hardware, while others have cleverly adapted commercial radios to indoor measurements [143, 158].

While prior measurement campaigns focus on link level characterization and propagation modeling, to the best of our knowledge, our work is the first measurement-centric study aimed specifically at design and validation of outdoor 60GHz picocell architectures.

4.6 Conclusion

Our measurements and simulations dispel some common myths on 60 GHz outdoor mobile communication, and show that outdoor 60 GHz picocells augmenting existing cellular networks indeed have the potential to deliver orders of magnitude increase in network capacity. Our findings lead us to identify the need for new, novel architectures and system designs such as "picoclouds," as part of a new research area on 60GHz outdoor networks. We conclude by discussing some of the significant challenges that remain.

Cross-layer Modeling. In our experiments, off-the-shelf hardware provides insight into propagation, blockage and interference in simple scenarios. However, today's hardware limitations imply that future research involving more complex scenarios must rely on simulations until more capable platforms (*e.g.,* with externally programmable beamsteering) become available. Therefore, significant effort is necessary to develop models at different layers accurate enough to capture the cross-layer interactions unique to 60 GHz communication.

Specifically, the lowest layer involves tying the physics of propagation and diffraction to link-level models. This includes link availability and data rate as a function of pedestrian configurations, for group motion as well as independent individual motion. Another layer involves signal processing models and algorithms that account for hardware constraints, *e.g.*, RF beamforming with a large number of antenna elements. We can then build to higher layers, including abstractions for user tracking and basestation coordination, and network-level resource management for seamless support of TCP flows.

User Tracking. To establish/maintain links, basestations must track users continuously and build a rich path inventory. GPS location works only for LoS paths, but its localization error is too large to enable highly directive beams. Brute-force beam scans incur large overhead, which prevent online adaptation. A potential solution is compressive adaptation [116], where basestations periodically send 60GHz beacons in random beam directions, and then leverage user feedback to estimate directions of available paths to each user. However, significant effort is required to translate such ideas to practical system designs.

Control Plane. A robust control plane is needed for coordination between basestations and users during bootstrapping, handoff and array (re)alignment. A potential candidate is to use existing LTE services (Figure 4.1) for this purpose. The resulting control channel is reliable, but the design must be robust to latencies as high as 80ms [66], especially if LTE bands are congested.

Picocloud Architecture & Protocol Design. In the picocloud architecture, basestations must coordinate to maintain user connectivity and minimize interference. This requires new, cross-layer design spanning the PHY, MAC and network layers. Frequent switching between basestations to handle blockage and mobility, together with rate adaptation switching between different paths, requires careful coordination to support applications such as TCP. Given the high frequency of blockages, it is particularly important to avoid naive handoff strategies that could lead to ping-pong effects and race conditions. Such considerations, together with the large per-user rate, call for careful designs that account for backhaul and storage constraints.

Hardware Design. Advances in system-on-chip design enable packing multiple radios into mobile devices. The integration of 60GHz radios poses little difficulty in terms of interference with existing radios, since the frequency band is so far from current WiFi and cellular frequencies. However, another consideration is the integration of (possibly multiple) antenna arrays to receive signals effectively regardless of the device orientation. Limiting radio energy cost for mobile devices is another. While signal processing and RF design for high-speed mmwave links are challenging, our advantage lies in leveraging existing and ongoing hardware development for IEEE 802.11ad.

Chapter 5

Capturing User Mobility at Scale

5.1 Introduction

For quite some time, the holy grail of mobile networking research has been the search for large-scale, fine-grained mobility traces of human movement. Such traces can provide the basis for a large range of applications, ranging from practical applications like traffic prediction, urban planning, and network management, to research applications like input for mobile network simulations and design of cellular protocols and smartphone energy management systems. Unfortunately, access to such mobility traces has been elusive for both privacy and logistical reasons. This has led to numerous alternatives, including synthetic mobility models ranging from random waypoint [70] to obstacle-based models [69], to adoption of unconventional sources of mobility datasets, *e.g.* movement traces from the SecondLife virtual reality platform [80].

Most recently, attention has turned towards geosocial mobility traces, traces of "check-in" events gathered from location based social networks (LBSN) such as Foursquare, Gowalla and Brightkite [41, 102]. These datasets are attractive

as mobility traces, because they are easy to obtain, and provide data for large user populations, *e.g.,* Foursquare has 40 million users with more than 4 billion checkins. In fact, researchers are already relying on geosocial mobility traces to predict human movement [42, 130, 103], infer friendships based on visited locations [25, 129], and improve the efficiency of content delivery networks [131].

So is this it? Have we solved our long running quest for realistic mobility traces? Before accepting these geosocial traces as ground truth, we need to address critical questions of validity and representativity. For example, just how reflective are "check-in" traces of our true mobility patterns, and how significantly would any potential discrepancies impact applications and systems that rely on these datasets? Furthermore, can we correct any inherent biases and produce realistic mobility traces for public consumption?

In this chapter, we take concrete steps towards addressing these questions. First, we examine potential biases in geosocial traces, by comparing a GPS-generated mobility trace to a Foursquare checkin trace[1] for identical users in a large user mobility study. Surprisingly, by comparing Foursquare "checkin" events to visits in the GPS trace, we find that there is only a small subset of common events in two traces: 75% of all Facebook checkins are extraneous or fake events that do not match real mobility, and validated Foursquare checkins only cover roughly 10% of all "visit events" in a user's actual movements.

Our next step is developing an automated *two-phase sanitization* process to correct these biases and produce mobility traces much more representative of true mobility patterns. First, we characterize distinctive features of extraneous (fake) checkins and apply machine learning techniques to identify and filter extrane-

[1]We focus on Foursquare checkins due to its popularity.

ous checkins. Second, we leverage interpolation techniques to fill in the missing locations visited by but not reported by users. By applying it to a publicly available trace of geosocial checkins from the Gowalla network, we demonstrate that our two-phase sanitization can effectively mitigate the negative impact of biased geosocial traces and produce realistic human mobility traces.

The key contributions of this work are as follows:

- First, we perform a large user mobility study on Foursquare to quantitively examine the discrepancies between geosocial checkin traces and ground-truth user mobility (GPS traces). We find geosocial traces are heavily biased, with 75% of all checkins being extraneous (fake), and real checkins only capturing 10% of users mobility events. We also find clear correlations between extraneous checkins and Foursquare's social incentives, *i.e.* Badges and Mayorships.
- Second, we propose a two-phase sanitization technique to correct biases in geosocial traces, including machine learning models to identify and filter extraneous checkins, and interpolation techniques to statistically "recover" missing user visit events. We use a trace-driven application to show that these biases have significant impact on applications, and that they are effectively removed by our sanitization process.
- Third we empirically apply our data sanitization mechanism onto a different geosocial dataset from Gowalla. This is possible because our sanitization process uses high level principles gathered from our measurement study: a) observations of general behavioral patterns that generate extraneous checkins, and b) general statistics of human mobility patterns and distributions. We demonstrate that our techniques can be used by third-parities to sanitize their own geosocial traces in practice.

- Finally, we recognize that our approach of capturing user mobility via modified geosocial traces is not suitable for all mobile applications. We explore this limitation in detail and discuss the suitability for a range of applications.

Our study follows our preliminary work [155] on the discrepancies between geosocial datasets and GPS datasets, but makes significant new contributions on how to sanitize the geosocial datasets for the good of third party applications and research communities. To the best of our knowledge, this is the first deep study of such problems and the first to offer both tools and sanitized datasets to the research community.

5.2 Methodology & Datasets

Here, we briefly introduce background on LBSNs, explain our methodology, and describe datasets used in our study.

5.2.1 Background & Motivation

Today's LBSNs allow users to share their social activities along with their locations, *e.g.* checking in to nearby Points-of-Interest (POI). The most popular LBSN is Foursquare, with over 40M registered users and 4 billion "checkins" (as of September 2013). Other popular sites include US-based Yelp and Gowalla (now defunct), and China-based JiePang.

LBSNs incentivize user checkins using (virtual) rewards. For example, the FourSquare user who checks-in to a location most frequently over a 60-day window is designated the "Mayor." In addition, users obtain "badges" for achieving certain

requirements, *e.g.* checked-in to five different coffeeshops. Other LBSNs have similar incentives.

Despite their growing acceptance as substitutes for large-scale mobility traces, little is known about how well these datasets actually reflect our true mobility patterns. Recent work has raised concerns of fake checkins and user biases, but there is no data on the severity of the problem [17, 86, 47]. Our goal is to not only use detailed measurements to examine potential biases in geosocial mobility traces, but to develop techniques to correct these biases, thus producing a realistic mobility trace that matches physical movement. Specifically, we answer four key questions.

- How well do datasets of LBSN check-in events reflect users' true mobility patterns?
- What caused any discrepancies between the trace and real mobility?
- How do we sanitize the geosocial datasets to make them useful to these applications and systems?
- Can our techniques be generalized and applied to "curate" existing LBSN datasets?

5.2.2 Methodology

Our approach consists of three steps: collecting real-world checkins and true mobility traces, identifying discrepancies between the two datasets, and finally sanitizing the checkin datasets to reflect true mobility patterns. We summarize our methodology, corresponding challenges, and our solutions.

Collecting checkins & true mobility traces. Our first step is to collect parallel physical mobility (GPS) traces and Foursquare checkin traces for a single set of users. For this, we developed a smartphone app that simultaneously collects a user's Foursquare checkins and her fine-grained (*i.e.* per-minute) GPS location data (with consent). In six months, our app attracted 200+ Foursquare users worldwide, with nearly 3 million GPS entries and 14,000+ checkin entries. We discuss more details on the dataset in §5.2.3.

Identifying & understanding discrepancies. Next, we compare "checkin" events from the Foursquare dataset to "visit" events extracted from the GPS trace (§5.3). We develop a spatio-temporal matching algorithm that identifies a matching visit event for each checkin event. We ensure that the algorithm is robust against measurement noise and human artifacts, and validate its accuracy using data from a controlled user group. Our analysis reveals that 75% of Foursquare checkins do not match any visits in the GPS dataset (i.e. *extraneous checkins*), while 90% of actual visited locations in the GPS dataset are missing from the checkin trace (*missing checkins*).

Next, we identify potential causes and motives for both extraneous and missing checkins. By examining the data, we find that "extraneous checkins" are easily explained by a small number of prototypical "cheating" behaviors that are well motivated by FourSquare incentives. Similarly, data analysis reveals intuitive reasons for why users neglect to checkin to "missing checkins."

Curating LBSN traces. Our goal is to "curate" LBSN datasets to remove its biases and making it representative of real user mobility. This is highly challenging, given the small overlap between the checkin and GPS datasets. Our approach

is a two-phase process, where we first detect and remove extraneous checkins (§5.4), then interpolate the remaining checkin dataset to recover missing checkins (§5.5). Detection involves identify a list of promising features that distinguish extraneous checkins from honest ones, and use these features to build an effective classifier. Interpolation requires accurately predicting the number of missing checkins given the current dataset, and then interpolating new visit events, such that the temporal and spatial distributions of the interpolated dataset matches distributions in the GPS trace.

5.2.3 Datasets

Our study uses three groups of datasets, two collected by us, and one publicly available online.

Data Collection. Our goal is to collect matching physical mobility (GPS) traces and Foursquare checkin traces for a single set of users. We built a measurement smartphone app for both Android and iOS which generates two matching traces: a per-minute GPS trace of the user's location, and a trace of the user's checkin events polled using Foursquare's open API. We process the GPS trace to detect "visits" to points of interest (POI), and define a visit as the user staying at one location for longer than some period of time, *e.g.* 6 minutes[2]. When GPS signals are not available, *e.g.* indoors, the app uses the phone's WiFi radio and accelerometer to determine if the user is stationary or moving, similar to [74]. We obtained human subject IRB approval for our study.

[2]While our mobile app can only detect stays longer than 6 minutes, we found that using larger thresholds, *e.g.* 7−20 minutes just reduces the number of stays slowly, and have negligible impact on our later findings.

