University of California
Santa Barbara

Practical Mobile Sensing of the Environment

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

Doctor of Philosophy
in
Computer Science

by

Yanzi Zhu

Committee in charge:

Professor Haitao Zheng, Co-Chair
Professor Ben Y. Zhao, Co-Chair
Professor Upamanyu Madhow

September 2019

The Dissertation of Yanzi Zhu is approved.

Professor Upamanyu Madhow

Professor Ben Y. Zhao, Committee Co-Chair

Professor Haitao Zheng, Committee Co-Chair

July 2019

Practical Mobile Sensing of the Environment

by

Yanzi Zhu

To my family and friends,
Without whom none of my successes would be possible

Acknowledgements

I greatly thank my advisors Prof. Heather Zheng and Prof. Ben Y. Zhao for their guidance throughout my PhD studies. It is by their directions and advices that I accomplish all achievements these years. It is by their patience and high standard that I develop an independent, critical, and open mindset. I especially thank Heather who always reminds me of focusing on the high-level pictures, makes me think out of the box, and encourages me when I was down and struggled. Her generous support not only sheds light on my research, but also trains me towards a confident and self-aware person in life.

I also deeply thank my committee member Prof. Upamanyu Madhow for his generous support, valuable feedbacks and suggestions on my research.

I am grateful to work with all my collaborators, Yibo Zhu, Zengbin Zhang, Ana Nika, Zhijing Li, Yuanshun Kevin Yao, Zhujun Xiao, Yuxin Chen, Bolun Wang, Xinyi Zhang, Maryam Eslami Rasekh, and Zhinus Marzi. I shall remember the projects we fought for, the experiments we struggled with, the deadlines we chased, the days and nights we worked together, and the lunch and dinner we enjoyed.

I am also grateful to have my colleagues and friends, Shiliang Tang, Qingyun Liu, Bimal Viswanath, Xiaohan Zhao, Gang Wang, Jenna Cryan, Emily Wilson, Huiying Li, Max Liu, Jie Ren, Fang Felix Fu, He Shao, Fan Sun, Miles Boucher and many more who have accompanied me through this fun journey.

Finally, I would like to thank my beloved family for their endless love and support. I thank my parents Mr. Quanhong Zhu and Mrs. Yan Ge who have raised me and encouraged me to pursuit my dreams. I thank my girlfriend Ms. Suqin Hou who has always been by my side. And I thank my cat Lucifer who has enlightened my days.

Curriculum Vitæ

Yanzi Zhu

Education

2019	Ph.D. in Computer Science, University of California, Santa Barbara
2014	B.S. in Electrical Engineering, University of Minnesota, Twin Cities

Publications

- **Yanzi Zhu**, Zhujun Xiao, Yuxin Chen, Zhijing Li, Max Liu, Ben Y. Zhao and Haitao Zheng. "Et Tu Alexa? When Commodity WiFi Devices Turn into Adversarial Motion Sensors." *Submitted to Proceedings of the Network and Distributed System Security Symposium (NDSS)*, 2020.
- Zhujun Xiao, **Yanzi Zhu**, Yuxin Chen, Ben Y. Zhao, Junchen Jiang, and Haitao Zheng. "Addressing Training Bias via Automated Image Annotation." *arXiv preprint arXiv:1809.10242*, 2018.
- Zhijing Li, Zhujun Xiao, **Yanzi Zhu**, Irene Pattarachanyakul, Ben Y. Zhao, and Haitao Zheng. "Adversarial Localization against Wireless Cameras." *Proceedings of Workshop on Mobile Computing Systems and Applications (HotMobile)*, Tempe, AZ, Feb. 2018.
- **Yanzi Zhu**, Yuanshun Yao, Ben Y. Zhao and Haitao Zheng. "Object Recognition and Navigation using a Single Networking Device." *Proceedings of International Conference on Mobile Systems, Applications, and Services (MobiSys)*, Niagara Falls, NY, Jun. 2017.
- Zhijing Li, Ana Nika, Xinyi Zhang, **Yanzi Zhu**, Yuanshun Yao, Ben Y. Zhao and Haitao Zheng. "Identifying Value in Crowdsourced Wireless Signal Measurements." *Proceedings of International Conference on World Wide Web (WWW)*, Perth, Australia, Apr. 2017.
- Maryam Eslami Rasekh, Zhinus Marzi, **Yanzi Zhu**, Upamanyu Madhow and Haitao Zheng. "Noncoherent mmWave Path Tracking." *Proceedings of Workshop on Mobile Computing Systems and Applications (HotMobile)*, Sonoma, CA, Feb. 2017.
- **Yanzi Zhu**, Yibo Zhu, Ana Nika, Ben Y. Zhao and Haitao Zheng. "Trimming the Smartphone Network Stack." *Proceedings of Workshop on Hot Topics in Networks (HotNets)*, Atlanta, GA, Nov. 2016.
- Ana Nika, Zhijing Li, **Yanzi Zhu**, Yibo Zhu, Ben Y. Zhao, and Haitao Zheng. "Empirical Validation of Commodity Spectrum Monitoring." *Proceedings of ACM Conference on Embedded Networked Sensor Systems (SenSys)*, Stanford, CA, Nov. 2016.

- **Yanzi Zhu**, Yibo Zhu, Ben Y. Zhao and Haitao Zheng. "Reusing 60GHz Radios for Mobile Radar Imaging." *Proceedings of Annual International Conference on Mobile Computing and Networking (MobiCom)*, Paris, France, Sept. 2015.
- Yibo Zhu, **Yanzi Zhu**, Zengbin Zhang, Ben Y. Zhao, and Haitao Zheng. "60GHz Mobile Imaging Radar." *Proceedings of Workshop on Mobile Computing Systems and Applications (HotMobile)*, Santa Fe, NM, Feb. 2015.
- **Yanzi Zhu**, Peiran Suo, and Kia Bazargan. "Binary Stochastic Implementation of Digital Logic." *Proceedings of ACM International Symposium on Field Programmable Gate Arrays (FPGA)*, Monterey, CA, Feb. 2014.

Abstract

Practical Mobile Sensing of the Environment

by

Yanzi Zhu

Future of autonomous agents like drones and robots need accurate, robust and compact sensing solutions to map and understand their surrounding environments. Today's solutions are insufficient. Vision- and acoustic-based solutions are sensitive to lighting conditions and background noise, while more robust laser- and radar-based solutions require expensive and bulky hardware. Alternative solutions are necessary.

Networking radios provide a great opportunity. The same radio waves that we use for networking can also be used to perform mobile sensing. As these radio waves bounce off different objects, they carry information about the objects, which can be extracted by a capable receiver. Since these radios are already installed on mobile devices to enable networking, the cost of radio-frequency (RF) sensing is minimal.

The development of mobile RF sensing systems must also consider the impact of adversarial sensing attacks. As radio waves bounce around us, they also carry information of our physical status. The radio waves, when intercepted by attackers, can potentially reveal our private information. We need to develop robust defenses against such attacks.

In this dissertation, we explore the above two aspects of mobile RF sensing. Different from existing works that rely on specialized hardware, we focus on commodity networking hardware that is installed on mobile devices for networking. Along this line, we make two key contributions. First, we design, implement and evaluate an environmental mobile imaging system using 60GHz networking radios. Leveraging highly directional 60GHz signals, our system can reliably detect the location, size, shape and material of the

nearby objects, while navigating itself in unknown environment. Second, we propose and evaluate a silent reconnaissance attack. This leverages the presence of nearby commodity WiFi devices to track users inside private homes and offices, without compromising any WiFi network, data packets, or devices. We then evaluate potential defenses, and propose a practical and effective defense based on AP signal obfuscation.

In summary, this dissertation explores two key aspects of RF sensing for future mobile agents. It addresses a key challenge faced by today's mobile environmental sensing systems. It then identifies a new reconnaissance attack that invades our physical security and privacy, and proposes practical defenses against the attack. We hope our work sheds light on the development of future mobile sensing systems.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The mobile computing ecosystem will involve a variety of autonomous mobile agents, ranging from drones and robots to self-driving cars. Many companies like Amazon, DHL and UPS have already drawn plans for drone deliveries in regularized environments [1, 2, 3]. Startups like Ziplines and QuiQui have also announced medicine delivery solutions using drones [4, 5]. In the near future, we will have advanced cleaning robots that organize our rooms, personal companion robots that guide visually impaired people, and robotic housekeepers that perform time-consuming and heavy-lifting tasks.

For these mobile agents to become fully automated, they must first be able to accurately sense their surroundings. For example, delivery drones must recognize obstacles to avoid collisions, and image and identify nearby objects. The corresponding sensing system should satisfy the following requirements. It should locate nearby objects and recognize their size and shape accurately, *e.g.*, at centimeter-level accuracy. The system should also work robustly under various conditions, *e.g.*, in the dark, in the rain, and in both indoor and outdoor scenarios. Finally, the sensing solution should be compact and cost-effective so it can be placed on a variety of mobile devices.

Today's solutions are insufficient. Traditional solutions like sonar [6], radar [7, 8],

and LIDAR [9, 10, 11] all use specialized hardware. Adding extra hardware on mobile devices increases their cost and size, which is not suitable for autonomous mobile agents. Although recent developments of radar systems [12, 13, 14] use customized signals on WiFi frequencies, these systems still miss the goal of achieving high imaging accuracy using a small device. Another direction is to use mobile devices' cameras with advanced computer vision techniques [15, 16]. But they work poorly in darkness and they cannot distinguish objects from their backgrounds when both have similar colors (*e.g.*, fatal accidents about Tesla [17, 18]).

Networking radios provide a great opportunity. The radio waves solely used for wireless communications can be used for sensing. As these radio waves penetrate through and reflect from the nearby objects, they carry the information about the objects. By analyzing the physical properties of these signals, a capable receiver can extract rich information about the objects. These networking radios are compact, energy-efficient and are currently deployed on mobile devices [19, 20]. As such, a mobile sensing system built on top of the radio signals can serve as a low-cost and robust alternative to existing solutions.

On the other hand, the development of mobile sensing systems must consider the impact of being misused. As radio signals bounce off us, they carry our physical information like locations. Capable attackers can intercept such signals and track our physical status, allowing them to commit serious physical crimes. Examples include surveillance on banks prior to robbery, burglary to homes and offices, and even planning attacks against government agencies. Even worse, due to the broadcasting nature of wireless signals, attackers can achieve sensing by passively listening to existing transmissions without any risk of detection. These potential adversarial uses of RF sensing bring significant security risks and privacy concerns that need to be defended against.

With these in mind, this dissertation addresses both aspects of RF-based environmen-

tal sensing: the first half of the dissertation designs accurate and robust sensing systems, while the second half discovers potential sensing attacks and develops robust defenses. In terms of the sensing system design, this dissertation differs from the existing works by only using *commodity networking radios*. We make two key contributions. *First*, we demonstrate an accurate and compact environmental imaging system that uses a pair of 60GHz networking radios. The system can robustly recover the nearby objects' location, size, shape and material.

Secondly, in terms of sensing attacks and defenses, we propose and evaluate a silent reconnaissance attack where an adversary can listen to ambient WiFi signals from the existing WiFi devices to track people in homes and offices. Different from existing works that use specialized hardware [21], our attack only needs a compact sniffer with a WiFi networking radio and a single antenna. We believe this is the first in a new class of silent reconnaissance attacks that are notable because of their passive nature and general applicability. Finally, we propose a promising defense against the attack by signal obfuscations at the access point (AP).

Next, we provide a detailed summary of both aspects.

1.1 Design of Robust Mobile Imaging Systems

We first present our design and implementation of an accurate, robust, and compact mobile sensing system that maps the surrounding objects and obstacles. To achieve this, existing imaging solutions using camera [15, 16], sonar [6], radar [7, 8, 12, 13, 14], and LIDAR [9, 10, 11] are inadequate. Instead, we show that 60GHz networking radios can lead to a highly accurate sensing system.

The 60GHz high-frequency band was approved for wireless communications by the Federal Communications Commission (FCC) in 2013, as an unlicensed band like WiFi.

Sensing using 60GHz signals has three main advantages over the alternatives. *First*, 60GHz signals are stable and predictable. Since 60GHz signals are highly directional, they suffer less from interference and have minimal multipath effects. This allows us to accurately model the signal propagations. Second, 60GHz signals remain robust in different lighting conditions and in the rain [22]. The radios are inexpensive and small enough to fit into small mobile devices like drones. Finally, from radar theory [23], 60GHz signals have small wavelength and thus should achieve higher accuracy than WiFi.

Leveraging these physical properties of 60GHz radios, we addressed three key challenges in designing an environmental sensing system: *(1)* achieving centimeter-level object imaging accuracy; *(2)* tolerating device tracking errors; and *(3)* accomplishing sensing using a single device.

Achieving centimeter-level imaging accuracy. Although 60GHz-based sensing systems should ideally achieve 12x more precision than those based on WiFi signals due to the small wavelength of 60GHz signals, in many cases such precision is still inadequate. Considering the size of small mobile agents like drones, the imaging resolution is no better than 1 meter when the agent is just several meters away from the target objects. To solve this, we take an alternative approach similar to the synthetic aperture radar (SAR [23]). Our intuition is to leverage the device mobility to emulate a virtual antenna array with a large aperture. Our initial design requires two cooperative devices: one moves around as a receiver, and another stays stationary as an anchor and transmits 60GHz signals. By aggregating the scattering signals from the objects along an x-meter trajectory, we virtually create an x-meter antenna array, which largely improves the imaging accuracy. We show via controlled experiments and simulations that our design can locate and image an object accurately, with errors below 10cm.

Tolerating device tracking errors. As we deploy the above system in practice,

we find the SAR-based design fails to produce accurate imaging results. The key reason is that it models any object surface as a collection of individual points, and images the objects by locating each individual point. Doing so requires the mobile device to track its trajectory with the accuracy of the 60GHz wavelength (5mm), which is infeasible. As a result, any estimation error will lead to inconsistent estimation of the signal phase at the receiver. These phase errors then translate into large errors in object location and imaging results.

We address this challenge by proposing a new object surface model that treats the entire object surface as a single unit, *e.g.*, straight and curved lines (from top view). Since the signals reflected from an object's surface highly correlate with the objects' location, shape, size and material, we developed a series of *RSS*[1] *Series Analysis* (RSA) algorithms that extracts the information only from RSS amplitudes. Instead of using signal phase, which is sensitive to device tracking errors, we use the RSS measurements and the corresponding signal directions, *i.e.* angle of arrival (AoA), provided by the 60GHz physical layer module. Since both RSS and AoA are reasonably robust against errors in device trajectory estimation, so is our proposed sensing system. Using testbed experiments, we confirm that even in the presence of movement deviations as large as ±10cm (within 1m moving distance), our sensing system can achieve less than 5cm error in location and surface width estimation, 100% accuracy in determining surface curvature (flat, concave, convex), and can narrow down to top 3 material choices.

Accomplishing sensing using a single device. Our initial design implements sensing using two cooperative mobile devices. We then improve our design by using a single mobile device, where a pair of transmitting and receiving 60GHz radios are co-located. This eliminates the complex coordination among multiple devices. We also design a device navigation module using solely 60GHz radios and integrate it with the

[1]RSS is short for Received Signal Strength.

sensing module, creating a compact, low-cost and practical solution using a single mobile device.

For imaging, co-locating the two networking radios on the same device creates two new challenges. *First*, the previous imaging algorithm no longer works, as the transmitting radio moves with the receiver and can no longer serve as a static anchor. *Second*, the sensing system will have limited visibility. Due to the co-location and the highly directional 60GHz signals, we can only obtain specular reflections from an object's surface, within a limited range of angles, *i.e.* at a limited set of locations.

To address these challenges, we propose a new object model that breaks down an object's surface into small, connected flat units. Since each unit reflects signals like a mirror, the transmitting and receiving signal directions are highly correlated with the orientation of each small unit. Using this correlation, by measuring the continuous changes of these signal directions along the device movement, we can first recover the object surface shape. We then resize and shift the recovered shape to the correct position based on their RSS. Similar to our previous design, we only use RSS and signal directions, *i.e.* AoA and angle of transmission (AoT). And our design is resilient to device trajectory deviations.

For navigation, we leverage existing algorithms [24, 25, 26] but introduce two key innovations. *First*, we propose to compute a safety zone around the mobile device using RSS signals measured by the 60GHz radio. *Second*, we compute the moving trajectory that the device should follow to image its nearby objects within the safety zone.

We integrate the imaging and navigation modules and prototype them using commercial-off-the-shelf 60GHz radios donated by Facebook, and show that our single-device sensing system works as accurately as our previously proposed two-device system. Our system is comparable to the state-of-the-art camera-based system, and it works robustly in low-light conditions, both indoor and outdoor.

Overall, our proposed sensing systems are the first practical, accurate mobile imaging systems that use 60GHz networking radios. They provide the basic primitives towards a detailed environmental sensing system for autonomous mobile agents.

1.2 Potential Sensing Attacks and Defenses

The second part of this dissertation explores the potential misuses of RF sensing and develops defenses against them. In particular, we focus on attacks that aim to detect and track users in their private homes and offices. This belongs to a common category of reconnaissance attacks, a precursor to many serious crimes.

To achieve the goal, the attackers need a reliable, compact and cost-effective method, just like the autonomous mobile agents. On top of these, the method should be *undetectable*. This makes the attack stealthy and thus extremely powerful. Although there are other methods, like observing the lights on and off, thermal imaging, microphone eavesdropping and network traffic analysis. these are either ineffective, or easily thwarted.

We find RF sensing is an attractive alternative for the attack. Using WiFi as an example, walls and buildings today are not built to insulate against WiFi signals, and these signals can be overheard by outside receivers. Also, since WiFi signals bounce off our bodies, human motions near a WiFi transmitter can change the signal propagation from transmitters to outside receivers. Leveraging these, we develop a new set of physical reconnaissance attacks that secretly turn a transmitting WiFi device into a tracking device. Although there are other sensing methods that actively transmit signals to track humans, they can be easily detected. Instead, our approach is undetectable, as attackers only need to sniff the ambient WiFi signals from an existing infrastructure (*e.g.*, wireless networking devices).

Attacks by sniffing the ambient WiFi signals. We present a new set of silent

reconnaissance attacks that leverages the presence of the pervasive WiFi ambient signals to silently track humans in their homes and offices. We demonstrate that even when the WiFi networks are encrypted and the WiFi devices are completely secured, by just sniffing the signals and modeling their variations, our attacks can accurately detect and locate humans to individual rooms (an overall 99.7% human detection rate). To realize the attack, all the attacker needs is a mobile device with a WiFi networking radio placed outside of the target property and only listen to existing WiFi signals. The resulting attack is simple and yet highly effective, posing a significant threat to our physical security and privacy.

Defenses via router-based signal obfuscation. We then explore potential defenses against our proposed attacks. Since the attacks heavily rely on the quantity and quality of the sniffed signals, we can either reduce the amount of transmitted packets that can be overheard by the attack sniffer, *e.g.*, by geo-fencing and packet rate limiting, or add noise to signals (*i.e.* obfuscation), *e.g.*, power randomization. But we find these methods are either ineffective or impractical.

We instead propose a practical defense that uses WiFi APs to obfuscate signals by actively injecting cover packets on behalf of the WiFi devices. Doing so only needs firmware/software changes to the routers and only adds small overheads to existing communication channels. This defense can effectively confuse the attacker sniffer by reducing the user detection rate by half and raising the false alarm rate significantly. While our proposed defense could be countered by more resourceful attackers, they require bulky, specialized hardware that would easily raise suspicion and cost.

1.3 Organization

The remaining content is organized as follows. Chapter 2 describes the design of a series of object imaging systems that identify and map surrounding objects using commodity 60GHz networking radios. In Chapter 3, we demonstrate in details a new type of silent reconnaissance attacks that use WiFi networking radios to passively listen to the ambient WiFi signals and track nearby people in rooms. We then analyze possible defenses and present our practical defenses by only changing the firmware of routers. Finally, we conclude the thesis in Chapter 4.

Chapter 2

Robust Mobile Imaging Systems

In this chapter, we present the details on our designs of an accurate, robust environmental sensing system. Unlike traditional radar approaches that leverage specialized hardware, our design focuses on reusing the networking radios on the mobile devices. In particular, we leverage the 60GHz networking radios to build a mobile sensing system that maps the location, shape, and material of the nearby objects.

Our initial design (§2.1) demonstrated the feasibility to image the nearby targets with only 60GHz networking radios. In our design, we leveraged two cooperative devices, one transmitting signals as an anchor while the other capturing the reflected signals from the nearby objects. To reach our goal of the centimeter-level imaging accuracy, we leveraged the device mobility to help construct a virtual antenna array along the moving path based on traditional radar algorithms. The resulting design should image the objects accurately with the location and size errors $<$10 centimeters. But in the practical deployment, we discovered that the traditional synthetic aperture radar algorithms are very sensitive to device trajectory tracking errors (§2.2). Due to the 60GHz's small wavelength, these traditional algorithms can only tolerate the tracking errors that are comparable to or less than 5 millimeters. Doing so is clearly impractical. We addressed the problem by

designing a new 60GHz ray-tracing model and proposing a set of novel 60GHz imaging algorithms, named RSA.

Although the RSA design is already accurate, robust and compact, it requires the coordination of two devices. We further advance the system to work independently on a single device, by co-locating the transmitting and receiving networking radios on the same device (§2.3). But doing so voids the anchor-based model from RSA. So we design an alternative, new 60GHz imaging algorithm that works on a single device. The design integrates the device navigation with imaging, and it remains robust to the device tracking errors.

2.1 60GHz Mobile Radar Imaging

[1] Mobile computing is undergoing a significant shift right before our eyes. In the past, the user was the center of the mobile network, and her movements determined the operational properties of the mobile network. But this is changing with the arrival of autonomous mobile agents for a variety of applications. Today, semi-autonomous drones are carrying out military missions in lieu of manned-flights, while vacuum robots search for dirt in our homes. In the near future, intelligent cars will be fully in control of delivering us to our destinations, and first responder robots will be first on scene to find and rescue victims in disasters [28].

One of the critical challenges limiting the growth of these autonomous devices is the lack of accurate sensing systems, *e.g.*, a mobile imaging radar system that captures the position, shape and surface material of nearby objects. These devices often operate in less than ideal sensing environments: at night or in dark rooms, or while moving at moderate speeds. Yet the desired level of accuracy is very high, and errors in sensing can

[1] The content in this section is published in [27].

produce dire consequences. For example, Google's self-driving cars are reported to use maps with inch-level precisions [29], while devices that assist the visually impaired must have errors smaller than 10cm [30, 31].

These constraints dramatically reduce the set of possible solutions. Traditional imaging systems rely on visible light imaging using cameras and object recognition. Unfortunately, they perform poorly in dark or low-light conditions, and lack the precision desired by these applications. Another approach relies on specialized hardware such as large lens radar for accurate signal detection and processing. But these devices are neither portable nor cost-effective for commodity devices. Finally, acoustic solutions have been used successfully for sensing over very short distances [32], but are easily disrupted by background noise and fail over longer distances.

60GHz imaging radar. An intriguing and still unexplored solution is a digital imaging radar system using reflective properties of narrow beamforming wireless links. A radar system using high frequency RF signals (*e.g.*, 60GHz) has a number of key advantages over existing alternatives. First, 60GHz links are directional and highly focused, making them relatively immune to interference from environmental factors. Second, 60GHz beams exhibit good reflective properties, and work reliably regardless of lighting conditions under most indoor or outdoor conditions. Finally, 60GHz radios are relatively inexpensive, and small enough to be included in today's smartphones and tablets.

In this section, we present early results in our efforts to design and evaluate a digital imaging radar system using reflections from 60GHz wireless beams. Such a system faces a fundamental challenge, that it is technically infeasible to build an accurate imaging radar using wireless hardware on a static mobile device. A simple rule from imaging radar theory [23], defined by Equation (2.1), holds for accuracy (radar resolution) and antenna size (aperture). For smartphone-sized antennas, even the most high frequency radios (5–120GHz) can produce resolutions no better than 1 meter, clearly insufficient

for our needs.

$$resolution = wavelength \times distance/aperture \tag{2.1}$$

Virtual antenna arrays. We take an alternative approach, by using user mobility to emulate a virtual antenna array with large aperture. Our design includes the user's mobile device as a receiver, with a decoupled transmitter either embedded in the infrastructure or "deployed" on-demand by the user (*e.g.*, dropped by a drone). By taking measurements of the same reflected signal at multiple locations, we can emulate the signals received by different elements of a large antenna array. In addition, we can further improve the resolution of our "virtual antenna" using 60GHz transmissions. Since 60GHz has a carrier wavelength of 5mm (12× shorter than WiFi and cellular), using 60GHz links means a user can obtain fine-grain resolution with just small movements in the measurement area.

In the remainder of this section, we present Nightcrawler, a 60GHz-based mobile radar system that leverages user mobility to emulate a large-aperture antenna array. We describe details of our design, including mechanisms for object detection, object imaging, and controlling precision. We present experimental results on a real 60GHz testbed, and show that we can achieve high precision (∼1cm) imaging with as little user movement as half a meter.

Our work is a promising first step in the development of high precision, wireless imaging radar systems. Initial results show promising accuracy, as well as added potential for using loss profiles to infer the *surface material* on detected objects. Ongoing work focuses on tolerating location errors for the transmitter, as well as extending imaging to multiple objects.

2.1.1 Conventional vs. Mobile Radar

Before presenting our design of a high precision radar system, we need to first describe the principle and hardware requirements behind conventional imaging radars. We will then explain the differences between personal mobile radar systems and conventional imaging radar systems, and the challenges that arise as a result.

Traditional radar imaging. Imaging radars detect the presence, position, and shape of an object by emitting directional RF signals and capturing/analyzing the portion of signal reflected by the object. Specifically, a radar estimates its distance to the object by measuring the round trip time of the reflected signal, either directly using a highly precise clock, or indirectly by transmitting frequency modulation (FM) pulses and measuring the frequency offset of the reflected signal [23]. The radar also uses highly directional RF signals to "scan" the object. Because the signals reflected from the object and its nearby spaces carry different signal strengths, the radar can identify the object's position and shape with high precision. Finally, high-end radars can identify object material using dispersion analysis, where they emit RF signals at various carrier frequencies and collect reflection results. Since different materials have different reflection profiles across frequencies, one can estimate material type by analyzing reflection results.

Overall, traditional imaging radars have strong requirements on radio hardware, *e.g.*, they require specialized FM circuits and highly directional dish antennas. These are easily met for applications where radar size and cost are not an issue, such as military radar systems or radio telescopes for use in astronomy.

Why mobile radar imaging is hard. Our goal in this work is to design radar imaging systems to enable commodity mobile devices to recognize their surrounding environments. This is highly challenging, due to tight constraints on radio size, functionality and cost. *First*, the small form factor of mobile devices puts a hard limit on both antenna size

(which determines aperture) and signal directionality. As shown by the Radar Theory in Equation (2.1), the small antenna size severely limits the maximum imaging resolution. For smartphone-sized antennas (2.5cm aperture), the maximum imaging resolution for an object of 10m away is 1m using 120GHz transmissions or 24m at 5GHz. *Second,* today's mobile devices are not equipped with FM pulse circuits, which are required for distance estimation by traditional radar imaging. Adding such circuits would significantly increase costs for budget-conscious mobile radio chipsets. Similar cost constraints prohibit the inclusion of hardware solutions to perform dispersion analysis for material detection or clock-based distance computation[2].

2.1.2 60GHz Imaging Radar

To overcome challenges of size and cost in mobile devices, we propose to leverage human mobility to extend the reach of a single mobile antenna. We propose *Nightcrawler*, a mobile radar imaging system using commodity 60GHz networking chipsets[3]. Using commodity chipsets, Nightcrawler performs object imaging using just signal measurements, and improves imaging resolution far beyond the theoretical limit defined by Equation (2.1). It achieves this by leveraging *user mobility* and unique RF propagation properties of *60GHz transmissions*. This section describes our core ideas and sets the context for details of our prototype in §2.1.3.

Leveraging 60GHz. Today's mobile devices are equipped with multiple wireless interfaces, *e.g.*, cellular, WiFi, Bluetooth, and 60GHz radio [19]. We implement Nightcrawler using 60GHz radios because its unique propagation properties present three significant advantages for our application.

[2]To measure round trip time accurately, *i.e.* with 1cm accuracy, the clock precision must be at least 0.033ns, which is extremely hard to realize on smartphones and laptops.

[3]Low-cost 60GHz chipsets are available today on the mass market, *e.g.*, WiloCity chipsets cost $37.5 and has a 23$m$ range [19, 22].

- 60GHz has a carrier wavelength of 5mm, more than 12x shorter than WiFi and cellular. According to Equation (2.1), the required antenna aperture for 60GHz is at least 12x smaller than WiFi/cellular for the same imaging resolution.

- 60GHz's short wavelength also makes its propagation much more stable/predictable. With minimum multi-path effects, signal strength remains stable over time, and is strongly correlated with propagation distance. This increases the robustness of our imaging design. For example, our imaging system can easily distinguish between a line-of-sight signal and a reflected signal that traveled over a longer distance, and use this fact to detect the presence of objects in local neighborhood.

- The object reflection profile is more stable at 60GHz. For example, the signal reflection loss has strong correlation with the object material. This enables Nightcrawler to narrow down the material type using signal strength measurements.

Mobility enabled virtual antenna array. Nightcrawler exploits the fact that as a user moves, her mobile device can take signal measurements at multiple locations, emulating a virtual antenna array whose antenna aperture is significantly larger[4]. This enables highly directional signal reception by a mobile device similar to those required by conventional radar imaging, and overcomes the limitation imposed by the size of mobile devices.

User mobility also increases the system's detection range and ability to detect surface curvature of objects. Surfaces with different curvatures reflect the signal to different directions in the space. Measuring reflections from different locations helps the radar capture the curvature of each of the object's multiple faces.

Decoupling transmitter and receiver. Given the small size of mobile devices, any mobile radar system cannot rely on just a single device to serve as both transmitter and

[4]Aperture of virtual array is equal to distance traveled by the user.

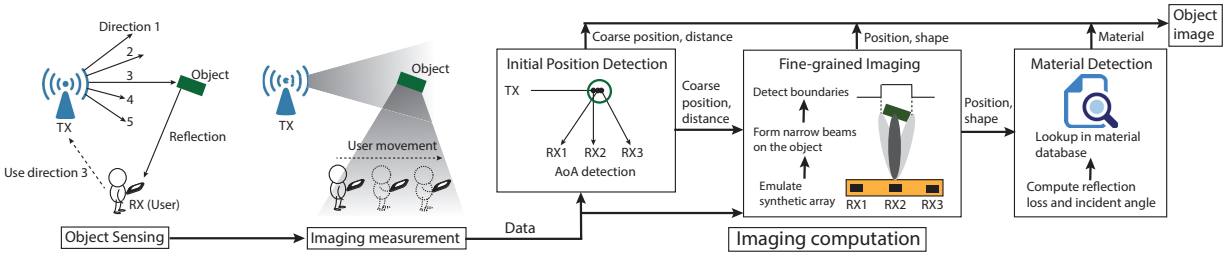


Figure 2.1: The high-level overview of the Nightcrawler radar imaging system.

receiver. Our design for a mobile radar system involves the primary mobile device, which acts as a receiver, and a decoupled transmitter, which can be either infrastructure-based, or a separate mobile device.

For example, an imaging system to assist the visually impaired may include an app on the user's smartphone, which coordinates with one or more transmitters embedded in the walls or ceiling. In contrast, an autonomous device (*e.g.*, first responder robots) can "deploy" a secondary transmitter device.

Once deployed (or periodically for infrastructure devices), the transmitter (TX) sends 60GHz beacons that reflect off of nearby objects[5]. Each beacon includes the angle of transmission, and if possible the transmitter's location. Users hold a mobile device equipped with a 60GHz receiver (RX), and move in pedestrian speeds. Each RX periodically scans[6] and records signal strengths for beacons across different directions. Nightcrawler processes these data on the fly to identify, locate and image objects in the local area.

[5]The beacon transmitters rotate their beam direction periodically to cover multiple objects or larger objects.

[6]Today's 60GHz antenna arrays can adjust beam direction every $50us$. So each RX can scan multiple directions in real time.

2.1.3 Nightcrawler: a First Look

We now describe our initial design. Seen in Figure 2.1, a primary device (RX) and decoupled transmitter (TX) start from "sensing" mode to identify the presence of any object. Upon detection, they switch to "imaging" mode to build a physical map of the object(s). We assume that the RX knows its relative position from the TX.

Object Sensing

Nightcrawler devices sense objects using the bootstrapping procedure defined by IEEE 802.11ad, the standard for 60GHz transmissions [33]. The TX operates in the directional mode, steers its beam to different directions, *e.g.*, in sectors of 3° in width, and embeds the direction in the signal. Operating in the omni-directional mode, the RX measures RSS and reports a list of TX beam directions where RSS exceeds the noise level[7]. The RX then identifies and removes from the list the set of TX beam directions whose transmissions did not experience any reflection. The remaining list of directions, if any, are those where the transmission was reflected, implying that at least one object exists in the local neighborhood.

To identify TX beam directions that did not experience reflection, the RX uses simple geometry to locate a set of candidate LoS beam directions based on the relative position of TX and RX and their antenna radiation patterns. It then validates each candidate direction by comparing its RSS to the model-predicted value without any reflection. If a direction gets (partially) reflected, its RSS will be lower than the model-predicted value due to longer propagation path and possible reflection loss.

[7]This step is slightly different from 802.11ad where the RX only reports the direction with the strongest RSS.

Object Imaging

After detecting the presence of objects, Nightcrawler devices enter the "imaging" mode. Intuitively, Nightcrawler should use the above collection of "reflected TX beam directions" to drive imaging. That is, the TX focuses its transmissions on these directions (by rotating its beam repeatedly across them in a round-robin fashion) while the RX locates and images object(s) in each direction. To improve imaging efficiency, it is desirable to identify a subset of the directions that cover all the potential objects. In our preliminary work, we leave this optimization to future work and simply assume that the reflected direction set only has a single direction.

With this in mind, our following description on Nightcrawler assumes that during imaging, the TX focuses its beam on the targeted direction and transmits the same beacon signal repeatedly. The RX, while moving, operates in the directional mode and steers its beam around to capture signals at each measurement location. This is done using the *antenna alignment* procedure defined by 802.11ad — the RX steers its beam across various directions and reports the direction with the strongest RSS. Once the movement distance is sufficient, the RX executes the imaging algorithm on the measurement data to locate and image the object.

The Nightcrawler imaging algorithm includes three steps: (1) *coarse position estimation*, (2) *fine-grained imaging*, and (3) *material detection*. We now describe them in more details.

1. Coarse position estimation. Nightcrawler first estimates the object's relative position and distance to the RX. This narrows down the search space for the next step, which applies a more sophisticated approach to perform detailed imaging. The RX estimates the object position by extracting the angle of arrival (AoA) of the beacon signal. At each measurement location, Nightcrawler derives the AoA as the strongest

receive beam direction. Since the TX embeds the beam direction in each beacon signal, the RX can estimate the object position as the intersection of the TX beam direction and the AoA.

Ideally, Nightcrawler should identify object position reliably from measurements at a single location. In practice, AoA detection can be noisy due to hardware artifacts, imperfect reflection from uneven surface, and the fact that each TX beam is not narrow enough. For example, our testbed results show that when using a TX beam of 10° beamwidth, the noise in AoA estimation can lead to up to $1m$ position error when the object is $6m$ away from the RX.

Nightcrawler overcomes this challenge by performing "majority vote" on measurements collected at multiple locations. Specifically, Nightcrawler considers data from N locations, each producing an estimated object position. It then identifies a cluster of $[N/2]+1$ positions with the minimum MSE among themselves, and computes the center of the cluster, *i.e.* the position with the minimum MSE to all the positions, as the final object position. This solution, while simple, can effectively improve the positioning accuracy. Our testbed results in §2.1.4 show that with N=9, the position error in the above example reduces from $1m$ to below $10cm$.

2. Fine-grained imaging. This step derives the precise position and shape of the object by implementing a large aperture virtual antenna array from aggregating signal measurements at different locations. Specifically, Nightcrawler identifies the object shape by detecting its boundaries as well as surface curvature, *i.e.* flat, convex or concave.

Detecting object boundaries. Inspired by airplane radars that implement synthetic aperture radar (SAR) to detect object size [23], Nightcrawler uses a small and moving RX antenna to emulate elements of a large array. The resulting synthetic array has a very narrow beam pattern and can identify signals at fine-grained directions. Thus the RX can observe a sharp decrease in RSS along the object boundaries, and locate these

boundaries with errors bounded by the (very narrow) beamwidth of the synthetic array.

A key component of our design is how to aggregate measured signals across locations to emulate the large array. This is done by "reverse-engineering" the process of a phased array focusing its beam. Specifically, let the estimated object position in the previous step be X_0. Nightcrawler picks a set of reflection "focus points" near X_0 as the potential boundary positions. Given a target image resolution r, any two neighboring focus points should be within a distance of $r/2$. For each focus point, Nightcrawler applies a *focus* process to derive the RSS of signals reflected by the small area of width r around the given focus point. This is done by first shifting the phase of signals collected at each measurement location by its distance to the focus point and then summing up all the signals across locations. After applying this on all the focus points, the RX obtains a reflected RSS map along the object itself. The object boundaries are the two focus points where the RSS drops sharply.

Note that Nightcrawler emulates the large array without synchronizing TX and RX. This is because all the measurements are done by a single receiver RX. As long as the TX sends the same beacon signal (per TX beam direction) during imaging, the RX can eliminate any phase offset caused by differences in measurement time.

Inferring surface curvature. Nightcrawler recognizes the object's surface curvature based on a simple intuition — signals reflected by a flat surface display a standard sector shape that can be reconstructed based on the antenna pattern and the signal propagation distance, while signals reflected by a convex (concave) surface display a wider (narrower) sector shape. Driven by this intuition, Nightcrawler infers the surface curvature by the RX constructing the beam pattern of the received signal. Specifically, as the RX moves, it measures the RSS at different segment of the signal beam and aggregates them to build the received beam pattern.

While our first design identifies the type of surface curvature (flat/convex/concave),

our ultimate goal is to discover detailed surface feature such as the curvature radius. This requires more sophisticated models on 60GHz signal reflection, which we leave to future work.

3. Material detection. Finally, Nightcrawler infers the object material based on the RSS loss due to reflection. At 60GHz, the reflection loss correlates strongly with the material type and the incident angle. Existing measurements have built a comprehensive database on 60GHz reflection loss, covering 38 common materials and different incident angles [34]. Our own measurements on five different materials also align with existing findings.

