RF-Based Indoor Localization Around Corners

by

Peng Cao

B.S., Peking University (2020)

Submitted to the Department of Electrical Engineering and Computer Science
in partial fulfillment of the requirements for the degree of

Master of Science in Electrical Engineering and Computer Science

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

May 2022

Author . . . .
Department of Electrical Engineering and Computer Science
May 13, 2022

Certified by . . . .
Dina Katabi
Professor of Electrical Engineering and Computer Science
Thesis Supervisor

Accepted by . . . .
Leslie A. Kolodziejski
Professor of Electrical Engineering and Computer Science
Chair, Department Committee on Graduate Students

RF-Based Indoor Localization Around Corners

by

Peng Cao

Submitted to the Department of Electrical Engineering and Computer Science
on May 13, 2022, in partial fulfillment of the
requirements for the degree of
Master of Science in Electrical Engineering and Computer Science

Abstract

Unmanned robots are increasingly used around humans in factories, malls, and hotels. As they navigate our space, it is important to ensure that such robots do not collide with people who suddenly appear as they turn a corner. Today, however, there is no practical solution for localizing people around corners. Optical solutions try to track hidden people through their visible shadows on the floor or a sidewall, but they can easily fail depending on the ambient light and the environment. More recent work has considered the use of radio frequency (RF) signals to track people and vehicles around street corners. However, past RF-based proposals rely on a simplistic ray-tracing model that fails in practical indoor scenarios. This thesis introduces CornerRadar, an RF-based method that provides accurate around-corner indoor localization. CornerRadar addresses the limitations of the ray-tracing model used in past work. It does so through a novel encoding of how RF signals bounce off walls and occlusions. The encoding, which we call the *hint map*, is then fed to a neural network along with the radio signals to localize people around corners. Empirical evaluation with people moving around corners in 56 indoor environments shows that CornerRadar achieves a median error that is 3x to 12x smaller than past RF-based solutions for localizing people around corners.

Thesis Supervisor: Dina Katabi
Title: Professor of Electrical Engineering and Computer Science

Disclaimer

This thesis is based on a paper published in ACM IMWUT 2022: Shichao Yue*, Hao He*, Peng Cao*, Kaiwen Zha, Masayuki Koizumi, Dina Katabi. CornerRadar: RF-Based Indoor Localization Around Corners. (* denotes equal contribution.) I am one of the joint first authors of this paper, and I contributed to all aspects of this paper, including the system and algorithm design and implementation, data collection, empirical evaluation and paper writing.

Acknowledgments

I would like to express my deepest gratitude to my thesis advisor, Prof. Dina Katabi, for her enthusiasm, patience, diligence and enormous wisdom. She helped me every aspects of my research, teaching and encouraging me to address the impactful problems. She set an example of what a good researcher should be like. I cannot imagine having a better advisor.

I am also more than grateful to the amazing colleagues in NETMIT and CSAIL. It is a great honor to meet, discuss with and collaborate with them. They can always offer me insightful points and warm companionship.

Finally, I want to thank all my family members and friends. It is their love and support all along makes me to grow up to the person I am.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The problem of detecting and tracking people around corners has attracted significant interest in recent years [48, 15, 20, 40, 7, 24, 6, 34, 33]. It is motivated by the introduction of robots and unmanned vehicles into indoor spaces to transport products in factories and warehouses, deliver food in restaurants, and clean rooms in hotels and hospitals [52, 4, 17]. These robots are typically equipped with camera and low-cost LiDAR devices to allow them to navigate around and detect visible people. However, they may collide with a person who suddenly appears in front of the robot as it turns a corner. If a robot could localize humans around corners and track their movements, it could better plan its trajectory and avoid potential collisions.

While the research community has made important advances, there is still no practical solution for indoor localization of people around corners. Past work falls into two categories: optical methods and RF-based methods. Several optical methods emit ultra-short light pulses and measure their time of return [58, 59, 24]. Such methods require expensive equipment with very high temporal precision and high-power laser beams that exceed the eye safety limit [30]. Other optical solutions track people around corners using the visible shadows they may leave on the floor or a sidewall [34, 33]. Such approaches however are fragile and can easily fail depending on the environment and lighting conditions.

A more natural solution would use Radio-Frequency (RF) signals, as they experience less diffusion when they bounce off walls due to their much longer wavelengths.