Datasets	# of users	Avg days per user	# of checkins	# of visits	GPS points
Primary	244	14.2	14K	31K	2.6M
Baseline	47	20.8	665	6.3K	558K
Gowalla	107K	90.0	6.4M	N/A	N/A

Table 5.1: Statistics of the datasets.

Primary Datasets. Our main dataset comes from Foursquare users who installed our app from Google Play and Amazon App stores. Between January and July 2013, we received data from 244 users worldwide, where each user's measurement data covered an average of 14 days. This produced two mobility traces (Table 5.1):

- **Checkin Trace** contains 14,297 Foursquare checkins events. Each event includes a timestamp, the name of a POI, its venue category and GPS coordinates.
- **GPS Trace** contains 2,600,000 sets of GPS coordinates. It captures each user's GPS location on a per-minute basis, and a list of 30,835 POI "visits."

These users are "organic," *i.e.* Foursquare users who started to use our app voluntarily. Their geographic distribution matches those of Foursquare, *i.e.* the top 3 areas (USA, Brazil and South Asia) contribute to 90%+ of the total users in both datasets [15]. Furthermore, we compared our checkin trace with existing Foursquare checkin traces collected by prior work [23, 102], and found consistent statistical properties, including distributions of per-user checkins, inter-checkin times, numbers of badges, numbers of friends, and user nationality. This lends support to our belief that our dataset is reasonably representative.

Baseline Dataset. We also collected a smaller dataset (also 2 parallel traces) by recruiting 47 undergraduate volunteers from the communication department at

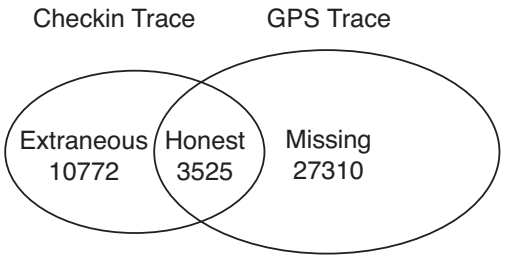


Figure 5.1: Matching results of the Primary dataset.

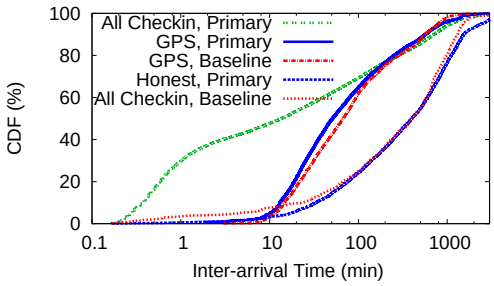


Figure 5.2: Inter-arrival time.

Checkin Type	Correlation			
	#Friends	#Badges	#Mayors	#Checkins/Day
Superfluous	0.22	0.07	**0.34**	0.15
Remote	0.18	**0.49**	0.16	0.15
Driveby	-0.10	-0.21	-0.08	**0.21**
Honest	-0.09	-0.42	-0.23	-0.40

Table 5.2: Correlation between users' profile features and ratio of each type of checkins.

our university. These students installed our smartphone app and used FourSquare normally. Later, we will use this dataset as a control group to validate the results of our checkin classification mechanisms.

Gowalla. Gowalla is a LBSN similar to Foursquare, allowing users to check in at the places they visit. The dataset[3] contains 6M checkins from 107K users collected from Feb 2009 to Oct 2010. It does not contain any true mobility trace data. We will use this dataset to further examine our data curation mechanism (§5.6).

[3] `http://snap.stanford.edu/data/loc-gowalla.html`

5.3 Validating Checkins

Using our checkin and GPS datasets, we now validate how well Foursquare checkins correspond to each user's physical mobility patterns. For this we develop a spatio-temporal matching algorithm that "matches up" each Foursquare checkin POI to a visit in the GPS trace, and use the matching result to quantify the discrepancies between the two datasets. We also perform detailed analysis to identify driving reasons behind these discrepancies, and describe our plan to correct them.

5.3.1 Matching Checkins to Visits

We first introduce the matching algorithm, and then discuss the matching results and our findings.

Matching Algorithm. Our algorithm matches a user's Foursquare checkins to her GPS visits based on the associated GPS coordinates and timestamps. To minimize the impact of measurement noises due to inaccurate GPS reports and minor time offset between a checkin and the corresponding physical visit, we apply thresholds to control the precision of data matching.

- **Spatial Matching:** For each checkin event C_i in a user's Foursquare trace, we identify from the same user's GPS trace a set of visits $\{V\}$ whose physical locations are within α meters from C_i's location. $\{V\}$ can contain one or multiple visits, or the null set.

- **Temporal Matching:** If $\{V\}$ is non-null, find the visit v_j from $\{V\}$ whose timestamp is closest to that of checkin C_i. If the difference between the two timestamps[4] is less than β, then v_j matches c_i.

Ideally, there would be a one-to-one mapping between checkin events and GPS visits. Our algorithm ensures that each checkin event has at most one matching visit. If multiple checkins are matched to the same visit v_j, (*i.e.* a user checks in at multiple POIs when visiting one), we match v_j to the geographically closest checkin event.

We have experimented with a wide range of α and β values, and found that the results are most consistent for values $\alpha = 500m$ and $\beta = 30min$. Our approach is designed to capture an upper limit on possible event matches. Thus both values are intentionally generous to increase the probability of matching checkins to visits.

Matching Results. We first ran the matching algorithm on the Primary dataset with ordinary Foursquare users. Its checkin trace contains 14297 checkins while the GPS trace includes 30835 visits. Figure 5.1 shows the matching result as a Venn diagram.

- **Honest Checkins**: 3525 checkins events match up with GPS visits. These checkins events correspond to GPS readings that show the user was indeed at the physical location matching her Foursquare checkin event. This represents a very small portion of both checkins and GPS visits.
- **Extraneous Checkins**: 10772 checkin events (75% of total checkins) do not match any visits in the GPS trace.

[4]Each visit has a start time T_s and an end time T_e. We calculate Δt, the timestamp difference between v and a checkin with timestamp (T_c) as follows: if $T_s \leq T_c \leq T_e$, $\Delta t = 0$; Otherwise, $\Delta t = min(|T_c - T_s|, |T_c - T_e|)$.

- **Missing Checkins (or Unmatched Visits)**:27310 visits in the GPS trace (89% of all visits) do not match any Foursquare checkins.

We wish to validate that our "honest checkins" are an accurate representation of real checkin activity. For this, we compare it against our Baseline dataset of undergraduate volunteers. Since these volunteers were participating to fulfill a research requirement, they were much less likely to be influenced by Foursquare rewards. We use several mobility metrics to compare the two datasets, including inter-arrival time distribution, movement distance distribution, event frequency, speed distribution and POI entropy [42, 41].

Figure 5.2 plots the inter-arrival time distribution results of GPS and checkin traces from both datasets, as well as the honest checkins from the Primary dataset. It shows that GPS traces from both datasets match up near perfectly. In addition, the entire checkin trace from the baseline matches up perfectly with the honest checkin set from the primary dataset, while the set of all checkins from the primary data shows significant differences. Other metrics led to the same conclusions (results omitted for brevity), thus confirming that our matching algorithm did accurately identify the set of honest Foursquare checkin events.

Overall, our matching result reveals shockingly large discrepancies between the checkin and GPS datasets. The discrepancies appear in both the raw data (Figure 5.1) and the statistical mobility patterns they represent (Figure 5.2). We confirm later in §5.5.3 that these discrepancies translate to signicant deviations in results of applications using these traces. Next, we study the underlying causes of both extraneous and missing checkins.

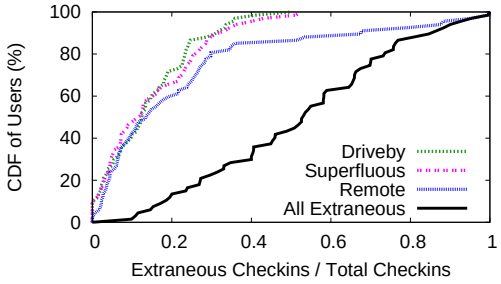


Figure 5.3: User's ratio of extraneous checkins.

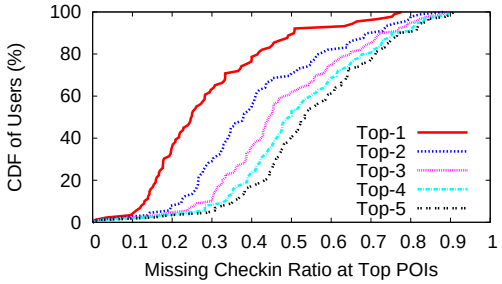


Figure 5.4: Ratio of missing checkins at top-n most visited POIs.

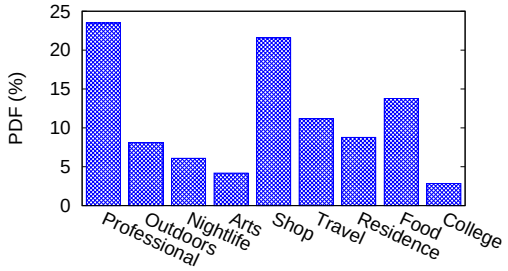


Figure 5.5: Breakdown of missing checkins by POI category.

5.3.2 Extraneous Checkins

Extraneous checkins (those without a matching visit) occur when users misrepresent their physical location. Here, we categorize observed extraneous checkins based on user behavior, then try to infer possible incentives motivating them.

Types of Extraneous Checkins. We manually inspected our pool of 10772 extraneous checkins, and found that 90% could be classified into one of three types of user behavior. The remaining 10% do not display any distinctive features.

- **Superfluous Checkins**. When visiting and checking in to one POI, some users also check in to multiple nearby POIs from the same physical location.

We found 2176 superfluous checkins in our dataset (15% of all checkins and 20% of extraneous checkins).

- **Remote Checkins.** These are checkins to POIs more than 500 meters away from a user's actual GPS location. 500m is beyond any reasonable GPS or POI location error, and the user is clearly falsifying her location. Our dataset has 5715 remote checkins (40% of all checkins and 53% of extraneous checkins).
- **Driveby Checkins.** These occur when users checkin to nearby POIs while moving at a moderate or high speeds. Computing speed from our GPS trace, we treat a checkin as driveby if its speed exceeds 4mph. There are 1782 driveby checkins (13% of all checkins).

Incentives for Extraneous Checkins. There could be many potential causes for extraneous checkins. To infer the key causes, we measure the correlation between a user's extraneous checkins and her user features, *i.e.* number of friends, number of badges, number of mayorships, and number of checkins per day. Table 5.2 lists the Pearson's correlation score between each user's features and the ratios of her checkins (superfluous, remote, driveby and honest). The score is between -1 and 1, where -1 means perfect negative correlation and 1 means perfect positive correlation.

We observe that both remote and superfluous checkins correlate strongly with Foursquare's reward mechanisms (badge and mayorship). This suggests that social rewards are key incentives for extraneous checkins. For example, to collect badges, Foursquare users need to checkin at various new POIs that could be far from their actual location. Similarly, mayorship of a POI requires more frequent checkins at that location than other users. This motivates her to check in even

when not physically visiting the POI. Foursquare allows remote checkins to count towards badges, but not mayorships. This is clearly visible from the correlation results. Since badges require checkins at specific POIs, superfluous checkins do not help. Finally, we note that honest checkins have negative correlation with all four features. This indicates that "honest" users tend to be less active (*i.e.* less checkins per day, less badges).

Our results show that superfluous and remote checkins dominate extraneous checkins, and are likely motivated by Foursquare's user badges and mayorships. Since these rewards play a large role in encouraging user engagement in Foursquare and other LBSNs, superfluous and remote checkins will likely remain a key component of geosocial traces.

Per-user Prevalence. If extraneous checkins are endemic to specific subsets of users, then we can focus on removing those users. We compute for each user the portion of her checkins that are extraneous. To our surprise, the CDF in Figure 5.3 shows that extraneous checkins are widespread. Nearly all users produced extraneous checkins, and for 20% of users, extraneous checkins accounted for up to 80% of their checkin events.