The key element is to accurately determine the amount of RSS loss due to reflection and the reflection incident angle. To derive the reflection loss, Nightcrawler first computes the signal propagation distance (TX $\rightarrow$ object $\rightarrow$ RX) and applies the Friis free-space model to derive RSS without any reflection loss (RSS^*). It then subtracts from RSS^* the measured RSS value to derive the reflection loss. Computing the signal incident angle is easy given the relative position between TX and RX.

Imaging Overhead vs. Precision

Nightcrawler's imaging computation overhead is low. Our MATLAB implementation finishes in less than $15ms$ for all test cases. We expect that a good native C implementation on mobile devices should be comparable if not faster. Therefore, Nightcrawler's overall overhead and delay are dominated by its signal measurements.

Nightcrawler's measurement delay depends on user walking distance. The further the user walks, the larger the imaging delay. But user walking distance also directly affects the size (or aperture) of the synthetic array and thus imaging resolution. So there exists a tradeoff between imaging response time and resolution.

We should also pay attention to measurement frequency, *i.e.* the number of mea-

surement locations for a given walk distance. Ideally we should minimize measurement frequency to save energy. However, since the number of measurement locations maps to the number of elements in the synthetic array, we need sufficient number of measurements to remove array artifacts such as side lobes. Our initial analysis suggests that for pedestrian speeds up to $1m/s$, the measurement frequency of 1 per $40ms$ (or 1 per $4cm$ movement) is sufficient to produce a high-quality synthetic array.

2.1.4 Initial Feasibility Study

We perform initial evaluation on Nightcrawler using both testbed measurements and system simulations. We use commercial off-the-shelf 60GHz radios to conduct microbenchmark experiments on Nightcrawler, and to evaluate its end-to-end imaging performance under simple scenarios. We also run simulations to identify potential performance of Nightcrawler under general scenarios.

Testbed Measurements

Our testbed consists of two HXI Gigalink 6451 60GHz radios[8], one as the transmitter (TX) and the other as the mobile receiver (RX). Compared with an ideal Nightcrawler system, the testbed has two hardware limitations. *First*, since there is no suitable 60GHz steerable antenna array on the market, we emulate beam steering by setting a horn antenna on a mechanical rotator and adjusting its beam direction in units of 0.5°. This can provide accurate results because 60GHz signal strength is largely determined by directionality and signal patterns of the main beam lobe, and our horn antenna's main lobe pattern closely aligns with that of a 10x10 array [22]. Since 60GHz propagation is stable over time (verified by others [35, 36] and our own measurements), at each location the RX can accurately measure RSS across different directions despite its slower beam

[8]http://www.hxi.com/

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

Figure 2.2: 5 different objects used in our testbed measurements.

steering speed. *Second*, the HXI radio reports RSS without any phase information, so in the computation we set the phase of signals measured at all RX locations to the same value. This makes it difficult to perfectly focus the beam during boundary detection, and can potentially degrade the imaging performance.

Our experiments consider a simple scenario of object recognition. We place an object in the middle of a room. The TX is $2m$ away from the object and emits a fixed beam towards the object. The RX starts from an arbitrary location in the room, and as she walks around, Nightcrawler identifies the object position and shape. We test five objects with different size and surface curvature, shown in Figure 2.2. We also experiment with pedestrian users as objects. By default, the user walks $45cm$ and performs one RSS measurements every $1cm$. As mentioned earlier, we assume the RX knows her relative position to the TX.

Position & Distance Accuracy. We first examine the accuracy of the coarse position detection described in §2.1.3 with $N = 9$. Table 2.1 lists errors in estimated position, distance and surface orientation when the RX is $3m$ away from the object. Across the five different objects, the position offset ranges between $1.7cm$ and $12cm$ while the distance offset is even smaller ($< 0.4cm$)[9]. This translates into less than 1° orientation error. Furthermore, we observe that the accuracy is higher for objects with planar surfaces, compared to those with convex surfaces. This is because signals reflected

[9]The distance offset is the projection of the position offset along the line of object→RX.

Objects in Figure 2.2	Position offset	Distance offset	Orientation error
(a) Desktop (Metal)	1.7cm	0.1cm	0.2°
(b) Monitor (Plastic)	6.9cm	0.1cm	0.6°
(c) Board (Wood)	5.5cm	0.1cm	0.5°
(d) Convex Box (Plastic)	12.3cm	0.4cm	1.0°
(e) Cylinder (Metal)	10.4cm	0.3cm	/

Table 2.1: Performance of Nightcrawler's position estimation.

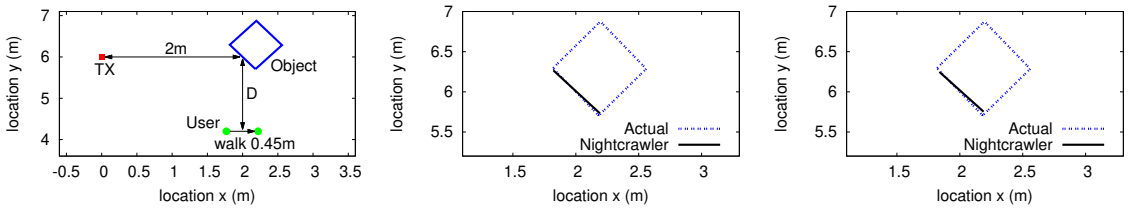
by convex surfaces become more scattered compared with planar surfaces, leading to larger variance in estimated reflection points. We also repeat the experiments by varying the RX to object distance between $2m$ and $6m$ and obtain similar results. Overall, Nightcrawler achieves an 10cm-level accuracy which should be sufficient for most mobile applications.

Boundary detection performance. Table 2.2 lists the performance of Nightcrawler's boundary detection in terms of the offset in object width. Here we compare three objects of similar size but different materials. Despite the lack of phase information, Nightcrawler already achieves $5cm$ and less error in object width estimation. Later in §2.1.4 our simulation result confirms that when phase information is available, the error in width detection is cut in half. In addition, we also observe that the width accuracy for the metal object is slightly better than those of the plastic and wooden objects. This is mostly because the smoother metal surface enables stronger signal reflection. Finally, we see that the closer the user (RX) is to the object, the more accurate the imaging. This aligns with the Radar Theory in Equation (2.1) as well as the common expectation on imaging — as a user gets closer, she sees the object more clearly.

End-to-end imaging results. By combining the results on position, boundary and surface curvature, Nightcrawler can produce a detailed map of the object surface. Figure 2.3 plots the imaging result of a metal object at different user-to-object distances.

Object Width (Material)	Object-RX distance		
	3.5m	4.8m	6m
24.5cm (Metal)	1.5cm	3.0cm	3.0cm
26cm (Plastic)	4.0cm	5.0cm	4.5cm
22cm (Wood)	4.0cm	4.0cm	4.5cm

Table 2.2: Accuracy of Nightcrawler's boundary detection, in terms of the offset in detected object width.



(a) Imaging setup (b) $D = 3.5m$ (c) $D = 6m$

Figure 2.3: Testbed results: Nightcrawler images a metal object when varying the user-to-object distance D.

The thin blue dash line in Figure 2.3(b)(c) marks the true object shape, while the thick black line is the imaging result of a surface. We see that Nightcrawler can identify the physical surface almost perfectly. Notice that in this example the user's walking path is in parallel with the TX transmitting direction. This is not necessary. In our experiments, the walking direction does not affect the results much as long as the path is relatively straight. It is the user-to-object distance and walking distance that matter the most.

Tracking moving pedestrian. We also evaluate Nightcrawler when the object is a moving pedestrian traveling at 1m/s towards the RX (see Figure 2.4). Here the RX user travels $0.8m$ in total during imaging. In the first $0.4m$, the RX detects a human $2.3m$ away (with a $6.9cm$ offset); in the second $0.4m$, the human is $1.5m$ away and the position offset reduces to $0.27cm$. This preliminary result shows that Nightcrawler can potentially identify and track moving pedestrian using signal reflection.

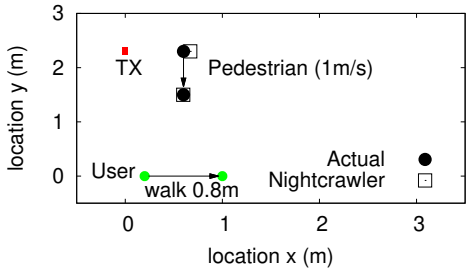
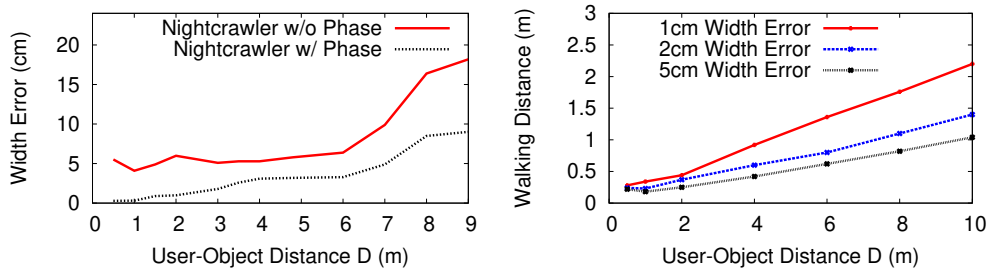


Figure 2.4: Testbed results: Nightcrawler detects and locates a pedestrian user.



(a) Benefits of phase information (b) How Far Should User Walk?

Figure 2.5: Simulated Nightcrawler radar imaging results for a metal object.

Simulation Results

We perform simulations to examine Nightcrawler in absence of testbed artifacts. Our simulation reproduces the scenario in Figure 2.3(a). The metal object surface is represented by dense discrete points and does not introduce any reflection loss. The propagation follows the Friis free-space model for 60GHz transmissions.

Is phase information beneficial? Figure 2.5(a) compares the imaging error on object width with and without phase information. The simulation results without phase information are similar to our testbed measurements. When phase information is available, Nightcrawler's error reduces by 50%. Therefore, a practical implementation of Nightcrawler can benefit significantly from obtaining signal phase information from the underlying 60GHz chipset.

Impact of array elements. Due to cost and sizing limits, mobile 60GHz chipsets are likely to use small number of array elements, *e.g.*, the Wilocity chipset has a 2×8 array, which leads to weaker directivity. We compare Nightcrawler performance using different arrays with 2×8, 6×6 and 10×10 elements, and found that they perform similarly if signal phase information is available.

Impact of object size. We examine a broad range of object sizes between $5cm$ and $1m$, and vary the user walk distance between $0.5m$ and $1m$. Our results, omitted for brevity, show that the absolute imaging error is independent of the object width, as long as the object is not too wide so that its edges fall out of the scope of a single 60GHz beam. To cover these objects, Nightcrawler needs to rotate the TX beam during the measurement process (see §2.1.3).

How far should users walk? Nightcrawler seeks to achieve high-resolution imaging by a user walking a short distance. Figure 2.5(b) plots the required walk distance versus the resulting width error under different user-to-object distances. Since the virtual antenna aperture scales with the walk distance, it is no surprise that the further the user walks, the higher the accuracy is. A practical implementation of Nightcrawler should exploit this tradeoff to achieve robust, efficient and high responsive object imaging. Overall, the result is very encouraging — even when the user is $8m$ away from the object, traveling just $1m$ can achieve $2cm$ imaging accuracy.

2.1.5 Related Work

Sonar and radar systems. Sonar and radar systems are deployed to detect the speed and position of moving targets, or to measure the contour of the terrain [37]. Portable radar devices are available to detect concealed weapons in airports [38]. To provide high-resolution imaging, these systems require either special hardware, *e.g.*, X-Ray or lenses

too large for mobile devices [39]. Different from existing works, Nightcrawler achieves high-resolution imaging using 60GHz networking chipsets that are being integrated into today's mobile devices. While our design is inspired by the SAR method used by airplane radars [23], our key contributions include the novel application of the SAR concept to mobile 60GHz scenarios and the detailed system design and experimentation.

Camera-based systems. Many have developed image-based object recognition systems [40, 41, 42]. These methods, however, cannot accurately measure distance between user and object. Google's Project Tango [43] detects an object's position and shape using three bulky cameras, including an infrared depth camera and a fish-eye lens. Yet it only works in environments with good visibility, and cannot reliably identify object material. Nightcrawler overcomes these challenges by leveraging 60GHz networking chipsets in mobile devices. We show that reflections of 60GHz signals can reveal key physical properties of the object surface even without any light.

RF-based systems. Recent works on WiFi-based systems [44, 12, 45, 46] target coarse-grained human or object tracking, *e.g.*, detecting relative movement of human body, recognizing predefined user gestures [47], or scanning tumors or weapons on human body [48]. Nightcrawler differs from these works by performing detailed imaging on objects, including its shape, surface curvature and material. Nightcrawler chooses 60GHz as the underlying RF technology because compared with WiFi, 60GHz offers much smaller wavelength and much more stable (and predictable) signal propagation. This largely boosts the imaging performance, enabling Nightcrawler to identify, locate and image various objects with high precision.

2.1.6 Summary

We present the initial design of Nightcrawler, a 60GHz imaging radar that locates and images objects in local neighborhood. Our initial evaluation under simple scenarios confirms the feasibility of Nightcrawler in performing high-resolution object imaging. As ongoing work, we seek to improve and further experiment on Nightcrawler. In particular, we consider the following directions.

Handling device positioning errors. Our basic design assumes the RX knows her position to the TX and tracks her position precisely when walking. In practice, any positioning error translates into inaccurate phase shifts during boundary detection (see §2.1.3), and can largely affect imaging performance. Addressing this challenge requires mechanisms for reliable ranging and motion tracking (*e.g.*, [49]) and those for identifying and correcting phase errors.

Identifying curvature details. We take a data-driven approach to extract surface curvature details — collect a large measurement on different surfaces, identify key features and then develop efficient classification algorithms.

Imaging multiple objects. When multiple objects are in range, Nightcrawler can potentially image them simultaneously. Doing so requires the RX to first narrow down a subset of "reflected TX beam directions" that cover all the objects (see §2.1.3). The TX then beams along the subset of directions during the imaging measurement process.

2.2 Object Imaging via Cooperative 60GHz Radios

[10] Mobile computing is evolving. For decades, mobile computing centered around the user and her movements, whether it was on foot, or on vehicles such as buses or cars.

[10]The content in this section is published in [50].

However, the next generation of mobile computing and its challenges will likely be defined in the context of a variety of autonomous mobile agents, including drones, self-driving cars, or semi-autonomous robots. Today, autonomous drones are scanning large crop fields and farm livestock, unmanned helicopters are delivering supplies to soldiers in the field, while water-proof drones patrol the underground sewer system in Barcelona [51]. In the near future, flying drones will deliver our mail, packages and groceries, self-driving cars will drop us off at work, and first responder robots will be first on scene to rescue victims of disasters [28].

A key challenge for the widespread deployment of these autonomous devices is the environmental sensing system, *e.g.*, a mobile imaging radar system that captures the position, shape and surface material of nearby objects. These systems must provide accurate and robust information about the device's surrounding at night or in dark areas (*e.g.*, tunnels), while moving at moderate speeds. Highly accurate results are critical, and errors can produce dire consequences. For example, Google's self-driving cars use maps with inch-level precisions [29], while devices that assist the visually impaired must have errors smaller than 10cm [30, 31]. Finally, to be placed on a variety of autonomous devices, the imaging system should be compact, lightweight and cost-effective.

None of the existing solutions meet these needs. Traditional visible light imaging systems (*e.g.*, cameras) perform poorly in dark or low-light conditions, and lack the precision desired by these applications. Acoustic solutions have been used successfully for ranging over short distances [52, 32], but are easily disrupted by background noise and fail over longer distances. Prior works on RF imaging use WiFi bands to track human motion and activity [12, 44, 45, 48], detect metal objects [48], and map large obstacles [46]. But they require costly specialized hardware or large antennas unsuitable for mobile devices. A recent project reuses WiFi communication devices with multiple antennas to image objects, but its precision is fundamentally limited by WiFi's large

wavelength [53]. Finally, while today's millimeter wave imaging systems can offer accurate object imaging [54, 55, 56, 57], they all require specialized hardware, *e.g.*, large lens radars and FMCW circuits, and do not fit the size or cost constraints of commodity mobile devices.

RF imaging via 60GHz networking radios. One attractive approach is RF imaging radar that reuses commodity 60GHz networking radios to "image" the environment by capturing 60GHz transmissions reflected by nearby objects. Such a high frequency RF radar system has several key advantages over alternatives. *First*, 60GHz links are highly directional, making them relatively immune to interference from environmental factors such as ambient sound or wireless interference. *Second*, 60GHz beams exhibit good reflective properties, and work reliably in a wide range of lighting conditions in both indoor and outdoor locations. *Finally*, 60GHz radios are relatively inexpensive ($< \$40$ [19, 22]), and small enough to be included in today's smartphones and tablets.

The real challenge of building accurate mobile RF imaging is achieving high accuracy within a small device. A simple rule from imaging radar theory [23], defined by Equation (2.1), holds for antenna size (aperture) and the optimal accuracy (radar resolution). For smartphone-sized antennas, even the most high frequency radios (5–120GHz) can produce resolutions no better than 1 meter, clearly inadequate.

Our initial work in this space explored the possibility of using device mobility to emulate a *virtual antenna array* with large aperture (§2.1). This design uses the mobile device as a receiver, with a decoupled transmitter either embedded in the infrastructure or "deployed" on-demand by the user, (*e.g.*, mounted on a nearby drone). By taking measurements of the same reflected signal at multiple locations and applying the *Synthetic Array Radar (SAR)* algorithm [58], the system emulates the signals received by different elements of a large antenna array. Using 60GHz beams is especially advantageous here. Since 60GHz has a carrier wavelength of 5mm (12× shorter than WiFi/cellular),

a user using 60GHz links can obtain fine-grain resolution with just small movements in the measurement area (§2.1).

Practical limitations of SAR. The goal of our work is to design, build, and deploy an accurate mobile imaging system for practical applications. Through experiments on an experimental testbed, we quickly identified *two* fundamental limitations with the SAR approach to imaging radar in real-world mobile settings. *First*, SAR is highly sensitive to receiver trajectory tracking noise. Any deviation from the path produces significant error in the predicted points on the reflection surface. This impact becomes particularly notable when the deviation is greater than λ, the RF wavelength. Whether the receiver is a handheld device, a robot, or a flying drone, its movement is likely to deviate from the targeted straight line trajectory, and deviations are likely much greater than λ for 60GHz links, which is 5mm. *Second*, high resolution imaging via SAR requires knowledge of the beam's phase information (ϕ). But any mm-level error in the receiver's position or trajectory introduces large errors in computing ϕ, and thus using phase information actually adds large errors into the SAR imaging result.

Ensuring accurate positioning and movement tracking to the level of millimeters is difficult using commodity hardware. Thus, accurate mobile 60GHz imaging requires a new imaging approach robust to device positioning and trajectory errors.

60GHz imaging via RSS series analysis. Our observation is that the SAR algorithm is sensitive to positioning errors because each error propagates when computing positions on each object's surface. As a more robust alternative, we propose an approach that identifies the location, overall shape, size and material of the target object, by comparing the measured distribution of 60GHz received signal strength (RSS) values against RSS value distributions predicted from our general surface-reflection model. Since key surface properties (*e.g.*, width and curvature) are strongly correlated with reflected

RSS distributions, we can accurately determine the overall surface shape of the target object. We call this approach *RSS Series Analysis*, or RSA for short. Finally, our work leverages a unique advantage of 60GHz radios – as the receiver moves and (re)aligns its beam, it reports the (strongest) receive beam direction and the corresponding RSS value [33]. Such directional RSS measurements carry ample information of the reflection surface to enable high precision imaging.

RSA is robust against small device positioning and tracking errors because unlike SAR, it does not image an object by locating the individual points on the reflected surface. Instead, RSA focuses on how the collection of these points creates a distribution of RSS values at different observation locations. Such a distribution not only captures the overall shape of the target object, but also tolerates local deviations and errors in device positioning and tracking. Using testbed experiments, we find that an RSA-based 60GHz system can achieve accurate imaging results in the presence of positioning errors as large as 10cm.

In the remainder of this section, we present our 60GHz-based mobile imaging system. We describe techniques for automatically estimating object location, orientation, surface curvature, surface boundaries, and even the surface material of nearby objects. We also present a detailed workflow of a practical implementation of our 60GHz imaging, including techniques for detecting the presence of objects and planning the receiver movement. Finally, we use detailed experimental measurements on a local 60GHz wireless testbed to validate the utility and accuracy of our techniques. Our testbed results on 12 common household objects (of 5–30cm in width) show that our proposed imaging system can image these objects at a high precision (*e.g.*, ∼5cm in object location and surface boundaries) with just small movement (∼1m) by the receiver.

Limitations. Our work provides a first step in the development of high precision RF imaging (re)using 60GHz networking chipsets. Our current design has several limitations;

some fundamental to the choice of 60GHz radios, while others can potentially be reduced via a better design.

First, to stay robust against device positioning/tracking errors, our RSA imaging does not use any phase information from the radio[11]. As a result, we are unable to recognize fine-grained details on an object, *e.g.*, the individual keys on a computer keyboard. Instead we can identify the overall rectangular shape of the keyboard. Similarly we were unable to identify very small objects like keyrings. Improving imaging precision in the presence of device positioning errors is an area of open research. *Second*, the working range of our imaging system is determined by the underlying 60GHz radios and the object surface material[12]. While our solution does not need actual high speed transmissions, accurate RSS measurements require signals to be sufficiently stable. Using an off-the-shelf, low-cost mobile 60GHz chipset (from Wilocity), we found that the imaging range[13] is at least 10m for metal objects, and 5m for cardboard boxes (like those found in Amazon packaging). *Finally*, because 60GHz signals cannot penetrate walls or most objects, our imaging system works when both transmitter and receiver have line-of-sight to the target object. Thus to perform effective imaging, especially in 3D, mobile devices must have a way to intelligently navigate across complex spaces. This is another active research problem.

2.2.1 Mobile 60GHz Radar

Here we set the context for mobile imaging using 60GHz transmissions. First, we begin by identifying key challenges facing mobile imaging systems, and explain why

[11]In some cases, not requiring phase can be an advantage of our system, since many commercial-off-the-shelf (COTS) radios do not report phase but only RSS.

[12]Objects of different materials introduce different degree of signal loss. Metal objects in general introduce no loss while wood objects introduce 12dB loss in signal strength.

[13]Here the imaging range defines the distance between the object and the receiver, assuming the transmitter and receiver are of equal distance to the object.

60GHz radios provide an attractive solution. We then describe initial designs on 60GHz imaging radar using synthetic array radar (SAR) algorithms, and the limitations they face in real deployment settings.

Mobile Imaging Radar and 60GHz

Mobile imaging radar systems face additional technical challenges compared to their traditional counterparts. Traditional imaging radars detect the position and shape of an object by emitting RF signals and analyzing the reflected signal [12, 44, 45, 48, 46, 54, 55, 56, 57]. They typically make use of specialized hardware such as FM circuits and highly directional, large dish antennas, and thus are not suitable for mobile devices. Instead, to be placed on a variety of autonomous devices from smartphones to drones, the imaging system should be severely constrained in size in both the processing hardware and the antenna, which severely limits the maximum imaging resolution (see Equation 2.1). For smartphone-sized antennas ($2.5cm$ aperture), maximum imaging resolution for an object of $10m$ away is $1m$ using 120GHz transmissions or $24m$ at 5GHz. Furthermore, mobile radar systems target commodity devices, which rules out costly FM pulse circuits. Similarly, cost constraints prevent the use of fine accuracy positioning devices, or dispersion analysis for material detection (specialized transmitters).

Instead, mobile imaging radar can (re)use existing wireless networking chipsets on mobile devices, but leverage human or device mobility to greatly extend antenna aperture. This can provide resolution better than the limit defined by Equation (2.1). Next, we describe key components of such a system.

Leveraging 60GHz radios. Today's mobile devices are equipped with multiple wireless interfaces, *e.g.*, cellular, WiFi, Bluetooth, and 60GHz radio[14]. Among them,

[14]Qualcomm is producing low-cost 60GHz chipsets at or below previous prices of $37.5, with a range of $23m$ or more [19, 22]. HP recently released a laptop equipped with the Intel 60GHz chipset [59].

60GHz is ideal for mobile imaging for three reasons.

- Carrier wavelength of 60GHz is 5mm, over 12x shorter than WiFi/cellular. This translates into 12x smaller required antenna aperture than WiFi/cellular under the same imaging resolution.
- 60GHz's short wavelength leads to more predictable propagation, i.e. minimal multi-path effects and signal strength is strongly correlated to propagation distance. The system can easily detect the presence of objects by distinguishing between line-of-sight (LoS) and reflected signals.
- The object reflection profile is more stable at 60GHz. Since reflection loss is strongly correlated to object material [34], the radar system can determine the material type of the reflection surface using signal strength measurements.

Emulating virtual antenna arrays with mobility. A mobile device can emulate a large aperture virtual antenna array by moving and taking signal measurements at different positions along its trajectory[15]. This allows a small mobile device to produce high-resolution imaging results despite its small aperture antenna. For example, a device can take signal measurements along a 1 meter trajectory and achieve an (optimal) resolution at 60GHz of 15mm, from a distance of 3 meters away. Finally, user mobility also increases the system's ability to detect surface curvature of objects, as reflected signals at different locations help capture the curvature of each of the object's multiple faces.

Decoupling transmitter and receiver. Given the small size of mobile devices, the power of a radar system is limited. Under the limited power, decoupling the transmitter and receiver, a.k.a *bistatic radar system*, can significantly improve radar range over a single transceiver (monostatic) [47]. We consider a mobile radar system including the

[15]Aperture of a virtual antenna array is equal to the distance traveled by the device.

primary mobile device acting as a receiver and a decoupled transmitter. For example, a system to assist the visually impaired may include an app on the user's smartphone, and transmitters embedded in the walls or ceiling. The transmitter/receivers duties can also be split across multiple mobile devices, *e.g.*, multiple drones scanning underground tunnels.

The transmitter (TX) sends 60GHz beacons that reflect off of nearby objects. Each beacon includes the angle of transmission, and if possible the transmitter's relative location to the receiver. Each RX moves and periodically scans and records signal strengths for beacons, and processes these data on the fly to identify, locate and image nearby objects.

A Synthetic Array Radar (SAR) System

Our earlier work proposed a 60GHz imaging system (§2.1), where the receiver estimates object location and surface boundary using the Synthetic Array Radar (SAR) algorithm [58]. Applying SAR on measurements along a trajectory emulates the process where a large array focuses its narrow beam on different points of the object surface. Controlled testbed measurements achieve centimeter level accuracy in detecting object location and surface boundaries.

The SAR algorithm. The imaging process is driven by the traditional SAR algorithm for bistatic radar [60]. TX transmits a simple sine wave, which is reflected by the object towards RX. RX measures the reflected signal at different locations as it moves. To understand SAR, consider a simple case where the object is a point. Let N represent the number of signal measurements taken by RX. The complex signal $r_i(t)$ measured at RX location i is $r_i = A_i e^{-j\phi_i}$ where A_i is the product of the transmit and receive antenna field radiation pattern and total propagation/reflection loss at i, and ϕ_i is the change in phase. Assuming signal reflection does not introduce any phase change, $\phi_i = \frac{2\pi}{\lambda} d_i$ where d_i is

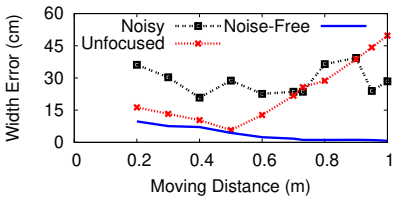
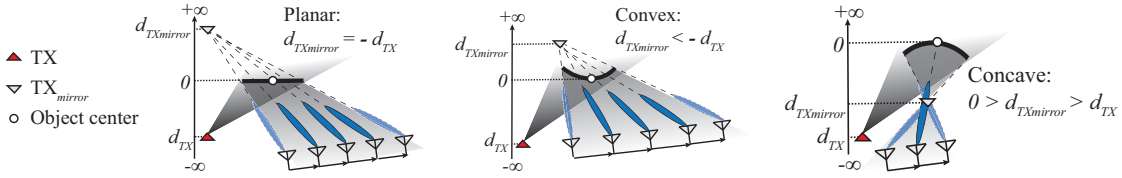


Figure 2.6: Experimental results demonstrate the limitations of SAR.



(a) Planar surface (b) Convex surface (c) Concave surface

Figure 2.7: An abstract view of the 60GHz signal reflection and RX's signal measurements as it moves.

the total propagation distance. SAR computes the relative power $\mathcal{P}(p)$ at any point p in space using the RSS $|r_i|$ and phase shift ϕ_i at different locations: $\mathcal{P}(p) = \left|\sum_i r_i e^{j\frac{2\pi}{\lambda}\hat{d}_i}\right|$, where $\hat{d}_i$ is the distance from TX to each RX location i through the point p. If this point is a point on the object surface, i.e. $\hat{d}_i = d_i$, then the summation is constructive and $\mathcal{P}(p)$ is large. Otherwise because of destructive interference, the relative power becomes small. Thus SAR determines the object location and shape by searching for the strongest $\mathcal{P}(p)$ values across space.

Limitations of SAR

Our initial design (§2.1) makes two "idealistic" assumptions on device positioning: (1) TX and RX have perfect knowledge of their relative position; (2) RX moves in a perfect trajectory *e.g.*, a straight line. However, in practice these two assumptions do not hold, and the imaging performance degrades significantly.

Limitation 1: Sensitivity to trajectory noise. It is well-known that SAR is highly sensitive to trajectory noise – when moving, RX often deviates from the targeted

path, and its trajectory cannot be tracked accurately. Such noise translates into errors in computing $\hat{d}_i$ and thus affects $\mathcal{P}(p)$. The impact becomes highly visible when the error is comparable to or larger than RF wavelength λ. For 60GHz, $\lambda = 5$mm. Thus even a few millimeters deviation in trajectory can largely affect the imaging result.

We perform experiments to examine this artifact. Figure 2.6 plots the imaging performance in terms of the error in derived surface boundary, for different object-to-RX distances. We compare two systems: "noise-free SAR" where the RX moves in a straight line and "noisy SAR" where we introduce random deviations (up to 5mm) to the actual RX trajectory. We see that in the presence of noise, the imaging error magnifies by at least 4 folds to 40cm. We also observe that errors in TX-RX positioning have similar effect (results omitted due to space limits). SAR cannot tolerate such small errors, let alone the 10cm error typically seen from the trajectory of mobile devices like drones.

To address this problem, one may consider using motion sensors to record the trajectory precisely. But commercial sensors cannot achieve millimeter-level accuracy. For example, accelerometer reports only the acceleration of device, and the translation information can only be obtained by integrating the result twice, resulting in poor performance [61]. GPS is known to have meter-level errors. Another approach by traditional SAR is to estimate the movement noise [62]. This can be effective for aircraft radars because the movement noise comes from air turbulence and can be approximated to the level of their operating wavelength (more than 10m). But for our targeted 60GHz mobile scenarios, the movement noise is much more random and harder to predict at the millimeter level.

Limitation 2: Dependency on phase information. To achieve high resolution, SAR requires the knowledge of the phase information ϕ_i. However, since the positioning/trajectory errors will corrupt the phase transition process, using the phase information actually introduces large errors in imaging. An alternative solution is to use

"unfocused" SAR which assumes $\{\phi_i\}_{i=1}^{N}$ are all identical, *i.e.* $\phi_i = 0$, and only uses RSS to compute $\mathcal{P}(p)$ [27]. This reduces the impact of trajectory errors, but sacrifices imaging resolution: the longer the receiver trajectory, the more the "uniform phase approximation" error amplifies and degrades imaging accuracy. Figure 2.6 shows that unfocused SAR performs slightly better than noisy SAR but far worse than noise-free SAR. Note that while (noise-free) SAR always benefits from longer moving distance (larger antenna aperture), unfocused SAR is highly sensitive to this parameter. After the trajectory distance exceeds some threshold, compounded error from the "uniform phase approximation" overcomes the gain of larger apertures, and imaging performance deteriorates quickly. This threshold is object dependent and hard to identify *a priori*[16], making the performance of unfocused SAR unpredictable. Similarly, existing work reported that unfocused SAR can be 10 times worse than SAR [63].

Summary. These results highlight the fact that SAR-based systems are highly sensitive to device positioning errors. Because of 60GHz's small wavelength (5mm), even small deviations in position translate into large distortions in phase transition results, and significant errors in imaging quality. Given these fundamental limitations, we must explore SAR alternatives to achieve the high accuracy demanded by next generation autonomous devices.

2.2.2 RSS Series Analysis (RSA)

Our proposal to address these limitations is *RSS Series Analysis* (RSA), a new 60GHz imaging algorithm. Unlike SAR, RSA images an object using *only* RSS measurements recorded along the receiver's trajectory. We summarize RSA here and present the detailed algorithm in §2.2.3. RSA offers two advantages over prior work on RF imaging [53, 64, 22]:

[16]Our measurements show that the threshold scales linearly with the object width and RX-object distance, thus hard to identify *a priori*.

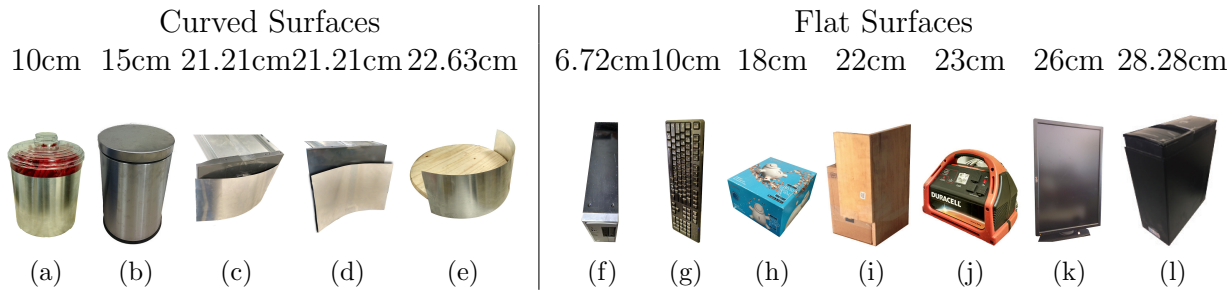


Figure 2.8: Objects used in our experiments. The number on top of each object is the width of the object. The left five objects (a)-(e) have curved surfaces and the right seven objects (f)-(l) have flat surfaces.

- RSA can discover a rich set of object surface properties at high resolution (cm level). These include object surface location, orientation, curvature, boundary and material.
- RSA is highly robust against device positioning and trajectory tracking noise. Testbed results show that it can tolerate deviations as large as 10cm without degrading imaging quality.

Core Concept

RSA achieves high-precision imaging by combining receiver mobility with the high directionality of 60GHz beamforming. Specifically, RSA treats each object surface as a continuous medium that reflects a directional 60GHz signal towards the directional receiver RX. As RX moves and continually (re)aligns its beam to maximize received signal strength, the measured RSS value and its receive beam direction (angle of arrival (AoA)) carry information of the object surface. By analyzing these *directional* RSS measurements across multiple RX locations, RSA recovers important properties of the object surface, including position, curvature, boundary and material. At a high level, RSA works in 3 sequential steps.

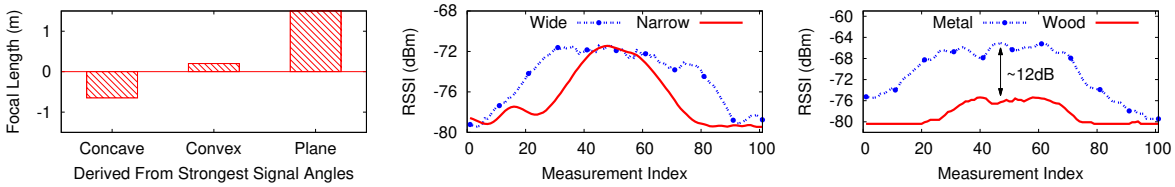
1. Surface curvature & center position. Consider a scenario in Figure 2.7(a) where TX points towards and reflects its beam off a flat object surface. As it moves, the directional receiver RX maximizes RSS by pointing the receive beam towards the mirror point of TX respect to the object surface, *i.e.* TX_{mirror}. This is a hypothetical point that would have originated the signals if there was no reflection, which can be computed as the intersection of AoAs, *i.e.* the strongest RSS direction, for different points on the RX trajectory. While in practice the AoA reported by RX might deviate slightly due to non-ideal antenna patterns, imperfect reflection and measurement artifacts, one can still locate TX_{mirror} by intersecting the series of (noisy) AoAs collected as RX moves.

Now consider the scenario where the object surface is curved, either convex (Figure 2.7(b)) or concave (Figure 2.7(c)). We can still locate TX_{mirror} by intersecting the reported AoAs. Following the mirror and lens equation [65], a surface's curvature type is determined by its focal length f:

$$\frac{1}{f} = \frac{1}{d_{TX}} + \frac{1}{d_{TX_{mirror}}} \tag{2.2}$$

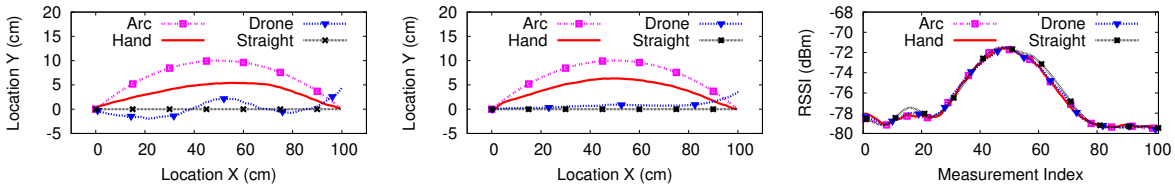
where d_{TX} and $d_{TX_{mirror}}$ are defined in Figure 2.7. Both values are under sign convention, *i.e.* positive if behind the object, and negative when in front of the object. The surface is convex if $f > 0$, concave if $f < 0$, plane if $f \to \infty$, and $|f|$ is half of the curvature radius. Therefore, we can identify surface curvature by computing d_{TX} and $d_{TX_{mirror}}$. This requires information of the position and surface orientation of the object center, which can be estimated by intersecting the TX center beam direction with the reported AoAs.

2. Surface boundary. Once curvature is determined, RSA detects surface boundary by exploiting the unique effect of 60GHz directionality on signal reflection. When RX is within the area of "object coverage area" (Figure 2.7(a)), the corresponding RSS is



(a) Plane vs Convex (b) Narrow vs Wide (c) Metal vs Wood

Figure 2.9: The observed RSS series are strongly correlated with the object surface properties.