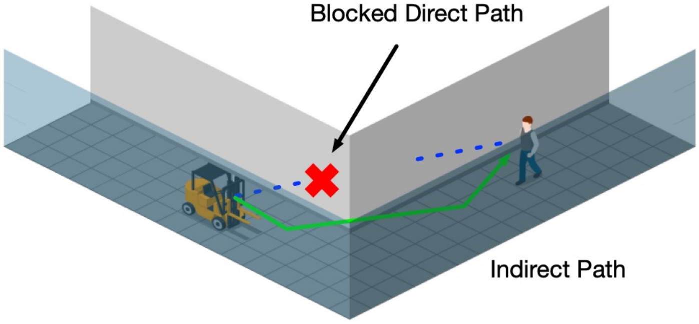


Figure 1-1: RF localization with/without direct paths. The direct path illustrated by the dashed blue line is blocked due to traversing multiple walls. The radio on the robot receives the RF signal along the indirect path that bounces off sidewalls, illustrated with the green line.

However, despite a rich literature on RF-based indoor localization, the vast majority of past solutions do not work around corners [63, 41, 19, 25, 56, 11, 5, 26, 42, 66, 1, 2, 39, 21, 29, 28, 62, 27, 64]. The reason is that these techniques typically assume that since RF signals can traverse walls, there always be a direct RF path from the tracked person to the radio receiver (e.g., the robot). Yet, when people are hidden around corners, the RF signals have to traverse multiple walls, and their structural support material, which causes a drastic attenuation of the direct RF path, as shown in Figure 1-1. When the direct propagation path is blocked, the core principle underlying most RF-localization techniques does not hold, leading to major localization errors [47].

This problem has been recognized by recent papers on around-corner RF-based localization [65, 15, 20, 40, 47]. While these papers differ in the details of their techniques, all of them address the above problem by adopting ray tracing, a technique widely used in computer graphics [14, 12]. Specifically, they assume that RF signals bounce off walls like visible light, and do not traverse them. Given no direct path, they use the floor plan to backtrack the indirect RF path to the person behind a corner, as shown by the green line in Figure 1-1. However, RF signals are not the same as visible light. Though RF signals can be reflected by occluders just like visible light, they can also propagate *through* those occluders. Simply adapting ray tracing from the visible domain and assuming no direct RF propagation is not robust.

This thesis introduces CornerRadar, a novel solution for RF-based indoor localization that works around corners. In contrast to past papers on localization with RF signals, which assume either the presence of a direct RF propagation path or its absence and fail to work when the environment does not comply with the chosen assumption, CornerRadar avoids making any such hard assumptions. Instead, CornerRadar takes into account both possibilities and uses a data-driven approach to decide which possibility is more likely.

Specifically, the design of CornerRadar has two components: a *Hint Map* and a *Localization Convolutional Neural Network (CNN)*. The Hint Map is a novel encoding that summarizes how RF signals propagate in a given spatial layout. It takes into account both the possibility of reflecting off walls following ray-tracing rules and propagating through them. The Localization CNN takes, as input, the received RF signals and the Hint Map and localizes people around corners. Once trained, CornerRadar's CNN can generalize to new environments never seen during training.

We have implemented CornerRadar and evaluated its performance empirically in 56 indoor environments, with people walking around corners and behind walls. Our evaluation scenarios are diverse in their layout, the speed of walking people, and the distance and orientation of the radio with respect to the corner of interest. Our results show that CornerRadar is highly accurate, and its median errors are 3x to 12x smaller than past RF-based localization techniques, which tend to fail either due to the presence or absence of the direct RF path. Further, CornerRadar's median localization errors along the x and y axes are 16.8 cm and 13.8 cm respectively, even in new environments unseen by the CNN during training. To the best of our knowledge, no past work on RF-based indoor localization has shown such low errors for tracking people around corners.

Chapter 2

Related Work

Past work on seeing around corners has two lines of research: optical methods and RF-based methods.

2.1 Optical Around-Corner Sensing

The past decade has seen significant work in computer vision and graphics on novel methods for sensing objects around corners [58, 59, 24, 32, 22, 23, 34, 33, 9, 7, 46]. Early work on non-line-of-sight (NLOS) imaging [58, 59] has relied on costly and complicated streak cameras, and is limited to small-scale scenes. Some later work like [24] images hidden objects by examining the intensity response of a laser; however, it expects the object to be rigid and its shape known in advance. Further, these methods require high power laser beams that can exceed the eye safety limit [47]. The authors of [32, 22, 23] reconstruct the shape of a hidden object by leveraging NLOS correlography. However such methods are applicable only to small objects, as big objects do not cause the self-interference upon which correlation-based techniques rely. A few papers [34, 33] detect hidden people by analyzing the shadows they leave on the floor or a sidewall; yet the shadows need to be observable from the camera's point of view, which works only at very close to the corner and requires a light source far behind the person. The authors of [9, 7, 46] use conventional intensity images for NLOS tracking and localization, but those methods are limited to highly reflective targets [9, 7], sparse dark backgrounds [9], or scenes with additional occluders [46]. The work in [7] analyzes the subtle spatio-temporal radiance variations to detect the presence of a hidden person; however, it cannot localize the person. In summary, none of the existing optical methods can accurately localize people around corners in general indoor settings.