These results raise serious concerns about the prevalence of unreliable checkins in the user population. They also mean we cannot target specific users in an effort to filter out extraneous checkins, unless we are willing to sacrifice a significant number of honest checkins. Filtering out users who generate 80% of all extraneous checkins would also filter out 53% of honest checkins!

5.3.3 Missing Checkins

For missing checkins, the obvious question is which locations are missing, and why? Our intuition is that users typically forget to check in at specific (perhaps routine) places that they visit frequently, *e.g.* home, office, and gas stations. If so, then a small number of places could account for the large majority of missing checkins. To validate this, we identify for each user the top-n most visited POIs, and examine the portion of her missing checkins attributable to these top POIs. Figure 5.4 plots the CDF of this ratio across all users for their top-5 visits. The results confirm our hypothesis. For roughly 60% of all users, 5 locations account for more than half of their missing checkins. For 20% of users, just a single location accounts for more than 40% of missing checkins.

We also looked at the types of POIs in those missing checkins. Figure 5.5 shows the break down of missing checkins locations into 9 categories based on Foursquare's POI classification. The three POI categories with the most missing checkins are *Professional*, *Shops* and *Food*. These are related to people's routine activities: going to work, grocery shopping and meals. According to prior studies [47, 86], users typically do not checkin at places that they think are "boring" or "private." However, these frequently visited places are critical parts of a user's mobility pattern.

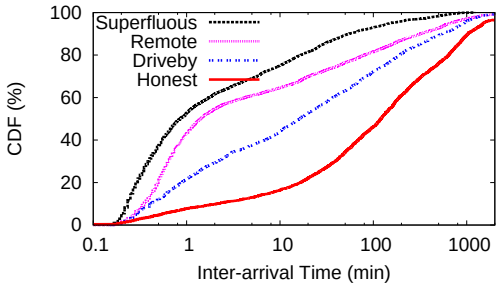
Overall, our analysis shows that missing checkins often cover frequently visited locations in a user's daily life, and are therefore critical components in a human mobility trace. Their absence means geosocial traces are missing a large majority of each user's mobile history.

5.3.4 The Need for Sanitization

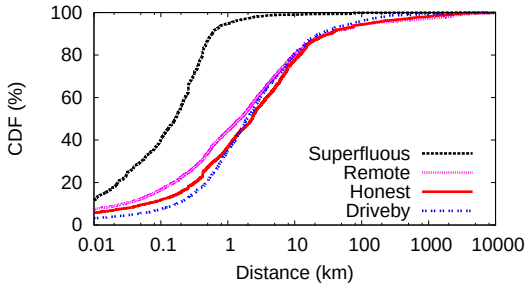
The prevalence of missing and extraneous checkins raises serious concerns on whether checkin traces truly reflect human mobility, significantly devaluing these geosocial mobility traces. Our second goal is to develop sanitization techniques to correct these biases in geosocial traces, thereby producing "clean" checkin traces that truly reflect human mobility patterns, *i.e.* stastistically resemble ground-truth GPS traces in key mobility metrics. We propose and validate a two-phase checkin sanitization process, first detecting and filtering out extraneous checkins (§5.4) and then statistically recovering the missing locations (§5.5). We also show empirically that our techniques can be applied to other datasets such as the Gowalla dataset (§5.6).

5.4 Detecting Extraneous Checkins

Recall from §5.3.2 that extraneous checkins are prevalent in the user population, and cannot be removed by removing specific users. A closer look reveals that extraneous checkins display distinct features that distinguish them from honest checkins. More importantly, we can efficiently extract and quantify these features by collecting checkin and GPS traces from a small number of Foursquare users. Here, we first describe key features identified by our analysis, and then use them to build machine learning based detectors that filter out extraneous checkins.



(a) Inter-arrival Time



(b) Movement Distance

(c) Checkin Categories

Figure 5.6: Example features of checkins.

5.4.1 Features

We identified four groups of features that could distinguish extraneous checkins: temporal features, spatial features, venue categories and user information. We have both features based on the checkin events themselves, and also based on the users who performed the checkins.

Temporal Features. Extraneous checkins are likely to display distinct temporal features from normal user checkins, *e.g.,* bursts, daily or weekly patterns, and checkin frequency. Take burstiness for example. Figure 5.6(a) plots the CDF of the inter-arrival times for all extraneous checkins and honest checkins. Most

extraneous checkins are separated by small inter-arrival times (< 10 minutes), and 35% of the checkins arrive less than a minute after the previous checkin! In contrast, most honest checkins arrive well beyond 10 minutes after the prior checkin. This result is quite intuitive. Our features include *Arrival Delay after Previous Checkin*, *Hour of Day of Checkin*, and *Day of Week of Checkin*.

Spatial Features. Similarly, extraneous checkins also display unique movement features, some specific to the checkin event, and some specific to the user who generated the checkin. Take for example *Distance from Last Checkin* and *Movement Speed since Last Checkin*. Figure 5.6(b) shows the CDF of distance between consecutive checkins. As expected, superfluous checkins, *i.e.* user checking in to multiple nearby venues, show small inter-checkin distances. In contrast, remote checkins display large movement speeds between two consecutive checkins (not shown here for brevity), because remote checkins are often generated between two far away locations in a short amount of time. Driveby checkins also shows higher speed because users' didn't stay in these checkin venues. Our spatial features include *Distance from Last Checkin, Movement Speed from Last Checkin,* and *Distance from User's Daily Center of Checkins* as our spatial features.

Venue Preference. We observe that people tend to make extraneous checkins in certain venue categories – users tend to fake checkins more frequently at Food, Travel and Outdoor places (see Figure 5.6(c)). Besides *Venue Category*, we also consider how frequently a user checkins at each venue, *i.e. Venue Revisit Frequency*.

User Information. Our previous analysis suggests that extraneous checkins have strong correlations with user's *Number of Badges* and *Number of Mayors*.

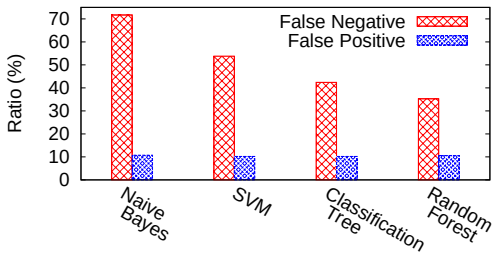


Figure 5.7: Detection accuracy on different machine learning classifiers.

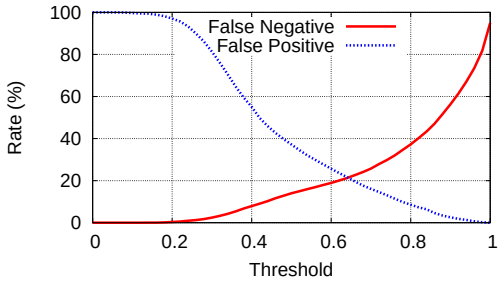


Figure 5.8: False Positives vs False Negatives (Random Forest).

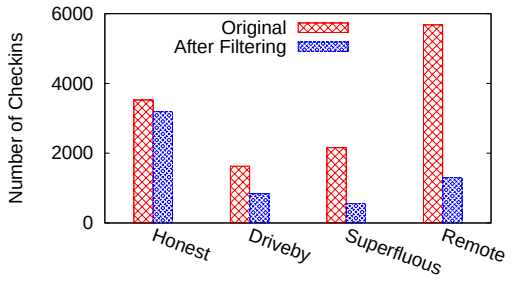


Figure 5.9: Checkin type distribution in the resulting data.

In addition, we also consider *Total Number of Checkins, Average Daily Checkins,* and *Daily Distance Traveled.*

5.4.2 Detection and Results

Our goal here is to remove as many extraneous checkins as possible regardless of the checkin type. With all the features we build for the checkins, we test a series of machine learning algorithms including k-Nearest Neighbor (kNN), Support Vector Machines (SVM), Naive Bayes, Classification Tree, Regression Tree and its improve versions *i.e.* Bagged Trees and Random Forest. Since Random For-

est and Bagged Tree generally perform similarly, we only present Random Forest results for brevity.

We utilize 1/3 of our dataset as training set to find the best settings. In Figure 5.7 we show the accuracy of multiple detectors. Here we limit the False Positive (FP) rate to be 10%. Among all the approaches, random forest performs the best, with a False Negative (FN) rate of just above 30%.

Breakdown of Results. While we were able to remove close to 70% of extraneous checkins, we need to understand the composition of the extraneous checkins that avoided detection. Figure 5.9 compares the number of different types of checkins *before* and *after* applying the detection algorithm (fixing FP to be 10%). The results show that the number of remote checkins and superfluous checkins are significantly reduced, by 77% and 74% respectively. Driveby checkins are the ones with smallest detection ratio (49%). This is not surprising. By definition driveby checkins share the most similarities with honest checkins, since the users are actually near the physical vicinity of the checkin location. Unless we can capture in real time the travel speed when a user checks in, driveby checkins are the hardest to identify. On the other hand, the presence of driveby checkins in a sanitized dataset is somewhat tolerable, because they do correctly mark the user's location at a particular point in time. Later in §5.5.3 our results show that the unfiltered portion of checkins only has minor impact on our sanitization result.

We observe that several features have a heavier impact on the detection process, namely *Total Number of Checkins*, *Average Daily Checkins*, *Daily Distance Traveled by User*, *Arrival Delay after Previous Checkin* and *Movement Speed from Last Checkin*. This means that a user's average behavior and a checkin's temporal/spatial properties can serve as good indicators of extraneous checkins.

In summary, we find that just applying popular machine learning models can significantly reduce the number of harmful extraneous checkins, while preserving most of the honest checkins. Among all the models tested, Random forest is the best, possibly because it trains multiple trees to capture different types of extraneous checkins. This means that we can potentially improve filtering accuracy with more advanced models, *e.g.,* multi-level or ensemble classifiers, which we leave to a future work.

5.5 Interpolating Missing Checkins

After filtering out extraneous checkin events, our next goal is to design an interpolation algorithm to recover real visit events missing from the trace. Our goal here to interpolate the sparse *Filtered Set* to make it *statistically consistent to the GPS dataset* in several meaningful metrics, including temporal and spatial statistics. These temporal and spatial statistics of human mobility are widely used in many models and applications [70, 69, 122].

In this section, we present details of our interpolation algorithm, including components for both temporal and spatial interpolation, and discuss two implementations, one based on a greedy algorithm and one based on conditional probability. We evaluate their performance by comparing the resulting dataset (*Sanitized Set*) with our GPS dataset over temporal, spatial and time-distance joint distributions. Finally, we also demonstrate its efficacy for trace-driven applications by using it and our GPS data to drive simulations of a mobile ad hoc network.

5.5.1 Interpolation Methods

Our interpolation approach includes two steps, *i.e.* temporal interpolation and spatial interpolation. In temporal interpolation, we modify the sanitized dataset to match the GPS dataset, by estimating the number of missing visits between two successive checkins, and the visiting time and stay duration of each missing visit. In spatial interpolation, we assign a movement distance between any two successive visits generated from the temporal interpolation, to ensure that the two datasets also match in spatial distributions.

Note that we assume prior knowledge of some statistical distributions of user movement, which we can draw from a small GPS ground-truth dataset[5]. Our targeted distributions include movement time[6], movement distance and stay durations. We perform the interpolation using a subset of our GPS dataset (*GPS Training Set*) to train the method, and the rest for testing (*GPS Testing Set*). We repeat the experiments using the Geolife dataset [156] as the training set, and observe the consistent result (omitted for brevity). We further discuss the generality of this approach in §5.6.

We explain the interpolation algorithm by example. In Figure 5.10, we show two successive checkins in the temporal domain, *i.e.* C_A at time T_A and C_B at time T_B. Since checkins do not have stay duration information, we assign C_A and C_B each with a stay duration (D_A and D_B as shown) randomly drawn from the stay duration distribution of *GPS Training Set*. Thus, we get two *visits* $V_A = \{T_A, D_A\}$ and $V_B = \{T_B, D_B\}$, which start at time T_A (T_B) and stays for

[5]This small ground-truth dataset can be collected by measurements from a small set of users or by leveraging existing datasets such as Geolife [156], CRAWDAD GPS [**?**] and OpenStreetMap [**?**].

