Object Recognition and Navigation using a Single Networking Device

Yanzi Zhu, Yuanshun Yao, Ben Y. Zhao and Haitao Zheng
Department of Computer Science, UC Santa Barbara
{yanzi, yao, ravenben, htzheng}@cs.ucsb.edu

ABSTRACT

Tomorrow's autonomous mobile devices need accurate, robust and real-time sensing of their operating environment. Today's solutions fall short. Vision or acoustic-based techniques are vulnerable against challenging lighting conditions or background noise, while more robust laser or RF solutions require either bulky expensive hardware or tight coordination between multiple devices.

This paper describes the design, implementation and evaluation of Ulysses, a practical environmental imaging system using colocated 60GHz radios on a *single* mobile device. Unlike alternatives that require specialized hardware, Ulysses reuses low-cost commodity networking chipsets available today. Ulysses' new imaging approach leverages RF beamforming, operates on specular (direct) reflection, and integrates the device's movement trajectory with sensing. Ulysses also includes a navigation component that uses the same 60GHz radios to compute "safety regions" where devices can move freely without collision, and to compute optimal paths for imaging within safety regions. Using our implementation of a small robotic car prototype, our experimental results show that Ulysses images objects meters away with cm-level precision, and provides accurate estimates of objects' surface materials.

1. INTRODUCTION

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 [20], and retailers like 7-up have performed drone deliveries on a smaller scale [33]. Meanwhile, Uber has already deployed a pilot program for self-driving passenger pickup vehicles in Pittsburgh [17]. 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 [18, 23].

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.

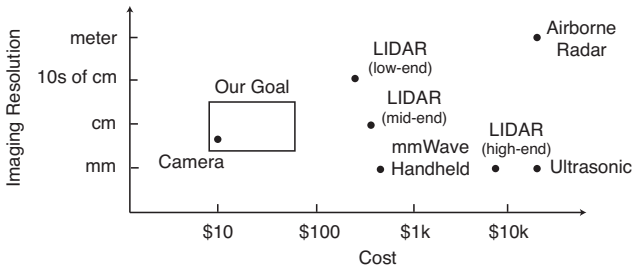


Figure 1: Comparing existing commercial single-device imaging products.

Of the numerous imaging products available today, few if any are appropriate for autonomous mobile devices. In Figure 1, we classify potential imaging solutions based on their precision and cost, including products based on sonar [40], computer vision [1], radar [9, 34] and LIDAR [2, 8, 45]. Nearly all of these approaches require specialized hardware that make them too costly (*e.g.,* $>$ \$250) and unwieldly for mobile devices[1]. 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 [35, 30]).

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 [11, 12, 13]. 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 [26] or two independently moving WiFi or 60GHz radios [22, 57].

[1]The cheapest mmWave handheld imager costs \$500, weighs 5lb [9], and the cheapest low-end LIDAR costs \$250 and offers 1D imaging at a resolution of tens of cms [8].

MobiSys '17, June 19–23, 2017, Niagara Falls, NY, USA.

DOI: http://dx.doi.org/10.1145/3081333.3081339

Imaging via a Single Networking Device. Our goal is to further advance the state of imaging systems based on commodity networking devices. Like [57, 22, 26], we 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 paper describes the design, implementation and evaluation of *Ulysses*, an object imaging system using a single compact mobile device and on-board 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* reflection 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 [57], 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 [1] and dual-device 60GHz imaging radar [57]. 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 mobile devices. We discuss the limitations of our current design in §7, and potential solutions to be explored in upcoming work.

2. 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.

2.1 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 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. These 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.

2.2 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 [48, 55].

Figure 3 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 $\times$ 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 [51].

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

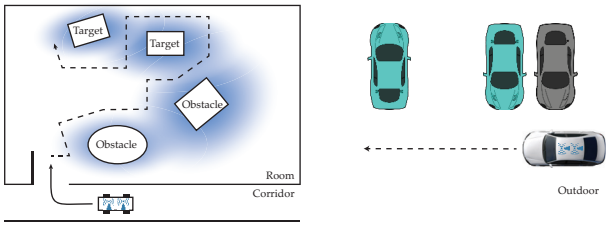


Figure 2: Illustration of two target scenarios where a robot explores an unknown room to image target objects, and a car drives by a set of parked cars to image them.