(a) Noisy trajectory (top view) (b) Noisy trajectory (front view) (c) Stable RSS pattern for 6.7cm wide object

Figure 2.10: The measured RSS series remains stable across all four (noisy) 3D trajectories.

strong because RX can align its beam to capture the (strong) reflected signals. But when RX moves outside of this area, the quality of its beam alignment (and RSS) degrades quickly. Thus shape of observed RSS values across different RX locations is strongly correlated to the object surface boundary. Using the estimated surface curvature, center location and orientation, we can model this correlation to enable reliable detection of surface boundary.

3. Material. When a signal hits a surface, parts of it may be "absorbed," leading to a *reflection loss*. At 60GHz, this reflection loss has a strong correlation of the surface material and the incident angle [34]. In particular, the RSS of a reflected signal is the RSS of a LoS signal (of the same propagation distance) minus the reflection loss (all in dB). Once we know surface location and curvature, we can derive the reflection loss and incident angle, and thus identify the likely surface material(s).

Quantifying Correlation via Measurements

Our intuition is that RSS measurements along a trajectory are highly correlated to a number of properties of a reflection surface. We use a commodity 60GHz radio testbed (details in §2.2.5) to better understand these correlations. We experimented with twelve objects (listed in Figure 2.8) of different width (5cm–30cm), curvature, material (wood, metal, plastic) and surface roughness (smooth vs. rough)[17]. We varied the TX and RX locations to examine the impact of object placement.

We experimented with four movement patterns involving a 1-meter straight line trajectory in space: a drone flying[18], a user moving a mobile phone over a line, an arc, and a perfect straight line. All four trajectories use the same start and end positions, and take the same amount of time to finish. RSS measurements are taken every 1cm, leading to a total of $N = 100$ measurements per trajectory. Each experiment collects $\{RSS_i, AoA_i\}_{i=1}^{N}$, where RSS_i is the strongest RSS value as RX rotates its beam at location i and AoA_i is the corresponding receive beam direction.

Our experiments led to two key observations.

1. Strong correlation with object surface properties. Our experiments confirm a strong correlation between RSS measurements and object surface properties. We show in Figure 2.9 that the RSS patterns, either as RSS values or AoAs, can be used to distinguish objects of different surface curvature, surface boundary (i.e. width), and material. The groundtruth of focal length in Figure 2.9(a) is 0.6m, 0.6m, and infinity, respectively.

2. Robustness against trajectory noise. The RSS series (both RSS and AoA) are

[17] A surface is considered smooth if $h < \frac{\lambda}{8cos\theta}$ and rough if $h > \frac{\lambda}{8cos\theta}$ [66]. Here h is the min to max surface protuberance, $\lambda = 5mm$ and θ is the incident angle. For our objects, the plastic keyboard is "rough" and a monitor surface is "smooth."

[18] Using a high-end IRIS+ drone by 3D Robotics Inc., we captured its movement trajectory when configured to fly straight.

highly robust against trajectory noise. Figure 2.10 illustrates different views of the four trajectories, and the RSS values along the trajectories when imaging an object of 6.7cm wide. While the trajectories deviate from each other by as far as 10cm, their spatial RSS patterns align well. We experimented with other movement patterns and objects, and arrived at similar conclusions. A closer look shows that RSS values correlate most strongly with propagation distance d. But in practice, d is at least multiple meters, and trajectory errors are in centimeters. Thus trajectory errors have little impact on RSS.

While these results may not be representative, they validate our intuition that much about properties of the reflection surface can be found in RSS measurements along the movement trajectory. Next we present techniques to extract these properties from RSS data.

2.2.3 RSA Imaging Algorithm

Our RSA algorithm provides highly accurate imaging results on distance, curvature, boundary, and material detection, all while tolerating positioning and trajectory errors. It takes three inputs: a sequence of RSS measurements in RSS and Angle of Arrival tuples $\{RSS_i, AoA_i\}_{i=1}^{N}$, RX's trajectory (*i.e.* RX location i) and its relative position to TX, and TX's transmit beam direction and pattern. We will discuss in §2.2.4 the procedure to obtain these inputs and the sensing process for TX to focus its beam on the object.

Imaging an object takes four processing steps on the RSS data. We estimate location and orientation of the object center, then compute surface curvature, and boundaries, and finally identify a set of potential surface materials. We first describe these key components to image a single surface, and then the process to image multiple surfaces/objects. Finally, we describe how RSA mitigates noise from device localization, interference and

RSS measurements.

Estimating Object Center & Orientation

RSA starts by computing an initial estimate of the location and surface orientation of the object center, since it is input for subsequent steps. The intuition is simple: when TX's beam covers the object evenly and TX/RX are perfectly aligned, we can locate the object surface center at the intersection of the TX beam direction and each AoA. While the TX/RX alignment is imperfect (since TX fixes its beam), the intersection with each AoA_i is still a good approximation of the reflection surface.

Like [27], RSA estimates the object center by performing a "majority vote" on the set of intersection points. Given K ($K < N$) intersection points, RX identifies a cluster of $\lfloor \frac{K}{2} \rfloor + 1$ points with the minimum mean square error (MSE) among themselves. It approximates the object center as the center of the cluster, *i.e.* the position with minimum MSE to all other points in the cluster. To generate the K intersections, RSA picks a subset of AoAs from $\{AoA_i\}_i^N$ whose RSS_i is among the strongest (and above the noise level) and intersects them with the center direction of TX beam.

Since the incident and reflected angles are equal, we can compute the (candidate) direction of the object surface's principal axis with respect to each of the K AoAs. We derive the object center's orientation by computing the principal axis using majority vote over K candidates, then computing its perpendicular direction.

A key difference from [27] is that RSA iterates to improve its estimate of object center and orientation, using as input the curvature and boundary results from later steps. This helps to mitigate the impact of TX/RX positioning errors and other artifacts (§2.2.3).

Characterizing Surface Curvature

After object position comes surface curvature. We characterize an object's surface curvature based on the mirror and lens equation defined by Equation (2.2). We first compute the "TX mirror point" and then compute the focal length f from d_{TX} and $d_{TX_{mirror}}$. We compute it as the intersection of angle of arrivals for different points on the RX trajectory (Figure 2.7). To mitigate noise/artifacts in AoA measurements, RSA first smoothens the AoAs using a moving window, *e.g.*, of size 3 in our current design, then performs a majority vote on pair-wise AoA intersections to derive the mirror point.

Given the estimate of object center point and orientation, we calculate d_{TX} and $d_{TX_{mirror}}$ by projecting the TX and the TX mirror point to the principal axis (see Figure 2.7). If we set the object center as position 0, d_{TX} is negative and $d_{TX_{mirror}}$ is positive. RX then computes f based on Equation (2.2). The surface is convex if $f > 0$, and concave if $f < 0$, and $2f$ is the curvature radius. In theory, a flat/planar surface should have $f = \infty$. Yet in practical scenarios, $f >$1 meter is sufficient to identify most objects with a flat surface.

Computing Surface Boundary

The next step is to compute the surface boundary, *i.e.*, the width of the surface if the object was projected to the plane of RX's movement trajectory. We exploit the strong correlation between the RSS sequence $\{RSS_i\}_{i=1}^{N}$ and the object surface, and propose a simple RSS model for surface reflection. After adding surface curvature and center location as parameters, this model generates a direct one-to-one mapping between a specific surface boundary and the sequence of RSS values captured by RX. Thus we can estimate the surface boundary by searching for a surface profile whose model-predicted RSS sequence matches those observed by RX.

A RSS model for surface reflection. We develop a new surface reflection model, which takes into account the reflection property of a "fixed-size" reflection surface. Consider the TX transmission towards the object as a collection of sharp rays, each reaching a point p on the object is reflected towards RX. Here we consider a general scattering reflection scenario where the point p uniformly scatters signals in space according to the Lambertian reflection model [67, 68]. We also consider a far-field scenario where the propagation distance is much larger than wavelength (*i.e.* > 100 times larger). In our case, the wavelength of 60GHz is $5mm$, and the overall propagation distance in our system should be at least $0.5m$. Using the complex baseband representation under far-field approximation, the 60GHz received signal at RX location i is

$$r_i(t) = \int_{\mathbb{P}} \underbrace{\frac{\lambda\sqrt{G_p(i)}e^{-j\frac{2\pi}{\lambda}d_p(i)}}{4\pi d_p(i)}}_{\text{overall propagation}} \underbrace{\Gamma_p(i)e^{-j\phi_p(i)}}_{\text{reflection}} u(t)dp \tag{2.3}$$

where for the p^{th} reflected path arriving at location i, $G_p(i)$ is the product of the corresponding transmit and receive antenna field radiation pattern, $d_p(i)$ is the total propagation length, $\Gamma_p(i)$ is the amplitude reflection coefficient, $\phi_p(i)$ is the corresponding change in phase, $u(t)$ is the complex baseband transmitted signal, and $\mathbb{P}$ represents the object surface in 3D space. The key to this model is the constraint of the fixed size reflection surface, captured by the integral over $\mathbb{P}$.

Because our design targets the overall shape of the object, we simplify Equation (2.3) by assuming the surface is relatively smooth, *i.e.* ignoring the fine-grained details. Therefore we consider a uniform reflection pattern: *i.e.* $\Gamma_p(i) = \Gamma, \phi_p(i) = \phi, \forall p \in \mathbb{P}, i = 1..N$. Then we can pull out the $\Gamma_p(i)$ term, and derive RSS as:

$$RSS_i = P_t \cdot \Gamma^2 \left| \int_{\mathbb{P}} R_p(i)dp \right|^2 \tag{2.4}$$

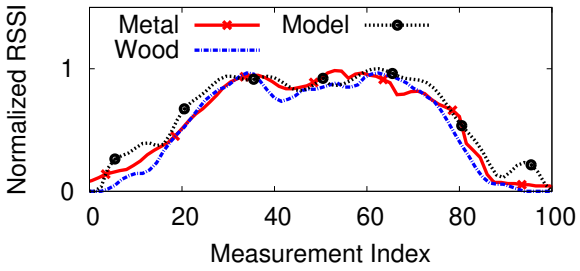


Figure 2.11: Comparing measured and predicted RSS patterns.

where $R_p(i) = \frac{\lambda\sqrt{G_p(i)}e^{-j\frac{2\pi}{\lambda}d_p(i)}}{4\pi d_p(i)}$. Given the object curvature and center location, and locations of TX and RX(i), we can calculate $R_p(i)$. Given the object surface boundary or width, we can construct $\mathbb{P}$ and then derive RSS_i[19]. To remove the contribution of Γ which is unknown, we can normalize RSS_i across i.

We verified this model using testbed experiments on objects in Figure 2.8. Example results in Figure 2.11 show that normalization effectively separates the contribution of materials from that of the surface boundary, *i.e.* two objects of the same width but different materials have the same normalized RSS series. The measured RSS series closely matches the series predicted by our model.

Fitting measurements to model. We determine the surface boundary by matching the observed RSS values to a range of RSS series produced by the model. In this process, we consider a range of possible surface width values. For each candidate width w, we construct the physical surface $\mathbb{P}$, use the model to predict the (normalized) RSS series, and compare it with our (normalized) measured RSS series. To compute "similarity" between two series (or curves), we experimented with multiple metrics, including MSE, MSE of the derivatives, and MSE of the dynamic time warping algorithm [69]. Among

[19]For efficiency, we approximate $\mathbb{P}$ as a collection of points whose interspacing $\leq$ 5mm.

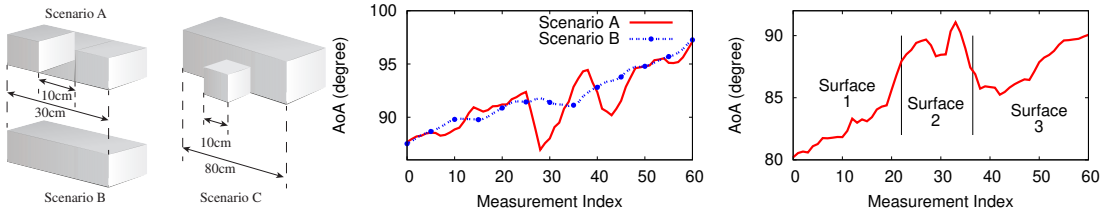
these, MSE of the derivatives is the best:

$$\eta = 1/\sum_{i=1}^{N} \left(\Delta_i^{model} - \Delta_i^{real}\right)^2 \tag{2.5}$$

where Δ_i^{real} and Δ_i^{model} are the derivatives of the normalized RSS at i using the measured values and the modeled values, respectively. This metric works well because computing surface width means detecting the two edges, which lead to fast RSS degradation at the corresponding RX locations. The RSS derivatives effectively capture such RSS variation. We leave the task of finding the optimal metric to future work.

Minimizing search space. We can significantly reduce the search space for the surface boundary size, by looking at only widths that can exist within the triangle formed by the start and end points of the RX trajectory, and the TX mirror point (see Figure 2.7(a)). RX can detect if the width w is large enough for the triangle to bound the reflection surface (steep dropoff in RSS before and after the surface boundaries). For very large surfaces, RX might need to extend its trajectory to detect the surface boundary. Assuming the surface is bounded by the triangle, we can estimate the maximum value of w using geometry. We then search for the true value of w starting from the max down to $1cm$ in the unit of $0.1cm$. These settings are sufficient for the target imaging precision. Going for a higher granularity adds extra computation complexity but little improvement of imaging quality. Currently our search takes less than $3s$ for all our twelve test objects using a matlab implementation on a standard MacbookPro. As future work, we can further prune the search space using sophisticated methods such as cutting planes.

Measuring curved surfaces. For curved surfaces, the reflected RSS series display a different pattern: a convex surface will *scatter* signals to a wider area while a concave will *gather* signals towards a smaller area (Figure 2.7). Thus using the above method, we will likely image a narrow, convex object as a wide object.



(a) Three Surface Scenarios (b) AoA Patterns for Scenario A and B (c) AoA Patterns for Scenario C

Figure 2.12: Detecting and imaging multiple objects. (a) The three scenarios considered: two surfaces separated by a gap, a single continuous surface, and one small surface in front of a big one. (b) The AoA pattern changes abruptly when two surfaces are separated. (c) The AoA pattern displays three segments when a small surface is in front of a big one.

To address this, we apply a slightly different algorithm. Upon determining that the object surface is non flat, RSA computes the surface boundary using $\{R\hat{SS}(\theta)_i\}_{i=1}^{N}$, *i.e.* the RSS measured at a *fixed* receive beam direction θ across all positions along the RX trajectory. Here $\theta = AoA_j$, $j = \text{argmax}_{i=1..N} RSS_i$, i.e. the AoA of the strongest RSS across all the RX locations. Intuitively, this direction θ is parallel to the surface's principal axis, thus $R\hat{SS}(\theta)_i$ includes less contribution of surface scattering (or gathering) but more impact of surface boundary. This way, we can apply the same surface reflection model by using the curvature detection result to construct $\mathbb{P}$. In §2.2.5, we show that this method is accurate and also robust against errors in the estimated curvature radius.

Identifying Potential Surface Materials

Finally, we seek to estimate the surface material based on the reflection loss Γ^2. Existing measurement studies on 60GHz propagation and reflection have built a table of Γ^2 values as a function of the surface material and the angle of incident [34]. We can estimate the angle of incidence given the estimated surface orientation and curvature. From Equation (2.4), $\Gamma^2 = RSS_i^{real} / \left(P_t \left(\frac{\lambda}{4\pi}\right)^2 \left| \int_{\mathbb{P}} R_p(i) dp \right|^2 \right)$. With Γ^2 and the angle of incidence, we can *narrow down* the material type using the reflection loss table. For

example, we can distinguish metal objects (0.3dB loss) from wood (12dB loss) or plastic objects (8dB loss).

To obtain a reliable estimate of Γ^2 for flat surfaces, we select a group of the strongest RSS measurement locations, and calculate the Γ^2 as above for each location. We then compute Γ^2 as their average. This helps to mitigate noise contributed by reflection artifacts near object boundaries.

Estimating Γ^2 for curved surfaces is more challenging because as signals scatter or gather, the above calculation becomes less reliable. Our current solution is to introduce a compensation factor that approximates the impact of signal scattering or gathering. Specifically, we input the already derived object curvature and width into the RSS model, use it to generate the RSS series (ignoring Γ^2), and record the maximum RSS value, γ_{curve}. We then input the width into the RSS model but treat the surface as flat, generate the RSS series and record the max, γ_{flat}. The final reflection loss estimate is $\Gamma^2 \cdot \gamma_{flat}/\gamma_{curve}$ where $\gamma_{flat}/\gamma_{curve}$ reflects the impact of signal scattering/gathering.

Imaging Multiple Surfaces/Objects

So far our discussion targets scenarios with a single object surface. We now discuss the feasibility of RSA for detecting and imaging multiple objects/surfaces. We consider two representative scenarios: (1) two nearby objects separated by some space; and (2) a smaller object in front of a larger one (see Figure 2.12(a)).

Intuitively, a key difference between single and multiple surface reflection should be the reflection angle, *i.e.* the AoAs. Using testbed measurements, we verified that for the above two multi-surface scenarios the AoA pattern is significantly different from a single surface of the same width. For example, Figure 2.12(b) plots the reported AoAs as a function of the measurement location (1–60) over a 60cm RX trajectory. For an even surface of 30cm in width, the AoA pattern grows smoothly as the RX moves. But when

this object is replaced by two 10cm objects separated by 10cm (same overall width), the AoA pattern changes abruptly (by 3–5°). Similarly, Figure 2.12(c) plots the AoA pattern when one 10cm-wide object is placed in front of a 80cm-wide one. It is segmented into three regions, corresponding to the uncovered portions of the larger object surface on each side and the small object in the middle.

These significant changes in AoA patterns suggest that multiple objects can be detected using AoA derivatives along the trajectory. For all object scenarios we tested[20], a threshold of 3° can reliably detect and extract multiple objects. We can then apply the single-object based RSA to each segment to image individual surfaces of moderate sizes. The key limitation here is that our imaging system lacks the precision to image small objects and fine-grained surface details, *e.g.*, individual keys on a computer keyboard. Instead, it should detect and image an Amazon package on the floor.

Above results also suggest that RSA can handle interference due to reflection from other objects. We treat the 80cm-width surface in scenario C as the background object, where our system can still identify and image the 10cm-wide object (surface 2) in the middle.

Handling Noise & Interference

We design RSA to stay robust to three types of noise or errors: *positioning error* for locating TX and RX, *trajectory noise* when RX's trajectory deviates from the ideal line, and *RSS measurement noise* caused by RF interference or background reflection.

TX/RX positioning errors. Errors in TX/RX positions can propagate to errors in locating object center point and TX mirror point. We address this by exploiting the fact that the measured RSS series is stable and strongly correlated with the object

[20]In our scenarios, when two objects are separated, the gap $\geq$ 10cm. The ability to detect 10cm gaps between objects is sufficient for most mobile applications like drones.

surface. After one round of imaging, RSA introduces *controlled perturbations* to explore possible small shifts in center location and surface boundary values that lead to a better match between the model predicted RSS profile and measured RSS data. Specifically, it shifts the TX/RX locations by up to $10cm$ and repeats the imaging process. This iterative search stops when the similarity metric (defined by Equation (2.5)) exceeds some threshold, or the boundary results of two consecutive iterations differ by 1cm or less, *e.g.*, convergence.

RX trajectory errors. The basic imaging algorithm assumes precise data on RX's trajectory. In practice, the movement itself is noisy — a high-end drone configured to fly a straight line can deviate by 4cm. Motion-tracking (via accelerometers or other sensors) can easily generate 10cm errors in less than a second [70, 61].

Trajectory errors can translate into errors in locating the object center and TX mirror point, and errors in RSS model. For the former, RSA denoises by applying "majority vote" across multiple RX AoA measurements (§2.2.3). For the latter, we found the impact on imaging quality to be minimal, and both RSS model and measurements are insensitive to trajectory errors $< 10cm$. In our scenarios, we configure RX to move in a line and rely on external trajectory control to keep the trajectory error less than $10cm$. In practice, any errors that do propagate will add to noise in estimates of object center and TX mirror point. These will be addressed together with any resulting noise from TX/RX positioning errors (see above).

Interference. Background reflection from other objects can be handled via the multi-surface detection and imaging process described in §2.2.3. The bigger challenge comes from possible corruption of RSS measurement values by RF interference from other 60GHz transmissions, *e.g.*, strong signals from a LoS transmitter to RX will distort the AoA values.

If angular separation between the interfering signal and reflected TX signals is sufficiently large, RSA can eliminate interference using 60GHz directionality. As RX scans across directions, it detects and decodes signals from different sources and only uses those from TX to construct the RSS series. In rare cases where the signals are closely aligned, the interference will likely affect data transmission between TX and RX during imaging, *e.g.*, high RSS but recurring packet errors. When this is detected, TX and RX can switch to another 60GHz channel or change physical location.

2.2.4 Implementation

We now present the detailed workflow of a practical implementation of RSA imaging. First, TX and RX determine each other's position. They scan for any objects, and once found, TX focuses its beam on the object and computes the RX movement direction and distance. RX moves, collects RSS measurements and images the object. The process does not require tight synchronization between TX and RX, only that TX signals remain consistent during imaging, *e.g.*, a simple sine wave, so that RSS is stable over time. In particular, for 802.11ad [33], our sine wave based design can directly use single carrier (SC) to send consistent 0s or 1s and generate a regular sine wave, or use one of the OFDM subcarriers for imaging.

TX/RX positioning. To determine each other's location, TX and RX can exchange their locations (if known), or apply existing mobile localization/ranging techniques based on RF or acoustic signals [71, 64, 52, 32]. We can also apply 60GHz localization in addition to improve localization accuracy to centimeter-level (if TX and RX have line of sight).

60GHz localization leverages the 802.11ad bootstrapping procedure and includes two steps. First, TX (in directional mode) steers its beam in different directions and embeds

its beam direction in the signal. RX (in omni-directional mode) receives signals over time and identifies the strongest signal strength r and TX beam direction α. If a LoS path exists between them, then RX can compute its distance to TX d from r (using the 60GHz Friis propagation model [72]). To detect whether LoS exists, TX compares α and r with those estimated by the external localization technique. If the discrepancy is large, especially if d is larger, the path is reflected. Otherwise, LoS exists and TX locates RX via α and d. Next, TX transmits in α direction. RX enters directional mode and scans for the strongest signal. RX can locate TX using the strongest receive direction at RX and d.

Object sensing & RX movement planning. TX and RX use the above two steps of 60GHz localization to sense nearby objects and compute the appropriate RX trajectory. There are two modifications from the sequence above. In step one, instead of reporting only the strongest RSS, RX reports a list of TX beam directions where the RSS exceeds the noise level. After pruning the list by removing the LoS directions, the remaining represent reflected signals. From these directions, RX identifies a set of TX beam directions $\{\alpha^k\}_{k=1}^{T}$ that TX should focus on based on their beam radiation patterns and steering granularity. If an object is too wide to be covered by a single TX beam, RSA can image the object by having TX steering sequentially in multiple segments and stitching the image results, or by TX modifying its antenna radiation pattern to form a wider beam (if possible).

In the second step, TX slowly steers its beam in each of these directions while RX measures AoA for each α^k direction. Ideally, for each α^k, RX should move perpendicularly to the corresponding AoA to detect object width. Furthermore, RSA uses the intersection of α^k and AoA to approximate the object location and thus the total propagation distance d_p. The projected RX movement distance is then the width of the TX beam pattern at distance d_p, which is sufficient to discover the object surface shape. Together the

recommended path and distance allow RX to create a virtual antenna array large enough to discover the object's surface while minimizing the travel distance.

Object imaging. TX focuses its transmission on each specific direction (while embedding the beam direction in its signal). As RX moves, it collects RSS measurements using the 802.11ad *antenna alignment* procedure. Collecting the (RSS, AoA) tuple across multiple directions (§2.2.3 and §2.2.3) does not require extra measurements, and is done by modifying 802.11ad to report additional data. Since for phased array, full-scope beam steering takes less than 1ms[21], RX can perform real-time measurements as it moves, even when TX rotates its beam across multiple directions to image multiple objects. Finally, RX analyzes the data to image the object(s).

Latency. We expect the imaging delay is dominated by those of RX movement and RSA data analysis. For latter, our current implementation finishes in less than 3s and can easily be further optimized, *e.g.*, using convex optimization during iterative search (§2.2.3).

2.2.5 Evaluation

We evaluate RSA in practical settings using off-the-shelf 60GHz radios. We study its utility and imaging quality in the presence of device localization and trajectory errors and background reflections. We also examine its error tolerance, and its sensitivity to different system/hardware configurations. Finally, we compare RSA with SAR and unfocused SAR, and perform a multi-object case study by emulating drones locating a target object using RSA.

[21]Phased array beam steering delay is as low as 50ns [73]. Scanning 360° in the steps of 1° takes 18μs.

Testbed and Experimental Setup

We consider two types of 60GHz beamforming radios. The first uses a pair of Dell D5000 dock (as transmitter) and 6430u laptop (as receiver), both equipped with a low-cost Wilocity 60GHz chipset designed for indoor mobile communications. The chipset uses a 2×8 rectangular antenna array, and operates under the IEEE 802.11ad standard [33]. Unfortunately, the chipset does not expose RSS values and the corresponding beam directions. Thus we use it only for understanding the range of our 60GHz imaging design when implemented on 802.11ad networking radios (§2.2.5).

The second and our main imaging testbed uses two HXI Gigalink 6451 60GHz radios, designed for outdoor communications. Since there are no suitable 60GHz steerable antenna arrays on the market, we emulate beam steering by setting a horn antenna (of 10° 3dB beamwidth) on an electronic controlled mechanical rotator. The horn antenna's main lobe pattern closely align with that of a 10×10 array with 1dBi elements and 21dBi gain, and the rotator physically adjusts the beam direction in units of 0.15°. The HXI radios use the On-Off-Keying modulation to generate sine waves in random on-off periods and reports RSS every 50ms. We note that under the same environment, the RSS of the HXI link is actually 17dB[22] weaker than that of the Wilocity chipset. This is because our HXI radio transmits at 0dBm and the cable that connects the horn antenna to the radio introduces 23dB loss (in order to enable mechanical antenna rotation).

The results of HXI radios with horn antennas should generalize to phased arrays, because our emulation matches phased arrays in three key aspects. *First*, 60GHz signal strength is largely determined by directionality and signal patterns of the main beam lobe (the side lobe is 13.26 dB weaker), and our horn antenna's main lobe pattern closely

[22]The HXI radios have 0dBm transmit power, 25dBi antenna gain per radio and 23dB cable loss due to the use of rotator. To compare with, the Wilocity chipset has 10dBm transmit power and 17dBi antenna gain per radio.

aligns with that of a 10x10 array [22]. *Second*, because 60GHz propagation is stable over time (verified by others [35, 36] and our own measurements), at each location RX can accurately measure RSS along different directions despite its slow beam steering speed. *Third*, the fine granularity of our rotator allows us to emulate beam steering of phased arrays, *e.g.*, in units of 1–3° required by the 802.11ad standard [74].

Experiment setup. Our experiments take place in a classroom of size 8m×12m with concrete walls. We place an object in the middle of the room with LoS to both TX and RX. We move both TX and RX to study imaging range and angle. By default, TX is 2m away from the object and RX is 3.5m away. We tried other distances with little impact on results as long as the total propagation path (from TX to object and then to RX) does not exceed the radio range. By default, the testbed steers beam at a 1° granularity.

We consider two types of RX movement: human waving smartphone and drone flying. The mechanical beamsteering means we cannot perform imaging in real time. Instead, we record five trajectories per type (by marking on paper) from actual movement of human users and our Iris drone, and use them to drive RSS measurements on our mechanical beamsteered radio. We align the trajectories so that the start and end points are 1 meter apart. Whether it's a user waving the device or a flying drone path, the maximum trajectory errors against a straight line are roughly $10cm$. To follow a specific moving trajectory, we mark the trajectory on the floor and pinpoint the receiver to each trajectory point using a plumb-bob.

We implemented the 60GHz localization mechanism (discussed in §2.2.4) for TX and RX. Across multiple scenarios, the measured TX/RX localization error is consistently between 2 and 6cm. We broaden our tests by adding random 2–6cm TX/RX position errors to our data analysis. In total, we have 60 noise instances per object/scenario to obtain statistically significant results.

Test objects. Like existing works on radar imaging [53, 46, 75, 76], we evaluate the utility and accuracy of our proposed design by imaging real objects of different size, curvature, and material. In addition to the objects listed in Figure 2.8 which are of 5cm–30cm in width, we also test smaller objects including a keyring and a small wrench (2.5cm in width). From these we seek to identify objects that our system can accurately image and those that it cannot.

Imaging Range & Angle

We first use the Wilocity radios to verify the working range of our 60GHz imaging system when implemented using today's 802.11ad mobile devices. For a pair of TX and RX, we block the LoS path between them, forcing the link to search and use NLoS paths, *i.e.* the reflection path. Using objects of different materials as the reflector, we measure the maximum propagation path length (TX to object to RX) such as TX can successfully transmit a 100MB file to RX. The path length is 20m for strong reflection material like metal, and 10m for weak reflection material like cardboard boxes (used for Amazon packaging). Assuming that TX and RX are of the same distance to the object, the corresponding imaging range (from object to RX) are 10m and 5m, respectively. We note that in our measurements the RSS is sufficient to support high speed communication (385Mbps) at these distances. When it comes to imaging, the RSS requirement can be much lower as long as the resulting RSS measurement is accurate. This means that the actual imaging range can be much longer.

As discussed earlier, due to extra cabling loss, our HXI link is 17dB weaker despite its stronger antenna gain. As a result, the imaging range is less than half of the Wilocity link, and we have to move the devices closer than we expect. Again this is due to artifacts of our testbed configuration.

Impact of imaging angle. Like [53], we found that the imaging quality depends on

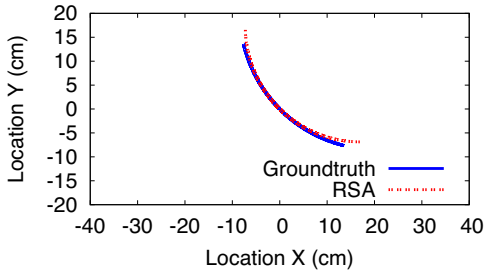


Figure 2.13: Result of imaging a curved surface.

the imaging angle, defined by the relative location of TX to the object and the trajectory of RX. Specifically, to identify the object, TX's beam should cover the object and the reflected signal should reach RX along its trajectory. Our system addresses this issue by performing object sensing and RX movement planning before running actual imaging (as discussed in §2.2.4). This also helps to reduce the amount of movement required to detect the overall object shape. For example, we have tested our algorithm by varying the object orientation relative to TX, *i.e.* the angle of incident, between 30° and 45°, and found that it always provides an ideal RX movement trajectory, and the subsequent imaging results remain consistent across these experiments. We also found that RX's actual movement can deviate from the ideal trajectory by at least 7° without noticeable impact on imaging quality.

Imaging Precision

Position/curvature/width. Table 2.3 lists RSA's overall imaging results in terms of errors in object center position and orientation, detected curvature type, shape deviation[23], and errors in surface boundary (width) estimation. We list the median and max values across all experiments while varying TX/RX position errors (60 instances) and RX trajectories (10 instances, 5 user-waving, 5 drones). The results for human waving

[23]Shape deviation is defined by the maximum difference between the actual object surface and the imaged object surface projected onto the object surface's principal axis.

and drone flying are consistent, so we did not isolate them.

We see that in the presence of position/trajectory noises, RSA achieves centimeter-level accuracy across all the objects, flat or curved. In terms of center locations, the max error is below 4cm for metal flat surfaces, and slightly larger (6-9cm) for curved surfaces and other materials (due to weaker reflection). Note that the depth of the object can be calculated based on the estimated object center position and the location of RX, which we found has a max error of $6cm$. The orientation error is always $< 1°$. RSA detects the curvature type accurately, and characterizes the shape of the surface within $0.68cm$ deviation for flat surfaces and $< 6cm$ for curved surfaces. The maximum error in surface width estimation is bounded $4.5cm$. Figure 2.13 plots the imaging result of a curved surface which captures the overall curved shape while being a slightly wider than the actual object.

Material. Using existing measurements on 60GHz reflection [77, 34], we built a reflection database of 39 materials. We added the profile of cardboxes using our own measurements. Using RSA estimated reflection loss and incident angle, we identify from the database the top three material candidates. Table 2.4 lists the result for four flat objects of different types (metal, plastic, wood and cardboard) and two curved objects, where RSA can successfully narrow down the material type.

Observed limitations. We also make the following key observations from our experiments. *First*, RSA reuses COTS 60GHz radios to image objects in the presence of device movement and tracking noises. To be robust to such noises (which existing designs like SAR fail to address), RSA has to sacrifice some degree of imaging precision without using the phase information[24]. As a result, our design seeks to identify the overall shape of an object surface (location, orientation, surface boundaries, material), rather than

[24]In practice, not requiring phase information can be an advantage, since most COTS radios do not report phase but only RSS values.

Ground truth				**Detected Position**			**Detected Shape**			
				Center location error		**Orientation**	**Detected**	**Shape deviation**		**Width error**
Objects in Figure 2.8	Shape	Radius	Width	Median	Max	**error (Max)**	**curvature**	Median	Max	Median	Max
(a) Aluminum Jar	Convex	+10.0	10.00	4.55	8.11	0.29°	Convex	1.59	2.22	1.00	2.33
(b) Steel cylinder	Convex	+15.0	15.00	4.43	4.98	0.51°	Convex	1.46	1.78	1.03	1.88
(c) Curved steel surface	Convex	+29.0	21.21	7.26	9.47	0.69°	Convex	1.47	1.93	2.67	4.22
(d) Curved steel surface	Concave	−29.0	21.21	6.64	7.65	0.90°	Concave	-1.64	-1.59	1.21	1.79
(e) Curved steel surface	Convex	+23.0	22.63	5.51	7.37	1.05°	Convex	4.46	6.35	0.91	3.59
(f) Metal desktop front	Flat	$+\infty$	06.72	2.82	3.64	0.49°	Flat	0.13	0.62	0.82	1.02
(g) Plastic keyboard	Flat	$+\infty$	10.00	4.44	7.31	0.81°	Flat	0.16	0.36	2.69	4.16
(h) Cardboard box	Flat	$+\infty$	18.00	4.31	5.13	0.72°	Flat	0.15	0.28	2.41	3.00
(i) Wood board	Flat	$+\infty$	22.00	3.83	8.73	0.44°	Flat	0.43	0.68	3.09	4.17
(j) Plastic battery case	Flat	$+\infty$	23.00	2.06	3.17	0.47°	Flat	0.28	0.48	1.70	3.51
(k) Plastic monitor	Flat	$+\infty$	26.00	4.07	6.29	0.45°	Flat	0.10	0.11	0.64	1.06
(l) Metal desktop side	Flat	$+\infty$	28.28	2.69	2.95	0.58°	Flat	0.12	0.12	1.78	3.16

Table 2.3: RSA imaging performance in terms of error in object center position and orientation, detected curvature type, deviation of overall shape, and error in object width (surface boundary). All the numbers are in the unit of *centimeter* except for the orientation error and curvature type.

Object (Material)	Estimated Reflection Loss	Top 3 Matches (out of 39)
Desktop (Metal)	0.3dB	**Metal**, Quartzite, Glass
Box (Cardboard)	6.1dB	**Cardboard**, Pertinax, Acrylic glass
Monitor (Plastic)	7.9dB	Chipboard, Fiberboard, **Plastic**
Board (Wood)	12.7dB	**Wood**, Brick, Breeze block
Cylinder (Metal)	0.3dB	**Metal**, Quartzite, Glass

Table 2.4: Results of RSA material detection.

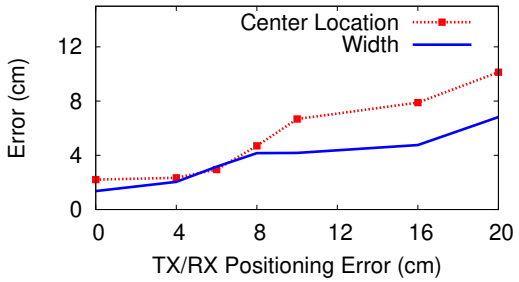


Figure 2.14: RSA's imaging errors scale gracefully with TX/RX position errors. RX trajectory noises are present for all the experiments.

fine-grained details such as individual keys on a computer keyboard. *Second*, our design is unable to accurately image small objects. It can detect and locate a wrench handle (of 2.5cm width), but the detected width varies between 1cm to 5cm. It cannot locate a keyring because the reflection is too weak to be captured by our HXI radios. To recognize these small objects, one could use a stronger radio, or move TX and RX much closer to the object, *e.g.*, <0.5m which becomes a near-field scenario and requires a new imaging design. *Third*, the accuracy of width estimation depends heavily on the RX movement distance. The amount of RX movement distance required to maintain high precision increases linearly with the sensing range and object size. We found that for our HXI testbed and all the test objects, 1 meter RX movement is sufficient. Our RX movement planning also predicts the same trajectory length (see §2.2.4).

TX/RX Positioning Error (cm)	center location error (SAR) / center location error (RSA)			width error (unfocused SAR) / width error (RSA)			width error (SAR) / width error (RSA)		
	Median	Min	Max	Median	Min	Max	Median	Min	Max
0	1.20	1.00	2.07	6.71	3.88	17.95	6.92	3.16	19.09
4	1.72	1.24	2.77	4.44	2.59	11.44	8.20	4.59	38.33
8	1.72	1.44	2.63	3.23	1.27	6.67	5.06	3.34	10.39
16	1.85	1.62	3.83	2.32	1.11	5.29	2.90	1.84	6.89
20	2.32	1.47	2.78	2.18	1.01	3.82	3.05	1.44	6.09

Table 2.5: Comparing RSA to SAR and unfocused SAR in terms of the ratio of error under SAR (or unfocused SAR) and error under RAS. The RX trajectory errors are present in all the experiments. Since the performance of unfocused SAR is sensitive to RX moving distance, we configure it as 0.5 meter while SAR and RAS use 1 meter.

Robustness to Noise

While Table 2.3 lists the imaging result when the TX/RX positioning error is bounded by $6cm$, we expand our noise model to explore RSA's noise tolerance. We found that RSA's performance is insensitive to trajectory errors when the deviation is bounded by $10cm$. Thus we focus on the TX/RX position errors. Specifically, we pick X as the maximum location error (deviation from the ground truth), draw a circle of radius X around the ground truth and randomly pick a point on the circle as TX's relative location to RX. We repeat this 20 times per X and report the maximum imaging errors. Using object (g) as an example, Figure 2.14 plots the maximum error in center location and width estimation for X between 0 and 20cm. We see that both errors grow gracefully with X, indicating that RSA is robust against TX/RX positioning errors. We observe this same trend for all objects.