[6]Movement time is different from inter-arrival time because it doesn't include stay duration.

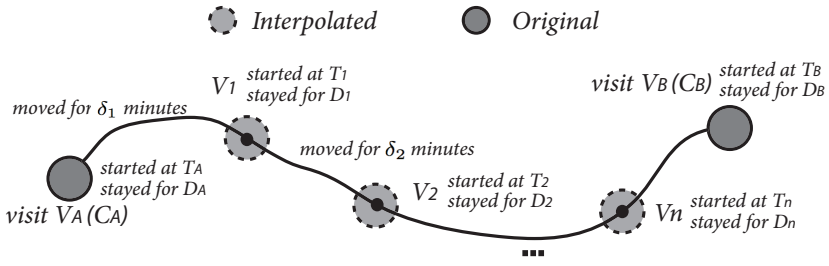


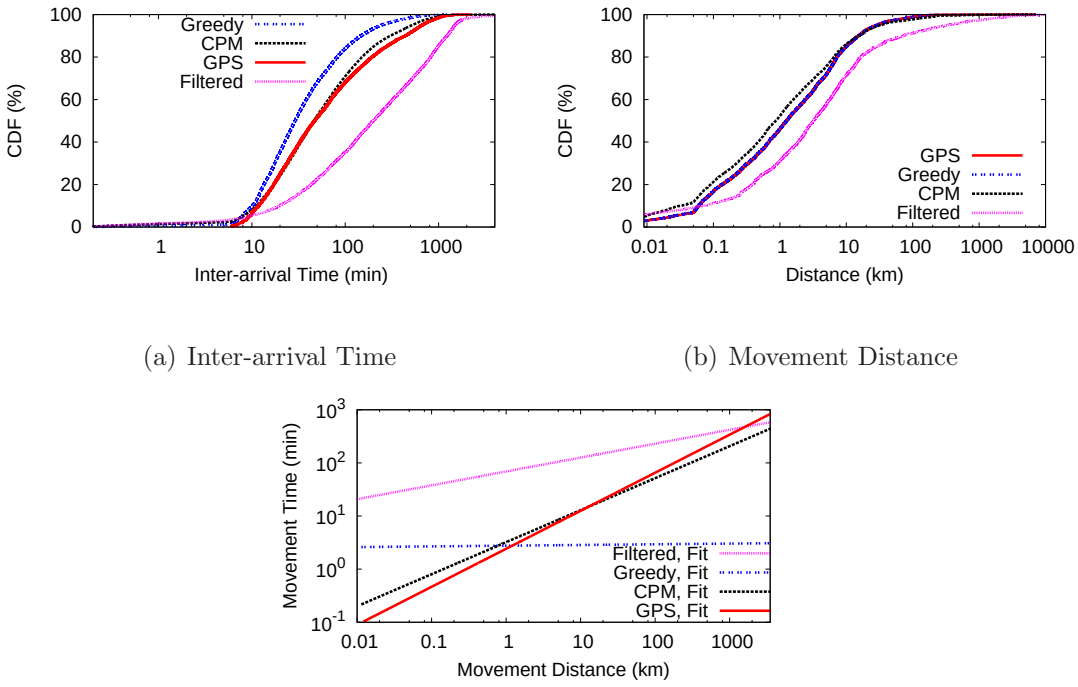
Figure 5.10: Interpolation methodology.

D_A (D_B). Then, we carry out the following two steps to interpolate missing visits between them.

In temporal interpolation, we insert a series of visits $V = \{V_i | V_i = \{T_i, D_i\}, i \in [1, n]\}$ between visit V_A and V_B. Then, for spatial interpolation, we assign a movement distance between two successive visits V_i and V_{i+1}, which is sampled from a movement distance related distribution of the GPS dataset. By applying the interpolation method between any two successive checkins in the *Filtered Set*, the resulting (*Sanitized Set*) will show temporal and spatial statistics similar to those of the GPS dataset.

We consider two possible implementations, *i.e.* a greedy method and a conditional probability-based method:

Greedy Method (*Greedy*). We greedily insert visits between V_A and V_B, until no visit can be inserted. Specifically, given the stay duration distribution $P(D)$, movement time distribution $P(\delta)$, and the movement distance distribution $P(d)$ of the *GPS Training Set*, we first sample a random movement time δ_1 from $P(\delta)$ and a random stay duration D_1 from $P(D)$, as shown in Figure 5.10. If $T_A + D_A + \delta_1 + D_1 < T_B$, we insert the visit V_1 at time $T_1 = T_A + D_A + \delta_1$; otherwise, it means no missing visit exists between V_A and V_B. Whenever a



(a) Inter-arrival Time

(b) Movement Distance

(c) Time vs. Distance

Figure 5.11: Interpolation results on filtered set.

visit V_i is inserted, we repeat this process to examine whether a visit V_{i+1} exists between V_i and V_B. With the new generated visit list $\{V_A, V_1, \ldots, V_n, V_B\}$, we assign a movement distance d (drawn from the movement distance distribution $P(d)$ of *GPS Training Set*) to any two neighboring visits.

Greedy method has some shortcomings. First, it tends to interpolate more visits with short inter-arrival times, since events with shorter inter-arrival times are more likely fit into any time gap. Second, it cannot reflect the fact that the movement distance between two visits highly correlates with movement time. That is, human tend to move further in a larger time period, and vice versa. Considering these factors, we consider an alternative interpolation method.

Datasets	Filtered Set	Greedy Method	CPM Method	GPS
Visits/day	3.5	16.3	9.0	9.9

Table 5.3: Density of visits in interpolated sets.

Conditional Probability Method (*CPM*). In temporal interpolation, we estimate the number of missing visits n before inserting any visits between V_A and V_B. Given a time gap $\Delta = T_B - T_A - D_A$, we estimate the number of missing checkins n based on conditional probability distribution $P(n|\Delta)$: the probability that n missing visits exisit between two successive visits with time gap Δ. We then start to insert visits V_i $(1 \leq i \leq m)$ between V_A and V_B in way similar to the greedy method, but limit the maximum number of insertion such that $m <= n$. We compute the distribution $P(n|\Delta)$ also using *GPS Training Set*.

In spatial interpolation, we estimate a movement distance between two visits by drawing a distance d from the distance conditional probability distribution $P(d|\delta)$, which is the probability of movement distance d given a movement time (δ drawn from *GPS Training Set*) between two neighboring visits. This captures the property that people tend to travel further in larger time periods.

To train these methods, we randomly choose 50% of our GPS dataset as *GPS Training Set*. We then interpolate the *Filtered Set* using the above methods and evaluate the performance by comparing with *GPS Testing Set*.

5.5.2 Interpolation Results

We now examine the performance of both methods. Before we show the distributions, we first examine the density of *visits* in the resulted datasets, as shown in Table 5.3. Comparing to the GPS dataset, visit density from the greedy method

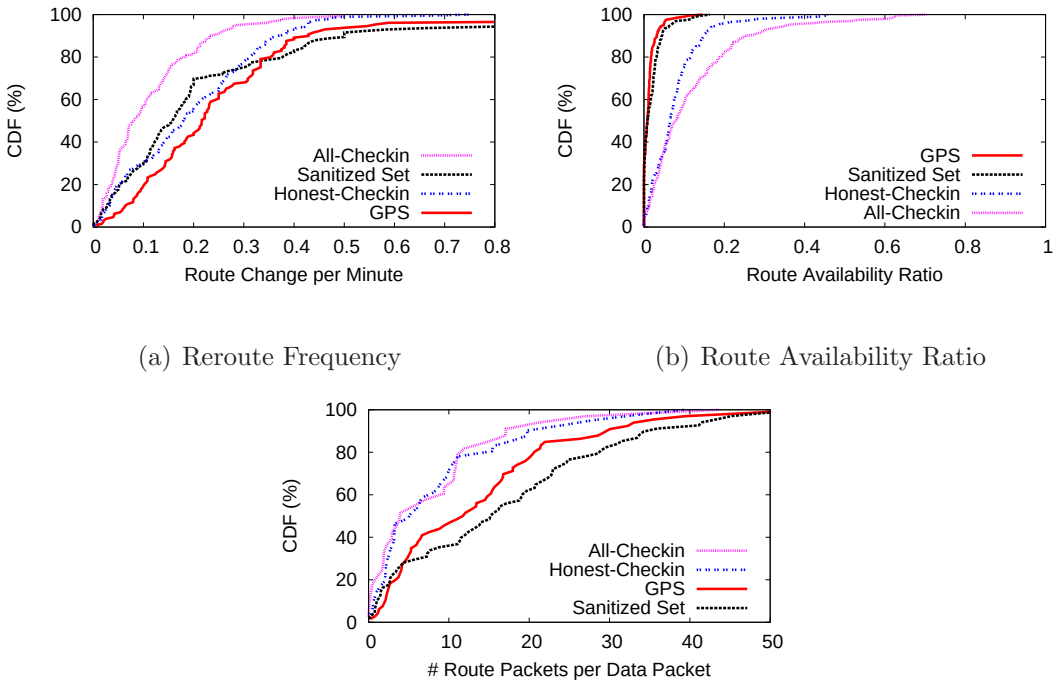
is 60% higher. This confirms that greedy method tends to over-interpolate visits to the dataset. Conditional probability method recovers a reasonable number of missing visits, with similar density to GPS dataset.

Spatial and Temporal Features. We now look at whether the interpolated sets also matches with the GPS dataset in temporal and spatial characteristics, *i.e.* inter-arrival time distribution, movement distance distribution, and time-distance joint distribution.

First, in generating inter-arrival time distribution, *CPM* outperforms *Greedy*. As shown in Figure 5.11(a), compared to *Filtered*, both interpolation methods significantly reduce the inter-arrival time between visits. Taking a closer look, we find that the inter-arrival time distribution of *CPM* is closer to the distribution of GPS dataset. This is because the conditional probability method can correctly predict the number of missing visits using the conditional probability distribution $P(n|\Delta)$, unlike the *Greedy* implementation.

Second, we find that the movement distance distributions from both methods match well with the GPS dataset (Figure 5.11(b)). *Greedy* here is slightly better, because the movement distances used in *Greedy* are directly drawn from the distribution of *GPS Training Set*, whereas in *CPM*, the distance also relies on movement time.

Third, *CPM* accurately captures the relation between movement time and movement distance in human mobility, as shown in Figure 5.11(c). Here we fit the movement time and movement distance in each dataset with $t = kd^{(1-\rho)}$ [122], where t is the movement time, d is the movement distance, and k and ρ are constants. We see that *Greedy*'s movement distance is independent of movement



(a) Reroute Frequency

(b) Route Availability Ratio

(c) Routing Overhead

Figure 5.12: MANET performance.

time. In *CPM*, we use the conditional probability $P(d|\delta)$ in spatial interpolation, thus its distribution is similar to the GPS dataset.

In summary, the conditional probability method generates a trace that matches the actual GPS trace in nearly all respects. Thus, we use the conditional probability method to generate the *Sanitized Set* for the remaining of the paper.

5.5.3 Application Level Impact

We study whether applications dependent on mobile traces can use our sanitized dataset, and produce results identical to using real, physical GPS traces. For our application, we choose mobile ad hoc network (MANET) model simulations,

Datasets	90%-tile Performance		
	Path Availability	Reroute Frequency	Route Overhead
Honest	0.15	0.37	20
All	0.25	0.23	17
Sanitized	0.06	0.50	34
User GPS	0.03	0.42	30

Table 5.4: Simulation results of MANET on different dataset.

because it is an application whose performance is highly dependent on low level properties of its input mobility traces. A popular application in MANETs is to drive a network simulator using one of several built in traffic models. In our case, we preserve the existing simulation process, and provide as input to the models our four traces: the sanitized dataset (*Sanitized Set*), GPS dataset (*GPS*), all our Foursquare checkins (*All-Checkin*) and honest checkin dataset (*Honest-Checkin*).