Figure 3: 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.

our scenarios. Heavy rain only adds 0.2dB signal loss for a 10m imaging range.

2.3 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 [9, 10]) 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 [22] 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 [46] and variations, and the RSA algorithm for 60GHz mobile imaging [57], which we discuss next.

M-SAR and phase-based solutions. Using colocated transmitter and receiver, M-SAR [46] 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. Prior work [57] 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 [26], near-field mmWave object imaging [36] and tracking [53] all rely on accurate phase construction, and fail to apply here.

RSA. Developed for dual-device 60GHz mobile imaging [57], 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).

2.4 Key Challenges

Challenge 1: Large phase noises. Like existing works on 60GHz mobile imaging [57], 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 [57]). Thus like [57]), we chose not to use phase measurements in our imaging design.

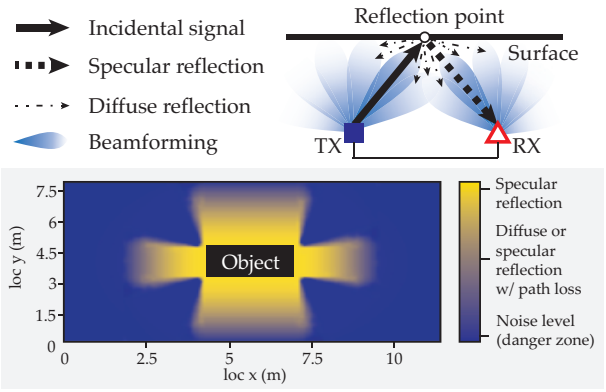


Figure 4: Colocated TX/RX leads to limited visibility of specular reflection, for both indoor and outdoor settings.

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 [57], 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 4, 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 4 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.

Rather than reinventing our own navigation algorithms, we seek to leverage existing navigation algorithms from the robotics community [15, 16, 39]. 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.

3. 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 that of [57].

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 [46] 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 §4.

1. Sensing specular reflection via beamforming. As shown by Figure 5, 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 [27], 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 §4), Ulysses extracts the angular and signal strength information of the reflection signal, producing a sensing map per location. Figure 5 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*.

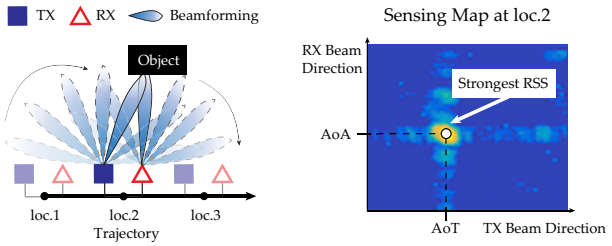


Figure 5: At each location, Ulysses scans objects by fine-grained beamforming. The result is a per-location sensing map that records the received signal strength as a function of TX and RX beam directions. The peak defines the $\{AoA, AoT, RSS\}$ tuple for imaging.

2. Imaging surface by projecting trajectory. The basic concept is shown by Figure 6. 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 [32]. 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 6, such projection can successfully reveal not only flat surfaces but also curved surfaces (details in §4).

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 [50]. 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.

4. ULYSSES DESIGN DETAILS

We present Ulysses' three modules in detail, starting from the sensing and imaging modules assuming a single object is present.

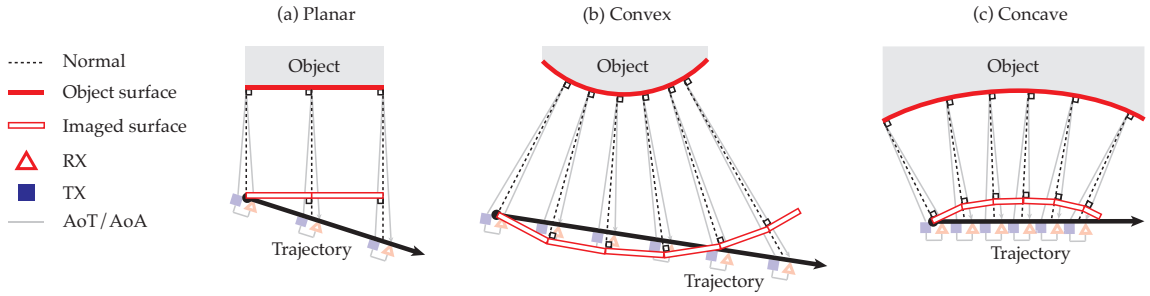


Figure 6: Estimate surface shape by projecting trajectory. At each measurement location on the trajectory, the captured $\{AoA, AoT, RSS\}$ is contributed by a small segment of the object surface. By projecting the trajectory segment guided by the two normal lines, we can estimate the shape of this surface segment. We then stitch these estimates up to build a continuous surface shape estimate.