2.2 RF-Based Around-Corner Sensing

There is a rich literature on RF-based indoor localization [63, 41, 19, 25, 56, 11, 5, 26, 42, 66, 1, 2, 39, 21, 29, 28, 62, 27, 64]. These methods work through walls and occlusions. However, while RF-based localization techniques vary widely in their details, they typically rely on the presence of a direct signal path from the tracked person to the radio receiver. They first disentangle the direct path from the indirect paths, then use the direct path to localize the target. Recent papers on around-corner RF-based tracking have recognized that when people are around a corner, the direct path tends to be blocked or very highly attenuated due to the need to traverse multiple walls [65, 31, 51, 15, 20, 40, 47]. To address this problem, all of these papers assume RF signals cannot traverse walls; they use ray tracing to estimate how RF signals bounce off walls, and backtrack the signal to the person around the corner. Different papers differ in the details of how they apply ray tracing. For example, [31, 65, 15, 40, 51] first process the RF signals to identify the main target, then use ray tracing to backtrack the center point in that target. Other papers [20, 47] apply ray tracing to all received RF signals first, then process the results to localize the target.

In particular, the authors of [47] have recently proposed an outdoor solution for allowing cars to track people and vehicles around street corners. Their method uses a Frequency Modulated Continuous Wave (FMCW) radar and processes the received RF signal using ray tracing, assuming no direct RF path. They then use a neural network to classify different hidden targets behind into person, bicycle or car.

CornerRadar differs from all of the past work on RF-based localization. In contrast to past work that assumes the presence of a direct path, CornerRadar can localize people even when the direct RF path is fully blocked. In contrast to past RF work on localizing people around corners, CornerRadar accounts for the fact that RF signals do not simply reflect off walls and that they can also traverse walls and arrive along the direct path. CornerRadar also accounts for the low spatial resolution of RF signals in comparison to visible light, which causes additional errors when past work applies ray tracing to RF signals, as explained in Chapter 4.

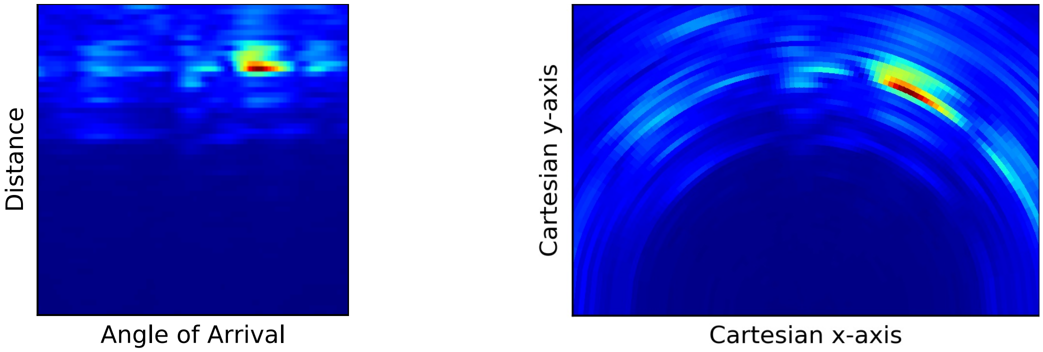
Chapter 3

Background

This chapter introduces background information that we use in the rest of the thesis. The description herein is kept at high level. For more information, readers can refer to past literature on localization using FMCW radio [1, 2, 36, 44, 61] and ray tracing [47, 65, 40].

3.1 FMCW Radio

As common in past work in many papers on RF-based localization [1, 2, 36, 44, 61], CornerRadar uses a multi-antenna FMCW radio. The radio has an antenna array with 12 antennas organized horizontally. The combination of FMCW and antenna array allows the radio to filter the received RF signal based on the distance it traveled and the spatial angle from which it arrives [50]. Thus, the received RF signal can be expressed as a time series of *RF-Snapshots*, where each snapshot is a 2D matrix, and each entry in this matrix, (i, j), refers to the RF signal received along a particular angle of arrival θ_i, and from a particular distance d_j. Figure 3-1a shows an example RF-Snapshot. The colors in the figure denote the signal power, where red refers to high power RF signal and dark blue refers to no RF power. Naturally, the original RF-Snapshot is in the polar coordinates. To allow for better visualization with respect to the floor map, we transform the RF-Snapshot into the Cartesian coordinates as shown in Figure 3-1b. Finally, as common in past work, we remove RF signals reflected off