MANET Simulation Setup. We obtain node/user movement traces directly from each of the above four traces. Each resulting node trace contains user movement direction and distance, movement time and speed, pause time when staying at a place, and total travel time. We feed the node traces into a NS-2 AODV simulator and configure the simulator by placing 200 mobile nodes in a $100km \times 100km$ area, each of which has an $1km$ communication range. These nodes form 100 random node pairs and each node pair communicates using constant bit rate (CBR) streams.

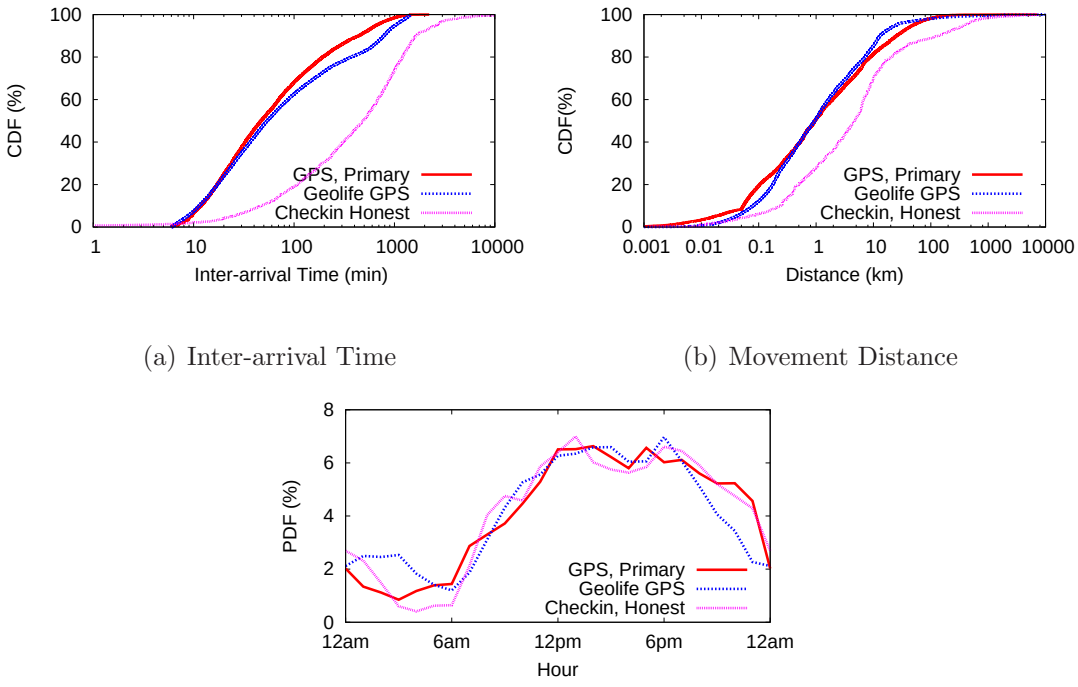
MANET Simulation Results. As stated earlier, our goal here is to examine whether the sanitized dataset is statistically consistent to the GPS dataset in mobility patterns, thus producing similar application-level results. We use three metrics to compare their application-level performance: reroute frequency, route

availability ratio, and routing overhead. Figure 5.12 plots the cumulative distribution and Table 5.4 lists the 90%-tile result. We make three key observations. *First*, compared to GPS, All-Checkin shows lower update frequency, higher availability and fewer routing overhead. This is due to the compound effect of missing and extraneous checkins. In particular, the existence of superfluous checkins produces much lower moving speed, which effectively lowers the update frequency and routing overhead. *Second*, removing the extraneous checkins is not enough – the results using honest checkins still deviate from those using GPS. *Third*, the results using Sanitized Set closely match that of GPS, confirming the necessarity of the interpolation on top of filtering out the extraneous checkins.

In summary, our MANET experiment shows that the resulting datasets from the sanitization process can be used in geosocial-based applications for better performance, which is similar to using a real GPS dataset from user measurements.

5.6 Public Datasets and Applications

Next, we discuss the generalizability of our techniques beyond our own dataset to public datasets, and the applications that can benefit from them. First, discuss our sanitization techniques in the context of general geosocial datasets, and compare some of our parameters against known traces of human movement. We then apply our methodology on a large-scale public checkin trace from Gowalla, which has already been widely used for research and applications [25, 42, 129]. Our goals are to validate our sanitization methodology and to generate a large-scale usable checkin dataset for the research community. Finally, we discuss limitations of our work, and consider its suitability for a range of applications.



(a) Inter-arrival Time

(b) Movement Distance

(c) Daily Pattern

Figure 5.13: Comparing feature distributions across multiple GPS traces and our honest checkin trace.

5.6.1 Generalizing Sanitization for Geosocial Data

The ideal outcome of our work is a robust and accurate sanitizing process for transforming potentially biased geosocial traces into realistic and useful mobility traces for the research community. To do so, we must consider the question of generalizability, *i.e.,* how much can we rely on a sanitization process for third-party datasets without ground-truth GPS traces for validation?

We believe our observations for detecting extraneous events is relatively generalizable. A quick look shows that the heaviest weights in classifiers are given to features on dynamics and user movement, *e.g.,* checkin interarrival times (for

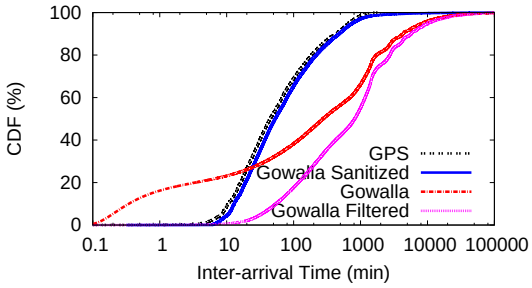
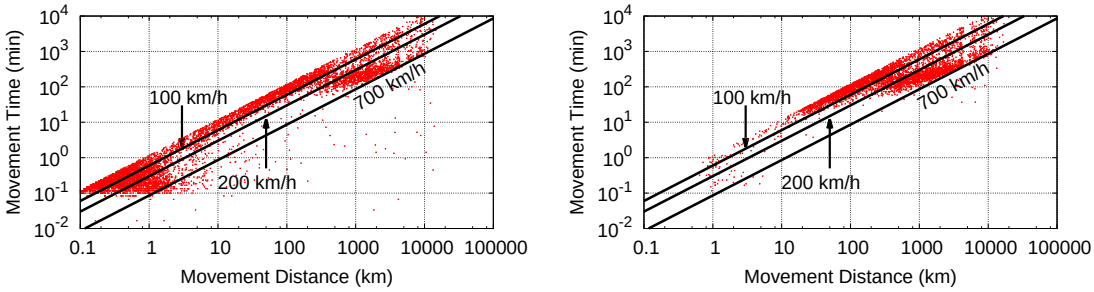


Figure 5.14: Inter-arrival time of checkins in Gowalla dataset.



(a) Before detection

(b) After detection

Figure 5.15: Movements with speed $>$ 50km/h.

detecting bursts) and movement speed between checkins. More importantly, main cause of extraneous checkins should be consistent across any location-based social network with similar incentive models (currently all LBSNs).

The biggest question of generalizability applies to our interpolation using feature distributions from the GPS trace. We believe that while exact values in the distribution may vary, inherent distributions of human movement are largely consistent across datasets. To verify this, we obtained a large GPS trace of 182 users gathered by Microsoft Research Asia (over 48,000 hours of data from April 2007 to August 2012) [156]. We plot key distributions in Figure 5.13. Clearly, our GPS dataset shows highly consistent distributions in key metrics to the Geo-

life dataset. The discrepancy between the honest checkin trace and the two GPS traces is especially clear in inter-arrival time and movement distances.

While a conclusive proof is likely infeasible, we believe this shows our sanitization techniques are sufficiently general for third party traces. Given our goal of reducing the inherent biases in geosocial traces, we believe a less than perfect process can still add tremendous value.

5.6.2 Sanitizing the Gowalla Dataset

We sanitize the publicly available Gowalla dataset, by first studying the existence of extraneous/missing checkins, then sanitizing the data to remove extraneous checkins and interpolating for missing checkins. By analyzing data statistics before and after, we show the effectiveness of our sanitization process in removing bias of geosocial traces and generating realistic human mobility data.

First, we check whether extraneous/missing checkins exist in the Gowalla dataset. Without the GPS traces of Gowalla users, we are empirically searching for clear evidence of extraneous/missing checkins. In particular, we found the following properties demonstrate the problem very well.

Gowalla Trace Inconsistencies. We start by studying inter-arrival time of Gowalla checkins in Figure 5.14. The red line shows that $\sim 20\%$ of checkins had inter-arrival times less than 1 minute, which is clearly too short to be "valid" visits in the common sense. Users cannot travel between separate PoI's within a few seconds. This is very similar to what we observed in our Foursquare checkin trace (Figure 5.6(a)). While we cannot confirm using measurement data, short inter-

arrival times (*i.e.,* bursts of checkins) are clear indicators of extraneous checkins in the Gowalla dataset.

Second, Figure 5.15(a) shows movement speeds of Gowalla users traveling faster than 50km/h. Normally, people travel at high speed for long-distance trips, via cars or flights, but generally move at low speeds for local destinations. Figure 5.15(a) shows that most Gowalla users follow this trend (note the log axes). However, points in the bottom left corner represent many checkins moving at extremely high speeds ($200km/h$ or higher). This is reasonable for cars, trains and air travel over long trips, but highly unlikely for trips less than $3km$. These observations clearly show that extraneous checkins do exist in the Gowalla dataset.

Finally, missing checkins are inevitable in any geosocial trace, because users often forget to check-in to each location they visit. A simple count of checkins per-user, per-day demonstrates this. On average, Gowalla users checked in to 1.8 locations per day, while our GPS trace showed 9.9 locations were visited per user per day.

Sanitizing Gowalla. Since we do not have a matching GPS data for Gowalla users, we apply general techniques learned from our data sanitization of the Foursquare checkin trace. First, we filter for extraneous events. We modify the detector to account for small differences between Gowalla and Foursquare datasets: the Gowalla data does not include counts of users' badges and mayorships (even though it uses similar incentives) or categories of locations[7]. Applying this detector on Gowalla classifies 23% of all checkins as extraneous. We call the post-removal dataset "Gowalla filtered."

[7]These features have minimal impact on accuracy. Removing them from the detector only reduces detection accuracy by 1% when applied to the Foursquare dataset.

Datasets	90%-tile Performance		
	Path Availability	Reroute Frequency	Route Overhead
Honest	0.33	0.24	22
All	0.24	0.29	34
Sanitized	0.08	0.38	31
User GPS	0.03	0.43	30

Table 5.5: Gowalla data results in MANET with different datasets.

Without a matching user GPS trace, we validate the effectiveness of the detection by looking for the inconsistencies we found earlier. First, Figure 5.14 shows that compared to the original Gowalla trace (in red), our detector successfully removes all of the short-interval checkins (filtered data in blue). Second, as shown in Figure 5.15(a) and Figure 5.15(b), our detector filters most of the extraneous checkins with unrealistically high speeds at short distances (bottom left corner). While we cannot guarantee we detected all extraneous events, we successfully filtered out nearly all abnormal checkins observable through our metrics.

Finally, we apply the conditional probability interpolation method on the Gowalla filtered dataset to recover the missing checkins. After interpolation, average visits per day in *Gowalla Sanitized* dataset rises from 1.8 to 8.4, which is close to the 9.9 we observed from our GPS user trace. Figure 5.14 also shows that the inter-arrival time distribution of Gowalla sanitized data match very well with distributions of our GPS dataset. The movement distance and time-distance joint distribution are also consistent with previous results, but omitted for brevity.

Application Level Impact. We also put Gowalla dataset and the sanitized set in our MANET simulations, and show the results in Table 5.5. The results are consistent with our Foursquare dataset results (Table 5.4), despite the fact that the Gowalla dataset contained less extraneous checkins (23% vs. 75%). The

Gowalla trace post-sanitization produces similar results to that of our GPS trace (and significantly different from the original Gowalla trace), thus demonstrating the effectiveness of the sanitization.