We then discuss how Ulysses images multiple objects simultaneously, followed by the navigation module.

4.1 Sensing via Beamforming

The colocated TX/RX perform sensing using the fine-grained beamforming module in 802.11ad [27]. 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 [25] to distinguish the contribution of the main lobe from those of the side lobes. Thus these side lobes have minimal impact on our system.

4.2 Imaging by Projecting Trajectory

After detecting the presence of specular reflection, the Ulysses device will move following a scheduled trajectory $\mathbb{T}$ (see §4.4 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 6). 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 use the average over the trajectory as the estimate of surface center. After estimating the surface curva-

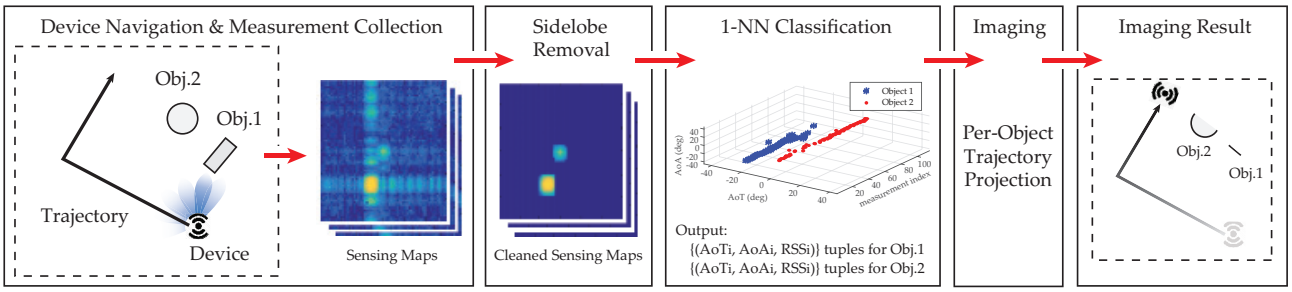


Figure 7: Ulysses can image *multiple* surfaces from a single trajectory. From a sequence of sensing maps, we extract per-object $\{AoA, AoT, RSS\}$ tuples via classification, and then image each object separately.

ture 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 [32], 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 [57]. 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 [31].

4.3 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 7 illustrates the process. Given a sequence of sensing maps collected along the trajectory, we first remove the contributions of side lobes using [25]. 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) [19] 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.

4.4 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.,* [15, 16, 39, 50]), 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 [50]. 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:

$$\frac{P_{TX} G_{TX} G_{RX}}{L_{path}(\gamma) L_{shape} L_{material}} = \text{noise floor} \quad (1)$$

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 [14, 24]. 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 [31] 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 (1) 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 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

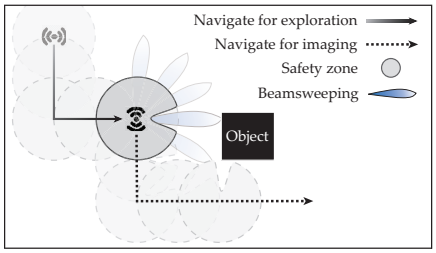


Figure 8: An illustration of Ulysses's navigation path to image an object and the corresponding safety zones.

object(s) that leads to the observed RSS value. For this we reuse (1) 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.

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 8 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.

5. 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 9(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 [3]. 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 [6] and follow the 802.11ad standard[2]. The Effective Isotropic Radiated Power (EIRP) is 32dBm, well-below the FCC regulation limit of 82dBm [5]. 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 [51].

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 10 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 [39]. 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 §4, 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 implementation, which we leave to future work. The navigation computation is nearly instantaneous.

6. 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 [1] and dual-device 60GHz mobile imaging [57].