RSA vs. SAR and unfocused SAR

We also compare RSA with SAR and unfocused SAR (as described by existing work [27]). Since both SAR algorithms do not offer curvature and material information, we only evaluate object center location error and width error. Specifically, for each sce-

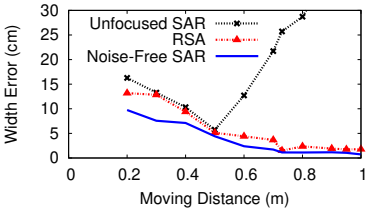
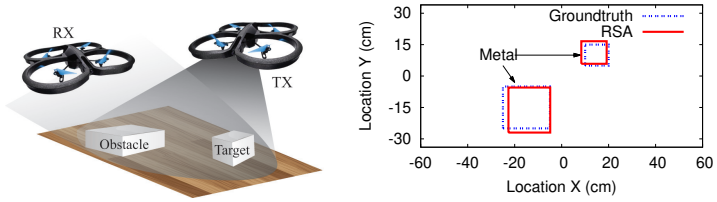


Figure 2.15: Impact of measurement configurations on RSA imaging performance, in terms of width error.



(a) Scenario setup (b) Imaging result (top view)

Figure 2.16: "Realistic" case study of RSA imaging: a drone seeks to locate the small metal object while avoiding a nearby obstacle.

nario defined by TX/RX positioning error, RX trajectory, and the object, we compute the amount of error reduction achieved by RSA, *e.g.*, $\frac{\text{width error (SAR)}}{\text{width error (RSA)}}$ or $\frac{\text{center location error (SAR)}}{\text{center location error (RSA)}}$. We report the min, median, and max values across the eleven objects, 10 trajectories and 20 instances of position errors for a given range X. Because unfocused SAR is highly sensitive to the choice of RX movement distance, we use 0.5 meter for unfocused SAR (which provides the best overall performance across all the objects), and 1 meter for RSA and SAR.

Table 2.5 lists the error reduction factor of RSA. For center point estimation, SAR and unfocused SAR have very similar performance [27], so we only report one. We see that RSA can effectively reduce the imaging error. This is particularly true for width estimation — the median reduction factor is 2–6.7 (over unfocused SAR) and 2.9–8.2 (over SAR); while the maximum value can reach 18 and 38 respectively. Also, width error reduction peaks at zero TX/RX position error, confirming RSA is highly robust against trajectory errors. Finally, RSA reduces errors in center point estimation by a factor of 1.2–2.32 (median). This is mostly due to the iterative search process (§2.2.3).

Microbenchmarks

RX movement distance. Figure 2.15 compares the object width error at different

RX movement distances. We see that RSA follows the same trend as noise-free SAR: imaging error reduces with moving distance and gradually converges to a stable value. This aligns with the theoretical limit in Equation (2.1) where increasing aperture (via RX movement) leads to higher imaging precision. Unfocused SAR, however, is highly sensitive to this parameter.

RSS measurement frequency. This factor translates into the choice of N, the number of measurements for a given movement distance. In practice, we want to minimize measurement frequency to save energy. Yet insufficient number of measurements reduces the accuracy of our model fitting. Our results in the above perform one measurement per 1cm. At a slow movement speed of 0.5m/s (1.1mph), the measurement frequency is once per 20ms. We found that the results remain the same even at a lower frequency of once per 80ms (*i.e.* once per 4cm).

Beam steering granularity. Using an electronically controlled mechanical rotator, our testbed can steer antenna beam in increments of 0.15°. While our experiments above use data from 1° steering, we also perform imaging under 0.15°, 3° and 5° steering to examine the impact of antenna hardware (steerable phased arrays). To separate its impact on localization, we use the same TX/RX localization result of 1° across all the experiments. Our results show that 1–3° steering is an efficient choice — increasing granularity to 0.15° reduces width error by <0.5cm while relaxing to 5° doubles the width error.

Case Study: Multiple Objects Detection

Consider a scenario in Figure 2.16(a), where a drone uses RSA imaging to locate an object, *i.e.* a square metal box of size 8cm×8cm. The target rests on a wood floor with a nearby larger (metal) object (size 18cm×18cm) as the obstacle. To locate (and pick

up) the object, a drone needs to recognize both objects, with the help of another drone as TX.

With two testbed radios emulating drones, they first perform 60GHz localization to locate each other. They then coordinate to sense the objects. Since the two objects are in proximity they can be covered by a single TX beam. After TX focuses its beam on the two objects (and sends beacon signals), RX moves in two directions sequentially to determine location, curvature, width and height, and material. The visual imaging result and the ground truth are shown in Figure 2.16(b) where RSA recognizes two flat metal objects, their overall shape/size, and the wood floor in between.

2.2.6 Related Work

Camera-based imaging. Camera is widely used for object recognition [40, 41, 42]. Detecting object position and shape, however, requires bulky, high-end cameras (*e.g.*, Google's Project Tango requires an infrared depth camera and a fish-eye lens). These mechanisms require good visibility and cannot reliably identify object material. RSA takes a low-cost RF-based approach leveraging mobile networking chipsets, and its 60GHz reflected signals reveal key properties of the object surface without any light.

Sonar and radar systems. These systems have been applied to many fields [37], from mapping terrain contour, tracking moving targets, to detecting concealed weapons at security checkpoints [38, 54, 55, 56, 57, 39]. They use special hardware like X-Ray or bulky lenses to achieve high precision, which are too large/expensive for mobile devices. RSA differs by using commodity 60GHz networking chipsets that are being integrated into today's mobile devices.

RF-based systems. Researchers have explored WiFi-band solutions to detect human motion, activity and gestures, and to detect (metal) objects [44, 12, 45, 46, 64, 47, 48].

A recent work built WiFi imaging using OFDM and large phased arrays (available on APs) and discussed the resolution limitation due to its large wavelength [53]. Our work considers 60GHz (mmwave) communications because it offers several desirable qualities for mobile imaging when compared with WiFi: tiny wavelength, high directionality, stable and predictable signal propagation. In addition to providing high-resolution (∼1–5cm), our imaging algorithm is also different by using just RSS measurements (rather than phase [53]) without requiring specialized hardware (*e.g.*, [44, 12]). Our work was inspired by recent 60GHz radar designs [27, 78] that apply SAR to detect object surface location and boundary in absence of noise. Our work develops a new imaging solution that is robust against noise and also detects surface curvature and materials.

2.2.7 Summary

Our proposed 60GHz mobile radar detects the location, orientation, curvature and surface boundaries of nearby objects using only signal strength measurements, and achieves cm-level precision.

Several limitations remain before we can realize a high precision, environmental mapping system using RF reflections. *First*, our imaging works when both TX and RX have line-of-sight to the target object because 60GHz signals cannot penetrate walls or most objects. To perform environmental mapping in 3D, TX and RX (*e.g.*, mobile devices) must have a way to intelligently navigate across space while coordinating their positions. *Second*, so far we only consider the general shape of static and regular object surfaces where reflected RSS is stable over time and predictable via a model. For moving or irregular objects, *e.g.*, humans, and detailed shape, *e.g.*, keys on keyboard, we need new models to define the correlation between object shape and reflected signal patterns. *Finally*, we need to develop an algorithm for TX and RX to reliably and iteratively scan

individual surfaces while moving in unknown environments. Such schemes must be robust and work reliably in large environments with complex objects (*e.g.*, caves, collapsed tunnels).

2.3 Imaging and Navigation on A Single Device

[25] Autonomous devices are the future of mobile computing. Today, Amazon's drone-based home delivery system (Prime Air) has already received regulatory approval in the UK [80], and retailers like 7-up have performed drone deliveries on a smaller scale [81]. Meanwhile, Uber has already deployed a pilot program for self-driving passenger pickup vehicles in Pittsburgh [82]. With improved hardware and advances in robotics, autonomous robots can do much more than their capabilities today. One might imagine more powerful versions of the Roomba robot cleaning tables and countertops at home, personal robot companions that walk alongside the elderly or visually impaired, and first responder robots that identify and rescue survivors from natural or man-made disasters [83, 84].

One of the significant roadblocks on the path towards this vision is object imaging and recognition. Autonomous devices require knowledge about objects in their surroundings, including distance to the object, size, surface curvature and other properties. Such information allows devices to recognize and distinguish between nearby obstacles and potential target objects for interaction. Additionally, a practical imaging system must be portable enough to mount on mobile devices, and provide robust results in a wide range of environmental conditions.

Of the numerous imaging products available today, few if any are appropriate for autonomous mobile devices. In Figure 2.17, we classify potential imaging solutions based

[25] The content in this section is published in [79].

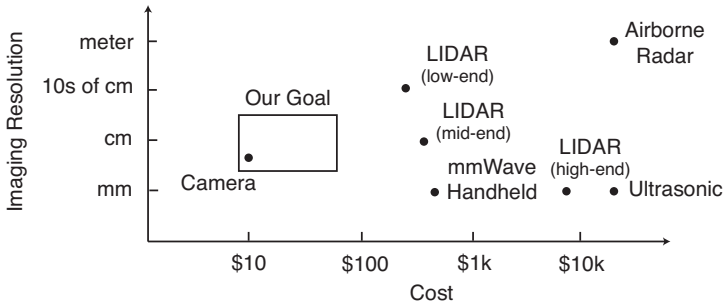


Figure 2.17: Comparing existing commercial single-device imaging products.

on their precision and cost, including products based on sonar [6], computer vision [15], radar [7, 8] and LIDAR [11, 9, 10]. Nearly all of these approaches require specialized hardware that make them too costly (*e.g.*, > \$250) and unwieldly for mobile devices[26]. The only exception is camera-based systems. Yet they are sensitive to lighting conditions and can fail badly when objects and their backgrounds have similar colors, (*e.g.*, the likely cause of the recent fatal accident in a driver-assisted Tesla [18, 17]).

Fortunately, researchers have made significant recent advances in radar systems, dramatically reducing their cost, size and weight. The first group of efforts developed specialized hardware operating in the WiFi bands (Frequency-Modulated Continuous Wave or FMCW radars). Leveraging the precise ranging capabilities of FMCW radars, researchers developed novel systems that detect and measure subtle human dynamics, *e.g.*, heartbeats and body movements [13, 14, 12]. The second group of efforts (re)uses commodity networking devices (non-FMCW hardware) to recognize static objects. Existing works use either two well-separated static WiFi radios [53] or two independently moving WiFi or 60GHz radios [85, 50].

Imaging via a single networking device. Our goal is to further advance the state of imaging systems based on commodity networking devices. Like [50, 85, 53], we

[26]The cheapest mmWave handheld imager costs \$500, weighs 5lb [7], and the cheapest low-end LIDAR costs \$250 and offers 1D imaging at a resolution of tens of cms [9].

focus on imaging static objects, but address the key limitation of requiring dual separate devices. We also integrate navigation with imaging, using the same networking radio to avoid obstacles. The result is a simple, single-device imaging system that recognizes details of objects meters away, using only the device's onboard networking radios. This supports a low-cost solution deployable in crowded spaces, robust to a variety of lighting and acoustic conditions, while avoiding the overhead and complexity of coordinating multiple devices.

Specifically, our approach is to use a pair of 60GHz networking chipsets, one transmitter (TX) and one receiver (RX), mounted on a single commodity mobile robot, separated by a (small) fixed distance (25–40cm). As the device moves by an object, the RX picks up reflections of signals sent by the TX, and uses changing angles of reflection along the path to reconstruct the surface shape, size, curvature, and material of the object. While the TX and RX beamform and analyze reflected beams to enable imaging, the device moves to emulate a large aperture antenna array. Finally, the mobile device can scan an area with multiple objects and map the area and the objects by navigating over a carefully computed path.

Our contributions. This section describes the design, implementation and evaluation of *Ulysses*, an object imaging system using a single compact mobile device and onboard 60GHz networking radios. Since phase noise from device tracking errors and colocating of the TX/RX pair introduce changes that invalidate existing radar and RF imaging algorithms, we must design a new imaging algorithm. Our work makes three key contributions:

- **A new imaging algorithm driven by specular reflection and beamforming.** Center to Ulysses is a new imaging algorithm that operates on *beamforming RSS*. As the device moves nearby an object, Ulysses captures the *specular* reflec-

tion off the object surface, uses the observed angular and amplitude values of the reflection to recognize tiny segments on the object surface, and then leverages device trajectory to assemble them and reconstruct the object surface details. In a nutshell, Ulysses emulates monostatic synthetic aperture radar (M-SAR) without using phase (for robustness), but using angular and signal strength information offered by the commodity 60GHz radios. Ulysses differs fundamentally from existing 60GHz imaging design [50], which explicitly avoids segment assembly.

- **Navigation based on 60GHz beamforming.** Ulysses also integrates robot navigation with imaging, focusing on defining "safety zones" where devices can move freely without collision, again using just the onboard 60GHz radios.

- **Prototype and evaluation using 802.11ad phased arrays and robot.** We prototype Ulysses using commodity 60GHz 802.11ad radios with phased arrays, which are cost effective ($5) and small enough (4.8cm $\times$ 2.4cm) for mobile devices. We then integrate Ulysses on a compact robotic car and evaluate it in multiple indoor and outdoor settings on objects of various sizes, shapes and materials. Our results show that Ulysses images objects meters away with cm-level precision, provides accurate estimates of the surface material, while safely and efficiently navigating in unknown, crowded environments. It is also robust against robot trajectory errors (up to 10cm). We also compare Ulysses to camera based imaging [15] and dual-device 60GHz imaging radar [50]. Ulysses achieves similar accuracy but eliminates the sensitivity to lighting and the need of two mobile devices.

Our prototype is primitive and limited by both hardware and device constraints. However, we believe these results demonstrate significant promise for this approach towards the development of an accurate, low-cost, and portable imaging system for single

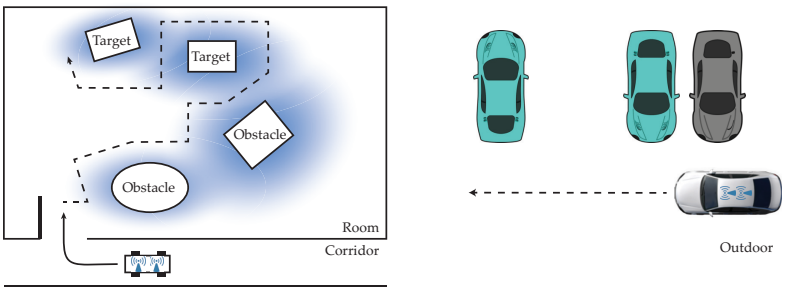


Figure 2.18: Illustration of two target scenarios where a robot explores an unknown room to image target objects, and a car drives around a set of parked cars to image them.

Figure 2.19: The actual size comparison of a drone, a robot car and our 60GHz array prototype. The array (16×8) is compact and both TX and RX can be mounted on a single mobile device.

mobile devices. We discuss the limitations of our current design in §2.3.6, and potential solutions to be explored in upcoming work.

2.3.1 Single Device Mobile Imaging

As background, we describe in this section our target scenarios and constraints, the trade-offs for using 60GHz networking radios, the reason why existing imaging solutions cannot be applied, and the key challenges facing our single-device imaging design.

Scenarios and Requirements

Our basic operating scenario includes a compact mobile device, *e.g.*, a robot car or a drone, exploring an unknown environment by imaging nearby objects. Figure 2.18 shows two illustrative examples. The first is a robotic car tries to pick up specific objects inside

an unknown room. Target objects are defined by size, shape and material, *e.g.*, a metal box of a specific size. To locate the objects, the robot navigates through the room while imaging each object or obstacle (without bumping into them). The second scenario is a vehicle (or a robot) tries to image the back of parked cars in an outdoor parking lot. For both scenarios, "imaging" an object means recognizing its size, shape and material.

The key requirements for our imaging system include:

- To be suitable for mobile devices, our imaging solution needs to be lightweight and low-cost, and compact enough to be mounted on a single compact mobile device like robot cars or drones (30–40cm in width).
- Our system must accurately image static objects, including recognition of object size, shape (curvature) and materials. The detailed information will greatly help with object recognition, especially in unknown environments.
- To satisfy today's application scenarios, our solution must attain accuracy (of location or size/shape) to a small number of centimeters, and image objects meters away.

A Case for 60GHz

Our goal is to achieve imaging using commodity networking radios on a single device. Both WiFi and 60GHz networking radios are attractive candidates because they are in unlicensed bands, and have commodity networking chipsets on the market today that are energy efficient and low-cost ($5–$30). We choose 60GHz radios because they offer high directionality (via real-time beamforming) in a small form-factor, so we can place both TX and RX radios on a single, compact mobile device. Furthermore, under high directionality, 60GHz propagation and reflection face minimum multipath effect, and

are stable and predictable for both indoor and outdoor scenarios, as shown by prior studies [22, 86].

Figure 2.19 compares the sizes of a drone, a robot car, and our 60GHz array prototype. The 60GHz array uses a standard 8×16 rectangular array and is 2.4cm × 4.8cm in size. The compact and light-weight design makes it feasible to deploy both transmitter and receiver on a single autonomous device. To achieve the same directionality using WiFi requires antenna size at least 12 times larger. Furthermore, our initial prototype already offers real-time fine-grained beamforming, *i.e.* switching beam every 0.4ms. Commercial 60GHz chips offer beam switching at a higher speed of 50ns [73].

Compared to WiFi, the key limitations of 60GHz radios are reduced range and sensitivity to blockage and rain. But since our goal is to image objects a few meters away (rather than maintaining high-speed communications), these limitations are tolerable under our scenarios. Heavy rain only adds 0.2dB signal loss for a 10m imaging range.

The Need for a New Imaging Algorithm

A key design question is "can we apply existing radar or RF imaging algorithms to our system?" Unfortunately, our analysis shows that existing solutions fail to apply. As starter, FMCW based solutions (*e.g.*, mmWave handheld imagers [7]) do not apply because they require specialized frequency-modulated hardware components that do not exist on commodity networking radios, and it will be difficult and costly to port them into networking radios. Next, prior design for WiFi mobile imaging [85] leverages the shadowing effect of WiFi propagation between two well-separated devices. This approach is not applicable to our scenario since it requires two independently moving mobile devices, and also 60GHz can hardly penetrate objects (thus there is no shadowing effect). Finally, the most relevant solutions are the monostatic synthetic aperture radar (M-SAR) algorithm [87] and variations, and the RSA algorithm for 60GHz mobile imaging (§2.2),

which we discuss next.

M-SAR and phase-based solutions. Using colocated transmitter and receiver, M-SAR [87] emulates a large antenna array by moving the device and aggregating both signal strength and phase measurements across locations. Unfortunately, M-SAR requires accurate phase construction along the device path to assemble the measurements. For 60GHz radios, this is only feasible when the device can track its trajectory to sub-millimeter-level accuracy. Our prior work (as presented in §2.2) have shown that even millimeter-level tracking errors translate into random, large phase noises and significant imaging errors. Similarly, prior works on WiFi object imaging [53], near-field mmWave object imaging [88] and tracking [89] all rely on accurate phase construction, and fail to apply here.

RSA. Developed for dual-device 60GHz mobile imaging (§2.2), RSA tolerates trajectory noises by operating only on RSS. Yet it fails to apply to our system because our *new* requirement of colocating TX and RX breaks the fundamental assumption of its design. Specifically, RSA images an object surface by capturing and modeling the *scattering* reflection contributed by the entire surface as one unit. This methodology works when TX and RX are widely separated and moving independently, but breaks down when TX and RX are colocated and move in unison (discussed next).

Key Challenges

Challenge 1: Large phase noises. Like existing works on 60GHz mobile imaging, our system faces the challenge that (random) errors in trajectory tracking translate into large phase noises across signal measurements. When phase measurements are used in imaging, we observe large imaging errors (similar to §2.2). Thus like §2.2, we chose not to use phase measurements in our imaging design.

In addition to not using phase, we face several new challenges by colocating TX and RX on a single mobile device, separated by a small, fixed distance.

Challenge 2: Moving TX. Colocation means that TX moves with RX as the device travels. This breaks the foundation of RSA, which assumed a static TX with fixed beam direction (during imaging) that helped generate an "anchor point" for the entire surface. RSA fails when TX moves in unison with RX.

More importantly, while RSA assumed scattering (specular+diffuse) reflection due to static TX and mobile RX, our system operates on specular (or direct) reflection thanks to colocation. As shown by Figure 2.20, signal reflected from a surface generally includes both *specular* and *diffuse* components — specular reflection is focused on a single direction and diffuse reflections are scattered over a range of directions. Since object surfaces are much larger than 60GHz's 5mm wavelength, specular reflection, when captured, is much stronger than diffuse reflection. This and the fact that TX/RX are in close proximity, co-moving and beamforming, indicate that our system will operate on specular reflection along the device trajectory, both indoor and outdoor.

Challenge 3: Limited visibility. With colocation, the proximity between TX and RX means (specular) reflections from the object can only be detected in a small angular window, and thus at a limited set of locations. Using a rectangular object as an example, Figure 2.20 also shows the measured reflection signal strength around the object. Clearly the object is only "visible" at a small set of locations, especially near the four object corners.

Challenge 4: Navigation via 60GHz beamforming. Since each object is only "visible" at certain locations due to specular reflections, the mobile device must travel to find these visible locations to perform imaging. That is, the mobile device must integrate its navigation with its imaging process.

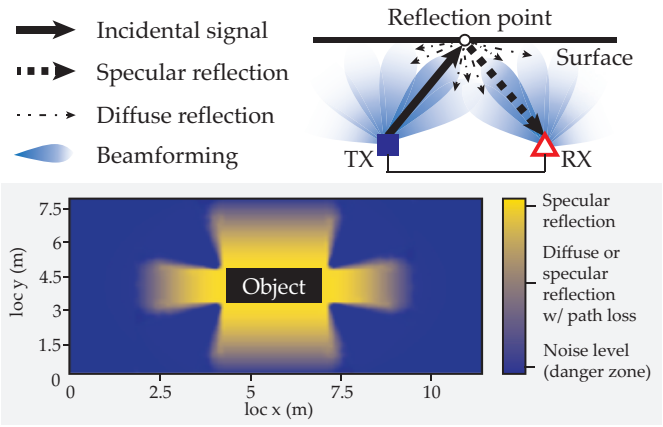


Figure 2.20: Colocated TX/RX leads to limited visibility of specular reflection, for both indoor and outdoor settings.

Rather than reinventing our own navigation algorithms, we seek to leverage existing navigation algorithms from the robotics community [24, 25, 26]. These algorithms typically assume cameras, sonar or LIDAR as sensors. Instead, we develop methods to compute "safety zones" for navigation based on 60GHz beamforming results, and to plan movement trajectories that facilitate the imaging process. To the best of our knowledge, we are the first to quantify safety zones using 60GHz beamforming.

2.3.2 Ulysses

To address the above challenges, we introduce *Ulysses*, a single-device imaging system that uses 60GHz directional beams to detect and image unknown objects in far-field scenarios. Our key insight is that as TX and RX move in unison on certain trajectories and perform fine-grained beamforming, RX can *continuously* capture specular reflection contributed by each *small* segment of object surface. The geometry of specular reflection creates a strong tie between its angular properties and the surface shape of these segments. Ulysses then integrates these angular properties with the device trajectory to assemble the estimated "segments" and image the object. Since Ulysses operates on the signal

strength and angular information of 60GHz beamforming signals, it is robust against trajectory errors (up to a few cms). We confirm this by performing signal measurements on different trajectories. While these trajectories can deviate from each other by as much as 10cm, their RSS and angular values vary little. This observation aligns with §2.2.

Core Concepts. Ulysses includes three core components:

- a *sensing* module that uses 60GHz beamforming to detect and extract specular reflection off objects;
- an *imaging* module that leverages the geometry of specular reflection and converts device trajectory into a reliable estimate of the surface shape; in essence, our design emulates M-SAR's point aggregation process [87] without using phase;
- a *navigation* module that uses 60GHz beamforming to safely and efficiently explore the unknown environment and to identify paths for capturing specular reflection off objects and imaging them.

In the following, we present the core idea of each component and leave their design details to §2.3.3.

1. Sensing specular reflection via beamforming. As shown by Figure 2.21, at each location, Ulysses scans for objects using *real-time*, *fine-grained* RF beamforming. This beamforming function is defined by the 802.11ad standard for 60GHz networking [33], and supported by all commodity 802.11ad chipsets. Using beamforming, Ulysses leverages the high directionality of 60GHz antenna array to capture reflection signals in each fine-grained TX/RX beam directions. That is, without physically rotating the hardware device, Ulysses can sense surrounding objects in real-time.

Once the reflection signal is identified as specular (details in §2.3.3), Ulysses extracts the angular and signal strength information of the reflection signal, producing a sensing

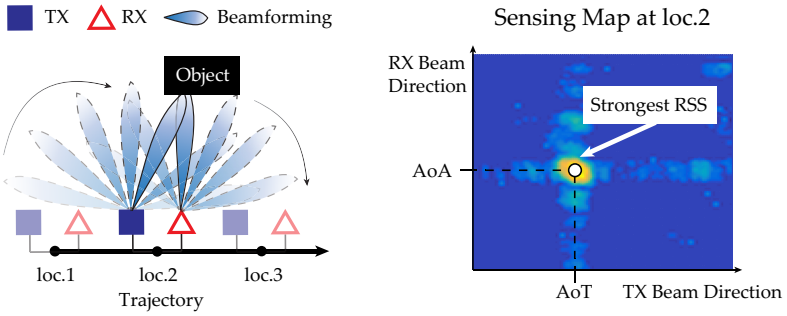


Figure 2.21: At each location, Ulysses scans objects by fine-grained beamforming. The result is a per-location sensing map that records the received signal strength as a function of TX and RX beam directions. The peak defines the $\{AoA, AoT, RSS\}$ tuple for imaging.

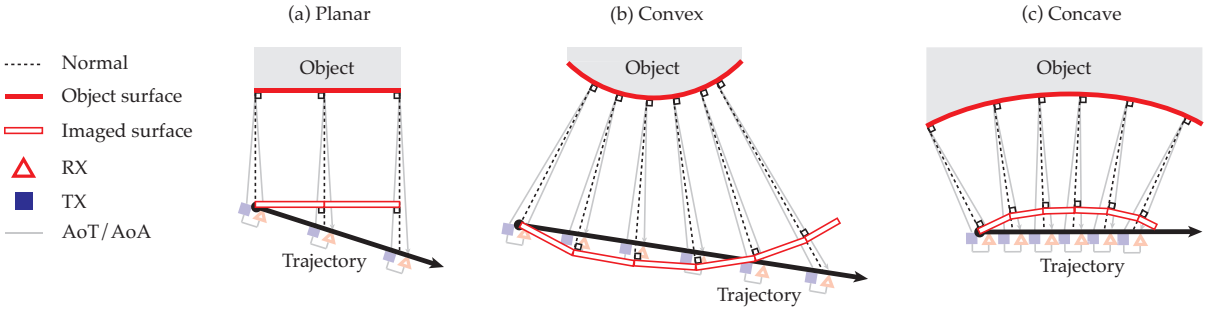


Figure 2.22: Estimate surface shape by projecting trajectory. At each measurement location on the trajectory, the captured $\{AoA, AoT, RSS\}$ is contributed by a small segment of the object surface. By projecting the trajectory segment guided by the two normal lines, we can estimate the shape of this surface segment. We then stitch these estimates up to build a continuous surface shape estimate.

map per location. Figure 2.21 shows an example sensing map, which records the signal strength as a function of the TX and RX beam directions. The peak on the map is used to extract the Angle of Arrival (AoA) and the Angle of Transmission (AoT), representing the RX and TX beam directions that lead to the strongest reflection signal, respectively. Since the strongest reflection comes from the center line direction of both the TX beam and the RX beam, each tuple of AoA, AoT and the corresponding received signal strength (RSS) captures the specular reflection off *a very small segment of the surface*.

2. Imaging surface by projecting trajectory. The basic concept is shown by Figure 2.22. As the device moves around an object and "lights" up each small segment of

the surface, the measured $\{AoA, AoT, RSS\}$ sequence allows us to compute the *normal line* of each small surface segment, *i.e.* a line that is perpendicular to the segment and represents its orientation.

Building an image of the object surface, however, requires both the normal line (orientation) and the location of each small segment. Since each segment location is the incident point of the reflection, ideally it can be estimated by intersecting AoA and AoT. Yet in practice the result is quite noisy due to both the quantization noise in beam steering (an inherent artifact of analog beamforming hardware design) and the trajectory noise in device movement. One might consider conventional RF ranging/positioning solutions, such as the time-of-flight method [90]. But under our far-field scenarios, these solutions only pinpoint the center of a surface rather than each tiny segment on a continuous surface.

Ulysses takes a different approach to image the surface without pinpointing each segment. Specifically, Ulysses first estimates the surface shape (size, curvature, orientation) without performing any ranging. It then estimates the center location of the object surface using *all* the beamforming measurements on the trajectory (thus achieving much higher accuracy), and "shifts" the estimated shape to the estimated surface center.

To estimate the surface shape, Ulysses leverages the fact that object surfaces are locally continuous, and thus the orientations of two neighboring surface segments are similar. Ulysses recovers the surface shape by projecting the device trajectory along the normal lines of every two segments. As shown by Figure 2.22, such projection can successfully reveal not only flat surfaces but also curved surfaces (details in §2.3.3).

3. Navigation by 60GHz beamforming. With colocation, a mobile device can only capture specular reflections at selected regions near an object. To navigate to these (unknown) locations while avoiding obstacles, Ulysses integrates 60GHz beamforming with prior work in robotic navigation [91]. Specifically, Ulysses leverages 60GHz reflection

models to compute safety zones when reflection signal is present or absent, and uses them to guide device navigation. It also leverages beamforming sensing results to plan efficient trajectory around the object(s) to perform imaging.

2.3.3 Ulysses Design Details

We present Ulysses' three modules in detail, starting from the sensing and imaging modules assuming a single object is present. We then discuss how Ulysses images multiple objects simultaneously, followed by the navigation module.

Sensing via Beamforming

The colocated TX/RX perform sensing using the fine-grained beamforming module in 802.11ad [33]. TX steers its beam towards each direction for a small period of time, *e.g.*, 25ms in our prototype, and sends out beacon packets repeatedly. RX steers its beam at a faster speed, *e.g.*, 0.5ms, and records the RSS value at each scanned direction. Upon capturing reflection signals, Ulysses identifies whether specular reflection is present, and extracts the $\{AoA, AoT, RSS\}$ tuple.

Detecting specular reflection. Being much stronger than diffuse reflection, specular reflection can be detected by examining the measured signal strength across locations. When moving from a region where specular reflection is invisible to a region where it becomes visible, the device will observe a sharp jump in the measured signal strength (at the strongest direction per location). Such variation is significantly stronger than those within each region. Thus Ulysses detects the presence of specular reflection if the signal strength variation (over space) exceeds some threshold.

Handling array sidelobes. Unlike laser, the beams of 60GHz arrays are not perfectly "clean" — it contains a strong main lobe and many weaker side lobes. The side lobes

can be reflected towards RX by the target object, other objects or backgrounds, leading to noises in the sensing results. In Ulysses, we apply the method in [92] to distinguish the contribution of the main lobe from those of the side lobes. Thus these side lobes have minimal impact on our system.

Imaging by Projecting Trajectory

After detecting the presence of specular reflection, the Ulysses device will move following a scheduled trajectory $\mathbb{T}$ (see §2.3.3 for trajectory planning) to image the corresponding object. When moving, the device performs the aforementioned beamforming sensing. The sensing granularity depends on the beamforming sensing time and the moving speed, but should be at least once every 1cm to ensure cm-level imaging accuracy.

Recovering surface shape from trajectory. The imaging module takes as input the trajectory $\mathbb{T}$ and a sequence of measurement tuples $\{AoT_i, AoA_i, RSS_i\}_{i\in\mathbb{T}}$, where i is a measurement location on $\mathbb{T}$. Each tuple i corresponds to a segment of the object surface, whose normal line is $\theta_i = (AoT_i + AoA_i)/2$. Similarly, the trajectory $\mathbb{T}$ is also segmented, where each segment $\mathbb{T}_i$ starts from measurement location i and ends at $i+1$. Next, at each location i, $\mathbb{T}_i$ is projected onto a line that is perpendicular to both θ_i and θ_{i+1}, creating a surface estimate $\hat{\mathbb{S}}_i$. Finally, all the estimated segments are assembled in space by aligning the starting point of $\hat{\mathbb{S}}_{i+1}$ with the end point of $\hat{\mathbb{S}}_i$, creating a continuous surface shape $\hat{\mathbb{S}}$.

As discussed earlier, Ulysses does not locate each individual surface segment. As each segment is small (<1cm), even sub-cm error in ranging will create unnecessary ambiguity in shape estimation. Instead, Ulysses assembles the estimated segments given that object surface is locally continuous. Doing so means the imaging result might miss subtle surface details, *e.g.*, keys on a keyboard. This is not a requirement for our target scenarios, and we leave it to future work.

Computing surface boundaries and curvature. The surface curvature can be easily determined by intersecting all the normal lines $\{\theta_i\}_{i\in\mathbb{T}}$. For flat surfaces, these lines should not intersect; for convex surfaces, they intersect at a location in the TX beam direction; and for concave surfaces, they intersect at a location in the reverse TX beam direction.

The estimate on surface boundaries, *i.e.* width, depends on the curvature. For flat surfaces, the shape estimate $\hat{\mathbb{S}}$ has the same length of the true surface. For curved surfaces, $\hat{\mathbb{S}}$ is either an enlarged (convex) or compressed (concave) version of the true surface (see Figure 2.22). But since the true and estimated surface shapes share the same curvature center, *i.e.* the intersection of $\{\theta_i\}_{i\in\mathbb{T}}$, we can resize $\hat{\mathbb{S}}$ properly by estimating the radii of the true and estimated shapes.

To compute the curvature center, we intersect every pair of $\{\theta_i\}_{i\in\mathbb{T}}$, and take a majority vote to mitigate noise. We calculate the radius of the estimated surface by applying majority vote on the distance between each measurement location and the curvature center, again to mitigate noise. Finally, we compute the radius of the true surface as the distance between the curvature center and an estimate on the object center (discussed below).

One exception is when the device is further away from a concave object than its curvature center, *i.e.* the curvature center is in between the trajectory and the object center. Now the reflection will appear as coming from a convex surface. This can be easily detected and corrected by flipping the estimated (convex) shape $\hat{\mathbb{S}}$.

Estimating object surface center and material. We now estimate the center location of the object surface, which allows us to not only determine the surface curvature and width, but also place the estimated shape $\hat{\mathbb{S}}$ at the proper location. For robustness and accuracy, we estimate the surface center using all the sensing results along the trajectory. Specifically, we compute the intersection of each AoA and AoT pair and

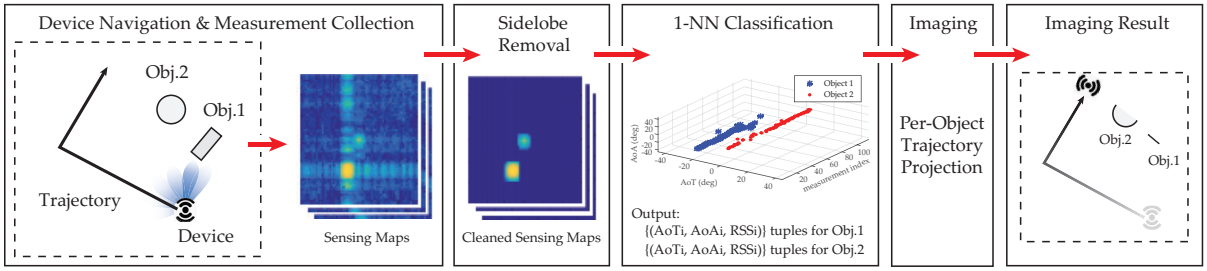


Figure 2.23: Ulysses can image *multiple* surfaces from a single trajectory. From a sequence of sensing maps, we extract per-object $\{AoA, AoT, RSS\}$ tuples via classification, and then image each object separately.

use the average over the trajectory as the estimate of surface center. After estimating the surface curvature and radius, we the refine the center estimate accordingly. Note that when the radio hardware offers access to high-precision timing information, we can also leverage the time-of-flight method [90], which can achieve $< 10cm$ accuracy that is independent of object-to-device distance.

Finally, given estimates of surface curvature, width and center location, we can determine the materials by computing the signal reflection loss (§2.2). That is, we first predict RSS (at the trajectory center) assuming signal reflection leads to zero loss, and compare it to the measured RSS value. The difference between the two is the reflection loss and the corresponding material can be found from the widely used material-loss table [34].

Imaging Multiple Objects

The above discussion assumes there is only one object in the search space. When multiple objects (including background walls, etc) are present, the device may capture reflections from multiple surfaces along its trajectory. In this case, Ulysses follows the same algorithm to image each object, but first applies the following method to detect and extract reflection signals for each surface.

Extracting per-object reflection signals. Here we leverage two insights. *First*, thanks to 60GHz high directionality, the (specular) reflection seen by each individual RX

beam generally only comes from a single surface. *Second*, along a continuous trajectory, the reflection from each object maintains a strong correlation over space. That is, the corresponding $\{AoT, AoA, RSS\}$ tuple per object varies smoothly along the trajectory.

With these in mind, we apply a classification-based method to separate contributions from different objects. Figure 2.23 illustrates the process. Given a sequence of sensing maps collected along the trajectory, we first remove the contributions of side lobes using [92]. If the cleaned sensing maps still contain multiple peaks, then multiple objects are observed. Next, from each of these peaks, we extract the $\{AoA, AoT, RSS\}$ tuples and apply 1-nn (1 nearest neighbor) [93] based classification to group the tuples along the trajectory to individual surfaces. Currently our classification uses equally weighted AoT and AoA, but not RSS. This is because the captured reflection signal per object can change from specular to diffuse reflection along the trajectory, creating large RSS variations that disrupt the classification. We leave further optimization of the classification to future work.

Navigation for Imaging

As mentioned earlier, a Ulysses device needs to navigate in an (unknown) environment safely to capture specular reflections off objects. The navigation design leverages rich literatures on robotic navigation/mapping to move within the unknown environment (*e.g.*, [24, 25, 26, 91]), and instead focuses on enabling navigation using the on-board 60GHz beamforming radios rather than sensors like sonar, Lidar/laser and camera that were used in conventional navigation systems. Ulysses' key contributions include methods to define safety zones for navigation and to schedule trajectory for imaging once an object becomes "visible".

Defining safety zones. A key input to robotic navigation is the *safety zone*. Defined with respect to the device's current location, it allows the device to move freely

without bumping into objects [91]. While prior works compute safety zones using sensors like sonar and camera, we are the first to define them using just 60GHz beamforming radios.