5.6.3 Applications and Limitations

Our approach of capturing user mobility via modified geosocial traces is not suitable for all mobile applications. We consider three broad classes of applications and discuss the suitability of our approach for each.

- Some applications require only high-level statistical distributions of human mobility, and can utilize our traces directly. For example, our traces can easily drive a wide range of *network simulations* for MANETs or opportunistic networks. This also holds for *mobile sensing* applications, such as crowdsourced sensing platforms focused on spatial coverage and frequency of user visits.
- Some applications require specific information like source and destination of the movements, but only output accumulated results over large populations. Examples include *urban planning*, which uses mobility trace to estimate population density at different locations and times, and *traffic planning*, which combine user mobility traces to public available information such as maps to predict specific travel routes. In these cases, we can bound the location error of the interpolated events, and in turn bound the error of final statistical results in the applications.
- Finally, for applications that focus on per-user details and require accurate event information, our datasets are not appropriate. Examples include real-time, *personalized venue recommendations*, *friend recommendations*, and *next*

destination prediction. While our interpolation produces statistically correct results, we cannot provide accuracy at the level of per-user visits.

5.7 Related Work

Human Mobility Tracking. Recently, researchers have been working on novel ways to gather detailed, timestamped user mobility traces. Recent efforts leverage the rich collection of sensors on smartphones to improve location accuracy [74]. But obtaining detailed traces requires significant overhead, and existing efforts still remain very limited in scale [157, 156]. To obtain large-scale human movement traces, others proposed relying on registration data from cellular or WiFi networks [144, 32, 152]. Such data approximates a user's location as the coverage area of her registered cellular basestation or WiFi AP. Unfortunately, since they rely on registrations, they also sample locations unevenly, leading to biased representations of human mobility [118].

Mobility Models. Many mobility models have been proposed in literature. A large group of models are random mobility models (*e.g.* Random Waypoint, Brownian Motion, Random Walk) and their variants (*e.g.* Gauss-Markov [38]). These models are elegant mathematically, but are too simplistic to capture detailed human mobility features. More realistic mobility models seek to capture different aspects of human mobility, such as geographical constraints (Obstacle [69], Freeway and Manhattan [30]) and social context (ORBIT [85], CMM [85]).

Other recent models have been built from real traces such as GPS traces and WiFi AP registration traces. [122] builds Random Waypoint Models with real traces. [144, 77, 68] study WiFi registration patterns. Of these, [68] only considers

registration patterns, but not exact physical mobility. [144, 77] both consider the problem in a geographical point of view, while [77] also models the hotspot feature of human mobility. [67, 42] models the regular home-work pattern, while [42] also considers the impact of social links. In [84], the authors proposed a model to capture multiple features of human mobility from real traces, including heavy-tail flights and pause times, power-law inter-contact times and hotspot locations. We use several of these to drive our simulations [77, 84, 122].

Geosocial Datasets. While prior work has proposed client-side methods to prevent fake checkins [61, 127], none have studied filtering extraneous events or interpolating events from samples of human movements. Methods have been proposed to interpolate human trajectory from sparse data samples, *e.g.* vehicular GPS records and cellular records [64, 147].

5.8 Conclusion

In this chapter, we describe a detailed effort to understand and improve the ability of geosocial human mobilty traces to capture human mobility patterns. Using a ground-truth GPS dataset from a user study, we identified significant discrepancies between the two datasets, *i.e.* that large amount of fake checkins exist while lots of visited locations are missing in the checkin dataset. To mitigate these problems, we proposed a two-phase sanitization method to filter out fake checkins and interpolating missing locations, and validate the effectiveness in a real application scenario. By applying our techniques to the public Gowalla dataset, we showed that it is possible to transform geosocial traces into the mobility traces needed by the research community.

Chapter 6

Conclusion

We conclude the dissertation by summarizing our works, sharing the lessons learned and discussing future directions.

6.1 Summary

The combination of sensing technology and network communication can enable a large variety of novel mobile applications, such as location based services, interactive games, collaboration applications among co-located users, motion monitoring of elder people and augmented reality. However, the limited localization capability and network capacity have hindered us from enabling these visions. To enable such services and scale to millions of people, we need to design highly accurate localization systems and high-capacity mobile networks.

In this dissertation, we identify the challenges in both areas and propose solutions for them. To be specific, we observe that traditional localization methods have a low level of accuracy, and requires a localization infrastructure and target device cooperation, which significantly limits their applicable scenarios. We propose two designs to address these problems. In the first design, we use acous-

tic signals instead of radio waves to address the accuracy issue, reaching a much higher accuracy in centimeter level without the aid of localization infrastructures. We propose novel signal processing algorithms to significantly improve the localization delay, thus enabling mobile applications to track user motion in real time. In the second design, we target a malicious scenario in which the wireless device to locate is uncooperative. To pinpoint the location of the device from the observed signal, we leverage the human body's blocking effect on wireless signals and build an efficient direction analysis algorithm upon it. Both our localization systems have practical system implementations and real world evaluations to demonstrate the effectiveness.

To address the network capacity issue, we also make contributions in two folds. First, we notice that existing mobile networks can hardly scale to provide orders of magnitude more bandwidth. There are two fundamental reasons: limited spectrum resource and high mutual interference. To address the problems, we make a drastically new design of next generation mobile networks utilizing 60GHz beamforming technology, which has 7GHz-wide license-free spectrum band, and highly directional transmission to avoid interference. Our work reveals the promising potentials of such a design, and quantify the system performance under real environments with pedestrian blockage, user mobility and interference. Second, we observe that user mobility could generate high dynamics in mobile networks, thus we need to understand large-scale human mobility patterns. We conduct a study on existing large-scale user mobility traces from popular location based social networks, *i.e.* Foursquare and Gowalla, and aim to understand whether these traces can reflect actual human mobility. Our work identifies large discrepancies between these traces and the actual GPS trace, which are probably due to the user incen-

tive systems of location based social networks such as mayorships or badges. We then design data analytic tools based on a list of distinctive features to sanitize the datasets.

6.2 Lessons Learned

Through solving the above problems, we have also learned some lessons on the research of mobile computing as follows.

Mobility Creates Both Challenges and Opportunities. Mobile computing aims to provide ubiquitous services and connections to mobile users, thus requiring the systems to handle various environment issues and mobility introduced errors. This is a great challenge to practical system design. For instance, in Chapter 2, our localization system face several challenges caused by mobility. First, user motion caused Doppler effect generates lots of measurement errors. Second, certain environments many have strong signal reflections from the walls or ambient noise, also causing measurement errors. Thus the impact of mobility should be always kept in mind while designing mobile systems.

However, we also find that mobility is not always a destructive factor to system design. In Chapter 3, we leverage the user's rotation to create a signal pattern that emulates an directional antenna to find out the location of a device. This design explores the diversity provided by user motion and build an effective algorithm on it. In Chapter 4, we find that even when the line of sight path from the basestation is completely blocked, it's still possible to build connection by reflection from walls. This is an example of leveraging the surrounding environment to achieve better performance.

Embracing the Power of Sensors. The rich sensor set on mobile devices are the most important resources to explore when designing mobile applications. In Chapter 2 and 3, we show that by leveraging different sensors, we can actually achieve much better performance than traditional solutions. For instance, in Chapter 2, we utilize the microphone and speaker to achieve high localization accuracy. Leveraging certain sensor information may also help to improve network performance. For example, in Chapter 4, one of the future works in the basestation design is to track mobile users and build directional 60GHz links with them. We can possibly leverage the sensors on mobile devices to assist this process.

6.3 Ongoing and Future Directions

So far our research has been focused on developing accurate localization systems and high performance mobile networks. Our existing localization systems focus on wireless/mobile device localization, and we have treated each mobile device as isolated entities. Moving forward, we plan to first expand our horizon of device localization to to the exploration of even richer information of surrounding environment, not only including devices and but also other objects such as a tables, buildings or human. We then plan to fusion the information collected from millions of devices together, utilizing data analytic tools to recognize common scenarios, and then configuring these device with the right functionality, to finally enable "intelligent" mobile computing. Our detailed visions and plans are as follows.

Mining Rich Context Information. In mobile computing, there is a urgent demand for a system to sense and digitize the surrounding environment of a

user, not only including the location of each object, but also its speed, shape and material. Such a system can be very beneficial to many applications. For example, a visually impaired person can avoid all possible collisions with such information, going anywhere without others help. For another example, a game designer can easily reconstruct any environment from the information captured by a smartphone, and integrate them into virtual reality games.

To meet this demand, we envision a personal RADAR system, which can collect the information of the objects in the vicinity of the user. By combing observations from different locations when the user move, the device can ultimately reconstruct the complete picture of the surrounding environment.

We are currently considering millimeter wave, such as 60GHz, as a candidate. as 60GHz signal has good reflection properties. We can project directional signal beams to the object's surface, and calculate the information from the signal that is echoed back, such as shape and surface material. Utilizing electrically steered antenna arrays, this process can be really fast.

To implement such a system is challenging. Traditional RADARs are normally equipped with large lenses. The larger the lens is, the higher the sensing accuracy is. However, limited by the size of mobile devices, 60GHz chipsets can not carry large lenses. One potential solution for this problem is to leverage the user's movement. Through capturing information in multiple locations, we can emulate a large lens by walking. This would require a precise localization system to connect all the "images" sampled at different location together. At the same time, the user body's movement such as shaking could also introduce measurement errors. We are using our 60GHz platforms to study these.

Empowering Mobile Computing with Big Data. Although mobile computing has been through years of fast growth, our devices are still not as "intelligent" as we expect. The devices are still working on user specified rules and can not configure themselves intelligently to fit the users need. For a simple example, whenever someone walks into a meeting room, his/her device should be able to learn the situation and turn itself into a meeting mode. This sounds like a simple task, but involves very challenging sensing and learning process, which our devices are not capable of currently.

To enable it, the device should first be able to tag the situation with sensed information, *, which room it is, whom the user is with, whether it is noisy or quiet*. Second, the device should be able to recognize the situation from its own previous experience or the experience of other devices which have been in the same situation. With the information collected from lots of devices, we can leverage data analytic tools to mine common scenarios, define operation rules based on the situation, and finally push these rules to each individual device to make it "intelligent".

To realize such a system, there are also many obstacles. First, the system requires every user to upload the sensed information to the cloud, which could be highly sensitive information. Thus, how to gather large-scale data from the user without hurting privacy is very challenging. Second, users normally have different types of behaviors even in the same situation, thus building adaptive models to fit different users' needs is another challenge to conquer.

Bibliography

[1] http://www.techjournal.org/2011/09/mobile-broadband-useage-is-set-to-explode-

[2] http://money.cnn.com/2012/02/24/technology/spectrum_crunch_solutions/index.htm.

[3] http://www.dailywireless.org/2011/05/06/att-moves-to-hotzones-and-picocells.

[4] http://www.siliconimage.com/news/releasedetails.aspx?id=649.

[5] http://news.cnet.com/8301-30685_3-57326718-264/wilocity-60ghz-wireless-revolution-begins-at-ces/.

[6] http://transition.fcc.gov/Bureaus/Engineering_Technology/Documents/bulletins/oet70/oet70a.pdf.

[7] By the numbers: 33 amazing yelp statistics. http://expandedramblings.com/index.php/yelp-statistics/.

[8] Cisco Small Cell Wireless Backhaul Ecosystem: A Flexible and Proven Deployment Toolkit. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/small-cell-solutions/brochure_c02-728436.html.

[9] Cisco visual networking index: Global mobile data traffic forecast update, 20132018. http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html.

[10] Fcc 13-112. http://hraunfoss.fcc.gov/edocs_public/attachmatch/FCC-13-112A1.pdf.

[11] Futuristic helmets use smart glasses, augmented reality. http://news.discovery.com/tech/gear-and-gadgets/futuristic-helmets-use-smart-glasses-augmented-reality-140531.htm.

[12] Google WiFi. http://wifi.google.com/.