[2] Facebook recently showcased similar devices at the Embedded Linux Conference [4] and deployed them in Downtown San Jose [7]. We hope that these devices will become commercially available soon.

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

Figure 9: Our testbed prototype, evaluation environments, and experimental objects.

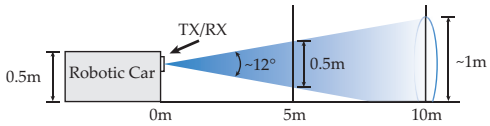


Figure 10: Since the TX/RX has a fixed height, the vertical imaging range depends on the vertical beam coverage. At 10m distance, any object placed within the 1m beam coverage can still be observed.

Experiment Configuration. We first perform experiments in three indoor environments (Figure 9(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 9 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 9(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.

6.1 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.

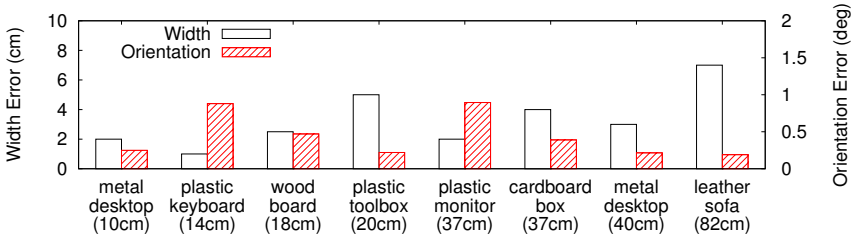
Imaging a surface via a straight line. Consider a simple scenario where Ulysses images a specific object surface by traveling on a straight line. In each experiment, we place an object in the center of the room and program the robot to move in a straight line. Due to random trajectory errors and measurement 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 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 11 (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 11 (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.

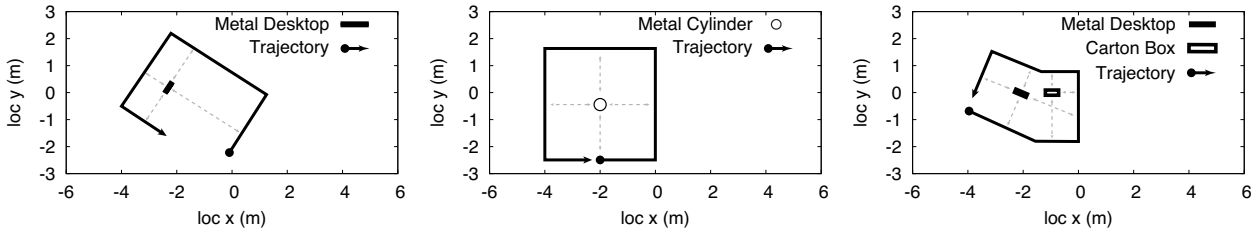
Finally, our material estimation is accurate. Across our experiments, the ground-truth material always falls in the top-3 candidates provided by our imaging algorithm. This level of accuracy aligns with prior works that must place TX/RX on multiple devices [56, 57].

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 12 shows three examples of our real-time navigation for imaging and the corresponding imaging results. Figure 12(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 12(d) shows that the estimated object is almost a duplicate of the ground



(a) Planar Surfaces (b) Curved Surfaces

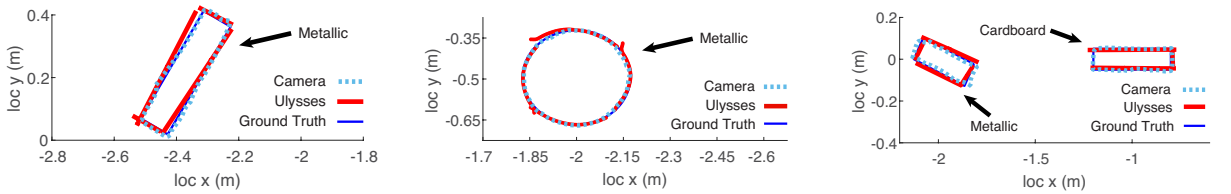
Figure 11: Object imaging benchmarks for (a) planar surfaces (5m away) and (b) curved surfaces (3m away). Overall, we achieve $< 8cm$ error in width and $< 1^\circ$ error in orientation. We later show that when imaging the entire object, the per-surface error will reduce after we assemble different surfaces.