1) Safety zone when no reflection is seen. While a Ulysses device has limited view on specular reflection, it will still capture (weak) diffuse reflections when in close proximity of an objection. Thus at locations where no reflection (above the noise floor) is seen, we build the safety zone as a circle whose radius γ is the *minimum* range that the device can capture any reflection in any beam direction from any object of reasonable size. That is, we determine γ based on the object that leads to the heaviest reflection loss.

We take an empirical approach to determine γ based on the following condition:

$$\frac{P_{TX}G_{TX}G_{RX}}{L_{path}(\gamma)L_{shape}L_{material}} = \text{noise floor} \tag{2.6}$$

where L_{path}, L_{shape} and $L_{material}$ are the path loss, the reflection loss due to shape and material, respectively. We compute γ by finding the heaviest L_{shape} and $L_{material}$ from any object. For L_{shape}, prior works have shown that for any given object, the sharp edge of the object leads to the weakest reflection, referred to as the wedge diffraction effect [94, 95]. Using testbed measurements on many household objects of different shapes and materials, we empirically validated this claim and found that L_{shape} is bounded by 24dB. For $L_{material}$ we use the widely known table of material vs. 60GHz reflection loss [34] and set it to $19.3dB$. This is the reflection loss of wooden objects which peaks among common household objects. We also manually measured other materials like leather and a deck of paper sheets, and found that they are no more than 19.3dB. Given the L_{shape} and $L_{material}$, we use Equation (2.6) to derive $\gamma \approx 1m$.

2) Safety zone when observing reflection. In this case, RX will observe reflection signals (above the noise floor) at some beam directions. At the beam directions where

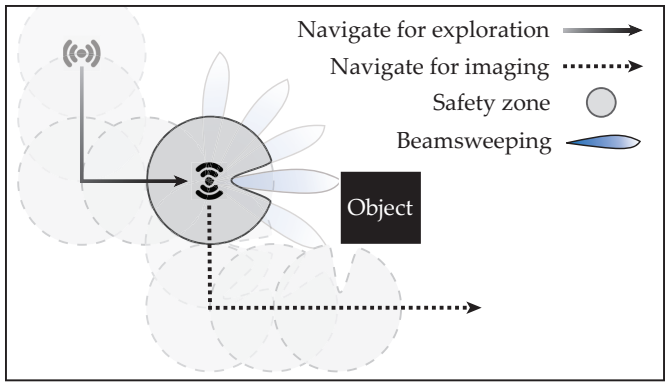


Figure 2.24: An illustration of Ulysses's navigation path to image an object and the corresponding safety zones.

reflection is absent, the safety zone is a round segment with $1m$ radius. At the beam directions where reflection is seen, the safety zone is a region between the current device and the object(s) since 60GHz waves cannot penetrate objects. Thus we approximate the safety zone as a round segment whose radius is the smallest separation between the device and the object(s) that leads to the observed RSS value. For this we reuse Equation (2.6) but replace the noise floor with the observed RSS.

One special case is that when an object is less than 1m away, the diffuse reflection signal becomes sufficiently strong and can be reliably captured at many RX beam directions. They can be utilized along with the specular reflection to pinpoint the object surface, *e.g.*, intersecting the AoAs and AoTs at many TX/RX beam directions. This is particularly useful when a robot rotates itself and suddenly faces an obstacle in close proximity and needs to avoid them.

Planning trajectory for imaging. When detecting reflection signals, a Ulysses device will move within its safety zone in a direction that is perpendicular to the normal line $\frac{AoA+AoT}{2}$. This not only allows Ulysses to identify the type of reflection (specular vs. diffuse) but also puts the device on an efficient trajectory for imaging (under specular reflection). Upon detecting diffuse reflection, the device will explore in other directions.

(a) Testbed (b) 3 Indoor and 1 Outdoor Environments (c) Objects

Figure 2.25: Our testbed prototype, evaluation environments, and experimental objects.

While traveling, a Ulysses device will periodically recalibrate the safety zone computation and adjust trajectory if necessary. In particular, when the trajectory coverage passes the edge of the object, it will observe a significant continuous drop of signal strength due to the aforementioned wedge diffraction effect. In this case, the device will re-compute the safety zone and rotate by 90° (or the closest value defined by the safety zone) to go around the object. Figure 2.24 illustrates how a device navigates to discover and image an object and the corresponding safety zones along the entire path. During the exploration segment of the trajectory, the safety zone is a full circle of 1m radius, which reduces into a partial circle when the device starts to image the object.

Finally, when multiple objects are present, Ulysses will choose a direction in the safety zone that is the average of the "optimal" trajectories across the objects, or even weighted by their RSS values. In some cases, it will image objects sequentially, sorted by their RSS values.

Avoiding walls. During navigation, the device will likely capture the reflection of a wall and attempt to image it by moving along it. This can be minimized based on the intuition that the wall is much larger than our target objects. Thus Ulysses includes a wall-avoidance feature that if enabled, will stop imaging an object if the detected shape is flat and more than $1m$ in width.

2.3.4 Implementation

Proof-of-concept hardware. We build a Ulysses prototype by placing two 60GHz radios on a robot car as the colocated TX and RX (see Figure 2.25(a)). Currently the two small 60GHz antennas are hard-attached to two wooden boxes so the prototype appears bulky. Yet it emulates a compact, mobile imaging robot where the arrays are placed on the robot front or top. We also do not use any software/hardware to synchronize the two radios.

Our robot car is from Nexus Robot [96]. We control its movement using the on-board Arduino chip (with a maximum speed of $1m/s$). The robot rotates at deg-level accuracy, thus we configure the robot to move in straight lines and avoid sub-deg-level rotation. Most of our rotations are 90° and we check the need for rotation every 0.5m. The trajectory error is random but bounded by $10cm$.

The 60GHz radios were donated by Facebook's Terragraph project [97] and follow the 802.11ad standard[27]. The Effective Isotropic Radiated Power (EIRP) is 32dBm, well-below the FCC regulation limit of 82dBm [100]. Each radio has an 8×16 rectangular phased array (6° horizontal and 12° vertical beamwidth), and is electronically steerable in the horizontal direction at a granularity of 1.5°. Each round of beamforming sweeps a 90° range (left and right 45°), and takes 1.6s due to the $0.4ms$ beam switching delay. To meet the real-time requirement (0.4s per measurement round for our robot car), we reduce the sweeping coverage from 90° to 45°. Note that the beam switching delay will be much smaller for production hardware, *e.g.*, 50ns in [73].

The imaging range depends on the surface material. For rough wood (>12dB loss), the horizontal imaging range is more than 10m. The vertical range depends on the vertical beam coverage (since our radios are mounted at a fixed height). Figure 2.26

[27]Facebook recently showcased similar devices at the Embedded Linux Conference [98] and deployed them in Downtown San Jose [99]. We hope that these devices will become commercially available soon.

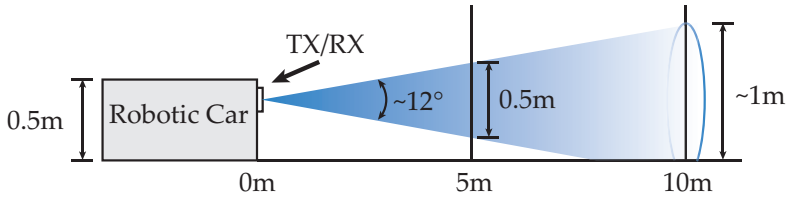


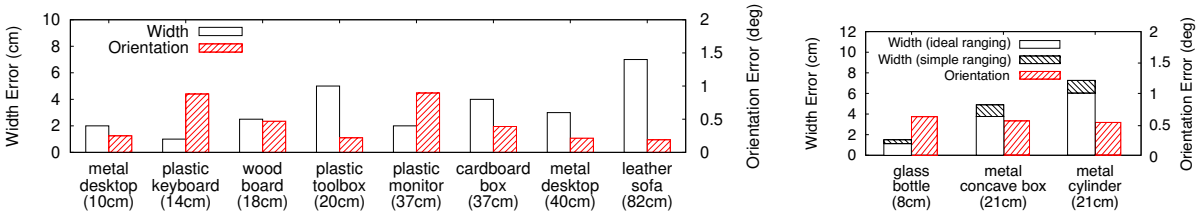
Figure 2.26: Since the TX/RX has a fixed height, the vertical imaging range depends on the vertical beam coverage. At 10m distance, any object placed within the 1m beam coverage can still be observed.

plots the vertical coverage of our array: 1m when the object is 10m away and 0.5m when 5m away.

Navigation. We configure the robot to move at a speed of 2.5cm/s and performs beamforming sweep (which finishes within 0.4s) every 1cm. While moving, the robot keeps track of its location, scans the surroundings via beamforming, and updates the planned path and its safety area. During bootstrapping, we use an exploration algorithm known to the robotic community [26]. Within the safety zone, it searches for the local RSS maxima and terminates when the device detects RSS above the noise floor.

Identifying specular reflection. As discussed in §2.3.3, Ulysses examines the measured RSS over space to identify the type of reflection (specular vs. diffuse). Our measurements have shown that when the reflection transitions between specular and diffuse dominated scenarios, one in general observes a gradual, consistent RSS change by 20dB over ∼10cm moving distance. Thus we empirically choose a 2dB threshold between any two consecutive to detect such continuous RSS change. This works well across all our experiments on many objects and environments.

Complexity. We implement all the computation in Matlab, running on a 2013 MacBook Pro laptop (2.4 GHz Intel Core i7 CPU and 8GB RAM). After collecting reflection measurements along a path segment, the imaging computation takes 0.5s–1s per meter for the given path. This can be further optimized by using a more efficient



(a) Planar Surfaces (b) Curved Surfaces

Figure 2.27: Object imaging benchmarks for (a) planar surfaces (5m away) and (b) curved surfaces (3m away). Overall, we achieve $< 8cm$ error in width and $< 1°$ error in orientation. We later show that when imaging the entire object, the per-surface error will reduce after we assemble different surfaces.

implementation, which we leave to future work. The navigation computation is nearly instantaneous.

2.3.5 Evaluation

In this section, we use real-life experiments to evaluate our Ulysses prototype, focusing on imaging accuracy, navigation efficiency and safety. We also compare Ulysses to camera based imaging [15] and dual-device 60GHz mobile imaging (§2.2).

Experiment configuration. We first perform experiments in three indoor environments (Figure 2.25(b)): a building corridor (of size $2.5m \times 50m$), a classroom (of size $8m \times 12m$) with randomly placed chairs as obstacles, and a standard office reception area ($5m \times 5m$) with leather seatings. We place one or multiple objects in these environments, some as target objects, some as obstacles. In total, we experiment with 11 household objects of various sizes (8–82cm in width), surface shapes (flat, convex, concave, complex), and materials (metallic, plastic, wood, glass, cardboard, leather). Figure 2.25 shows their physical pictures.

By default, we place our robot car based prototype either near the entry door of the classroom and the office or at the center of the corridor. Since our prototype cannot vary

the height of the TX/RX, we instrument the system to focus on imaging objects in the ground level. When placing the objects, we vary their locations and orientations to the starting location of the robot car. We change the initial orientation of the robot to create different first-views of the environment, such as a door, walls, obstacles, target objects or nothing. This allows us to test our design under different startup conditions. We also vary the separation between the colocated TX and RX from 25cm to 50cm (we are unable to go below 25cm due to the hardware case constraint). We verify via experiments that in this range, the amount of separation has minimum impact since they all allow the system to identify the normal line efficiently. Thus we only show the results for 40cm separation for brevity.

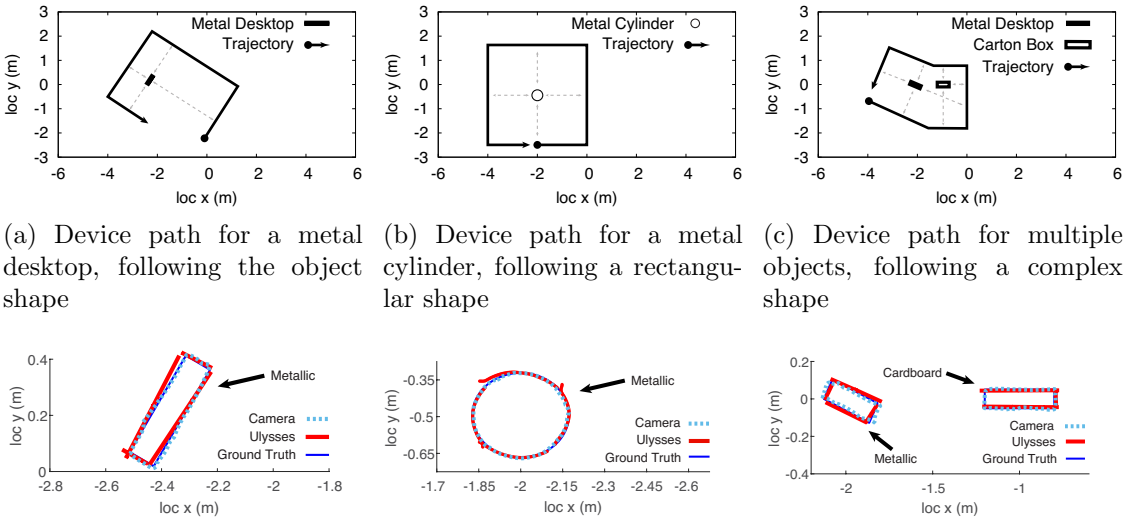
We next experiment with Ulysses in an outdoor parking lot to image the back of parked cars (Figure 2.25(b)). We set the prototype on top of a mobile cart to emulate a vehicle, and move the car at a slow speed to compensate the beam switching delay.

All of our results are produced under (uncontrolled) device movement errors. While programmed to move in a straight line, the trajectory deviation is random and can reach 10cm. The orientation deviation is bounded by 1° per run.

Imaging Accuracy

Our imaging system outputs the shape, orientation, and material of the target object surface. We quantify the accuracy by the absolute error in each metric.

Imaging a surface via a straight line. Consider a simple scenario where Ulysses images a specific object surface by traveling on a straight line. In each experiment, we place an object in the center of the room and program the robot to move in a straight line. Due to random trajectory errors and measurement noises, the imaging outcome varies slightly across multiple runs. We report the median value over 6 runs per configuration. For sensitivity analysis, we vary the object to robot distance between 1m and 5m, and



(a) Device path for a metal desktop, following the object shape

(b) Device path for a metal cylinder, following a rectangular shape

(c) Device path for multiple objects, following a complex shape

(d) Imaging result of a metal desktop

(e) Imaging result of a metal cylinder

(f) Imaging result of multiple objects

Figure 2.28: Ulysses navigates around the object and images the target(s) accurately.

the object surface to trajectory angle between 0 and 10° (so that the robot can capture specular reflection).

For flat (or planar) surfaces, the estimates of surface width and orientation are not affected by the object-to-device distance and orientation. Thus Figure 2.27 (a) shows the results for the 9 planar surfaces that are 5m away from the robot. The median width error ranges between 2–5cm except for the large ottoman (it is of 82cm in width and the error is 7cm). The surface orientation error is bounded by 1°.

Our system can always identify the curvature type, i.e. planar, convex and concave. For curved surfaces, the orientation error is always bounded by 1° across all the configurations. Yet the accuracy of width estimation depends on the object-to-device distance. As discussed earlier, we need to estimate the location of surface center in order to resize the shape estimate. Since the accuracy of our current ranging method decreases with the object distance, the error in the ranging result propagates into the width estimation. Figure 2.27 (b) shows the width and orientation errors for the three curved objects that

are 3m away from the robot. With our simple ranging method, the median width error of the metal trash can (convex) is 7cm, which increases to 8.8cm when the object-to-device distance is 5m. But by improving the ranging accuracy to $< 10cm$, the width error at both 3m and 5m will reduce to 6cm and 6.7cm, respectively.

Finally, our material estimation is accurate. Across our experiments, the ground-truth material always falls in the top-3 candidates provided by our imaging algorithm. This level of accuracy aligns with prior works that must place TX/RX on multiple devices [27, 50].

Imaging objects via navigation. Next we consider practical scenarios where the robot navigates around the object(s) to image them. For this we consider both single object and multiple object scenarios. Figure 2.28 shows three examples of our real-time navigation for imaging and the corresponding imaging results. Figure 2.28(a) shows that to image a metal desktop (of 40cm in width), the Ulysses device takes a rectangular trajectory following the shape of the object. This is because when reaching the edge of the object, detected by observing a significant and consistent drop of signals, the device will rotate 90° to circle around the corner. Figure 2.28(d) shows that the estimated object is almost a duplicate of the ground truth ($< 3cm$ error in width). Another example in Figure 2.28(b) shows the navigation path around a circular object (the trash can) assuming the robot car can only rotate by 90°. In this case, the trajectory is rectangular, and does not follow the object surface shape. Yet the estimated object still closely aligns with the ground truth.

Figure 2.28(c) shows the scenario where our Ulysses device images two objects simultaneously. The corresponding trajectory planned by our system has a more complex shape, as the device rotates its moving direction slightly after identifying a different surface. The imaging result is accurate except it misses one side of carton box. This is because the corresponding reflection from this surface is blocked by the metal desktop.

Error Type	single-object		multi-object	
	median	max	median	max
Surface boundary	2cm	5cm	2.5cm	4cm
Curvature radius	3.5cm	5cm	-	-
Orientation	0.47°	0.86°	0.58°	0.98°
Object center	2cm	7cm	1.5cm	4cm

Table 2.6: Overall imaging errors under single- and multi-object scenarios, when the device navigates around the object to image the entire object.

Finally, Table 2.6 summarizes the distribution of the object imaging errors (surface boundaries, orientation, and object center location, and curvature radius for circular objects) across all of our experiments (using our simple ranging method). The maximum errors are bounded by 5cm for surface boundaries, 1° for orientation, and 7cm for object center location, while the median errors are bounded by 2.5cm, 0.58°, and 2cm, respectively. One interesting observation is that when we assemble the estimated surface segments to construct the entire object, the inherent geometry dependence across them also helps to correct the imaging error on individual surfaces. For example, the estimated surface segments for the trash can that is 5m away can have width errors up to 12cm, which reduce to below 5cm after assembly.

Impact of measurement granularity. Since each beamforming sweep takes 400ms, the measurement granularity (on the trajectory) varies with the robot speed. So far we configure the robot to move at 2.5cm/s, mapping to one measurement per 1cm. We then increase the speed to 5cm/s, mapping to one measurement per 2cm or half the granularity. In this case, the imaging results do degrade slightly, *i.e.* $< 2cm$ error in size estimation. However, this is only a limitation to our current hardware. Since commercial 60GHz chipsets will support a significantly faster beamforming sweep ($50ns/beam$ [73]), this will no longer be an issue.

Ulysses vs. RSA and camera-based imaging. We also compare Ulysses to

RSA, the dual-device 60GHz imaging system, and the camera-based imaging system. First, we implement the RSA algorithm on our platform, following the same scenarios of Figure 2.27, except that we place TX on a separated device 4m away from the object and well-separated from RX. Ulysses and RSA provide similar imaging results ($< 5cm$ error in width, and $< 1^\circ$ error in surface orientation) thanks to 60GHz's directionality. But Ulysses outperforms RSA by using a single device, thus greatly simplifying navigation and eliminating device coordination overhead.

Second, we use a smartphone app called `123D Catch` [15], a popular camera-based imaging solution from Autodesk. It captures a series of photos, analyzes them on the cloud, and reconstructs the captured environment. Like Ulysses, this is a single-device imaging solution. Since `123D Catch` does not offer navigation, we set the smartphone to follow the same navigation paths of Ulysses. Figure 2.28 compares the imaging results of both systems, which are very close to each other. But when we turn off the light, or use it in a foggy day, `123D Catch` fails completely and Ulysses is not affected. We also noticed that the camera app takes minutes and even hours to produce imaging results. Instead, Ulysses is of low computational cost and runs in real-time (<1s).

Outdoor results. We also evaluate Ulysses in an outdoor parking lot, with the goal of imaging the back of parked cars to identify shape, size and material. Our results are promising, indicating cm-level accuracy in this scenario. Figure 2.29 shows an example imaging result of a parked car, where Ulysses correctly identifies the shape, size (to the cm-level accuracy) as well as the surface material.

Safe and Effective Navigation

Next we evaluate the safety and efficiency of our navigation design using 60GHz beamforming. We run experiments in three indoor environments with obstacles and objects on the floor. The device has zero start-up knowledge of the environment and

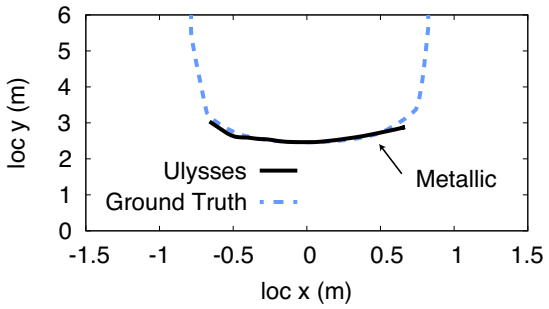
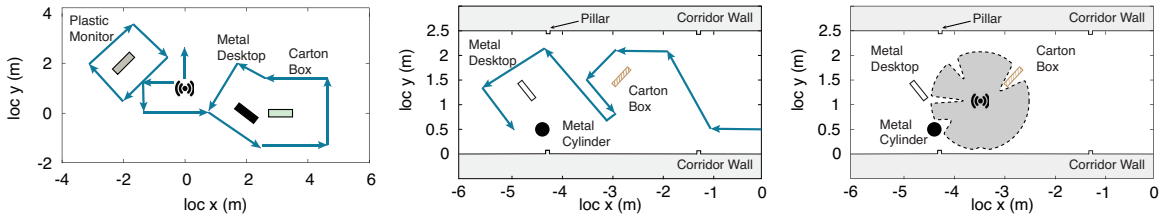


Figure 2.29: Imaging result of the back of a parked car.



(a) Path in the classroom (b) Path in the corridor (c) Estimated safety zone

Figure 2.30: Ulysses navigates in (a) the classroom and (b) the corridor without colliding into objects/walls. In (c) we plot an example of the estimated safety zone (the shaded region) at a specific location.

cannot "see" the target object initially. In each experiment, we place three objects with 1m to 5m distance from each other, and perform 10 experiments per indoor environments. Across all 30 experiments, our prototype successfully navigates around all the obstacles and objects without any collision.

Figure 2.30(a) illustrates an example of the overall navigation path. The robot car starts in the center of the classroom facing away from all three objects, and seeks to "find the carton box". It first explores local areas in four directions within the 1m safety circle, and detects reflection in one direction and then adjusts its trajectory to go around the object to image it. The device then identifies the imaged object as a plastic monitor. This is not its target, thus the robot moves to discover and image the other two objects together.

Figure 2.30(b) shows another example in the corridor, a smaller environment due to

the walls on the side. The robot's mission is to find the "metal desktop". At first the robot sees the wall reflection and attempts to image it. After moving 1m and finding the object is at least 1m in width, the robot gives up on that. It then finds two reflections and decides to image the one with stronger RSS. When approaching the wall, it turns left to avoid collision. After identifying the carton box, it senses new reflection signals and moves to image the other two object and locates the target.

Accuracy of safety zones. We also measure the safety zone at individual locations and confirm that they all do not overlap with any object or obstacle. Figure 2.30(c) shows a specific example in the narrow corridor of 2.5m in width where the device is in the middle of three objects. Although being conservative, the safety zone accurately captures the impact of the walls, the objects, and even the small pillars on the wall (5cm in width). As future work, we plan to compare our 60GHz based safety zone estimations to those using sonar and/or camera [24, 25, 101, 102].

2.3.6 Limitations and Future Work

Handling large trajectory errors. Operating on angular and RSS measurements of reflections, Ulysses is robust against moderate trajectory tracking errors (our robot achieves $< 10cm$ deviations over straight lines). In this work we limited our experiments to 2D movements where the trajectory tracking error is moderate. Under 3D movements, *e.g.*, drone flying, the tracking error might be (much) larger, leading to noisy angular and RSS measurements, thus affecting imaging and navigation performance.

A potential solution is to integrate SLAM into Ulysses [40, 103], which iteratively estimates the robot's current movement and the corresponding imaging outcome, using prior trajectory data and imaging results.

Handling device rotation errors. Moving in straight lines, our robot makes small

rotation error ($< 1°$). Thus in our experiments, we did not observe any impact on Ulysses imaging and navigation. But under larger rotation errors, the accuracy of AoT and AoA computation may be affected. We plan to empirically examine this potential artifact, and improve our design if necessary.

Handling multipath reflection. 60GHz's high directionality and colocation of RX/TX effectively limit the chance that RX captures multiple reflections in a single beam. Yet, this can happen if two objects are placed in close proximity and have the same orientation with respect to the Ulysses device. Separating these signals requires very high channel sampling rate which is costly and hard to achieve.

Multipath reflection will have much less impact on imaging than navigation, because multipath is location-dependent and will appear on very few locations along a trajectory. One can potentially identify these "noisy" locations and compensate accordingly. Navigation requires computing safety zone for each location, thus the impact of multipath will be more visible. As multipath is location-specific (within a few cms), one potential solution is to derive per-location safety zones by integrating measurements at nearby locations.

Duty cycling 60GHz radios. Currently Ulysses assumes 60GHz radios are always available for imaging. In practice, we need to dutycycle/schedule imaging tasks on 60GHz radios to reduce energy consumption and/or allow radios to perform necessary communication tasks. We plan to study the tradeoff between imaging accuracy and the amount of 60GHz radio usage, with and without communication tasks.

Improving image resolution. Our current prototype achieves an image resolution of a few cms. We think this is partially due to the beam width, the beam steering accuracy and granularity of our 60GHz prototype, and the lack of precise timing information from the radios. Moving forward, we expect imaging accuracy to improve with better

hardware availability and better software access to radio data. Finally, Ulysses does not assume TX and RX are tightly synchronized. We plan to study whether adding TX/RX synchronization can help improve imaging performance.

2.3.7 Related Work

Reflection-based RF imaging and tracking. Recent works have leveraged signal reflection to perform imaging and target tracking in both WiFi and 60GHz bands. In the WiFi bands, researchers have used commodity WiFi chips [85, 53, 64, 104] or specialized FMCW hardware [13, 14, 12] to localize/image static objects, or measure human body dynamics as well as hand/finger motions. Others use commodity WiFi radios to recognize predefined hand gestures [47, 105], often leverage machine learning methods to distinguish different gestures and motions.

Existing efforts in 60GHz applied radar design to achieve precise object imaging and motion tracking of small targets. [88] uses FMCW hardware and applies SAR with sparse measurements in absence of device movement noises, while [89] uses three separate, static 60GHz radios to track subtle pen movements on a tablet. Both designs assume short object-device distance (30–50cm), rely on phase and do not face any device movement noise. Another direction is to use two mobile 60GHz radios to image static objects meters away at cm-level accuracy [50, 27], focusing on being robust to device path noises. Finally, the most relevant mmWave handheld imager is *Walleye* [7], which uses specialized hardware, costs $500, and weighs 5lb.

Acoustic-based tracking and imaging. Recent works use smartphone's speaker-mic pair to localize targets with cm- and mm-level accuracy [106, 107], while another develops new measuring methods to image objects using an audible frequency [108]. These acoustic systems are very appealing in short distance scenarios, but are sensitive

to environmental noises. On the other hand, today's acoustic imaging products use ultrasound and are costly, *e.g.*, $45K [6].

LIDAR and vision-based solutions. Today, the state of the art in mobile imaging products is LIDAR, used by Google self-driving cars [10], with costs up to $75,000. Low-cost LIDAR prototypes are in the works, but must sacrifice range, precision and coverage (from 2D imaging to 1D). To our best knowledge, the cheapest version [9] still costs $250 per device, providing 1D imaging with 10s of centimeter precision; another recent version [11] improves precision to a few cms but costs $400.

Another widely studied area is vision-based solutions that use commodity cameras, *e.g.*, the commercial app 123D Catch [15], which we compare with in this work. As discussed earlier, these solutions face the challenge of being sensitive to lighting conditions and not being able to distinguish objects of similar colors.

In terms of imaging techniques, existing vision-based systems assume a stationary lighting source (TX) like [109, 110]. Using TX's location as a reference, these systems construct images of objects by exploiting either special illumination patterns or the image correlation of multiple reflection points observed at various locations. Our system uses a similar spatial correlation based approach, but differs by leveraging the directional 60GHz signals on a moving TX. One of our key contributions is the geometric method that integrates a sequence of observations (and spatial patterns) collected under a moving TX source.

Robotic navigation. Ulysses leverages the rich literature on robotic navigation in unknown environments (*e.g.*, [24, 25, 26, 91]). Our (new) contribution here is to define safety zones using the onboard 60GHz beamforming radios rather than sonar and camera [24, 25, 101, 102].

2.3.8 Summary

This section describes our experiences in designing, implementing and evaluating an object imaging system for mobile devices using commodity 60GHz radios. Experiments on our prototype validate the feasibility of our colocated radio design, and confirm that our imaging and navigation algorithms can leverage onboard 60GHz radios for robust and accurate results. While the current prototype is limited by hardware constraints, we believe our results show significant promise for a low-cost, compact, single-device imaging system. We hope this work and followups will play a role in autonomous devices in the near future.

Chapter 3

Silent Reconnaissance Attacks and Defenses

Wireless devices are everywhere, constantly bombarding us with transmissions across a wide range of RF frequencies. Many of these invisible transmissions reflect off our bodies, carrying off information about our location, movement, and other physiological properties. While a boon to professionals with carefully calibrated instruments, they may also be revealing our physical status to potential attackers nearby.

Our work (§3.1) demonstrates a new set of silent reconnaissance attacks that leverages the presence of commodity WiFi devices to track users inside private homes and offices, without compromising any WiFi network, data packets, or devices. We show that just by sniffing existing WiFi signals, an adversary can accurately detect and track movements of users inside a building. This is made possible by our new signal model that links together human motion near WiFi transmitters and variance of multipath signal propagation seen by the attacker sniffer outside of the property. The resulting attacks are cheap, highly effective, and yet difficult to detect. We implement the attack using a single commodity smartphone with a WiFi networking radio and only a single antenna. We deploy it in 11

real-world offices and residential apartments, and show it is highly effective. Finally, we evaluate the potential defenses, and propose a practical and effective defense based on AP signal obfuscation (§3.2).

3.1 Adversarial Sensing Under Ambient WiFi

With near-ubiquitous deployment of WiFi-enabled smart devices (*e.g.*, security cameras, voice assistants, and smart appliances), our homes and offices are filled with many WiFi devices[1]. The ubiquity of these devices and their sheer density means that they will fill the air around us with radio frequency (RF) signals, wherever we go.

Unfortunately, the RF signals emitted by these devices pose a real security and privacy risk to all of us. They are constantly interacting with (*e.g.*, reflecting off) our bodies, carrying information about our location, movement and other physiological properties to anyone nearby with sufficient knowledge and curiosity. In this work, we explore a new set of passive reconnaissance attacks that leverages the presence of *ambient WiFi signals* to *silently* track users in their homes and offices, even when the WiFi network, data packets, and individual devices are completely secured and operating as expected. We show that by just sniffing existing WiFi signals, an adversary outside of the target property can accurately detect and track movements of any users down to their individual rooms, regardless of whether they are carrying any networked devices.

We believe this is the first in a new class of silent reconnaissance attacks that are notable because of their passive nature and general applicability. This attack can be highly effective, incurs low cost (only cheap commodity hardware), and yet remains *undetectable*. The attacker does not need to compromise/access the WiFi network or individual devices, decode packets or transmit any signals. All they need is to place

[1]The worldwide number of WiFi-enabled IoT devices is expected to reach 5 billion by 2025 [111], and the number of WiFi connected devices will reach 22.2 billion by 2021 [112].

a single, off-the-shelf, minimally equipped WiFi receiver outside of the target property. More importantly, this attacker receiver only needs a *single* antenna, and simply measures the signal strength of existing WiFi signals, without even decoding any packets.

Unaddressed, these reconnaissance attacks put our security and privacy at significant risk. The ability for an attacker to continuously and automatically scan, detect and locate humans behind walls at nearly no cost and zero risk (*e.g.* attacker waits for notifications remotely) will enable attackers to launch strong physical attacks and commit serious crimes. Such threat broadly applies to our homes, businesses, factories, government facilities and many others. Examples include burglary to homes and offices, kidnapping and assault of targets in their homes, "casing" a bank prior to robbery, and even planning attacks against government agencies.

Why WiFi sensing? We note that there are some simple approaches to inferring user presence that do not require the use of sophisticated RF sensing. For example, attackers could infer user presence by observing lighting or acoustic conditions inside an area, or use thermal imaging. These attacks are well understood and easily disrupted by time-controlled lighting or sound systems [113], or insulated walls designed to prevent heat leakage and naturally block thermal imaging [114]. Finally, attackers can infer user presence from increased WiFi network traffic. Yet this is highly unreliable, as growth of IoT devices increases traffic levels in the absence of users. It is also easily thwarted using cover traffic [115]

Instead, we describe a new class of physical reconnaissance attacks enabled by inherent properties of WiFi signal propagation: 1) user movement near a WiFi transmitter changes its signal propagation in a way that can be observed by nearby receivers, and 2) walls and buildings today are not built to insulate against WiFi signals, thus signals sent by devices inside a property can often be overheard by outside receivers. Leveraging these, we design the attack such that, whenever a WiFi device transmits, it unknowingly

turns into a tracking device for our attack. In this context, our attack could be viewed as an adversarial analogy to WiFi-based device-free human sensing (*e.g.*, see-through-wall systems that actively transmit customized RF signals towards the target [44]). Yet our attack differs significantly (§3.1.1), because we use a novel model on multipath signal dynamics to remove dependence on active transmissions (only passive sensing), customized hardware (only a commodity, single antenna receiver), and knowing precise locations of WiFi devices inside the property.

Motion detection via multipath signal dynamics. The core of our attack design is a new model on signal dynamics that links together human motion near WiFi transmitters and variance of multipath signal propagation seen by a sniffer outside of the property. Specifically, when a human target makes a movement (*e.g.*, sitting down, walking, opening/closing doors) near a WiFi device x, the motion changes the multipath signal propagation from x to the attacker sniffer S. Our new signal model allows S to accurately capture such signal dynamics and use them to pinpoint the target to her specific room. The more WiFi devices inside the property, the more accurate the tracking becomes.

Our proposed attack does not assume any prior knowledge of the WiFi network and devices inside the target property, including their locations. Our attack can discover devices and estimate their coarse locations using their WiFi signals, and the attack continues to function even after any of these devices are relocated.

We build a complete prototype of the attacker system on commodity smartphones, and experimentally show that the attack (using a single smartphone) is not only highly accurate (detecting and localizing users to an individual room), but also highly general (effective across a wide range of 11 different physical settings, including both office buildings and residential apartments).

Defense via AP-based signal obfuscation. We explore robust defenses against our proposed attack and other passive sensing attacks. We consider four immediate defenses: reducing leakage by geo-fencing and rate limiting, signal obfuscation by MAC randomization, and power randomization at WiFi devices, and find that they are either impractical or ineffective. We then propose a practical alternative using *AP-based signal obfuscation*, where the WiFi Access Point actively injects customized cover signal for its associated devices. This defense effectively creates noise to the signal measurements, such that the attacker is unable to identify change due to human motion. Our defense is easy to implement, incurs no changes to devices other than the AP, but reduces the human detection rate to 47% while increasing the false positive rate to 50%. Such ambiguity renders the attack useless in practice.

In the rest of the section, we describe our efforts to understand the feasibility, challenges, and defenses surrounding the proposed attack. In short, our key contributions include:

- We identify a low-cost, undetectable human sensing attack using just a single sniffer with a single antenna, and design a new multipath signal variance model for motion detection.
- We prototype the attacker system on a commodity smartphone and validate the attack in real-world settings.
- We propose and evaluate a practical and effective defense against such attacks using AP-based signal obfuscation.

Limitations. Currently, our attack detects human presence in each room over time by detecting and localizing targets to individual rooms. It is unable to identify fine-grained features such as speed, activity type and user identity, or separate humans from

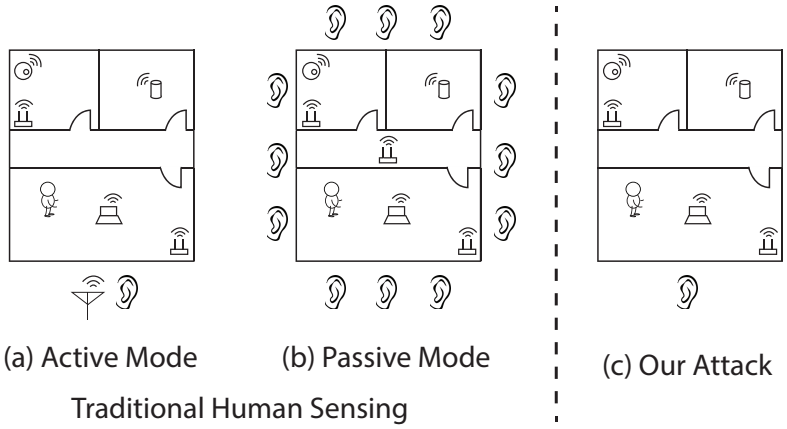


Figure 3.1: Traditional human sensing designs either (a) relies on active transmissions by (customized) attacker devices, or (b) deploys one or more advanced sniffers (laptops/USRPs) with multiple antennas; (c) Our attack uses a single smartphone (with a single antenna) as the passive sniffer, and turns commodity WiFi devices inside the property as motion sensors.

large animals. Despite such limitations, our work identifies a realistic, low-cost, and undetectable reconnaissance attack using passive WiFi sensing. We hope our work brings more attention to this important and understudied topic.

3.1.1 Background: Device-free Human Sensing

Some details of adversarial sensing attacks are reminiscent of the problem of "device-free human sensing." A natural question is: *can we simply reuse existing work on device-free human sensing systems to launch adversarial sensing attacks*? To answer this question, and to better understand how these attacks in the context of prior work, we review in detail existing works in device-free human sensing.

The task of "device-free human sensing" makes no assumptions on whether targets are carrying networked devices. Sensing is generally achieved by capturing and analyzing RF signals reflected off or blocked by human bodies. To be clear, this is quite different from the task of "device localization," in which the target is a networked device that communicates and synchronizes with the sensing system, *i.e.* sending and/or receiving

signals (*e.g.*, [116, 117, 118, 119, 120]).

Existing works on device-free human sensing can be categorized into two broad groups: *active mode* and *passive mode*.