[13] IEEE 802.11 Task Group AD. http://www.ieee802.org/11/Reports/tgad_update.htm.

[14] Improving qoe with an intelligent look into wireless network capacity. http://www2.alcatel-lucent.com/techzine/improving-qoe-with-an-intelligent-look-into-wireless-network-capacity/.

[15] Infographic - Who is the Average Foursquare User? http://blog.spotistic.com/infographic-who-is-the-average-foursquare-user/.

[16] Market trends: Gaming ecosystem, Gartner 2011.

[17] On foursquare, cheating, and claiming mayorships from your couch. http://blog.foursquare.com/post/503822143/on-foursquare-cheating-and-claiming-mayorships-from.

[18] Walgreens tests google's augmented reality for loyalty app. http://adage.com/article/datadriven-marketing/walgreens-tests-google-s-augmented-reality-loyalty-app/293961/.

[19] WiFi data offload: Turning challenges into opportunities. http://wifidataoffload.wordpress.com/2010/12/04.

[20] AT&T now has 460 mobile data offload points in NYC. http://www.4gtrends.com/articles/37936/, June, 2011.

[21] M. Abouelseoud and G. Charlton. The effect of human blockage on the performance of millimeter-wave access link for outdoor coverage. In *Proc. of VTC*, 2013.

[22] M. Abouelseoud and G. Charlton. System level performance of millimeter-wave access link for outdoor coverage. In *Proc. of WCNC*, 2013.

[23] A. Aggarwal, J. Almeida, and P. Kumaraguru. Detection of spam tipping behaviour on foursquare. In *Proc. of WWW*, 2013.

[24] M. R. Akdeniz, Y. Liu, S. Rangan, and E. Erkip. Millimeter wave picocellular system evaluation for urban deployments. *arXiv:1304.3963*, 2013.

[25] M. Allamanis et al. Evolution of a location-based online social network: analysis and models. In *Proc. of IMC*, 2012.

[26] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In *Proc. of MobiCom*, 2009.

[27] R. T. Azuma et al. A survey of augmented reality.

[28] P. Bahl and V. N. Padmanabhan. Radar: An in-building rf-based user location and tracking system. In *Proc. of INFOCOM*, 2000.

[29] P. Bahl and V. N. Padmanabhan. RADAR: An in-building RF-based user location and tracking system. In *Proc. of INFOCOM*, 2000.

[30] F. Bai, N. Sadagopan, and A. Helmy. The important framework for analyzing the impact of mobility on performance of routing protocols for adhoc networks. *Ad Hoc Networks*, 1(4), 2003.

[31] A. Balasubramanian, R. Mahajan, and A. Venkataramani. Augmenting mobile 3G using WiFi. In *Proc. of MobiSys*, 2010.

[32] M. Balazinska and P. Castro. Characterizing mobility and network usage in a corporate wireless local-area network. In *Proc. of MobiSys*, 2003.

[33] J. J. Barton, T. Hsieh, B. Johanson, V. Vijayaraghavan, A. Fox, and T. Shimizu. The meetingmachine: Interactive workspace support for nomadic users. In *Proc. of WMCSA*, 2003.

[34] T. Beigbeder, R. Coughlan, C. Lusher, J. Plunkett, E. Agu, and M. Claypool. The effects of loss and latency on user performance in unreal tournament 2003. In *Proceedings of the 3rd ACM SIGCOMM Workshop on Network and System Support for Games (NetGames)*, pages 144–151, 2004.

[35] X. Bian, G. Abowd, and J. Rehg. Using sound source localization in a home environment. *Pervasive Computing*, pages 281–291, 2005.

[36] R. Bracewell. *Pentagram Notation for Cross Correlation*. The Fourier Transform and Its Applications. New York: McGraw Hill, 1965.

[37] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, 2000.

[38] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless communications and mobile computing*, 2(5):483–502, 2002.

[39] Y. Chan, W. Tsui, H. So, and P. Ching. Time-of-arrival based localization under nlos conditions. *IEEE Transactions on Vechicular Technology*, 55(1):17–24, 2006.

[40] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. Fm-based indoor localization. In *Proc. of MobiSys*, 2012.

[41] Z. Cheng et al. Exploring millions of footprints in location sharing services. In *Proc. of ICWSM*, 2011.

[42] E. Cho, S. A. Myers, and J. Leskovec. Friendship and mobility: user movement in location-based social networks. In *Proc. of SIGKDD*, 2011.

[43] J. Chung, M. Donahoe, C. Schmandt, I.-J. Kim, P. Razavai, and M. Wiseman. Indoor location sensing using geo-magnetism. In *Proc. of MobiSys*, 2011.

[44] G. Cohn, D. Morris, S. Patel, and D. Tan. Humantenna: using the body as an antenna for real-time whole-body interaction. In *Proc. of SIGCHI*, 2012.

[45] L. Correia and J. Reis. Wideband characterisation of the propagation channel for outdoors at 60 ghz. In *Proc. of PIMRC*, 1996.

[46] D. Costello and J. Forney, G.D. Channel coding: The road to channel capacity. *Proc. of the IEEE*, 95(6):1150–1177, 2007.

[47] H. Cramer, M. Rost, and L. E. Holmquist. Performing a check-in: emerging practices, norms and "conflicts" in location-sharing using foursquare. In *Proc. of MobileHCI*, 2011.

[48] J. Fink and V. Kumar. Online methods for radio signal mapping with mobile robots. In *Proc. of ICRA*, 2010.

[49] A. Flach and K. David. A physical analysis of an accident scenario between cars and pedestrians. In *Proc. of VTC*, 2009.

[50] H. T. Friis. A note on a simple transmission formula. *Proc. of IRE*, 1946.

[51] S. Geng, J. Kivinen, X. Zhao, and P. Vainikainen. Millimeter-wave propagation channel characterization for short-range wireless communications. *IEEE TVT*, 58(1):3–13, 2009.

[52] M. Ghaddar, L. Talbi, and T. Denidni. Human body modelling for prediction of effect of people on indoor propagation channel. *Electronics Letters*, 40(25):1592–1594, 2004.

[53] L. Girod, M. Lukac, V. Trifa, and D. Estrin. The design and implementation of a self-calibrating distributed acoustic sensing platform. In *Proceedings of the 4th International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 71–84, 2006.

[54] M. Gonzalez, J. Gomez, M. Lopez-Guerrero, V. Rangel, and M. de Oca. GUIDE-gradient: A guiding algorithm for mobile nodes in WLAN and Adhoc networks. *Wireless Personal Communications*, 57:629–653, April 2011.

[55] D. Han, D. Andersen, M. Kaminsky, K. Papagiannaki, and S. Seshan. Access point localization using local signal strength gradient. In *Proc. of PAM*, 2009.

[56] R. C. Hansen. *Phased array antennas*. John Wiley & Sons, 2nd edition, 2009.

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

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

[59] M. Hazas, C. Kray, H. Gellersen, H. Agbota, G. Kortuem, and A. Krohn. A relative positioning system for co-located mobile devices. In *Proceedings of the 3rd International Conference on Mobile Systems, Applications, and Services (MobiSys)*, pages 177–190, 2005.

[60] T. He, C. Huang, B. Blum, J. Stankovic, and T. Abdelzaher. Range-free localization scheme for large scale sensor networks. In *Proc. of MobiCom*, 2003.

[61] W. He, X. Liu, and M. Ren. Location cheating: A security challenge to location-based social network services. In *Proc. of ICDCS*, 2011.

[62] K. C. Ho and W. Xu. An accurate algebraic solution for moving source location using TDOA and FDOA measurements. *IEEE Transactions on Signal Processing*, 52(9):2453–2463, 2004.

[63] F. Hoffmann and J. Scott. Location of mobile devices using networked surfaces. In *Proc. of UbiComp*. 2002.

[64] S. Hoteit et al. Estimating real human trajectories through mobile phone data. In *Proc. of MDM*, 2013.

[65] P. Hu, L. Li, C. Peng, G. Shen, and F. Zhao. Pharos: enable physical analytics through visible light based indoor localization. In *Proc. of HotNets*, 2013.

[66] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck. A close examination of performance and power characteristics of 4G LTE networks. In *Proc. of MobiSys*, 2012.

[67] S. Isaacman, R. Becker, R. Cáceres, M. Martonosi, J. Rowland, A. Varshavsky, and W. Willinger. Human mobility modeling at metropolitan scales. In *Proc. of MobiSys*, 2012.

[68] R. Jain, D. Lelescu, and M. Balakrishnan. Model t: an empirical model for user registration patterns in a campus wireless lan. In *Proc. of MobiCom*, 2005.

[69] A. Jardosh et al. Towards realistic mobility models for mobile ad hoc networks. In *Proc. of MobiCom*, 2003.

[70] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. *Mobile Computing*, 5:153–179, 1996.

[71] L. Jones and S. Lederman. *Human hand function*. Oxford University Press, USA, 2006.

[72] F. A. Z. K. D. Katabi and R. C. Miller. 3d tracking via body radio reflections. In *Proc. of NSDI*, 2014.

[73] K. Kelleher. *The Microwave Engineers' Handbook and Buyers' Guide*. New York: Horizon Press, 5th edition, 1964.

[74] D. Kim et al. Sensloc: sensing everyday places and paths using less energy. In *Proc. of SenSys*, 2010.

[75] M. Kim and N. Chong. RFID-based mobile robot guidance to a stationary target. *Mechatronics*, 17(4-5):217–229, 2007.

[76] M. Kim and N. Chong. Direction sensing RFID reader for mobile robot navigation. *IEEE Transactions on Automation Science and Engineering*, 6(1):44–54, 2009.

[77] M. Kim, D. Kotz, and S. Kim. Extracting a mobility model from real user traces. In *Proc. of INFOCOM*, 2006.

[78] G. Kortuem, C. Kray, and H. Gellersen. Sensing and visualizing spatial relations of mobile devices. In *Proceedings of the 18th Annual ACM Symposium on User interface Software and Technology (UIST)*, pages 93–102, 2005.

[79] M. Kushwaha, K. Molnár, J. Sallai, P. Volgyesi, M. Maróti, and A. Lédeczi. Sensor node localization using mobile acoustic beacons. In *Proceedings of the IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS)*, pages 491–500, 2005.

[80] C.-A. La and P. Michiardi. Characterizing user mobility in second life. In *Proc. of WOSN*, 2008.

[81] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, et al. Place lab: Device positioning using radio beacons in the wild. *Pervasive Computing*, pages 116–133, 2005.

[82] A. LaMarca and E. de Lara. Location systems: An introduction to the technology behind location. *Synthesis Lectures on Mobile and Pervasive Computing*, 3(1):1–122, 2008.

[83] B. Langen, G. Lober, and W. Herzig. Reflection and transmission behavior of building materials at 60ghz. In *Proc. of PIMRC*, 1994.

[84] K. Lee, S. Hong, S. J. Kim, I. Rhee, and S. Chong. Slaw: A new mobility model for human walks. In *Proc. of INFOCOM*, 2009.

[85] S. Lim, C. Yu, and C. R. Das. Clustered mobility model for scale-free wireless networks. In *Proc. of LCN*, 2006.

[86] J. Lindqvist et al. I'm the mayor of my house: examining why people use foursquare-a social-driven location sharing application. In *Proc. of SIGCHI*, 2011.

[87] H. Liu, Y. Gan, J. Yang, S. Sidhom, Y. Wang, Y. Chen, and F. Ye. Push the limit of wifi based localization for smartphones. In *Proc. of MobiCom*, 2012.

[88] J. Liu, B. Priyantha, T. Hart, H. S. Ramos, A. A. Loureiro, and Q. Wang. Energy efficient gps sensing with cloud offloading. In *Proc. of SenSys*, 2012.

[89] C. Lopes, A. Haghighat, A. Mandal, T. Givargis, and P. Baldi. Localization of off-the-shelf mobile devices using audible sound: architectures, protocols and performance assessment. *ACM SIGMOBILE Mobile Computing and Communications Review*, 10(2):38–50, 2006.