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

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

truth ($< 3cm$ error in width). Another example in Figure 12(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 12(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.

Finally, Table 1 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 ($50ns/beam$ [51]), 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 [57], and the camera-based imaging system. First, we implement the RSA algorithm on our platform, following the same scenarios of Figure 11, 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 [1], 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 12 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 13 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.

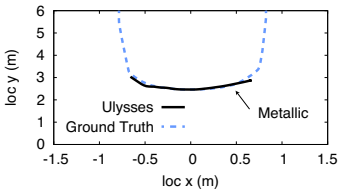


Figure 13: Imaging result of the back of a parked car.

6.2 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 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 14(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 14(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 14(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 [15, 16, 42, 43].

7. 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 paper 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 [21, 44], which iteratively estimates the robot's current movement and the corresponding imaging outcome, using prior trajectory data and imaging results.

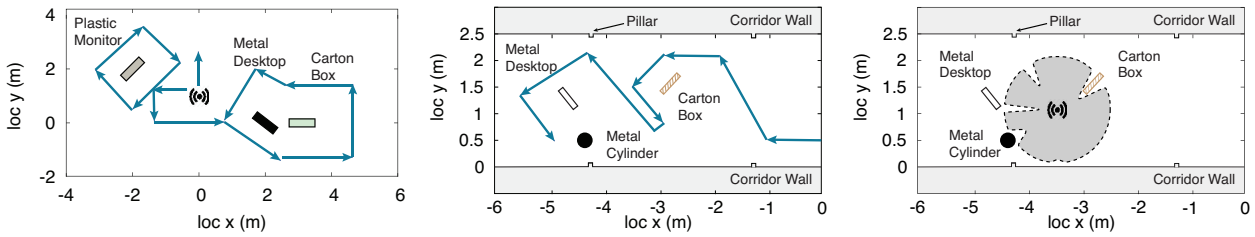
Handling device rotation errors. Moving in straight lines, our robot makes small rotation error ($< 1^\circ$). Thus in our experiments, we did not observe any impact on Ulysses imaging and navigation. But under larger rotation errors, the accuracy of AoT and AoA computation may be affected. We plan to 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



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

Figure 14: 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.

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.

8. 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 [22, 26, 29, 47] or specialized FMCW hardware [11, 12, 13] 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 [41, 49], 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. [36] uses FMCW hardware and applies SAR with sparse measurements in absence of device movement noises, while [53] 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 [57, 56], focusing on being robust to device path noises. Finally, the most relevant mmWave handheld imager is *Walleye* [9], 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 [38, 52], while another develops new measuring methods to image objects using an audible frequency [37]. 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 [40].

LIDAR and vision-based solutions. Today, the state of the art in mobile imaging products is LIDAR, used by Google self-driving cars [45], 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 [8] still costs $250 per device, providing 1D imaging with 10s of centimeter precision; another recent version [2] 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 [1], 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 [54, 28]. 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.,* [15, 16, 39, 50]). Our (new) contribution here is to define safety zones using the onboard 60GHz beamforming radios rather than sonar and camera [15, 16, 42, 43].

9. CONCLUSION

This paper 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.

Acknowledgment

We would like to thank our shepherd Fadel Adib and the anonymous reviewers for their useful feedback, and the Facebook Terragraph team for providing the 60GHz testbed on which our experiments were carried out. This work is supported by NSF grants CNS-1518812 and CNS-1317153. Any opinions, findings, and conclusions or recommendations expressed in this material do not necessarily reflect the views of any funding agencies.

10. REFERENCES

[1] http://www.123dapp.com/catch.
[2] http://www.slamtec.com/.
[3] http://www.nexusrobot.com/.
[4] http://events.linuxfoundation.org/events/embedded-linux-conference/.
[5] https://apps.fcc.gov/edocs_public/attachmatch/FCC-13-112A1_Rcd.pdf.

[6] 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/.

[7] San josé partners with facebook for high-speed outdoor wi-fi. https://gcn.com/articles/2016/04/18/san-jose-facebook.aspx.

[8] ACKERMAN, E. Sweep is a $250 LIDAR with range of 40 meters that works outdoors. IEEE Spectrum, April 2016.

[9] ADAMS, C., HOLBROOK, D., AND SENGSTEN, R. A handheld active millimeter wave camera. In *Proc. of HST* (2010).