Active sensing. Most of the existing works fall into this group, where the sensing device continuously transmits RF signals towards the target area (Figure 3.1a). As some signals get reflected off the target body, they are captured by the sensing device to infer the target status (*e.g.*, [44, 14, 121, 122]). To facilitate sensing/inference, the RF signals are often custom-designed to capture information of the target, *e.g.*, frequency-modulated continuous wave (FMCW) signal [44, 14] that largely differs from WiFi transmissions. We note that some prior works on active sensing (*e.g.*, [123, 124, 125]) are branded as "passive sensing" to refer to device-free human sensing, although their sensing device is actively transmitting signals.

When considering our adversarial scenario in the context of active sensing, the key property is "detectability." Since the attacker device must *continuously* transmit RF signals, it is easy to detect, localize and remove these devices.

Passive sensing. In a passive mode, sensing devices only listen to existing transmission signals, but do not transmit signals themselves. They have no control of the RF signal used for sensing. The state-of-the-art design [121] deploys multiple sniffers to listen to WiFi signals sent by multiple transmitters in the target area, and uses these signals to detect and estimate user location. Specifically, when a user blocks the direct line of sight (LoS) path from a transmitter to a sniffer, the original signal will diffuse around the user. By building a precise propagation model on signal diffusion on the LoS path, [121] is able to derive the user location. However, doing so requires precise location of the transmitters (cm-level). Such requirement is impractical under our adversarial scenario.

Similarly, an earlier work [126] detects the presence of user when it disturbs the direct

path between a WiFi access point (AP) and a sniffer. Again, the attacker must obtain AP locations a priori and must deploy multiple sniffers around the target area to detect user presence (see Figure 3.1b).

Key observation. While some existing human sensing systems can be turned into attacks, they impose a hefty cost and risk for the attacker, significantly limiting the applicability of the attack. This motivates us to consider a new, passive human sensing attack that can be launched by a minimally equipped attacker and remains undetectable. Along these lines, our proposed attack only requires a single commodity WiFi receiver (with a single antenna) outside of the target area (Figure 3.1c). As we will explain in §3.1.3, this is made possible by building a new model to detect motion using dynamics of multipath signal propagation from each anchor to the sniffer, rather than those of the direct path as in [121, 126].

3.1.2 Attack Scenario and Adversarial Model

We start by describing the attack scenario, the adversarial model, and the type of signals that can be captured by the attacker sniffer.

Attack scenario: one sniffer and many anchors. As shown in Figure 3.1c, our attack leverages the ubiquity of commodity WiFi devices, ranging from routers, desktops, printers, to IoT devices like voice assistants, security cameras, and smart appliances. These devices are often spread over each room of our homes and offices [127, 128], and generally flood the surroundings with periodic WiFi packets. We refer to these WiFi devices as *anchors* in our attack.

Our attack also leverages the fact that WiFi signals are designed for significant coverage and can penetrate most walls. Thus an attacker can place a sniffer outside the target property to passively listen to existing signals sent by anchors, without compromising

them or the network. Because WiFi protocols do not encrypt source and destination MAC addresses, the sniffer can isolate packets sent by each anchor, even under MAC randomization [129, 130, 131].

Our attack is effective if the sniffer can capture signals from at least one anchor per room of interest. The actual number of sniffers required depends on the size and shape of the target property and wall materials. Across all of our experiments with 11 office buildings, residential apartments and single family houses (described later in §3.1.4), a single sniffer is sufficient to cover our target scene.

Our attack does not work when WiFi signals do not leak to outside of the property, *e.g.*, when the property has thick, concrete exterior walls. The attacker can detect this (and walk away) when either the sniffer sees too little WiFi signals, or the detected anchors are outside of the target area (§3.1.4).

Adversarial model. We make the following assumptions about the adversary.

- The adversary makes no assumptions about the number, location, or movement speed of human targets being tracked.
- The adversary does not have physical or remote access to WiFi devices in the target property, or the property itself.
- Similar to the evil maid attack [132], the attacker can physically move *outside* the target property, either outside exterior walls or along public corridors, without attracting suspicion. This temporary access is necessary only for initial bootstrapping the attack, and not required for the prolonged sensing phase.
- To avoid detection, the attacker only performs passive WiFi sniffing, and avoids using any bulky or specialized hardware, *e.g.*, directional antennas, antenna arrays, or USRP radios [133]. Instead, they use commodity mobile or IoT devices, *e.g.*,

smartphones or smart street lights. The sniffer device only needs a single (built-in) antenna.

Note that while some smartphones (including the ones used in our attack implementation) have multiple antennas, their firmware only exposes aggregate signal received across multiple antennas, effectively giving the same amount of information as devices with a single antenna.

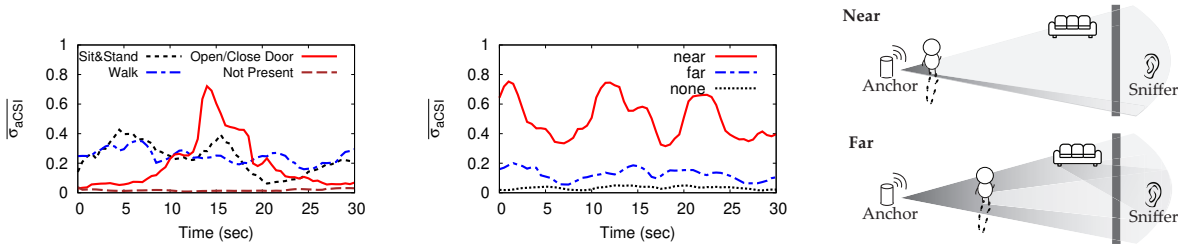
- The adversary partitions the target property into "regions" or virtual rooms around the anchors to detect user presence. When the adversary has access to rough floor plans of the target property[2], the attacker detects user presence down to their individual rooms.

We intentionally choose a resource-limited attacker to demonstrate the generality of this attack. Lower resource requirements imply that the attack can be successful in a wider range of practical scenarios.

Signals captured by the sniffer. For each anchor x, the sniffer S can extract two metrics from its raw WiFi signals (even if the packets are encrypted). The first is *amplitude of channel state information (aCSI)* that measures the received signal strength (RSS) on each of the many frequency subcarriers used in a WiFi transmission. Since human movements change the multipath signal propagation from x to S, x's aCSI values seen by S will fluctuate over time. The second one is RSS, or the mean aCSI value over all the subcarriers. This aggregation makes RSS relatively insensitive to human movements.

It should be noted that a passive sniffer with a single antenna is *unable* to extract advanced signal features including phase of CSI (fCSI), Angle of Arrival (AoA) and Time of Flight (ToF) [134, 124]. Tracking fCSI and ToF requires the sniffer to actively

[2]The rough floor plan can often be derived from publicly available data, thanks to real estate websites and apps, *e.g.*, Zillow and Redfin, and public building documents.



(a) $\overline{\sigma_{aCSI}}$ captures user movement (b) $\overline{\sigma_{aCSI}}$ when the user is near, far from the anchor, or absent (c) Illustration of $\overline{\sigma_{aCSI}}$'s near-far phenomenon

Figure 3.2: Observations on how human movements affect an anchor's $\overline{\sigma_{aCSI}}$ seen by the sniffer. (a) $\overline{\sigma_{aCSI}}$ w/ and w/o user presence; (b)-(c) When a user moves near an anchor x, some signal paths from x to the sniffer are more frequently affected, so $\overline{\sigma_{aCSI}}(x)$ rises. As she moves away from x and has less impact on the signal propagation, $\overline{\sigma_{aCSI}}$ reduces.

synchronize with the transmitter [135], and estimating AoA requires the sniffer to have an antenna array [116, 136]. As mentioned earlier, while some smartphones are equipped with multiple antennas, their firmware only reports a single effective CSI but not per-antenna CSI values. Furthermore, AoA estimation requires antennas to be separated by half a wavelength (12.5cm for WiFi). Thus a reasonable array of 4 antennas will be at least 19cm in width. These physical limitations rule out the feasibility of using phase, ToF and AoA in our sensing design.

3.1.3 Turning WiFi Devices into Motion Sensors

Our attack is driven by a new aCSI variance model that links human motion near any anchor to temporal dynamics of the anchor's multipath signal propagation seen by the attacker sniffer. Whenever an anchor transmits WiFi signals, it unknowingly turns into a motion sensor for our attack. These "motion signals" are immediately seen by the attacker sniffer, who then pinpoints the targets down to their exact room(s).

Unlike prior work on passive RF sensing [121, 126], our new model focuses on captur-

ing temporal dynamics of multipath signal propagation[3] from each anchor to the sniffer, rather than only the direct path. This lets the attacker detect any motion *around* each anchor that disturbs the multipath signal propagation, and also eliminates the need to obtain precise anchor locations and deploy multiple sniffers [121, 126].

In the following, we describe the basic observations that motivate us to pursue the attack, the key challenges it faces, and the key design concepts that make the attack feasible.

Correlation between Signal Dynamics and User Movement

(i) User movement → aCSI variance. In an office/home, human users are never completely stationary. Whether it is playing games, walking, opening doors, sitting down, standing up, the natural movements will disturb the multipath signal propagation of nearby WiFi transmitters (*i.e.* anchors), creating immediate, temporal variations in their aCSI values seen by the attack sniffer.

We propose to capture such temporal variation by a new *aCSI variance* metric:

$$\overline{\sigma_{aCSI}} = \frac{1}{|I_q|} \sum_{i \in I_q} \sigma_{aCSI}(f_i) \tag{3.1}$$

where $\sigma_{aCSI}(f_i)$ represents the aCSI *standard deviation* for subcarrier i (at frequency f_i) calculated by the sniffer over a short time window (*e.g.*, 5s). We also take efforts to reduce the impact of noise and artifacts in aCSI reports by the firmware, first denoising aCSI per subcarrier using the wavelet method [137], then removing outliers by only including subcarriers whose $\sigma_{aCSI}(\cdot)$ sequences are highly correlated. The set of subcarriers used in the above calculation (I_q) are the top 50% of most correlated pairs[4].

[3] WiFi signals emitted from an anchor, when reaching the sniffer, will go through rich multipath propagation, *e.g.*, reflections by furniture, walls and human.

[4] We did not notice any particular patterns for those uncorrelated subcarriers, *e.g.*, if they are due to

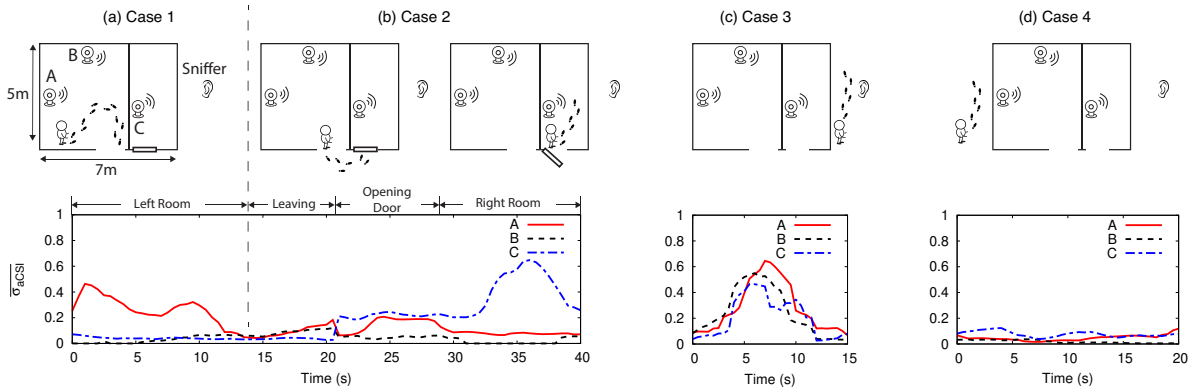


Figure 3.3: Four (simple) cases on user presence and the corresponding $\{\overline{\sigma_{aCSI}}\}$ traces from anchors A, B, and C.

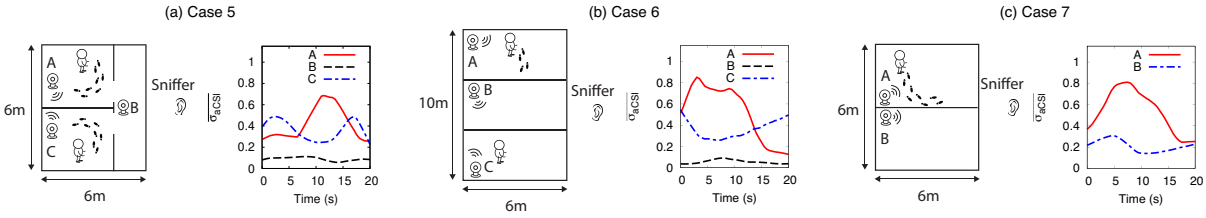


Figure 3.4: Three (complex) cases on user presence and the corresponding $\{\overline{\sigma_{aCSI}}\}$ traces.

Figure 3.2a plots several 30-second samples of an anchor's $\overline{\sigma_{aCSI}}$ seen by the sniffer, for scenarios of no human presence, a nearby user sitting down and standing up, opening/closing the door, and walking. Compared to no human presence, user movements lead to much higher $\overline{\sigma_{aCSI}}$.

We also find that user motion differs from equipment motion commonly seen in homes and offices, *e.g.*, an oscillating fan and a robot vacuum. The latter either is too weak to produce any notable impact on $\overline{\sigma_{aCSI}}$ or generates periodic signal patterns different from those of human motion (§3.1.6).

(ii) $\overline{\sigma_{aCSI}}$ depends on user-anchor distance. Another key observation is that when a target is far away from an anchor x, its movements will produce less disturbance to the signal propagation from x to the sniffer. This is demonstrated in Figure 3.2b,

multipath propagations rather than hardware artifacts. We leave this to future work.

which compares an anchor x's $\overline{\sigma_{aCSI}}$ when a human user (walking) is close to x, far from x (in a different room), or completely absent.

We think this is due to the fact that a target is "bigger" when it is closer (Figure 3.2c). As a target moves in the space between an anchor x and the sniffer, it blocks and diffracts some signal paths from x to the sniffer. When close to x, it affects more paths than when it is far away from x. Thus the received signals seen by the sniffer will display a larger temporal variation when the user is closer to x. This phenomenon can be modeled using an abstract, ray-tracing model on $\overline{\sigma_{aCSI}}$ (details in §A.1). Given a fixed time period, user movements near x create more path dynamics than those far from x, leading to a larger standard deviation in the received signal strength (per subcarrier).

We validate this observation by measuring $\overline{\sigma_{aCSI}}$ of multiple anchors (Table 3.3) in 11 test scenes (Table 3.2). As a target moves in the space between an anchor and the sniffer, we see a general tendency of $\overline{\sigma_{aCSI}}$ decreasing with the anchor-to-target distance. We experiment with different wall materials (*e.g.*, glass, wood), distance of anchor and sniffer (8m–15m), and sniffer placement (*e.g.*, on the floor, in the bush, underneath a plastic trash can), and observe the same trend.

(iii) $\overline{\sigma_{aCSI}}$ is a more robust motion indicator than $aCSI$. Prior work [121] localizes targets by modeling $aCSI$ of the direct path. This requires an accurate propagation model and the precise physical location of each anchor. Instead, our $\overline{\sigma_{aCSI}}$ based method targets multipath dynamics caused by user motion, thus only requires knowing the room each anchor resides, rather than its precise location inside the room.

Challenge: Sensing Ambiguity

The above discussion suggests that with a sufficient number of anchors in a room, the sniffer should be able to detect human motion in the room from its anchors' $\overline{\sigma_{aCSI}}$. For example, if any anchor's $\overline{\sigma_{aCSI}}$ is sufficiently large, *i.e.* motion detected, the room

should be marked as occupied.

But we also find notable ambiguity in such design, caused by two factors. *First*, $\overline{\sigma_{aCSI}}$ depends on the target-anchor distance and the motion pattern/strength. Yet the attacker has no knowledge of target behaviors or previous ground truth. *Second*, short physical distance to an anchor does not always translate into being the same room.

Next, we illustrate the problem of sensing ambiguity using real-world examples, including four basic cases with a single user and three complex cases with multiple users.

Case 1: Target staying in a room. Figure 3.3a shows the traces of $\overline{\sigma_{aCSI}}$ for three anchors: A and B in the left room, and C in the right room. The target user stays inside the left room and moves around anchor A. In this case, anchors B and C show no sign of targets nearby (very low $\overline{\sigma_{aCSI}}$) while anchor A has the largest $\overline{\sigma_{aCSI}}$ over time.

Case 2: Target moving across rooms. Following case 1, the target walks towards the room door (already open) at $t = 12s$, enters hallway at $t = 18s$, starts to open the right room door at $t = 24s$, closes it and enters the room at $t = 28s$. In this case, anchor A's $\overline{\sigma_{aCSI}}$ drops as the target moves away, followed by a short, minor increase due to the opening/closing of the right room door. Anchor C has a short, minor increase in its $\overline{\sigma_{aCSI}}$ due to the door opening/closing, followed by a significant increase as the target moves closer. Interestingly, as the target transitions between the two rooms, we can observe somewhat synchronized changes on anchor A and C (since they are triggered by the same event). But from per-anchor $\overline{\sigma_{aCSI}}$ traces, a naive design will mark both rooms as occupied.

Case 3: Sniffer blocked by external pedestrian. Pedestrians who move outside of the target area near the attack sniffer could also create aCSI variations. Yet such movements (near the common receiver) will create synchronized aCSI variations at all the transmitting anchors, regardless of any human presence. Again a naive design will

mark both rooms as occupied.

Case 4: External users walking around the house. When pedestrians move away from the sniffer, the impact on $\overline{\sigma_{aCSI}}$ is small even when they are close to the anchors (Figure 3.3d). This is because those movements have little impact on the multi-path propagation between the anchors (inside the two rooms) and the sniffer.

Case 5: Multiple users moving in neighboring rooms. Figure 3.4a shows an example where two targets are moving in two different rooms, each with an anchor device. In this case, both anchors (A and C) display large $\overline{\sigma_{aCSI}}$.

Case 6: Multiple users moving in distant rooms. A user walks around in room A when another user sits down near an anchor in room C (Figure 3.4b). We see that room A and C's anchors are triggered, but not the one in room B.

Case 7: Anchors on both sides of a wall. Figure 3.4c shows that when the user moves near anchor A, it triggers both anchor A and B (on the other side of wall). Here the simple design will mark both rooms as occupied (since both anchors are triggered), creating a false positive.

Design Concepts

Our analysis shows that instantaneous $\overline{\sigma_{aCSI}}$ observed at each individual anchor is insufficient to detect and localize user motion. We propose to overcome sensing ambiguity by analyzing the value and pattern of $\overline{\sigma_{aCSI}}$ across both time and anchors. The end result is a robust $\overline{\sigma_{aCSI}}$ model that links each human motion with signal dynamics of anchors in its actual room. Next, we outline the signal analysis process in two steps: 1) *detecting human motion* and 2) *mapping each motion to a room*. The detailed procedures are described later in §3.1.4.

Detecting human motion. If the number of detected anchors per room is reasonable[5], any user movement should "trigger" at least one anchor in the scene. But *how do we determine threshold $\sigma_p(x)$ necessary to trigger an anchor x*? This is not easy, because the adversarial has no ground truth on target presence. Also the threshold should be anchor-specific and could also vary over time.

Leveraging a common observation where a user will not stay and move in a single room forever, we propose to derive $\sigma_p(x)$ by finding "outliers." Assuming for anchor x the sniffer can measure $\overline{\sigma_{aCSI}}(x)$ over a long period of time (*e.g.*, hours or even days), it is reasonable to assume that x is mostly not triggered. Thus the sniffer can apply outlier detection methods like MAD [138, 139] to derive $\sigma_p(x)$ and adapt it over time.

Mapping each motion to a room. When multiple anchors in more than one room are triggered, the sniffer needs to decide whether they are triggered by users in one room (one source) or users in multiple rooms (multiple sources). This is because a target's movement could trigger anchors in neighboring rooms (case 7), and the same holds when multiple users move in two rooms (case 5 and 6). The sniffer needs to distinguish between them and determine the set of rooms that are actually occupied.

Again we leverage a common observation: human movements in different rooms are generally asynchronous, thus anchors triggered by separate sources will display different temporal patterns in $\overline{\sigma_{aCSI}}$ (case 5 and 6). But when a single source triggers anchors in neighboring rooms (case 7), these anchors' $\overline{\sigma_{aCSI}}$ will share a similar pattern. By computing the correlation of normalized $\overline{\sigma_{aCSI}}$ time series across anchors, we can determine whether they are triggered by sources in one room *i.e.* positively correlated. For example, the correlation between the two triggered anchors are -0.07, -0.03, and 0.32, in case 5, 6, and 7, respectively, and 0.23 during the door opening in case 2.

[5]Home/office WiFi devices naturally spread out in a room [128, 127]. One can assume 3-4 devices in a room of common size of $25m^2$.

Our attack can also use the floor plan (or room transition probabilities) to fine-tune the detection result (similar to [140]). For example, a user cannot "fly" from one room to another when the rooms are widely separated. If the sniffer observes two anchors in two widely separated rooms being triggered sequentially with little or no gap, it will report two users, one in each room, rather than a single user moving across rooms.

For other cases, our attack conservatively treats the rooms with at least one anchor triggered as occupied.

3.1.4 Attack Design

After presenting the key concepts, we now present the attack design in detail. As shown in Figure 3.5, the attack includes two phases: (1) identify and locate anchors during "bootstrapping," and (2) deploy the sniffer and perform "continuous human sensing."

(1) Bootstrapping. The attacker first needs to identify and locate the anchors in the target area. The unique feature of our motion detection is that is does not require precise location of anchors, only their individual room. In our attack, this is achieved by the attacker performing a brief passive WiFi measurement (using the sniffer) while walking outside the target property. Similar to the evil maid attack [132], the walking measurements are only necessary during initial bootstrapping.

Before feeding the collected measurements into a device localization algorithm, our attack introduces a novel *data sifting* procedure to identify the right measurement instances for anchor localization. As a result, the attacker can localize anchors down to their individual rooms using limited and often noisy signal measurements[6].

(2) Continuous human sensing. After locating a list of anchors, the attacker hides the same sniffer at a fixed location outside the target area. The sniffer continuously

[6]Because the attacker has little control on the available walking path and the propagation environment, the signal measurements will inevitably contain bias, noise and human errors.

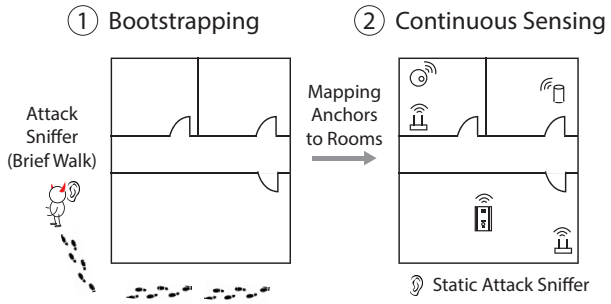


Figure 3.5: Our attack process includes a bootstrapping phase and a continuous human sensing phase.

monitors ambient WiFi signals, and uses them to locate and track human presence and movements inside. The sniffer also monitors each detected anchor, and any relocation of an anchor will trigger its removal from the anchor list, and possibly another *bootstrapping* phase to (re)locate the anchors.

Our proposed attack process is fully automated, and does not require any operations by the adversary, beyond the initial bootstrapping which involves a walk around the property to collect signal measurements. Note that this walking measurement could also be achieved by a robot/drone.

Continuous Human Sensing

In this phase, the sniffer will continuously collect $\overline{\sigma_{aCSI}}$ for each anchor and analyze them to detect, locate human presence to their individual rooms.

Detecting the presence of human motion. For each anchor x, when $\overline{\sigma_{aCSI}}(x) > \sigma_p(x)$, the sniffer declares the presence of motion near x, or "anchor x is triggered." To compute $\sigma_p(x)$, the sniffer applies median absolute deviation (MAD) [138, 139] on observed $\overline{\sigma_{aCSI}}(x)$ over time. Assuming "untriggered" $\overline{\sigma_{aCSI}}(x)$ follows a Gaussian distribution, we have

$$\sigma_p(x) = \lambda \cdot \text{MAD}(Z) + \text{median}(Z), \tag{3.2}$$

where λ is the conservativeness factor and Z is the long-term observation of $\overline{\sigma_{aCSI}}(x)$. By default $\lambda = 3$.

Assigning target(s) to rooms. When any anchor(s) get triggered, the sniffer analyzes their temporal $\overline{\sigma_{aCSI}}$ traces to determine the set of rooms that are actually occupied.

(i) If all the triggered anchors are in the same room, then the room is declared as occupied. Exit.

(ii) If most of the anchors are triggered, and their $\overline{\sigma_{aCSI}}$ time series are (consistently) positively correlated, then the sniffer is blocked by an external pedestrian next to the sniffer, and the sensing output is "uncertain." Exit.

(iii) Now consider all the triggered anchors. Start from the triggered anchor x with the highest $\overline{\sigma_{aCSI}}$. Mark x as "checked" and mark the room of x as occupied. Compute pair-wise correlation between x and any triggered anchor (y) in neighboring rooms. If x and y are highly positively correlated, mark y as checked. Repeat until all the triggered anchors are "checked".

Tracking targets. After generating a set of motion events, the sniffer can combine them with room transition probabilities built from the floorplan to estimate user trajectories. For example, the sniffer can track a security guard's patrol route from a sequence of detected motion events.

It should be noted that while our attack can detect whether a room is occupied or not, it cannot identify an individual out of a group of users in the same room. Thus accurate per-person tracking is **only** feasible when the number of users is small.

Monitoring anchor status. The sniffer also monitors each (static) anchor's RSS (see §3.1.4). Upon detecting a considerable RSS change for an anchor, the attacker either removes it from the anchor list or run bootstrapping to relocate anchors and recompute

its σ_p.

Impact of sniffer placement. The sniffer should be placed where it can capture aCSI signals from the detected anchors, while avoiding being too close to the anchors or at busy places with pedestrians frequently passing by. While one could further optimize the sniffer location, our current design randomly chooses a location that can capture signals from all the anchors.

Bootstrapping: Locating Anchors

During bootstrapping, the attacker uses the passive sniffer to identify and localize static anchors inside the target property. There are many device localization proposals, but since the sniffer stays passive and only has a single antenna, we choose to use RSS-based method [141, 142]. In this case, with a brief walk outside of the target's home/office, the adversary uses the sniffer to measure RSS of potential anchors along the trajectory. These spatial RSS values and the trajectory (recorded by the smartphone's IMU sensors) are fed into a log distance path loss model [143] to estimate the transmitter location. Each transmitter located inside the target scene area is added to the anchor list.

Why RSS but not aCSI? The localization uses RSS rather than aCSI, fCSI or AoA [144, 136] because of two reasons. *First*, our attacker sniffer only has one antenna, and cannot estimate fCSI accurately due to lack of synchronization with the transmitter. Recent work [136] estimates AoA from aCSI, but only if the sniffer has an antenna array and is in complete LoS to the targets, *i.e.* no wall. *Second*, as shown in §3.1.4, aCSI is sensitive to nearby target movements. As the adversary has no knowledge of the target status during bootstrapping, it cannot rely on aCSI for localization. In comparison, RSS is much more robust against target movements, thus a reliable input for localization under the adversarial scenario.

Identifying static anchors. RSS of a static transmitter, when captured by a static sniffer, should stay relatively stable, while those of moving ones will fluctuate over time. Thus before running spatial RSS measurements, the attacker will keep the sniffer static and measure the per-device RSS standard deviation (σ_{RSS}) for a period of time (*e.g.* 60s). Devices with large σ_{RSS} (>2.7dB in our work) are not used as anchors. This is repeated during the continuous sensing phase (see §3.1.4) to detect relocation of any anchor device. A complementary method is to infer the device type (and brand name) from the Organizational Unique Identifier (OUI) field of the MAC address [129] and ignore moveable devices like smartphones, wearables, laptops, and camera robots.

Finding high-quality RSS measurements. The localization accuracy depends heavily on the "quality" of RSS measurements (details in §A.2). Instead of searching for a new localization design, we apply a statistical data sifting algorithm to identify proper RSS data samples as input to the localization algorithm.

The attacker can filter out "bad" measurements using de-noising methods, *e.g.*, Kalman filter [145], wavelet filter [146] and feature clustering [142]. We find that these are insufficient under our attack scenarios because the propagation environment is highly complex and unknown to the adversary, making it hard to distinguish between noise and natural propagation effect. Similarly, features used by [142] to identify bad measurement rounds are too coarse-grained to effectively control localization accuracy. In fact, our experiments in §3.1.6 show that with [142], $> 50\%$ of the good measurement rounds it identifies all locate the device to a wrong room.

Instead, we propose *consistency-based data sifting* to identify proper data samples that will be used for model fitting. Our hypothesis is that, by the law of large numbers [147], *consistent* fitting results from many random sampling of RSS measurements, if exist, can reveal true signal propagation behaviors and produce high-fidelity localization results.

Given a round of measurements $\mathbb{R}$, we apply the Monte Carlo method [148] to ran-

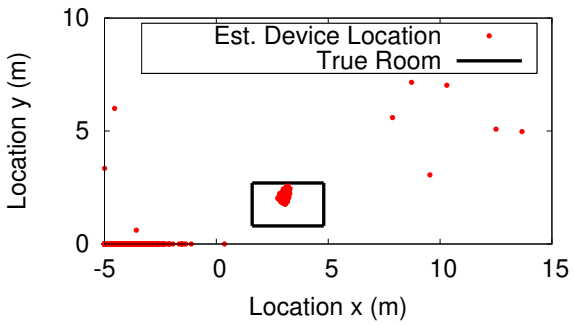


Figure 3.6: Improving accuracy of anchor localization using our proposed consistency-based data sifting. Each red dot is the anchor location estimated from a Monte Carlo sample of RSS measurements. The rectangle marks the actual room the anchor resides. In this example, a dominant cluster is present and is used to estimate the final anchor location.

domly sample a subset (80%) of $\mathbb{R}$ as the input to the model fitting. This is repeated by $N = 1000$ times, producing N localization results. We can find natural clusters among these N results from their locations and fitting mean square errors (MSE). If they form many small clusters with different room placements, then $\mathbb{R}$ is inconsistent and cannot be used for localization. If a dominant cluster exists and its averaged MSE is less than those of the other clusters, then $\mathbb{R}$ can be used for localization. An example is shown in Figure 3.6, which produces a single, dominant cluster, while the rest are widely scattered. When such a dominant cluster is present, we can estimate the anchor room location by aggregating the location data points of the cluster. In the example of Figure 3.6, all the data points are located in the top center of a single room. When the data points belong to different rooms, we choose the room with the most data points.

When multiple rounds of RSS measurements are available, the attacker can apply consistency check — if a localization result is consistent across multiple rounds, then it is a confident result. Formally, we assume a probability p_x that the localized human is actually located in room x. The more consistent our results are across multiple rounds of measurements, the higher p_x is:

Lemma 1 *If $p_x \geq p_i, \forall i \in$ rooms and the results over n rounds of measurements are consistent, the probability of the correct room estimation $\hat{p}_x \geq p_x$.*

Proof:

The probability of consistent results over n rounds of measurements is $\sum_i(p_i^n)$ and $\hat{p}_x = p_x^n / \sum_i(p_i^n)$.

Let $0 < m < n$. Then:

$$
\begin{aligned}
(\forall i)(p_x \geq p_i) &\Rightarrow (\forall i)(p_x^{n-m} p_i^m \geq p_i^n) \\
&\Rightarrow p_x^{n-m} \sum_i (p_i^m) \geq \sum_i (p_i^n) \\
&\Rightarrow {p_x}^n / \sum_i (p_i^n) \geq {p_x}^m / \sum_i (p_i^m)
\end{aligned}
$$

When $m = 1$, ${p_x}^m / \sum_i(p_i^m) = p_x$. Thus, $\hat{p}_x \geq p_x$.

■

As the condition in Lemma 1 meets all the environments we have tested, we can improve the room correctness. Across our experiments, we find that such consistency check across 4 rounds of measurements is sufficient to achieve a room placement accuracy of 92.6%. For environments that do not satisfy this condition (§3.1.6), this method does not guarantee an improvement.

Floor-level signal isolation. When the target property has multiple floors, the attacker needs to localize wireless anchors to a particular floor during bootstrapping. This is easily achieved using coarse angle of arrival (AoA) estimates captured by the smartphone with a simple cone cover to focus signals from a particular AoA. The received RSS from each anchor can be combined with the phone angle (via the built-in gyroscope) to localize each anchor to a floor. More details are discussed in §A.3.

3.1.5 Smartphone Implementation

We prototype our attacker system using a commodity smartphone as the sniffer. We implement the bootstrapping and continuous sensing modules each as an Android app, and deploy and experiment using two versions of Android phones, Nexus 5 and Nexus 6. Both smartphones are equipped with the Broadcom WiFi chipset. For spatial RSS measurements (during bootstrapping), we use the built-in IMU sensors (accelerometer and gyroscope) to detect user strides and build trajectory. The key system parameters are listed in Table 3.1.

Enabling continuous, passive sniffing of aCSI. Previously, aCSI can only be captured when the receiver actively communicates with the target transmitter [149]. Recent work [150] produces a firmware (Nexmon) that enables passive[7] sniffing, but only on a single customized transmitter at very low rates.

For our attack, we made a series of changes to the Nexmon firmware, so that the sniffer can run continuous passive sniffing and capture aCSI from multiple commodity WiFi devices simultaneously. In particular, we made changes to hardware buffer management to resolve the issue of buffer overflow facing the original Nexmon firmware.

One remaining artifact is that the firmware only reports aCSI at a limited speed, up to 8–11 packets per second. To save energy, we subsample sniffed packets based on this rate limit. Despite this artifact, our prototype sniffer is able to capture sufficient aCSI samples to successfully launch the attack.

Computation and energy cost. One strength of our attack is its simplicity. For our current smartphone prototype, the bootstrapping app runs 1000 rounds of Monte Carlo sampling and model fitting, which finishes in less than 25s per anchor. It takes less than 1s to compute average aCSI standard deviation. The app consumes 4.18 watts

[7]Passive sniffing means that the sniffer does not need to communicate with the target transmitter, thus remains completely undetectable.

Parameters	**Value**
MAD conservative factor λ	11
Threshold of σ_{RSS} for static anchors	2.7
Ratio of Monte Carlo sampling size	80%
Number of Monte Carlo sampling rounds (N)	1000

Table 3.1: Attack parameters used in our experiments.

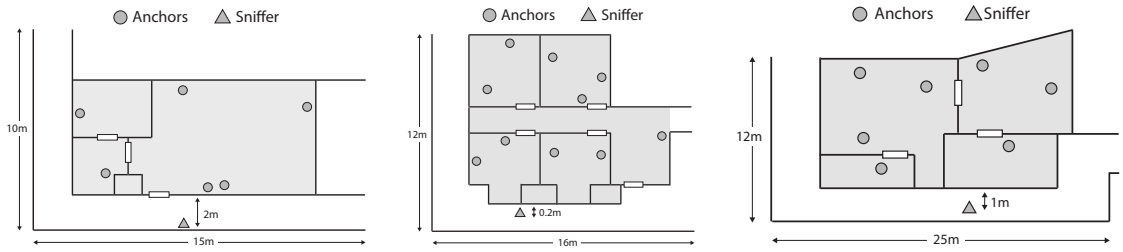
Sniffer Path	Test Scene	# of Rooms	Mean Room Size (m^2)
Indoor Hallway	1	6	14.19
	2	7	14.60
	3	8	13.65
	4	3	14.50
	5	3	9.51
	6	6	14.21
	7	5	16.75
	8	4	44.39
	9	2	69.83
Outdoor Sidewalk	10	2	47.20
	11	4	12.99

Table 3.2: Test scene configuration.

(bootstrapping) and 2.1 watts (continuous sensing), respectively. For Nexus 5 (with a built-in 2300mAh battery), this enables 4.1 hours of continuous sensing. Currently our app does not optimize for energy efficiency, which could be improved to extend sensing duration.

3.1.6 Evaluation

We evaluate our attack using experiments in typical office buildings and apartments. We first describe our experiment setup and test scenes, then present our evaluation on individual attack phases (bootstrapping and continuous sensing), followed by an end-to-end attack evaluation.



(a) Scene 4 (b) Scene 6 (c) Scene 8

Figure 3.7: Sample test scene floorplans, derived from the real estate websites or emergency exit maps, where shaded regions are the target property. We also show an instance of anchor placements where ◯s are the anchor devices, and △ is the static attack sniffer.

	Device Type	Exact Product	Mean Packet Per Second (pps), Idle	Mean Packet Per Second, Active
Static	Cameras (w/o Motion Detection)	AHD Security Camera	N/A	124
	Cameras (w/ Motion Detection)	Amcrest/Xiaomi IP Camera	≥0.5	108
	Home Voice Assistance	Amazon Echo, Google Home	2	16
	Smart TV (& Sticks)	Chromecast, Apple TV, Roku	6.64	200
	Smart Switches	LifeSmart Plug	≥2.44	≥3.33
	WiFi Router	Xiaomi/Cisco/Asus Routers	28.6	257
Mobile	Surveillance Robot	iPATROL Riley Robot Camera	N/A	124
	Smartphones	Samsung/Google/Apple Phones	≥0.5	≥6

Table 3.3: Summary of WiFi devices used in our experiments. Note that our attack will detect and recognize static anchors and only use them to detect/localize human motion.

Experiment Setup

We experiment at 11 typical offices and apartments that are accessible to us. The owners of each test volunteered for our experiments. The test scenes are of different sizes and configurations, and have different wall materials except for concrete[8]. The walking path available to the adversary also differs across experiments, from indoor corridors outside the apartment to outdoor pathways. Table 3.2 lists the test scene configuration while Figure 3.7 shows floorplan examples derived from publicly available data. Across all experiments, attack parameters remain unchanged (as listed in Table 3.1).

Inside each test scene, we either reuse existing WiFi devices or deploy our own WiFi

[8]Our attack does not work when the wall separating the targets and the adversary is made of concrete, which blocks the majority of the WiFi signals.

devices to emulate smart homes and offices. We use popular commodity products for smart offices and homes, *e.g.*, wireless security cameras, voice assistants, WiFi routers, and smart switches. In total, we have 31 WiFi devices, including 6 security cameras. These devices are naturally placed at locations where they are designed to be: security cameras at room corners, smart switches on the wall outlets, and WiFi routers in the center of the room for coverage. Our experiments use the 2.4GHz WiFi band due to its dominant coverage. We also test 5GHz WiFi and do not observe notable difference except its shorter coverage.