[90] G. Lovnes, J. Reis, and R. Raekken. Channel sounding measurements at 59 ghz in city streets. In *Proc. of PIMRC*, 1994.

[91] K. Maeda et al. Urban pedestrian mobility for mobile wireless network simulation. *Ad Hoc Networks*, 7(1):153–170, 2009.

[92] A. Maltsev, R. Maslennikov, A. Sevastyanov, A. Khoryaev, and A. Lomayev. Experimental investigations of 60 GHz WLAN systems in office environment. *IEEE JSAC*, 27(8), 2009.

[93] S. Mason, C. Berger, S. Zhou, and P. Willett. Detection, synchronization, and doppler scale estimation with multicarrier waveforms in underwater acoustic communication. *IEEE Journal on Selected Areas in Communications*, 26(9):1638–1649, 2008.

[94] D. Matic, H. Harada, and R. Prasad. Indoor and outdoor frequency measurements for mm-waves in the range of 60 ghz. In *Proc. of VTC*, 1998.

[95] Microsoft. Xbox Kinect. `http://www.xbox.com/kinect`.

[96] R. Nandakumar, K. K. Chintalapudi, and V. N. Padmanabhan. Centaur: locating devices in an office environment. In *Proc. of MobiCom*, 2012.

[97] A. Nasipuri and K. Li. A directionality based location discovery scheme for wireless sensor networks. In *Proc. of WSNA*. ACM, 2002.

[98] A. Natarajan, S. K. Reynolds, M.-D. Tsai, S. T. Nicolson, J.-H. Zhan, D. G. Kam, D. Liu, Y.-L. Huang, A. Valdes-Garcia, and B. A. Floyd. A fully-integrated 16-element phased-array receiver in sige bicmos for 60-ghz communications. *IEEE JSSC*, 46(5):1059–1075, 2011.

[99] D. Niculescu and B. Nath. VOR based stations for indoor 802.11 positioning. In *Proc. of MobiCom*, 2004.

[100] Nintendo. Nintendo Wii. `http://www.nintendo.com/wii`.

[101] S. Nishi and K. Tokuda. Development of mm-wave video transmission system-development of antenna. In *Proc. of APMC*, 2001.

[102] A. Noulas et al. An empirical study of geographic user activity patterns in foursquare. In *Proc. of ICWSM*, 2011.

[103] A. Noulas et al. Mining user mobility features for next place prediction in location-based services. In *Proc. of ICDM*, 2012.

[104] R. Pacchiano. Track down rogue wireless access points. WiFi Planet Tutorial, March 2006.

[105] A. Pages-Zamora, J. Vidal, and D. Brooks. Closed-form solution for positioning based on angle of arrival measurements. In *Proc. of PIMRC*, 2002.

[106] K. Pearson. Mathematical contributions to the theory of evolution. iii. regression, heredity, and panmixia. *Philosophical Transactions of the Royal Society of London. Series. A*, 187:253–318, 1896.

[107] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan. Beepbeep: a high accuracy acoustic ranging system using cots mobile devices. In *Proceedings of the 5th International Conference on Embedded Networked Sensor Systems (SenSys)*, pages 1–14, 2007.

[108] C. Peng, G. Shen, Y. Zhang, and S. Lu. Point&connect: intention-based device pairing for mobile phone users. In *Proc. of MobiSys*, 2009.

[109] Z. Pi and F. Khan. An introduction to millimeter-wave mobile broadband systems. *IEEE Comm. Magazine*, 49:101–107, 2009.

[110] Z. Pi and F. Khan. System design and network architecture for a millimeter-wave mobile broadband (mmb) system. In *Proc. of Sarnoff Symposium*, 2011.

[111] K. Poulsen. Wardriver pleads guilty in lowe's wifi hacks. SecurityFocus, June 2004. `http://www.securityfocus.com/news/8835`.

[112] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. The Cricket Location-Support System. In *Proceedings of the 6th Annual International Conference on Mobile Computing and Networking (MobiCom)*, pages 32–43, 2000.

[113] Q. Pu, S. Gupta, S. Gollakota, and S. Patel. Whole-home gesture recognition using wireless signals. In *Proc. of MobiCom*, 2013.

[114] J. Qiu, D. Chu, X. Meng, and T. Moscibroda. On the feasibility of real-time phone-to-phone 3d localization. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 190–203, 2011.

[115] D. Ramasamy, R. Ganti, and U. Madhow. On the capacity of picocellular networks. In *Proc. of ISIT*, 2013.

[116] D. Ramasamy, S. Venkateswaran, and U. Madhow. Compressive tracking with 1000-element arrays: A framework for multi-Gbps mm wave cellular downlinks. In *Proc. of Allerton*, 2012.

[117] S. Rangan, T. Rappaport, and E. Erkip. Millimeter-wave cellular wireless networks: Potentials and challenges. *Proc. of the IEEE*, 102(3):366–385, March 2014.

[118] G. Ranjan et al. Are call detail records biased for sampling human mobility? *ACM MC^{2}R*, 16(3):33–44, 2012.

[119] T. S. Rappaport, E. Ben-Dor, J. N. Murdock, and Y. Qiao. 38GHz and 60 GHz angle-dependent propagation for cellular & peer-to-peer wireless communications. In *Proc. of ICC*, 2012.

[120] T. S. Rappaport, F. Gutierrez, E. Ben-Dor, J. N. Murdock, Y. Qiao, and J. I. Tamir. Broadband millimeter-wave propagation measurements and models using adaptive-beam antennas for outdoor urban cellular communications. *IEEE TAP*, 61(4):1850–1859, 2013.

[121] J. Rekimoto and M. Saitoh. Augmented surfaces: a spatially continuous work space for hybrid computing environments. In *Proc. of SIGCHI*, 1999.

[122] I. Rhee et al. On the levy-walk nature of human mobility. *IEEE/ACM ToN*, 19(3):630–643, 2011.

[123] B. Roberts and K. Pahlavan. Site-specific RSS signature modeling for WiFi localization. In *Proc. of GLOBECOM*, 2009.

[124] J. Ryckaert, P. De Doncker, R. Meys, A. de Le Hoye, and S. Donnay. Channel model for wireless communication around human body. *Electronics Letters*, 40(9):543–544, 2004.

[125] J. Sallai, G. Balogh, M. Maroti, A. Ledeczi, and B. Kusy. Acoustic ranging in resource constrained sensor networks. In *Proceedings of the International Conference on Wireless Networks (ICWN)*, pages 467–474, 2004.

[126] A. Sani, L. Zhong, and A. Sabharwal. Directional antenna diversity for mobile devices: Characterizations and solutions. In *Proc. of MobiCom*, 2010.

[127] S. Saroiu and A. Wolman. Enabling new mobile applications with location proofs. In *Proc. of HotMobile*, 2009.

[128] A. Savvides, C. Han, and M. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *Proc. of MobiCom*, 2001.

[129] S. Scellato et al. Exploiting place features in link prediction on location-based social networks. In *Proc. of SIGKDD*, 2011.

[130] S. Scellato et al. Nextplace: a spatio-temporal prediction framework for pervasive systems. In *Proc. of Pervasive*, 2011.

[131] S. Scellato et al. Track globally, deliver locally: improving content delivery networks by tracking geographic social cascades. In *Proc. of WWW*, 2011.

[132] T. Schmidl and D. Cox. Robust frequency and timing synchronization for ofdm. *IEEE Transactions on Communications*, 45(12):1613–1621, 1997.

[133] J. Scott and B. Dragovic. Audio location: Accurate low-cost location sensing. *Pervasive Computing*, pages 307–311, 2005.

[134] S. Shah, S. Srirangarajan, and A. Tewfik. Implementation of a directional beacon-based position location algorithm in a signal processing framework. *IEEE Transactions on Wireless Communications*, 9(3):1044–1053, 2010.

[135] S. Singh, R. Mudumbai, and U. Madhow. Distributed coordination with deaf neighbors: efficient medium access for 60 GHz mesh networks. In *Proc. of Infocom*, 2010.

[136] S. Singh, R. Mudumbai, and U. Madhow. Interference analysis for highly directional 60-ghz mesh networks: The case for rethinking medium access control. *IEEE/ACM ToN*, 19(5), oct. 2011.

[137] P. Smulders and L. Correia. Characterisation of propagation in 60 ghz radio channels. *Electronics & communication engineering journal*, 9(2):73–80, 1997.

[138] Sony. Playstation Move. `http://us.playstation.com/ps3/playstation-move`.

[139] N. A. Streitz, J. Geißler, T. Holmer, S. Konomi, C. Müller-Tomfelde, W. Reischl, P. Rexroth, P. Seitz, and R. Steinmetz. i-land: an interactive landscape for creativity and innovation. In *Proc. of SIGCHI*, 1999.

[140] A. Subramanian, P. Deshpande, J. Gaojgao, and S. Das. Drive-by localization of roadside WiFi networks. In *Proc. of INFOCOM*, 2008.

[141] Y. Sun, J. Xiao, X. Li, and F. Cabrera-Mora. Adaptive source localization by a mobile robot using signal power gradient in sensor networks. In *Proc. of GLOBECOM*, 2008.

[142] Z. Sun, A. Purohit, R. Bose, and P. Zhang. Spartacus: spatially-aware interaction for mobile devices through energy-efficient audio sensing. In *Proc. of MobiSys*, 2013.

[143] X. Tie, K. Ramachandran, and R. Mahindra. On 60 ghz wireless link performance in indoor environments. In *Proc. of PAM*, 2012.

[144] C. Tuduce and T. Gross. A mobility model based on wlan traces and its validation. In *Proc. of INFOCOM*, 2005.

[145] E. Violette, R. Espeland, and G. R. Hand. Millimeter-wave urban and suburban propagation measurements using narrow and wide bandwidth channel probes. *NASA STI/Recon Technical Report*, 86:25683, 1985.

[146] D. Wagner and D. Schmalstieg. First steps towards handheld augmented reality. In *Proc. of ISWC*, 2003.

[147] L.-Y. Wei, Y. Zheng, and W.-C. Peng. Constructing popular routes from uncertain trajectories. In *Proc. of KDD*, 2012.

[148] T. Welch, R. Musselman, B. Emessiene, P. Gift, D. Choudhury, D. Cassadine, and S. Yano. The effects of the human body on UWB signal propagation in an indoor environment. *IEEE Journal on Selected Areas in Communications*, 20(9):1778–1782, 2002.

[149] K. Whitehouse and D. Culler. Calibration as parameter estimation in sensor networks. In *Proceedings of the 1st ACM International Workshop on Wireless Sensor Networks and Applications (WSNA)*, pages 59–67, 2002.

[150] J. Wiltse. Corrections to published curves for atmospheric attenuation in the 10 to 1000 ghz region. In *Proc. of ISAP*, 1997.

[151] C. Wu, Z. Yang, Y. Liu, and W. Xi. Will: Wireless indoor localization without site survey. *IEEE Transactions on Parallel and Distributed Systems*, 2013.

[152] J. Yoon et al. Building realistic mobility models from coarse-grained traces. In *Proc. of MobiSys*, 2006.

[153] H. Zhang, S. Venkateswaran, and U. Madhow. Channel modeling and mimo capacity for outdoor millimeter wave links. In *Proc. of WCNC*, 2010.

[154] Z. Zhang, D. Chu, J. Qiu, and T. Moscibroda. Demo: Sword fight with smartphones. In *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems (SenSys)*, pages 403–404, 2011.

[155] Z. Zhang et al. On the validity of geosocial mobility traces. In *Proc. of HotNets*, 2013.

[156] Y. Zheng et al. Mining interesting locations and travel sequences from gps trajectories. In *Proc. of WWW*, 2009.

[157] C. Zhou et al. Discovering personally meaningful places: An interactive clustering approach. *ACM TIS*, 25(3), 2007.

[158] X. Zhou, Z. Zhang, Y. Zhu, Y. Li, S. Kumar, A. Vahdat, B. Y. Zhao, and H. Zheng. Mirror mirror on the ceiling: Flexible wireless links for data centers. In *Proc. of SIGCOMM*, 2012.