[10] ADAMS, C., HOLBROOK, D., AND SENGSTEN, R. A handheld active millimeter wave camera. In *HST* (2010).

[11] ADIB, F., HSU, C.-Y., MAO, H., KATABI, D., AND DURAND, F. Capturing the human figure through a wall. *ACM Transactions on Graphics 34*, 6 (2015), 219:1–219:13.

[12] ADIB, F., KABELAC, Z., AND KATABI, D. Multi-person localization via rf body reflections. In *Proc. of NSDI* (2015).

[13] ADIB, F., KABELAC, Z., KATABI, D., AND MILLER, R. C. 3d tracking via body radio reflections. In *Proc. of NSDI* (2014).

[14] ANDERSON, H. R. A ray-tracing propagation model for digital broadcast systems in urban areas. *IEEE Transactions on Broadcasting 39*, 3 (1993), 309–317.

[15] BORENSTEIN, J., AND KOREN, Y. Real-time obstacle avoidance for fast mobile robots in cluttered environments. In *Proc. of ICRA* (1990).

[16] BORENSTEIN, J., AND KOREN, Y. The vector field histogram-fast obstacle avoidance for mobile robots. *IEEE Transactions on Robotics and Automation 7*, 3 (1991), 278–288.

[17] BREWSTER, S. Uber starts self-driving car pickups in Pittsburgh. TechCrunch, September 2016.

[18] CAMERON, D., AND CUADRA, A. Meet the future first responders. Washington Post, June 2015.

[19] COVER, T., AND HART, P. Nearest neighbor pattern classification. *IEEE Transactions on Information Theory 13*, 1 (1967), 21–27.

[20] CUTHBERTSON, A. Amazon drone deliveries receive U.K. approval. NewsWeek, July 2016.

[21] DAVISON, A. J., REID, I. D., MOLTON, N. D., AND STASSE, O. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence* (2007).

[22] DEPATLA, S., BUCKLAND, L., AND MOSTOFI, Y. X-ray vision with only wifi power measurements using rytov wave models. *IEEE Transactions on Vehicular Technology 64* (2015), 1376–1387.

[23] GERSHGORN, D. Google's robots are learning how to pick things up. Popular Science, March 2016.

[24] HACIVELIOĞLU, F., USLU, M. A., AND SEVGI, L. A matlab-based virtual tool for the electromagnetic wave scattering from a perfectly reflecting wedge. *IEEE Antennas and Propagation Magazine 53*, 6 (2011), 234–243.

[25] HOSOYA, K., PRASAD, N., RAMACHANDRAN, K., ORIHASHI, N., KISHIMOTO, S., RANGARAJAN, S., AND MARUHASHI, K. 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*, 1 (2015), 81–96.

[26] HUANG, D., NANDAKUMAR, R., AND GOLLAKOTA, S. Feasibility and limits of Wi-Fi imaging. In *Proc. of SenSys* (2014).

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

[28] IHRKE, I., KUTULAKOS, K. N., LENSCH, H., MAGNOR, M., AND HEIDRICH, W. Transparent and specular object reconstruction. *Computer Graphics Forum 29*, 8 (2010).

[29] KUMAR, S., GIL, S., KATABI, D., AND RUS, D. Accurate indoor localization with zero start-up cost. In *Proc. of MobiCom* (2014).

[30] LAMBERT, F. Understanding the fatal Tesla accident on autopilot and the NHTSA probe. Electrek, July 2016.

[31] LANGEN, B., LOBER, G., AND HERZIG, W. Reflection and transmission behavior of building materials at 60ghz. In *Proc. of PIMRC* (1994).

[32] LANZISERA, S., LIN, D. T., AND PISTER, K. S. J. RF time of flight ranging for wireless sensor network localization. In *Proc. of WISES* (2006).

[33] LIPTAK, A. 7-eleven just made the first commercial delivery by drone. The Verge, July 2016.

[34] LIU, K., WANG, X., SAMARABANDU, J., AND AKHTAR, A. Monostatic airborne sar using license exempt wimax transceivers. In *VTC* (2014).

[35] LOHR, S. A lesson of tesla crashes? Computer vision can't do it all yet. The New York Times, September 2016.