Table 3.3 summarizes these devices and their traffic patterns during idle and active periods. The packet rate varies from 0.5 packet per second (pps) to more than 100 pps. Even when idle, they still periodically transmit packets. It should be noted that to prevent attackers from inferring user presence by simply counting the packet rate of a device (if an Amazon Echo is sending more packets, it means that a human user is around), devices like home voice assistants, smart TVs, and motion-triggered cameras will need to send cover traffic when in idle state and the corresponding idle packet rate will be much higher than the listed number.

Bootstrapping. To benchmark our bootstrapping design, we collect, for each test scene, 50 walking measurements, each of 25–50 meters in length and 0.5–2 minutes in time. We also change anchor placements and repeat the experiments. In total, we collect more than 3000 RSS measurement traces, with more than 121,000 location-RSS tuples.

Continuous sensing. We place a static sniffer behind plants or at the corners (on the ground) outside of the target building within $2m$ to the building wall. We ask volunteers to carry out normal activities in each test scene and collect more than 41hrs of aCSI entries (7.8hrs of human presence, labeled). The volunteers are aware of the attack goals but not the techniques.

Evaluation of Continuous Human Sensing

We start from evaluating the *continuous sensing* component of our attack. Here we assume that the attacker knows the actual room where each anchor resides. By default, the attacker only uses anchors whose packet rate $\geq$ 11pps.

Performance metrics. Our goal is to evaluate whether the continuous sensing component is able to correctly detect user presence/motion in each room. We divide time into 5s slots, and run continuous sensing to estimate room occupancy in each slot based on aCSI variance values. We compare these estimates to ground truth values, and compute the detection rate and false positive rate as follows.

- *Detection rate* (DR) measures the probability of the attack reporting a room as being occupied when it is occupied, across all the slots.
- *False positive rate* (FP) measures the probability of a room not being occupied when our attack reports the room as being occupied.

Under our adversarial scenario, having a high detection rate is more important since the attacker does not want to miss the presence of any targets.

Results: sensing accuracy. Table 3.4 lists the detection rate and false positive rate when we vary the number of anchors per room. We see that the detection rate scales with the number of anchors per room, reaching 86.8%, 95.03% 99.85%, and 99.988% with 1, 2, 3, and 4 anchors per room, respectively. This trend is as expected since having more anchors increases the chance that a user movement triggers at least one anchor. Furthermore, the false positive rate is low (<3%) with a single anchor per room and increases slightly to 6.9% if the attacker wants to leverage all 4 anchors. Across our experiments, the false positives mainly come from the impulse noises in aCSI reported by the firmware. Thus having more anchors will lead to more false positives.

		# of WiFi Devices Per Room			
		1	2	3	4
Ours	DR	86.824%	95.034%	99.854%	99.988%
	FP	2.927%	4.082%	5.305%	6.935%
LiFS	DR	20.536%	37.040%	50.262%	60.821%
	FP	4.622%	4.961%	5.395%	5.886%
LiFS (unrealistic)	DR	43.568%	68.315%	82.289%	90.149%
	FP	4.622%	5.364%	6.443%	7.644%

Table 3.4: Detection rate (DR) and False positive rate (FP) of continuously human sensing, assuming accurate room placement of anchors. We compare our design to the state-of-art human sensing system (LiFS).

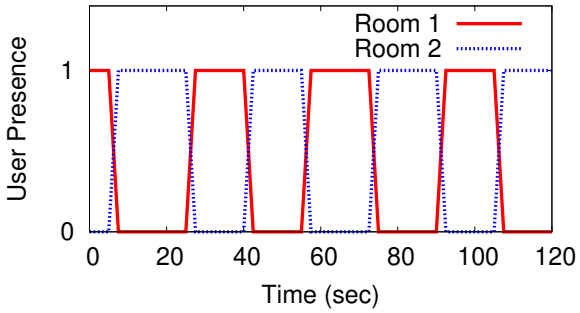


Figure 3.8: The attack sniffer can track fast user motion between rooms.

We also compare our system to the current state of the art of passive human sensing (LiFS [121]). For fair comparison, we add wavelet denoising to LiFS, confirming that it improves the sensing performance. Since LiFS requires each anchor's precise physical location in the room (which is not available to our attacker), we first use the room center as the input to LiFS, mapping to 1-2m localization error. LiFS also requires knowledge of the aCSI value when no user is present, which we use the same MAD based method to estimate. Results in Table 3.4 show that even with four anchors in the room, LiFS can only achieve a detection rate of 60.82%. Here the miss-detection happens when LiFS locates the human presence to a wrong room. We also run another version of LiFS that is unrealistic under our attack scenario, where each anchor's physical location error is random but bounded by 50cm (and without any room placement error). In this case, its

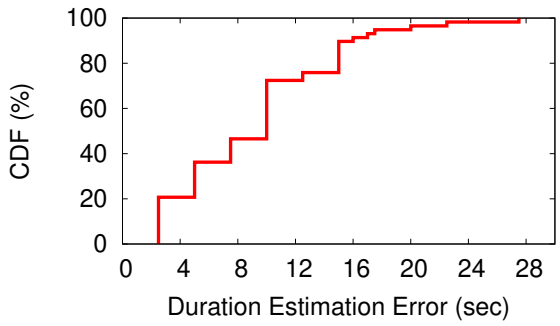


Figure 3.9: Error in motion duration estimation is small.

detection rate improves, but is still far from our attack, especially with smaller number of anchors per room.

Results: tracking responsiveness. We also examine whether our attack is able to track human movements in time. We start from an example scenario where a user moves back and forth between two connecting rooms, *i.e.* she walks in one direction for 18s, turns around and walks in the opposite direction, and repeats. Figure 3.8 shows the detected user occupancy of the two rooms (each with two anchors). We see that our detection is highly responsive to rapid human movements.

We also consider all the aCSI traces collected across our test scenes and examine the duration of individual movement events estimated by the attacker. We compare these estimations to the ground truth. Figure 3.9 plots the CDF of the duration estimation error, where for 80% of the cases, the error is less than 16 seconds.

Impact of anchor packet rate. So far, our results assume that anchors send packets at no less than 11pps[9]. To study the impact of anchor packet rate, we take the aCSI traces of WiFi security cameras (w/o motion detection) and sub-sample them to produce desired packet rates. Our experiments show that for a single anchor per room, the detection rate is 86.824% at its full rate (an equivalent aCSI rate of 11pps), and

[9]As discussed in §3.1.5, the sniffer's firmware reports CSI in an equivalent packet rate of 8–11pps.

reduces to 85.49% at 2pps, and 69.29% at 1pps. The false positive rate remains <5%. This means that each low-rate anchor can still "help" the attacker identify and locate targets. For a room with multiple low-rate anchors, the attacker will take the *union* of their detection results.

Impact of interference. During all experiments, we observe minimal impact on attack performance from interference by other WiFi transmissions out of our control or access. In the presence of strong interference, anchor packet rates will drop (due to CSMA contention) and thus human detection rate will drop as discussed earlier.

Non-human sources of motion. Smart homes and offices often have equipment that create motion even in the absence of human users. One relevant question of interest is whether these machines will be detected by the attack as human movement, leading to false positives? We experiment with a set of moving devices commonly seen in homes and offices, as well as pets, *e.g.* cats and dogs (see Table 3.5). For example, robotic vacuums are placed on the ground level and thus have minimal impact on the sniffed signals. The only device to affect $\overline{\sigma_{aCSI}}$ in our tests is an oscillating fan. Yet its motion is highly periodic and consistent, making it easy to distinguish as non-human. We note that certain cats and dogs can also affect $\overline{\sigma_{aCSI}}$ with their movement, and their movement patterns can be hard to distinguish from human motion. Overall, our experiments show that the attack can eliminate all non-human sources of motion, except for pets.

Impact of MAD conservation factor λ. While the above experiments all assume that the MAD conservative factor λ is 11, we also vary the value of λ (0.1–19) to study its impact on DR and FP. Results in Figure 3.10 confirm that a smaller λ means that the attacker is more conservative and seeks to detect every user motion/presence. As such, the false positive rate also increases. The result also confirms that having more anchors per room also increases detection confidence.

Motion source	Impact on $\overline{\sigma_{aCSI}}$	Distinguishable from human motion?
Server internal cooling fan	No	-
Standing fan	No	-
Oscillating fan	Yes	Yes
Robot vacuum	No*	No
Cats or dogs	Yes	No

Table 3.5: Impact of sources of non-human motion on our attack. (*) A robot vacuum only affects $\overline{\sigma_{aCSI}}$ of an anchor in close proximity when the anchor is placed on the floor.

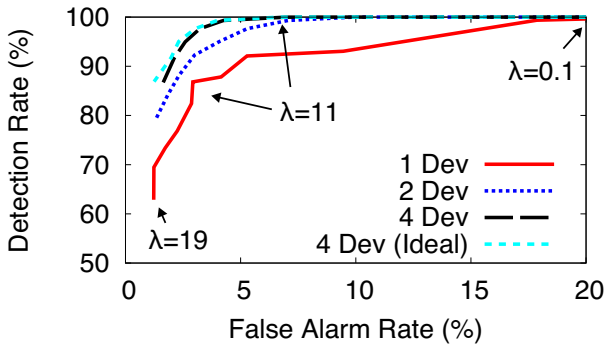


Figure 3.10: Impact of MAD conservative factor λ on the overall DR and FP.

Finally, we look at an ideal case assuming the attackers can *always* pick the best performed anchor with minimal false alarms. We see that it slightly improves the false alarm rate, but does not help much on improving the human detection rate. This means the false alarms do not come from a single anchor device. There may exist other methods to combine the signals across multiple devices in the same room to achieve better results, and we leave that as our future work.

Evaluation of Bootstrapping

We evaluate the bootstrapping phase (where the attacker detects and locates anchors) via two metrics: *absolute localization error* which is the physical distance between the ground-truth location and the attacker-estimated location, and *room placement accuracy*

which is the probability of placing an anchor to its exact room. Figure 3.11 plots, for each test scene, the quantile distribution of the absolute localization error and the room placement accuracy. We compare three systems: the model-fitting algorithm that uses all the measurements, the feature-clustering based data filtering proposed by [142], and our consistency-based data sifting method.

Gain of consistency-based data sifting. The results show that our proposed data sifting method can largely improve anchor localization accuracy compared to the two existing approaches. Our method largely reduces the localization error tail, and for more than 90% of the cases, the attacker places the anchor at the right room. Our method outperforms [142] by using fine grained, scene-specific features to filter data.

An interesting observation is that in scene 8–10, our method faces a similar (and even larger) absolute localization error than feature clustering but produces higher room placement accuracy. This is because our design directly analyzes the consistency of room placements, rather than raw localization errors. Smaller raw localization error does not always translate into higher room placement accuracy.

Impact of anchor placements. As expected, it is relatively harder to accurately locate anchors placed at room boundaries, *e.g.*, those plugged into wall outlets. In many cases, these boundary anchors create a dominant Monte Carlo cluster, but the data points in the cluster map to either of the two neighboring rooms. Our current design simply chooses the room with more data points, which could lead to room placement errors.

When the number of anchors is sufficiently large, the attacker can minimize the impact of such boundary anchors in two ways. First, the attacker can either user these boundary anchors "with caution" or not using them at all. Second, the attacker can leverage past human sensing results to discovery any strong correlation between anchors and adjust their room placements. We leave these to future work.

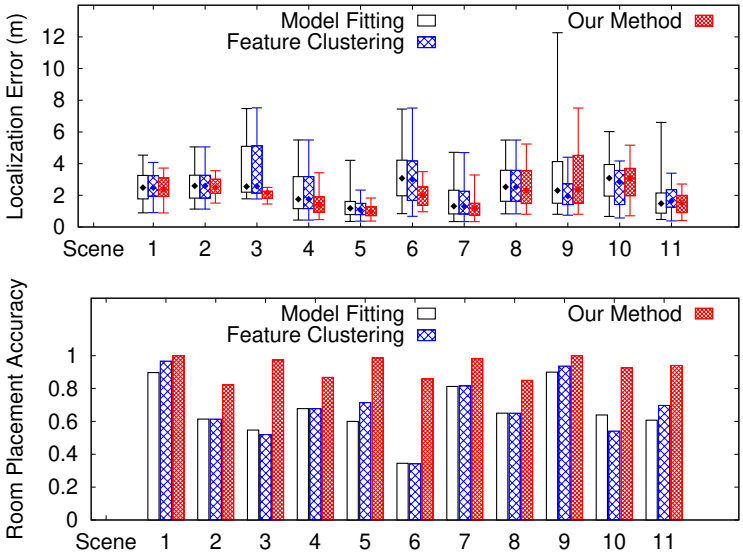


Figure 3.11: Bootstrapping performance: anchor localization accuracy in terms of absolute localization error (m) and room placement accuracy, per test scene.

Impact of anchor packet rate. The accuracy of our proposed anchor localization method is relatively insensitive to anchor packet rate. This is likely because RSS (of static anchors) is relatively stable over time. As long as the measurement trace covers >20m in distance and the RSS values are between -75dB and -30dB without strong bias, we observe little difference in localization (and room placement) accuracy.

End-to-End Attack Evaluation

Finally, we evaluate the end-to-end performance of our attack, combining the bootstrapping and continuous sensing phases. Like §3.1.6, we consider the detection rate and false positive rate for the task of detecting and localizing human users to their individual rooms. The results will include the impact of any misplaced anchors during bootstrapping, or errors in detecting/localizing users during continuous sensing.

Table 3.6 lists the detection rate and false positive rate for our attack design and those achieved via LiFS [121]. We also vary the number of WiFi anchor devices in each

		# of WiFi Devices Per Room			
		1	2	3	4
Ours	DR	80.603%	94.210%	98.780%	99.725%
	FP	3.595%	5.962%	8.386%	10.719%
LiFS	DR	14.153%	26.381%	36.954%	46.033%
	FP	14.024%	14.077%	14.493%	15.064%

Table 3.6: End-to-end performance of our attack vs. LiFS, in terms of detection rate (DR) and false positive rate (FP).

room. Compared to the results in Table 3.4 assuming accurate anchor room placement, the end-to-end attack sees minor drop in both recall and precision, especially with more anchors per room. Despite using a passive, minimally equipped attacker device, our attack still achieves high human sensing accuracy, *e.g.*, 99.7% detection rate at 10.71% false positive rate.

The impact of anchor localization errors is much more visible on LiFS, whose detection rate drops to 36.954% and 46.033% even with 3 and 4 anchors in each room, respectively. Overall, we see that while both using the same aCSI values per anchor, our proposed passive human sensing largely outperforms LiFS by not requiring precise anchor location to model signals on the direct propagation path.

3.2 Defending Against Our Proposed Attacks

We now explore robust defenses against our proposed attack and other passive sensing attacks that use the same hardware setup as ours, *i.e.* a single sniffing device with a networking radio and a single antenna. Our design insight is that our attack effectiveness depends heavily on the quantity and quality of the WiFi signals captured by the sniffer. Thus a defense reducing the amount of WiFi signal leakage to external sniffers or adding inconsistency to WiFi signals could render the attack ineffective.

3.2.1 MAC Randomization

The first solution coming to mind would be *MAC address randomization*, a well-known method for protecting mobile devices against tracking. Since the attack sniffer uses MAC address to isolate signals of anchors, MAC randomization can disrupt both bootstrapping and continuous sensing phases. However, recent work has shown that MAC randomization is disabled on most devices (<3% of adoption rate so far) [151] and can be easily broken to reveal the real MAC address [131, 152]. We note that Android 9.0 Pie switches to per-network MAC randomization [153], which does not apply any MAC randomization to static WiFi devices. Thus MAC randomization is not a plausible defense against our attack.

3.2.2 Geofencing WiFi Signals

Geofencing bounds signal propagation to reduce or eliminate WiFi signals accessible to the adversary. In our attack, when the area with a signal in our walking trace reduces from 25–50 meters to 10 meters or less, the anchor localization error increases significantly: raw errors more than double, and anchor room placement accuracy drops from 92.6% to 41.15%.

Geofencing is also extremely difficult to deploy and configure. The simplest form is to reduce the anchor's transmit power, which is almost always undesirable since it degrades connectivity. Another option is to equip WiFi devices with directional antennas, limiting signal spatial coverage. This is also undesirable as it requires upgrading to equipment with higher cost and larger form factors, as well as carefully configuring antenna directionality. Finally, the extreme solution is to block RF propagating beyond property walls by painting these walls with electromagnetic shielding paints. This is again impractical, since it blocks incoming WiFi/cellular signals.

If the area accessible to the attacker is limited, a potential solution is to customize WiFi signal coverage using 3D fabricated reflectors [154] or backscatter arrays [155] that create noise towards the area. Yet both remain open research problems.

3.2.3 WiFi Rate Limiting

While geofencing reduces spatial leakage of WiFi signals, rate limiting reduces their temporal volume. When anchors transmit less signals over time, the sniffer will not have sufficient data to compute $\overline{\sigma_{aCSI}}$. Results in §3.1.6 show that reducing anchor packet rates does lower the detection rate (when using a single anchor) but can be compensated by aggregating the results of multiple anchors.

In practice, rate limiting is undesirable for most network applications. As shown in Table 3.3, many WiFi devices, when idle, already transmit at more than 2pps. It is hard to rate limit further, rendering the defense ineffective.

3.2.4 Signal Obfuscation: Existing Designs

Signal obfuscation adds noise to WiFi signals, so the adversary cannot accurately localize anchors or detect user motion. Existing works have proposed both temporal and spatial obfuscations against RF sensing [142, 156].

In *temporal obfuscation*, WiFi devices (anchors) change transmit power randomly over time, injecting artificial noises to signals seen by the sniffer. Doing so, however, requires upgrading commodity WiFi devices to equipment with much higher cost and energy consumption. Also a more resourceful adversary can counter by deploying an extra static sniffer (during bootstrapping) to infer the injected signal power changes and remove them from the signal traces, as shown by [142].

In *spatial obfuscation*, a recent work [156] shows that by deploying a full-duplex radio

near each anchor x, one can obfuscate x's signal phase, RSS, CSI, and Doppler shift seen by any nearby sniffers with a single antenna. But full-duplex radios are of high cost, and there is no commodity product on the market. On the other hand, defending against our attack only needs to obfuscate RSS and aCSI, which can be observed by the sniffer.

3.2.5 Proposed: AP-based Signal Obfuscation

The above four immediate defenses are either ineffective or impractical. Instead, we propose a practical defense where the WiFi access point (AP) actively injects customized cover traffic for any of its associated WiFi device w that is actively transmitting. We design this defense to produce large ambiguity to the attack in two steps. *First*, our defense adds noise to the attacker's RSS measurements, so that during bootstrapping, the attacker will place most of the anchors to the wrong room and even outside of the property. *Second*, our defense largely reduces (and even removes) the $\overline{\sigma_{aCSI}}$ gap between no human presence and human motion, such that the attacker is unable to identify human motion.

AP inserting customized cover signal. As soon as the AP detects a transmission from w, it estimates w's transmission rate T_w and injects a cover traffic stream with the rate of ρT_w, at a randomized power level and with w's MAC address. ρ is the injection rate, a system parameter. Since the AP limits its cover traffic stream to be proportional to w's throughput, the CSMA protocol will randomly interleave packets from the two streams together. In the worst case (ρT_w is at or higher than available channel throughput), the cover traffic will reduce w's effective throughput by $1 + \rho$. Finally, the insertion of "fake" packets requires a careful design, so that it disrupts the attack rather than creating obvious "anomalies" or heavily affecting the WiFi network. The AP configures the sequence numbers of fake packets to (partially) interleaved with

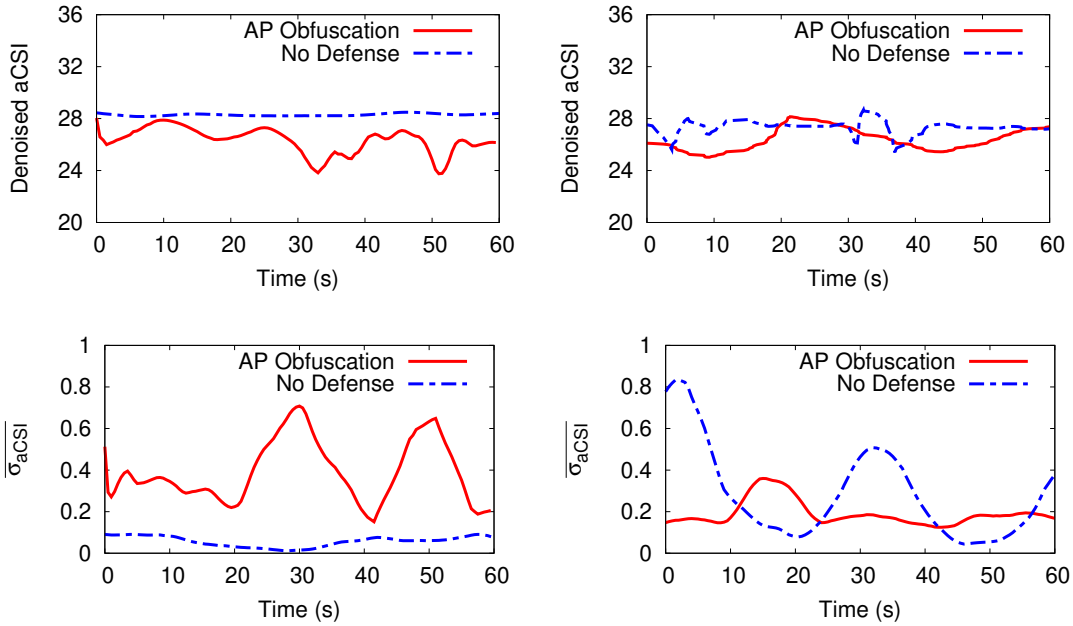
those of real packets, so that the attacker is unable to separate the two streams based on sequence number and packet arrival time.When sending fake packets, the AP's transmit power is randomized but close to that of w, so the mixed traffic follows natural (and complex) multipath signal variation. This makes it hard to distinguish real and fake packets from signal strength values alone.

This defense can be deployed on today's WiFi APs that support transmit power adaptation with minor changes. The major overhead is the extra consumption (a factor of ρ) of bandwidth and energy at the AP.

Results: impact on bootstrapping. With this defense, the attacker's RSS measurements of anchor w will display fluctuations, tricking the sniffer to think that w is moving and not use it as an anchor. Even if the adversary assumes w is stationary, the noisy RSS measurements (even after our data sifting) will lead to inaccurate room placement.

When evaluating this defense, we consider both our original attacker (with one smartphone) and an *advanced* attacker who adds an extra stationary sniffer and applies RSS signal subtraction to detect and remove any "injected" signal variations [142]. We configure our defense where the AP injects cover traffic of ρ with power randomization in the $10dB$ range. For both attackers, ρ=5% is sufficient to drop the accuracy of anchor room placement from 92.6% (without our defense) to 35.71%, except for the anchors close to the AP (in the same room). As we further increase ρ, the attacker will map most of the detected anchors to the AP's room.

Results: impact on continuous sensing. As the attacker sniffer calculates $\overline{\sigma_{aCSI}}(w)$ on randomly interlaced packets sent by anchor w and the AP, the value of $\overline{\sigma_{aCSI}}(w)$ with no human presence will increase significantly. Figure 3.12a shows a sample trace of aCSI (of a sub-carrier) and $\overline{\sigma_{aCSI}}$ with and without the defense. We see



(a) User not present (b) User in motion

Figure 3.12: aCSI and $\overline{\sigma_{aCSI}}$ with and without AP based signal obfuscation.

that our defense can effectively elevate the threshold $\sigma_p(w)$ for human presence detection. More importantly, the defense has much less impact on $\overline{\sigma_{aCSI}}(w)$ when there is actual human movement near the anchor w. The sample traces in Figure 3.12b show that $\overline{\sigma_{aCSI}}(w)$ actually drops (below the threshold) when using the proposed defense. In this case, human movement will not trigger any anchor in proximity, for both the original and the advanced attackers (who deploy an extra sniffer).

Table 3.7 lists the attack performance with our proposed defense (ρ=30%) and without any defense. We first consider the case where the attacker manages to obtain perfect anchor room placement. In this case, our defense increases the false positive rate from 7.9% to 48.28% while dropping the detection rate to 78.776%. Next, we consider the end-to-end attack scenario where the attacker performs both bootstrapping and continuous sensing. Our defense drops the human detect rate down to 47.48% while increasing the false positive rate to 49.5%. These results apply to both the original attacker and

	False positive rate		Detection rate	
	No defense	AP obf	No defense	AP obf
knowing anchor room placement	7.935%	48.284%	99.988%	78.776%
end-to-end attack	10.719%	49.598%	99.725%	47.481%

Table 3.7: The attack performance under AP-based signal obfuscation (best performance out of the original and the advanced attack with an extra sniffer).

the advanced attacker. This level of ambiguity renders the attack useless in practice.

Possible countermeasures. To overcome our proposed defense, the attacker must find ways to distinguish the obfuscation packets sent by AP from the original packets sent by an anchor w. As discussed earlier, doing so using packet sequence number and arrival time is infeasible due to our packet injection method. Doing so at the network traffic level is also difficult, since packet contents are encrypted, and we can shape traffic to resist traffic identification by attackers [115]. Finally, it is also difficult to separate the two streams using physical layer characteristics, because doing so requires much more sophisticated and bulky hardware. One option is to analyze per-symbol aCSI/RSS patterns. This is infeasible using commodity WiFi chips, as they only report per-packet aCSI/RSS values. Another option is to use a large antenna array (MIMO with at least 4–6 antenna elements, each separated by 6.25cm) to distinguish signals sent by w from those sent by the AP, since they come from different directions. The resulting sniffer (at least 31cm in length) would be conspicuous and easily raise suspicion.

3.3 Related Work

Human sensing by snooping signals. We categorize existing works into five groups. The first group applies *traffic analysis* to infer user presence and status in a home/office from their network traffic [157, 129, 158, 159, 160, 161, 162]. It requires

strong knowledge on device behaviors and can be easily countered by sending cover traffic, applying encryptions and traffic shaping. In contrast, our attack remains effective even when all network-level defenses are deployed, as long as WiFi devices still transmit packets.

The second group uses "specialized signals" such as RFID [163], visible light [164, 165], and acoustic [166, 167], that often correlate with human motion. But existing solutions require control of transmitters inside or outside of the target property, which is infeasible under our attack model.

The third group builds *fingerprints* of each predefined target location and/or activity, based on either aCSI [168, 169, 137], CSI [47, 170], RSS [123, 171, 172, 173], or raw signals [174]. Since the attacker under our model has no knowledge of the target users and access to the target property, building fingerprints becomes infeasible.

The fourth group uses advanced radio hardware (laptops or USRPs with antenna arrays or directional antennas) that communicate with the anchors inside the target property. This allows the sniffer to measure fine-grained CSI values (both amplitude and phase) [175], and use them to calculate AoA and doppler frequency shift (DFS) to detect human motion [134, 124, 170, 176, 177, 178]. Our attack differs by using a passive sniffer with a single antenna, which does not communicate/synchronize with the anchors. In this case, the sniffer cannot infer CSI phase, AoA or DFS.

The final group detects user motion using passive sniffers to collect and analyze physical RF signals [126, 178, 121]. As discussed earlier, both [126, 121] target user motion that disturbs the direct propagation path, requiring precise locations of the anchors. [178] uses multiple sniffers with bulky directional antennas to compute doppler shift of user motion. The sensing method used by our attack falls into this category, but targets multipath signal propagation from each anchor to the sniffer. We design a new aCSI variance model to reliably detect user motion, eliminating the need for precise anchor

location and antenna array at the sniffer.

Passive transmitter localization. Existing works often leverage bulky receivers with multiple antennas [44, 117, 144, 136, 116, 179] to estimate signal AoA, and applies triangulation across receivers to derive target location. Our anchor localization (during bootstrapping) uses a compact smartphone with a single antenna, and applies passive localization that fits spatial RSS measurements to a propagation model [180, 142, 181]. Our key contribution is the data sifting algorithm that identifies good RSS samples as input to the model fitting.

Defense against RF sensing. Existing works [182, 156, 183, 184] defend against eavesdropping on a transmitter by a jammer transmitting simultaneously, preventing the attacker from decoding packets or estimating CSI/AoA. This requires precise synchronization between the transmitter and the jammer [185] or a high-cost full-duplex obfuscator [156]. Our defense uses the AP to insert fake packets (rather than transmitting simultaneously), which is easy to deploy and effective against our proposed attack.

3.4 Summary

Our work shows that the ubiquity of WiFi (and general wireless) devices has an unexpected cost: reflected or blocked RF transmissions leak information about our location and activities. Our work describes a set of low-cost, stealthy reconnaissance attacks that can continuously monitor and locate human motion inside a private property, essentially turning WiFi devices inside into motion sensors. All this is done without compromising the WiFi network, data packets or devices, and only requires a commodity WiFi sniffer outside of the property. We validate the attack on a variety of real-world locations, and develop a new effective defense based on carefully tuned WiFi signal obfuscation by APs.

We believe our work points to the potential of more powerful information leakage

attacks via passive RF reflections. With more sophisticated signal processing techniques (and potentially new hardware), much more might be learned from the way ambient RF signals interact with our bodies and surroundings. We are pursuing this line of research to both better understand these attacks and to develop defenses to better safeguard our security and privacy.

Chapter 4

Conclusion

This dissertation presents two aspects when we develop an environmental mobile sensing system. We proposed the design of novel and practical mobile sensing systems using only the networking radios. We also identified the potential ways to misuse sensing using the ambient wireless signals from exiting infrastructures, and defended against it by the router-assisted obfuscation. We hope our work sheds light on the future development and research of mobile sensing.

Below we summarize our major contributions, followed by the discussions of the future research directions.

4.1 Summary of Contributions

Designing robust mobile imaging systems. We propose our practical sensing system designs that leverage 60GHz networking radios to image the nearby objects. By actively transmitting 60GHz signals and receiving the reflections from the objects, our system can accurately tell the objects' location, shape and surface material. In our design, we address several challenges that are unique to such high frequency signals.

First, operating at 60GHz, traditional radar algorithms like synthetic aperture radar no longer work in practice as they require the system to track the device trajectory under 5mm. To address this, we proposed a new imaging algorithm named RSS Series Analysis (RSA) that models an object's surface as a single unit (rather than a set of independent points). In our model, we only use the directionality of 60GHz signals (*i.e.* AoA) and the RSS measurements. Since both of these are insensitive to device movement deviations (<10cm), our algorithm is robust to small trajectory tracking errors.

Second, to enable environmental sensing on a single device, we need to co-locate the transmitting and receiving 60GHz networking radios on the same device. This means the sensing system has a very limited view of an object, as the co-location of the two radios restricts the system to receive only the specular reflections from an object. Also, we need a different imaging algorithm from the previously designed one (RSA) as it requires a static anchor along the device movement. We address the problem by combining imaging with device navigation. Specifically, we model an object's surface as a set of connected, flat units so each unit reflects signals as a mirror. By geometry, the measured strongest signal directions and the unit orientation are highly correlated. Thus from the continuous change of signal directions along the device movement, we can first recover the shape of the object's surface, and then shift and resize it to the correct position. Again, here we only rely on the RSS measurements and the transmitted and received signal directions (AoA and AoT), and so our method is insensitive to small movement deviations.

Our proposed sensing system can recover the object shape accurately, comparable to today's vision-based reconstruction algorithms. It works robustly in low-light conditions, and can narrow down the object's surface materials to 3 candidates.

Defining and defending against silent reconnaissance attacks. We also expose a novel set of passive sensing attacks, where an adversary can accurately tell the presence and location of people in rooms by analyzing the nearby ambient signals, either from the

networking or the sensing devices. The attack only requires a small device with a WiFi networking radio and a single antenna, passively sniffing the signals around it. It is simple to deploy and is quite effective as we achieve 99.77% human detection rate with only 10.7% false alarm rate.

To address such potential attacks, existing approaches are either ineffective or impractical. Instead we propose an AP-based defense where AP injects power-randomized packets as cover traffic when communicating with the associated WiFi devices. Since the constructed packets mimics the traffic from the WiFi device to our AP, they will be ignored by the legit devices, but will be picked up by the sniffer. The sniffer then includes all signals for human sensing, with the human motions buried inside. The analysis would either ignore the actual human motions by considering them the same as pure noise, or output that human motions always exist. This significantly degrades the human sensing performance to only 47.5% human detection rate and 49.6% false alarm rate.

4.2 Future Research Directions

This dissertation demonstrates the feasibility of reusing the networking radios for environmental mobile sensing, and presents one possible direction of sensing attacks and the ways to defend against the proposed attacks. Many open questions are yet to be explored, and we list a few below.

Sensing via sensor fusion. We show earlier that object imaging and navigation can be achieved on a single device with two 60GHz networking radios. Although our method is robust to various conditions, we rely on the networking radios which could be used for communications sometimes. So the sensing functions cannot be up 100%. Also, since 60GHz signals are highly directional, the system view can be limited and it is not suitable to sense a wide range of moving targets at far distance. Because of these shortcomings,

to build a reliable sensing system and deploy it in practice, we need different sensors to be complementary to each other, *i.e.* sensor fusion. Instead of adding all possible sensors to achieve this, we shall carefully choose a few that keep the system compact and cost-effective while they can fill the "gaps" for each other when others are disabled or unavailable.

Advanced sensing attacks and defenses. Previously we assume the attackers only want to know the presence and the location of human motions in rooms. With the signals bouncing off the walls and furniture, it is possible for the attackers to learn more than just tracking humans in rooms. For example, based on the received signals the attackers may identify the detailed structures of rooms and furniture inside. From the attackers' perspective, it would be more useful if they can infer the detailed activities of people inside, while staying stealthy.

On the defense side, our AP-based defense should also work to some extent if the attackers use a similar hardware setup to our silent reconnaissance attacks, which use a simple device with a single antenna to passively sniff signals, and if they apply a similar algorithm that correlate these signals with human motions. But for attacks that use more devices with advanced hardware like USRPs and with multiple antennas, which trades off with the stealthiness of the attack, we need a more advanced defense method. For example, we may leverage multiple APs to help obfuscate the spatial locations of the network transmitters, and these APs can craft fine-grained symbol-level obfuscations to each packet to avoid symbol-level sensing attacks.

Inserting virtual objects into the "reality". When we leverage more and more autonomous mobile agents to read the world, we learn the "reality" perceived by these sensing systems. An intriguing direction is to mimic the signals reflected from objects and trick the systems to create virtual objects. Doing so enables more immersive augmented

reality experience, as the virtual objects are created at the physical level rather than appended in the applications. One benefit is that multiple users no longer need to sync up at the application level to share experience on things virtually presented in one person's view. Also, it introduces no computation overhead to the original sensing systems. On the other hand, it is obvious that the attackers may also take advantage of this and fool the systems. While we design the exciting sensing systems, we should also consider the adversary and protect the benign users and systems.

Appendix

A.1 Understanding $\overline{\sigma_{aCSI}}$

The key motivation of our attack is that $\overline{\sigma_{aCSI}}$ decreases with the target-to-anchor distance. In the following, we use an abstract, ray-tracing model to explain this phenomenon.

A ray-tracing model on $\overline{\sigma_{aCSI}}$. For an anchor x, let P_i represent the received signal power on subcarrier i (wavelength λ_i) measured by the sniffer. It can be modeled as an aggregation of multiple signal paths $\mathbb{S} = \{s_j\}_{j=1..N}$ [186]:

$$P_i = \left\| \sum_{j=1}^{N} s_j(d_j, \Gamma_j, \lambda_i) \right\| \tag{A.1}$$

where d_j and Γ_j are the propagation distance and reflection/diffraction coefficient of the signal path $j = 1..N$, respectively. For simplicity, we assume the human user moves at a constant speed around x with a fixed distance r. Similar to a widely used model [180], we consider complete blockage (Γ_i= 0 or 1) and disregard the contribution of signal phase.

When the user moves near the anchor x (at distance r), some signal paths $\mathbb{S}(r)$ are blocked. We choose the time $t_0 < t_1 < t_2 < t_3$, where the blockage happens during period

$[t_1, t_2]$. Then the received signal power at time t is

$$P_i(t,r) = \begin{cases} \| \sum_{j \notin \mathbb{S}(r)} s_j(\cdot) \| & \text{if } t \in [t_1, t_2] \\ \| \sum_j s_j(\cdot) \| & \text{otherwise} \end{cases} \tag{A.2}$$

And the standard deviation $\sigma_i(r)$ over $[t_0, t_3]$ is:

$$\sigma_i^2(r) = \frac{1}{t_3 - t_0} \int_{t_0}^{t_3} (P_i(t,r) - \mu_i(r))^2 dt \tag{A.3}$$

At distances $r_2 > r_1$, the set of blocked signal paths $|\mathbb{S}(r_2)| < |\mathbb{S}(r_1)|$, meaning $P_i(t, r_2) > P_i(t, r_1), \forall t \in [t_1, t_2]$. Since $P_i(t|t \notin [t_1, t_2], r) > P_i(t|t \in [t_1, t_2], r)$, we can derive the mean received power μ_i over time $[t_0, t_3]$ and it satisfies $\mu_i(r_2) > \mu_i(r_1)$ for $r_2 > r_1$.

Combined with Equation (A.3), it is easy to derive that $|P_i(t, r_2) - \mu_i(r_2)| < |P_i(t, r_1) - \mu_i(r_1)|$, and so $\sigma_i^2(r_2) < \sigma_i^2(r_1)$. The averaged standard deviation over i sub-carriers, *i.e.* $\overline{\sigma_{aCSI}}$, follows the same trend.

Also, since we are looking at the standard deviation rather than the absolute power, it is invariant to the constant signal penetration loss (through walls).

A.2 Details on RSS Model Fitting

Our RSS model fitting uses the log distance path loss model, which is shown to be robust in indoor environments [141]. This model captures the relation between the RSS P_i and the sniffer's distance d_i to a WiFi transmitting device (TX) when the attacker

sniffer is at a location index i:

$$\begin{aligned} P_i &= (P_{TX} - P_{REF}) - 10\gamma \log_{10}(d_i/d_{REF}) + \text{noise} \\ &= P_{TX^o} - 10\gamma \log_{10} d_i + \text{noise} \end{aligned} \tag{A.4}$$

where γ is the path loss component, P_{TX} is the transmit power of the target device TX, P_{REF} is its reference power received at distance d_{REF}, and $P_{TX^o} = P_{TX} - P_{REF} + 10\gamma \log_{10} d_{REF}$. When the attacker detects that the sniffer and the target device TX are on the same floor level (see §3.1.4), we can approximate d_i by

$$d_i \approx \sqrt{(x_i - x_{TX})^2 + (y_i - y_{TX})^2}$$

where xs and ys are 2D coordinates. If TX is detected to be on a different floor,

$$d_i \approx \sqrt{(x_i - x_{TX})^2 + (y_i - y_{TX})^2 + (z - z_{TX})^2}$$

where z and z_{TX} are vertical heights of the sniffer and the target TX. The attacker will pre-calculate $(z - z_{TX})$ using our floor level detection (§3.1.4).