[36] MAMANDIPOOR, B., MALYSA, G., ARBABIAN, A., MADHOW, U., AND NOUJEIM, K. 60 GHz synthetic aperture radar for short-range imaging: Theory and experiments. In *Proc. of ASILOMAR* (2014).

[37] MOLERÓN, M., AND DARAIO, C. Acoustic metamaterial for subwavelength edge detection. *Nature Communications 6*, 8037 (2015).

[38] NANDAKUMAR, R., GOLLAKOTA, S., AND WATSON, N. Contactless sleep apnea detection on smartphones. In *MobiSys* (2015).

[39] NEWMAN, P., BOSSE, M., AND LEONARD, L. Autonomous feature-based exploration. In *Proc. of ICRA* (2003).

[40] ORALKAN, O., ERGUN, A. S., JOHNSON, J. A., KARAMAN, M., DEMIRCI, U., KAVIANI, K., LEE, T. H., AND KHURI-YAKUB, B. T. Capacitive micromachined ultrasonic transducers: Next-generation arrays for acoustic imaging? *IEEE transactions on ultrasonics, ferroelectrics, and frequency control 49*, 11 (2002), 1596–1610.

[41] PU, Q., GUPTA, S., GOLLAKOTA, S., AND PATEL, S. Whole-home gesture recognition using wireless signals. In *Proc. of MobiCom* (2013).

[42] SE, S., LOWE, D., AND LITTLE, J. Vision-based mobile robot localization and mapping using scale-invariant features. In *Proc. of ICRA* (2001).

[43] SE, S., LOWE, D. G., AND LITTLE, J. J. Vision-based global localization and mapping for mobile robots. *IEEE Transactions on Robotics 21*, 3 (2005), 364–375.

[44] SMITH, R. C., AND CHEESEMAN, P. On the representation and estimation of spatial uncertainty. *The International Journal of Robotics Research 5*, 4 (1986), 56–68.

[45] SOLON, O. Lidar: the self-driving technology that could help tesla avoid another tragedy. The Guardian, July 2016.

[46] SOUMEKH, M. *Synthetic aperture radar signal processing*. New York: Wiley, 1999.

[47] SUN, L., SEN, S., KOUTSONIKOLAS, D., AND KIM, K. Widraw: Enabling handsfree drawing in the air on commodity wifi devices. In *Proc. of MobiCom* (2015).

[48] SUR, S., VENKATESWARAN, V., ZHANG, X., AND RAMANATHAN, P. 60 ghz indoor networking through flexible beams: A link-level profiling. In *SIGMETRICS* (2015).

[49] TAN, S., AND YANG, J. Wifinger: Leveraging commodity wifi for fine-grained finger gesture recognition. In *MobiHoc* (2016).

[50] THRUN, S., BURGARD, W., AND FOX, D. *Probabilistic robotics*. MIT press, 2005.

[51] VALDES-GARCIA, A., REYNOLDS, S., NATARAJAN, A., KAM, D., LIU, D., LAI, J.-W., HUANG, Y.-L., CHEN, P.-Y., TSAI, M.-D., ZHAN, J.-H., NICOLSON, S., AND FLOYD, B. Single-element and phased-array transceiver chipsets for 60-GHz Gb/s communications. *IEEE Communications Magazine 49*, 4 (April 2011), 120 –131.

[52] WANG, J., ZHAO, K., ZHANG, X., AND PENG, C. Ubiquitous keyboard for small mobile devices: Harnessing multipath fading for fine-grained keystroke localization. In *MobiSys* (2014).

[53] WEI, T., AND ZHANG, X. mTrack: High-precision passive tracking using millimeter wave radios. In *Proc. of MobiCom* (2015).

[54] ZHENG, J. Y., AND MURATA, A. 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*, 8 (2000).

[55] ZHU, Y., ET AL. Demystifying 60GHz outdoor picocells. In *Proc. of MobiCom* (2014).

[56] ZHU, Y., ZHU, Y., ZHANG, Z., ZHAO, B. Y., AND ZHENG, H. 60GHz mobile imaging radar. In *Proc. of HotMobile* (2015).

[57] ZHU, Y., ZHU, Y., ZHAO, B. Y., AND ZHENG, H. Reusing 60GHz radios for mobile radar imaging. In *Proc. of MobiCom* (2015).