The goal of RSS modeling fitting is to estimate (x_{TX}, y_{TX}) as well as (γ, P_{TX^o}), using spatial measurement of RSS values $\{P_i\}$. The corresponding model fitting is formulated into a least square optimization problem:

$$\begin{aligned} \underset{\hat{x}_{TX}, \hat{y}_{TX}, \hat{P}_{TX^o}, \hat{\gamma}}{\text{minimize}} \quad & \sum_i (P_i - \hat{P}_i)^2, \\ \text{subject to} \quad & (\hat{x}_{TX}, \hat{y}_{TX}) \in \text{Candidate area}, \\ & \hat{P}_{TX^o} \leq 30dB, \\ & \hat{\gamma} \in [2, 6] \end{aligned} \tag{A.5}$$

The constraint on $\hat{\gamma}$ follows the well-known observations from empirical measurements [187]

while the value of $\hat{P}_{TX^o}$ is upper bounded by the maximum transmit power for WiFi frequency defined by the FCC.

We also experimented with other types of propagation models. Among them, only a complicated ray-tracing model accounting the floor plan of the target building [154] achieves a marginal gain over the above log distance model. Given its high complexity and computation cost, we did not include it in the final attack. Resourceful attackers can further improve the localization by switching to more sophisticated models.

Model fitting is sensitive to noisy RSS measurements. From Equation (A.4), we can clearly see that the signal propagation distance (between the victim and the attacker) d_i satisfies the following:

$$d_i \propto 10^{(P_{TX}-P_i+noise)/10} \tag{A.6}$$

This means given a determined transmitting power P_{TX}, the lower P_i the attacker observes, the more sensitive the model is to *noise* to estimate d_i. In practice, P_{TX} is certainly much higher than the noise level $-85dB$. Low P_i happens 1) when the attacker is far away from the victim device, and 2) when additional blockage exists to degrade the signal. In either way, we would have errors while measuring the true P_i (due to fading, environmental factors, and measurement hardware). This noise creates a large uncertainty to estimate d_i at every measuring location i. This is one major reason of the large variance of localization errors across multiple rounds of measurements when the attacker moves along the same trajectory.

Ideally, the fitting method (Equation (A.5)) needs data that is sampled evenly to avoid fitting bias. To obtain the unbiased RSS samples, *e.g.*, from $-40dB$ to $-80dB$, Equation (A.4) suggests that the attacker should move near the victim device *more often* than moving away from the victim device. This is because at close distance, moving by

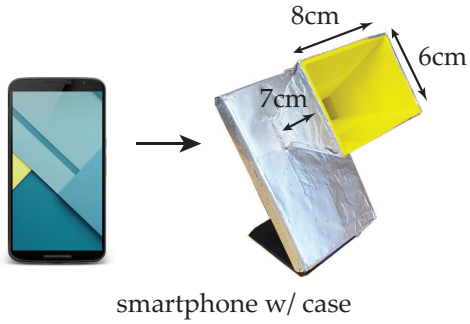


Figure A.1: Attacker app and our 3d-printed case prototype that emulates horns.

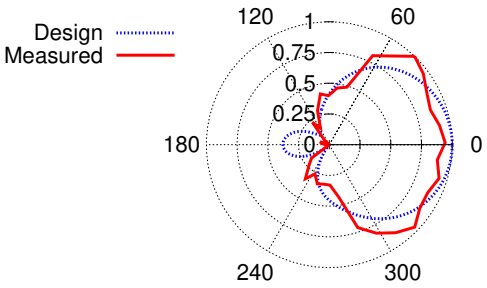


Figure A.2: Our design and the measured beam patterns closely match.

a small distance would create large RSS difference. But in practice, the attacker would not know how far away the victim device is.

A.3 Details for Floor-level signal isolation

To detect the floor level, our key intuition is that devices on the second or third floor transmit signals to the attacker receiver at a different direction comparing with the signals on the same floor. For example, the attacker who is 10m away from a victim house receives signals from the upper floor devices at least 15 degrees different from those on the ground level. So if the attacker can tell the signal incoming angle (*i.e.* angle of arrival, or AoA) with accuracy $< 15°$, the floor level of the victim device can be estimated[1].

The challenge is that today's low-cost, compact devices like smartphones cannot report AoA due to the hardware limitation, *i.e.* it has only one antenna. Even doing hardware hacks by adding more antennas, to get beams that are 15 degrees wide, we need at least 3m wide antenna size, according to the antenna theory. Clearly, the attacker cannot carry such a large, eye-catching device.

[1]Although AoA can also be used to locate devices, its accuracy must be < 5 degrees for meter-level localization at 10m distance [116]. This needs a large antenna array, which is not applicable in our scenarios.

Reshaping beams by horn antenna emulation. Inspired by [154], we address the above problem by reshaping the antenna beam. Specifically, we create a phone case, wrapped with aluminum foils that emulates the horn antenna (Figure A.1). The design is from MATLAB [188], and the measured beam pattern closely matches the design (Figure A.2).

Due to the fixed beam direction, the attacker needs to physically rotate the phone and record the angle changes via gyroscope. To mitigate impacts from the rotation artifacts, sensor noise, and the multi-path angles, we combine 4 or more locations of measurements that are in a meter range and average them[2]. We then derive the measured AoA as the direction that matches the beam pattern the most (in Figure A.2).

[2]We can combine the measurements since AoA is not sensitive to small movements (in a meter) when an attacker is meters away.

Bibliography

[1] "Amazon prime air." https://www.amazon.com/Amazon-Prime-Air/, 2019.

[2] T. Rehkopf, "DHL launches its first regular fully-automated and intelligent urban drone delivery service." https://www.dpdhl.com/en/media-relations/press-releases/2019/dhl-launches-its-first-regular-fully-automated-and-intelligent-urban-drone-delivery-service.html, 2019.

[3] "UPS drones are now moving blood samples over north carolina." https://www.wired.com/story/ups-matternet-drone-delivery-north-carolina/, 2019.

[4] L. Segall, "Meet QuiQui, the drug-delivering drone." https://money.cnn.com/2014/06/19/technology/innovation/quiqui-drone-drugs/, 2014.

[5] J. Bright, "Drone delivery startup zipline launches UAV medical program in ghana." https://techcrunch.com/2019/04/24/drone-delivery-startup-zipline-launches-uav-medical-program-in-ghana/, 2019.

[6] O. Oralkan, A. S. Ergun, J. A. Johnson, M. Karaman, U. Demirci, K. Kaviani, T. H. Lee, and B. T. Khuri-Yakub, *Capacitive micromachined ultrasonic transducers: Next-generation arrays for acoustic imaging?*, *IEEE transactions on ultrasonics, ferroelectrics, and frequency control* **49** (2002), no. 11.

[7] C. Adams, D. Holbrook, and R. Sengsten, *A handheld active millimeter wave camera*, in *Proc. of HST*, 2010.

[8] K. Liu, X. Wang, J. Samarabandu, and A. Akhtar, *Monostatic airborne SAR using license exempt WiMAX transceivers*, in *Proc. of VTC*, 2014.

[9] E. Ackerman, "Sweep is a $250 LIDAR with range of 40 meters that works outdoors." http://spectrum.ieee.org/automaton/robotics/robotics-hardware/sweep-lidar-for-robots-and-drones, 2016.

[10] O. Solon, "Lidar: the self-driving technology that could help Tesla avoid another tragedy." https://www.theguardian.com/technology/2016/jul/06/lidar-self-driving-technology-tesla-crash-elon-musk, 2016.

[11] "Slamtec." http://www.slamtec.com/, 2019.

[12] F. Adib, Z. Kabelac, D. Katabi, and R. C. Miller, *3d tracking via body radio reflections*, in *Proc. of NSDI*, 2014.

[13] F. Adib, C.-Y. Hsu, H. Mao, D. Katabi, and F. Durand, *Capturing the human figure through a wall*, *ACM Transactions on Graphics* **34** (2015), no. 6.

[14] F. Adib, Z. Kabelac, and D. Katabi, *Multi-person localization via RF body reflections*, in *Proc. of NSDI*, 2015.

[15] N. Lievendag, "Autodesk 123d catch." https://3dscanexpert.com/autodesk-photogrammetry-review-123d-catch/, 2016.

[16] N. Lievendag, "123D catch alternative: ReCap 360 mobile workflow." https://3dscanexpert.com/123d-catch-alternative-recap-360-sketchfab-mobile-android-workflow/, 2018.

[17] F. Lambert, "Understanding the fatal Tesla accident on autopilot and the NHTSA probe." https://electrek.co/2016/07/01/understanding-fatal-tesla-accident-autopilot-nhtsa-probe/, 2016.

[18] S. Lohr, "A lesson of tesla crashes? Computer vision cant do it all yet." http://www.nytimes.com/2016/09/20/science/computer-vision-tesla-driverless-cars.html, 2016.

[19] S. Shankland, "Wilocity: 2015 phones getting extra-fast 802.11ad networking." https://www.cnet.com/news/wilocity-2015-phones-getting-extra-fast-802-11ad-networking/, 2014.

[20] "Qualcomm bolsters Wi-Fi leadership with 60 GHz wireless for mobile, computing and networking." https://www.qualcomm.com/news/releases/2014/07/02/qualcomm-bolsters-wi-fi-leadership-60-ghz-wireless-mobile-computing-and, 2014.

[21] S. Martin, "Scientists now able to "SEE through walls" using Wi-Fi." https://www.express.co.uk/news/science/808289/x-ray-vision-wifi-Technical-University-of-Munich, 2017.

[22] Y. Zhu, Z. Zhang, Z. Marzi, C. Nelson, U. Madhow, B. Y. Zhao, and H. Zheng, *Demystifying 60GHz outdoor picocells*, in *Proc. of MobiCom*, 2014.

[23] Y. K. Chan and V. C. Koo, *An introduction to synthetic aperture radar (SAR)*, *Progress In Electromagnetics Research* **2** (2008).

[24] J. Borenstein and Y. Koren, *Real-time obstacle avoidance for fast mobile robots in cluttered environments*, in *Proc. of ICRA*, 1990.

[25] J. Borenstein and Y. Koren, *The vector field histogram-fast obstacle avoidance for mobile robots*, *IEEE Transactions on Robotics and Automation* **7** (1991), no. 3.

[26] P. Newman, M. Bosse, and L. Leonard, *Autonomous feature-based exploration*, in *Proc. of ICRA*, 2003.

[27] Y. Zhu, Y. Zhu, Z. Zhang, B. Y. Zhao, and H. Zheng, *60GHz mobile imaging radar*, in *Proc. of HotMobile*, 2015.

[28] M. Line, "Robot rescue: First-responders of the future." `https://www.foxnews.com/tech/robot-rescue-first-responders-of-the-future`, 2015.

[29] A. C. Madrigal, "The trick that makes Google's self-driving cars work." `http://www.theatlantic.com/technology/print/2014/05/all-the-world-a-track-the-trick-that-makes-googles-self-driving-cars-work/370871/`, 2014.

[30] N. Al-Salihi, *Precise positioning in real-time for visually impaired people using navigation satellites*, *International Journal of Engineering and Technology* **12** (2010), no. 2.

[31] L. A. Guerrero, F. Vasquez, and S. F. Ochoa, *An indoor navigation system for the visually impaired*, *Sensors* **12** (2012).

[32] Z. Zhang, D. Chu, X. Chen, and T. Moscibroda, *SwordFight: Enabling a new class of phone-to-phone action games on commodity phones*, in *Proc. of MobiSys*, 2012.

[33] "IEEE 802.11 task group ad." `http://www.ieee802.org/11/Reports/tgad_update.htm`, 2012.

[34] B. Langen, G. Lober, and W. Herzig, *Reflection and transmission behavior of building materials at 60GHz*, in *Proc. of PIMRC*, 1994.

[35] D. Halperin, S. Kandula, J. Padhye, P. Bahl, and D. Wetherall, *Augmenting data center networks with multi-gigabit wireless links*, in *Proc. of SIGCOMM*, 2011.

[36] 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.

[37] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, *Principles of modern radar*. The Institution of Engineering and Technology, 2013.

[38] J. F. Federici, D. Gary, R. Barat, and D. Zimdars, *THz standoff detection and imaging of explosives and weapons*, in *Proc. of Defense and Security*, 2005.

[39] T.-F. Tseng, J.-M. Wun, W. Chen, S.-W. Peng, J.-W. Shi, and C.-K. Sun, *High-resolution 3-dimensional radar imaging based on a few-cycle W-band photonic millimeter-wave pulse generator*, in *Optical Fiber Communication Conference*, 2013.

[40] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse, *MonoSLAM: Real-time single camera SLAM*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2007).

[41] D. G. Lowe, *Object recognition from local scale-invariant features*, in *Proc. of ICCV*, 1999.

[42] D. G. Lowe, *Distinctive image features from scale-invariant keypoints*, *International journal of computer vision* **60** (2004), no. 2.

[43] J. Mundy, "What is Project Tango? Google's new AR tech explained." `https://www.trustedreviews.com/news/what-is-project-tango-2941129`, 2017.

[44] F. Adib and D. Katabi, *See through walls with Wi-Fi!*, in *Proc. of SIGCOMM*, 2013.

[45] K. Chetty, G. E. Smith, and K. Woodbridge, *Through-the-wall sensing of personnel using passive bistatic wifi radar at standoff distances*, *Trans. on Geoscience and Remote Sensing* **50** (2012), no. 4.

[46] Y. Mostofi, *Cooperative wireless-based obstacle/object mapping and see-through capabilities in robotic networks*, *IEEE TMC* (2013).

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

[48] D. Bharadia, K. R. Joshi, and S. Katti, *Full duplex backscatter*, in *Proc. of HotNets*, 2013.

[49] S. Kumar, S. Gil, D. Katabi, and D. Rus, *Accurate indoor localization with zero start-up cost*, in *Proc. of MobiCom*, 2014.

[50] Y. Zhu, Y. Zhu, B. Y. Zhao, and H. Zheng, *Reusing 60GHz radios for mobile radar imaging*, in *Proc. of MobiCom*, 2015.

[51] "Meet the drones patrolling Barcelona's sewers." `http://www.cnet.com/news/meet-the-drones-patrolling-the-pipes-of-barcelonas-sewers/`.

[52] C. Peng, G. Shen, Y. Zhang, Y. Li, and K. Tan, *BeepBeep: A high accuracy acoustic ranging system using COTS mobile devices*, in *Proc. of SenSys*, 2007.

[53] D. Huang, R. Nandakumar, and S. Gollakota, *Feasibility and limits of Wi-Fi imaging*, in *Proc. of SenSys*, 2014.

[54] C. Adams, D. Holbrook, and R. Sengsten, *A handheld active millimeter wave camera*, in *Proc. of IEEE HST*, 2010.

[55] R. Appleby and R. Anderton, *Millimeter-wave and submillimeter-wave imaging for security and surveillance*, *Proceedings of the IEEE* **95** (2007), no. 8.

[56] J. Mackenzie and E. Brown-Kenyon, *Wide-bandwidth mobile radar for ISAR/SAR radar imaging*, in *IEE Colloquium on Radar and Microwave Imaging*, 1994.

[57] M. Sato and K. Mizuno, *Millimeter-wave imaging sensor*. InTech, 2010.

[58] M. Cheney and B. Borden, *Fundamentals of Radar Imaging*, vol. 79. SIAM, 2009.

[59] "HP Elite x2 1011 G1 datasheet." `http://www8.hp.com/us/en/ads/elite-products/elitex2-1011.html`.

[60] X. Qiu, C. Ding, and D. Hu, *Bistatic SAR Data Processing Algorithms*. Wiley, 2013.

[61] M. Niessner, A. Dai, and M. Fisher, *Combining inertial navigation and ICP for real-time 3D surface reconstruction*, in *Proc. of Eurographics*, 2014.

[62] V. C. Koo, T. S. Lim, and H. T. Chuah, *A comparison of autofocus algorithms for SAR imagery*, in *Proc. of PIERS*, 2005.

[63] K. Kulpa, M. Purchla-Malanowska, and M. P. Malanowski, *Improvement of resolution in real-time unfocused SAR algorithm*, in *Proc. of EuSAR*, 2004.

[64] S. Kumar, S. Gil, D. Katabi, and D. Rus, *Accurate indoor localization with zero start-up cost*, in *Proc. of MobiCom*, 2014.

[65] E. Hecht, *Optics*. Addison-Wesley, 2002.

[66] H. Zhang, S. Venkateswaran, and U. Madhow, *Channel modeling and MIMO capacity for outdoor millimeter wave links*, in *Proc. of WCNC*, 2010.

[67] S. Juds, *Photoelectric Sensors and Controls: Selection and Application, First Edition*. Taylor & Francis, 1988.

[68] V. Zhurbenko, *Electromagnetic Waves*. InTech, 2011.

[69] M. Müller, *Information Retrieval for Music and Motion*. Springer Berlin Heidelberg, 2007.

[70] "Sensor fusion on android devices: A revolution in motion processing." `http://davidcrowley.me/?p=370`.

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

[72] H. T. Friis, *A note on a simple transmission formula*, *Proc. of IRE* **34** (1946), no. 5.

[73] Valdes-Garcia *et. al.*, *Single-element and phased-array transceiver chipsets for 60-GHz Gb/s communications*, *IEEE Communications Magazine* **49** (2011), no. 4.

[74] "Draft standard - part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications - amendment 4: Enhancements for very high throughput in the 60ghz band." IEEE P802.11adTM/D9.0, July, 2012.

[75] F. Gu, Q. Zhang, H. Lou, Z. Li, and Y. Luo, *Two-dimensional sparse synthetic aperture radar imaging method with stepped-frequency waveform*, *Journal of Applied Remote Sensing* **9** (2015), no. 1.

[76] J. Grajal, A. Badolato, G. Rubio-Cidre, L. Ubeda-Medina, B. Mencia-Oliva, A. Garcia-Pino, B. Gonzalez-Valdes, and O. Rubinos, *3-D high-resolution imaging radar at 300 ghz with enhanced FoV*, *Microwave Theory and Techniques, IEEE Transactions on* **63** (2015), no. 3.

[77] A. Fricke, S. Rey, M. Achir, P. Le Bars, T. Kleine-Ostmann, and T. Kurner, *Reflection and transmission properties of plastic materials at THz frequencies*, in *Proc. of IRMMW-THz*, 2013.

[78] M. Seo, B. Ananthasubramaniam, U. Madhow, and M. J. Rodwell, *Millimeterwave (60 GHz) imaging wireless sensor network: Recent progress*, in *Proc. of ACSSC*, 2007.

[79] Y. Zhu, Y. Yuanshun Kevin, B. Y. Zhao, and H. Zheng, *Object recognition and navigation using a single networking device*, in *Proc. of Mobisys*, 2017.

[80] A. Cuthbertson, "Amazon drone delivers receive U.K. approval." `https://www.newsweek.com/amazon-drone-deliveries-get-uk-approval-483918`, 2016.

[81] A. Liptak, "7-Eleven just made the first commercial delivery by drone." `http://www.theverge.com/2016/7/23/12262468/7-11-first-retailer-deliver-food-drone`, 2016.

[82] S. Brewster, "Uber starts self-driving car pickups in Pittsburgh." `https://techcrunch.com/2016/09/14/1386711/`, 2016.

[83] D. Cameron and A. Cuadra, "Meet the future first responders." `https://www.washingtonpost.com/graphics/business/robots/`, 2015.

[84] D. Gershgorn, "Google's robots are learning how to pick things up." `http://www.popsci.com/googles-robots-are-learning-hand-eye-coordination-with-artificial-intelligence`, 2016.

[85] S. Depatla, L. Buckland, and Y. Mostofi, *X-ray vision with only WiFi power measurements using Rytov wave models*, *IEEE Transactions on Vehicular Technology* **64** (2015).

[86] S. Sur, V. Venkateswaran, X. Zhang, and P. Ramanathan, *60 GHz indoor networking through flexible beams: A link-level profiling*, in *Proc. of SIGMETRICS*, 2015.

[87] M. Soumekh, *Synthetic aperture radar signal processing*. New York: Wiley, 1999.

[88] B. Mamandipoor, G. Malysa, A. Arbabian, U. Madhow, and K. Noujeim, *60 GHz synthetic aperture radar for short-range imaging: Theory and experiments*, in *Proc. of ASILOMAR*, 2014.

[89] T. Wei and X. Zhang, *mTrack: High-precision passive tracking using millimeter wave radios*, in *Proc. of MobiCom*, 2015.

[90] S. Lanzisera, D. T. Lin, and K. S. J. Pister, *RF time of flight ranging for wireless sensor network localization*, in *Proc. of WISES*, 2006.

[91] S. Thrun, W. Burgard, and D. Fox, *Probabilistic robotics*. MIT press, 2005.

[92] K. Hosoya, N. Prasad, K. Ramachandran, N. Orihashi, S. Kishimoto, S. Rangarajan, and K. Maruhashi, *Multiple sector ID capture (MIDC): A novel beamforming technique for 60-GHz band multi-Gbps WLAN/PAN systems*, *IEEE Transactions on Antennas and Propagation* **63** (2015), no. 1.

[93] T. Cover and P. Hart, *Nearest neighbor pattern classification*, *IEEE Transactions on Information Theory* **13** (1967), no. 1.

[94] H. R. Anderson, *A ray-tracing propagation model for digital broadcast systems in urban areas*, *IEEE Transactions on Broadcasting* **39** (1993), no. 3.

[95] F. Hacıvelioğlu, M. A. Uslu, and L. Sevgi, *A MATLAB-based virtual tool for the electromagnetic wave scattering from a perfectly reflecting wedge*, *IEEE Antennas and Propagation Magazine* **53** (2011), no. 6.

[96] http://www.nexusrobot.com/.

[97] "Introducing facebook's new terrestrial connectivity systems: Terragraph and project aries." https://code.facebook.com/posts/1072680049445290/introducing-facebook-s-new-terrestrial-connectivity-systems-terragraph-and-project-aries/.

[98] http://events.linuxfoundation.org/events/embedded-linux-conference/.

[99] "San josé partners with facebook for high-speed outdoor Wi-Fi." https://gcn.com/articles/2016/04/18/san-jose-facebook.aspx.

[100] https://apps.fcc.gov/edocs_public/attachmatch/FCC-13-112A1_Rcd.pdf.

[101] S. Se, D. Lowe, and J. Little, *Vision-based mobile robot localization and mapping using scale-invariant features*, in *Proc. of ICRA*, 2001.

[102] S. Se, D. G. Lowe, and J. J. Little, *Vision-based global localization and mapping for mobile robots*, *IEEE Transactions on Robotics* **21** (2005), no. 3.

[103] R. C. Smith and P. Cheeseman, *On the representation and estimation of spatial uncertainty*, *The International Journal of Robotics Research* **5** (1986), no. 4.

[104] L. Sun, S. Sen, D. Koutsonikolas, and K. Kim, *WiDraw: Enabling handsfree drawing in the air on commodity WiFi devices*, in *Proc. of MobiCom*, 2015.

[105] S. Tan and J. Yang, *WiFinger: Leveraging commodity wifi for fine-grained finger gesture recognition*, in *Proc. of MobiHoc*, 2016.

[106] R. Nandakumar, S. Gollakota, and N. Watson, *Contactless sleep apnea detection on smartphones*, in *Proc. of MobiSys*, 2015.

[107] J. Wang, K. Zhao, X. Zhang, and C. Peng, *Ubiquitous keyboard for small mobile devices: Harnessing multipath fading for fine-grained keystroke localization*, in *Proc. of MobiSys*, 2014.

[108] M. Molerón and C. Daraio, *Acoustic metamaterial for subwavelength edge detection*, *Nature Communications* **6** (2015), no. 8037.

[109] J. Y. Zheng and A. Murata, *Acquiring a complete 3D model from specular motion under the illumination of circular-shaped light sources*, *IEEE Transactions on Pattern Analysis and Machine Intelligence* **22** (2000), no. 8.

[110] I. Ihrke, K. N. Kutulakos, H. Lensch, M. Magnor, and W. Heidrich, *Transparent and specular object reconstruction*, *Computer Graphics Forum* **29** (2010), no. 8.

[111] K. L. Lueth, "State of the IoT 2018: Number of IoT devices now at 7B - market accelerating." https://iot-analytics.com/state-of-the-iot-update-q1-q2-2018-number-of-iot-devices-now-7b/, 2018.

[112] S. Liu, "Number of wireless local area network (WLAN) connected devices worldwide from 2016 to 2021 (in billions)." https://www.statista.com/statistics/802706/world-wlan-connected-device/, 2019.

[113] B. Bennett, "The first 5 things to do with new smart lights." https://www.cnet.com/how-to/the-first-5-things-to-do-with-your-smart-lights/, 2018.

[114] "Thermal imaging FAQ." https://pr-infrared.com/about-thermal-imaging/thermal-imaging-faq.

[115] N. Apthorpe, D. Reisman, S. Sundaresan, A. Narayanan, and N. Feamster, *Spying on the smart home: Privacy attacks and defenses on encrypted IoT traffic*, *CoRR* **abs/1708.05044** (2017).

[116] J. Xiong and K. Jamieson, *ArrayTrack: A fine-grained indoor location system*, in *Proc. of NSDI*, 2013.

[117] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, *SpotFi: Decimeter level localization using WiFi*, in *Proc. of SIGCOMM*, 2015.

[118] M. Youssef, A. Youssef, C. Rieger, U. Shankar, and A. Agrawala, *PinPoint: An asynchronous time-based location determination system*, in *Proc. of MobiSys*, 2006.

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

[120] Z. Farid, R. Nordin, and M. Ismail, *Recent advances in wireless indoor localization techniques and system*, *Journal of Computer Networks and Communications* **2013** (2013).

[121] J. Wang, H. Jiang, J. Xiong, K. Jamieson, X. Chen, D. Fang, and B. Xie, *LiFS: Low human-effort, device-free localization with fine-grained subcarrier information*, in *Proc. of MobiCom*, 2016.

[122] L. Yang, Y. Chen, X.-Y. Li, C. Xiao, M. Li, and Y. Liu, *Tagoram: Real-time tracking of mobile RFID tags to high precision using COTS devices*, in *Proc. of MobiCom*, 2014.

[123] H. Huang and S. Lin, *WiDet: Wi-Fi based device-free passive person detection with deep convolutional neural networks*, in *Proc. of MSWIM*, 2018.

[124] K. Qian, C. Wu, Y. Zhang, G. Zhang, Z. Yang, and Y. Liu, *Widar2.0: Passive human tracking with a single Wi-Fi link*, in *Proc. of MobiSys*, 2018.

[125] M. Youssef, M. Mah, and A. Agrawala, *Challenges: device-free passive localization for wireless environments*, in *Proc. of MobiCom*, 2007.

[126] A. Banerjee, D. Maas, M. Bocca, N. Patwari, and S. Kasera, *Violating privacy through walls by passive monitoring of radio windows*, in *Proc. of WiSec*, 2014.

[127] "Smart layouts." `https://onefirefly.com/creative-services/smart-layouts`, 2019.

[128] "Smart home layout." `https://www.hornernetworks.com/smart-home-layout`, 2019.

[129] I. Sanchez, R. Satta, I. N. Fovino, G. Baldini, G. Steri, D. Shaw, and A. Ciardulli, *Privacy leakages in smart home wireless technologies*, in *Proc. of ICCST*, 2014.

[130] S. Siby, R. R. Maiti, and N. O. Tippenhauer, *IoTScanner: Detecting privacy threats in IoT neighborhoods*, in *Proc. of IoTPTS*, 2017.

[131] J. Martin, T. Mayberry, C. Donahue, L. Foppe, L. Brown, C. Riggins, E. C. Rye, and D. Brown, *A study of MAC address randomization in mobile devices and when it fails*, *CoRR* **abs/1703.02874** (2017).

[132] M. Rouse, "Evil maid attack." `http://searchsecurity.techtarget.com/definition/evil-maid-attack`, 2018.

[133] "USRP software defined radio (SDR) online catalog." `https://www.ettus.com/product/`, 2018.

[134] K. Joshi, D. Bharadia, M. Kotaru, and S. Katti, *WiDeo: Fine-grained device-free motion tracing using RF backscatter*, in *Proc. of NSDI*, 2015.

[135] D. Vasisht, S. Kumar, and D. Katabi, *Decimeter-level localization with a single WiFi access point*, in *Proc. of NSDI*, 2016.

[136] C. R. Karanam, B. Korany, and Y. Mostofi, *Magnitude-based angle-of-arrival estimation, localization, and target tracking*, in *Proc. of IPSN*, 2018.

[137] H. Yu, B. Yang, J. Liu, and G. Yu, *Passive human trajectory tracking study in indoor environment with CSI*, in *Proc. of NaNA*, 2018.

[138] F. R. Hampel, *The influence curve and its role in robust estimation*, *Journal of the American Statistical Association* **69** (1974), no. 346 383–393.

[139] P. J. Rousseeuw and C. Croux, *Alternatives to the median absolute deviation*, *Journal of the American Statistical association* **88** (1993), no. 424 1273–1283.

[140] D. Vasisht, A. Jain, C.-Y. Hsu, Z. Kabelac, and D. Katabi, *Duet: Estimating user position and identity in smart homes using intermittent and incomplete RF-data*, in *Proc. of UbiComp*, 2018.

[141] L. Li, G. Shen, C. Zhao, T. Moscibroda, J.-H. Lin, and F. Zhao, *Experiencing and handling the diversity in data density and environmental locality in an indoor positioning service*, in *Proc. of MobiCom*, 2014.

[142] Z. Li, Z. Xiao, Y. Zhu, I. Pattarachanyakul, B. Y. Zhao, and H. Zheng, *Adversarial localization against wireless cameras*, in *Proc. of HotMobile*, 2018.

[143] J. Seybold, *Introduction to RF Propagation*. Wiley, 2005.

[144] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse, *Multipath triangulation: Decimeter-level WiFi localization and orientation with a single unaided receiver*, in *Proc. of MobiSys*, 2018.

[145] F. Evennou and F. Marx, *Advanced integration of WiFi and inertial navigation systems for indoor mobile positioning*, *EURASIP J. Appl. Signal Process* **2006** (2006).

[146] S. Tan and J. Yang, *WiFinger: Leveraging commodity WiFi for fine-grained finger gesture recognition*, in *Proc. of MobiHoc*, 2016.

[147] P. K. Sen and J. M. Singer, eds., *Large sample methods in statistics*. Chapman & Hall, Inc., 1989.

[148] "HOWTO estimate parameter-errors using Monte Carlo." `http://www-personal.umd.umich.edu/~wiclarks/AstroLab/HOWTOs/NotebookStuff/MonteCarloHOWTO.html`, 2014.

[149] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, *Tool release: Gathering 802.11n traces with channel state information*, *ACM SIGCOMM CCR* **41** (2011), no. 1.

[150] M. Schulz, J. Link, F. Gringoli, and M. Hollick, *Shadow Wi-Fi: Teaching smartphones to transmit raw signals and to extract channel state information to implement practical covert channels over Wi-Fi*, in *Proc. of MobiSys*, 2018.

[151] C. Matte and M. Cunche, *Spread of MAC address randomization studied using locally administered mac addresses use historic*, *RR-9142, Inria Grenoble Rhône-Alpes* (2017).

[152] "Researchers break MAC address randomization and track 100% of test devices." https://www.bleepingcomputer.com/news/security/researchers-break-mac-address-randomization-and-track-100-percent-of-test-devices/, 2017.

[153] "Android P feature spotlight: Per-network MAC address randomization added as experimental feature." https://www.androidpolice.com/2018/03/08/android-p-feature-spotlight-per-network-mac-address-randomization-added-experimental-feature/, 2018.

[154] X. Xiong, J. Chan, E. Yu, N. Kumari, A. A. Sani, C. Zheng, and X. Zhou, *Customizing indoor wireless coverage via 3D-fabricated reflectors*, in *Proc. of BuildSys*, 2017.

[155] Z. Li, Y. Xie, L. Shangguan, R. I. Zelaya, J. Gummeson, W. Hu, and K. Jamieson, *Towards programming the radio environment with large arrays of inexpensive antennas*, in *Proc. of NSDI*, 2019.

[156] Y. Qiao, O. Zhang, W. Zhou, K. Srinivasan, and A. Arora, *PhyCloak: Obfuscating sensing from communication signals*, in *Proc. of NSDI*, 2016.

[157] H. Li, Y. He, L. Sun, X. Cheng, and J. Yu, *Side-channel information leakage of encrypted video stream in video surveillance systems*, in *Proc. of INFOCOM*, 2016.

[158] F. Zhang, W. He, X. Liu, and P. G. Bridges, *Inferring users' online activities through traffic analysis*, in *Proc. of WiSec*, 2011.

[159] Y. Cheng, X. Ji, T. Lu, and W. Xu, *DeWiCam: Detecting hidden wireless cameras via smartphones*, in *Proc. of Asia CCS*, 2018.

[160] S. V. Radhakrishnan, A. S. Uluagac, and R. Beyah, *GTID: A technique for physical device and device type fingerprinting*, *IEEE Transactions on Dependable and Secure Computing* **12** (2015), no. 5.

[161] T. OConnor, R. Mohamed, M. Miettinen, W. Enck, B. Reaves, and A.-R. Sadeghi, *Homesnitch: Behavior transparency and control for smart home iot devices*, in *Proc. of WiSec*, 2019.

[162] A. Acar, H. Fereidooni, T. Abera, A. K. Sikder, M. Miettinen, H. Aksu, M. Conti, A. Sadeghi, and A. S. Uluagac, *Peek-a-boo: I see your smart home activities, even encrypted!*, *CoRR* **abs/1808.02741** (2018).

[163] J. Zhang, G. Tian, A. M. J. Marindra, A. Imam, and A. Zhao, *A review of passive RFID tag antenna-based sensors and systems for structural health monitoring applications*, *Sensors* **17** (2017).

[164] C. Zhang and X. Zhang, *LiTell: Robust indoor localization using unmodified light fixtures*, in *Proc. of MobiCom*, 2016.

[165] T. Li, Q. Liu, and X. Zhou, *Practical human sensing in the light*, in *Proc. of MobiSys*, 2016.

[166] W. Mao, J. He, and L. Qiu, *CAT: High-precision acoustic motion tracking*, in *Proc. of MobiCom*, 2016.

[167] R. Nandakumar, A. Takakuwa, T. Kohno, and S. Gollakota, *CovertBand: Activity information leakage using music*, in *Proc. of UbiComp*, 2017.

[168] R. Nandakumar, B. Kellogg, and S. Gollakota, *Wi-Fi gesture recognition on existing devices*, *CoRR* **abs/1411.5394** (2014).

[169] Y. Wang, J. Liu, Y. Chen, M. Gruteser, J. Yang, and H. Liu, *E-eyes: Device-free location-oriented activity identification using fine-grained WiFi signatures*, in *Proc. of MobiCom*, 2014.

[170] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, *Understanding and modeling of WiFi signal based human activity recognition*, in *Proc. of MobiCom*, 2015.

[171] V. Srinivasan, J. Stankovic, and K. Whitehouse, *Protecting your daily in-home activity information from a wireless snooping attack*, in *Proc. of UbiComp*, 2008.

[172] M. Seifeldin, A. Saeed, A. E. Kosba, A. El-Keyi, and M. Youssef, *Nuzzer: A large-scale device-free passive localization system for wireless environments*, *IEEE Transactions on Mobile Computing* **12** (2013), no. 7.

[173] S. Sigg, M. Scholz, S. Shi, Y. Ji, and M. Beigl, *Rf-sensing of activities from non-cooperative subjects in device-free recognition systems using ambient and local signals*, *IEEE Transactions on Mobile Computing* **13** (2014), no. 4.

[174] N. Xiao, P. Yang, Y. Yan, H. Zhou, and X. Li, *Motion-Fi: Recognizing and counting repetitive motions with passive wireless backscattering*, in *Proc. of INFOCOMM*, 2018.

[175] T. Xin, B. Guo, Z. Wang, M. Li, Z. Yu, and X. Zhou, *Freesense: Indoor human identification with wi-fi signals*, in *Proc. of GLOBECOM*, 2016.

[176] S. Yousefi, H. Narui, S. Dayal, S. Ermon, and S. Valaee, *A survey on behavior recognition using WiFi channel state information*, *IEEE Communications Magazine* **55** (2017).

[177] W. Jiang, C. Miao, F. Ma, S. Yao, Y. Wang, Y. Yuan, H. Xue, C. Song, X. Ma, D. Koutsonikolas, W. Xu, and L. Su, *Towards environment independent device free human activity recognition*, in *Proc. of MobiCom*, 2018.

[178] K. Chetty, G. E. Smith, and K. Woodbridge, *Through-the-wall sensing of personnel using passive bistatic WiFi radar at standoff distances*, *IEEE Transactions on Geoscience and Remote Sensing* **50** (2012), no. 4.

[179] M. Kotaru and S. Katti, *Position tracking for virtual reality using commodity WiFi*, in *Proc. of CVPR*, 2017.

[180] Y. Ji, S. Biaz, S. Pandey, and P. Agrawal, *ARIADNE: A dynamic indoor signal map construction and localization system*, in *Proc. of MobiSys*, 2006.

[181] A. Goswami, L. E. Ortiz, and S. R. Das, *WiGEM: A learning-based approach for indoor localization*, in *Proc. of CoNEXT*, 2011.

[182] Y. S. Kim, P. Tague, H. Lee, and H. Kim, *Carving secure Wi-Fi zones with defensive jamming*, in *Proc. of Asia CCS*, 2012.

[183] S. Gollakota and D. Katabi, *iJam: Jamming oneself for secure wireless communication*, tech. rep., Computer Science and Artificial Intelligence Laboratory Technical Report, 2010.

[184] T. Wang, Y. Liu, Q. Pei, and T. Hou, *Location-restricted services access control leveraging pinpoint waveforming*, in *Proc. of CCS*, 2015.

[185] M. Khaledi, M. Khaledi, S. K. Kasera, and N. Patwari, *Preserving location privacy in radio networks using a stackelberg game framework*, in *Proc. of Q2SWinet*, 2016.

[186] Z. Yang, Z. Zhou, and Y. Liu, *From RSSI to CSI: Indoor localization via channel response*, *ACM Comput. Surv.* **46** (2013), no. 2.

[187] M. Tsai, *Path-loss and shadowing (large-scale fading)*, tech. rep., 2011.

[188] "Create horn antenna." `https://www.mathworks.com/help/antenna/ref/horn.html`, 2018.