(a) Polar

(b) Cartesian

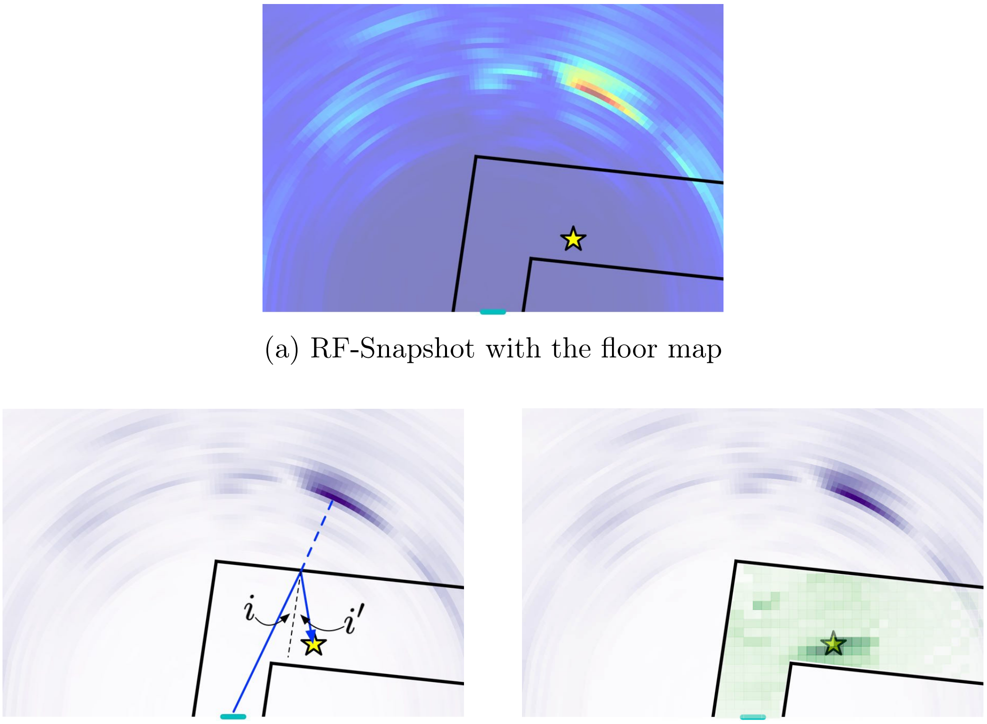
Figure 3-1: An RF-Snapshot in two coordinate systems. It shows the signal power received from a particular position in space. Red refers to high signal power, while dark blue denotes no signal.

static objects (e.g., walls and furniture) by subtracting consecutive snapshots. For clarity, all RF-Snapshots in this thesis are shown after such subtraction.

3.2 Ray Tracing

Ray tracing is commonly used in computer graphics to trace the propagation of visible light [14, 12]. It is also used in some past work on RF-based localization to backtrack a received RF signal to the original location it came from [47, 65]. To understand the process, let's use the example in Figure 3-2a, which shows an RF-Snapshot augmented with the floor map, the location of the radio (the cyan rectangle) and the true location of the person (the yellow star). The black lines in the figure refer to the walls. In this example, the target is hidden around a corner, and the direct path is blocked. However, because of the existence of an indirect path, some locations behind the wall have high signal power. The goal of ray tracing is to map those locations back to the original location of the person in the corridor.

To do so, walls are assumed to be perfect specular reflectors of RF signals. One can then draw a ray between the radio and the location of high RF power, as in Figure 3-2b. If the ray does not cross a wall, the signal is considered to be coming from the direct path. If the ray crosses a wall (as is the case in Figure 3-2b), the ray is reflected off the wall while ensuring that the angle of incidence i is equal to the angle



(a) RF-Snapshot with the floor map

(b) Peak-based ray tracing

(c) All-pixels ray tracing

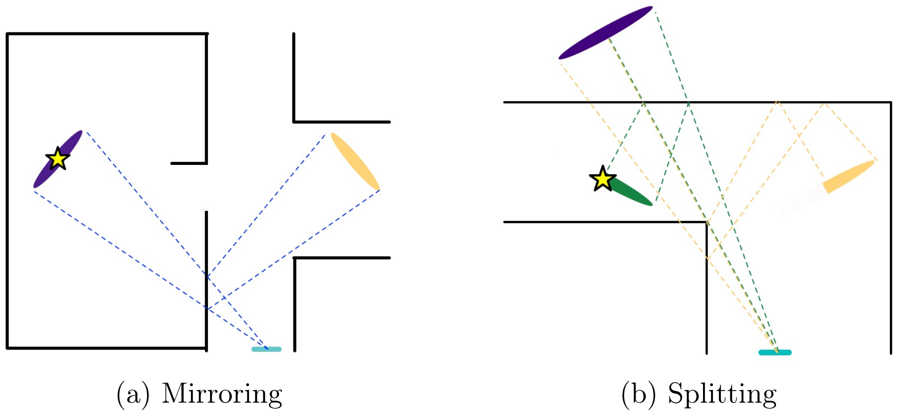
Figure 3-2: (a) An RF-Snapshot augmented with the location of the radio (the cyan rectangle), the true location of the person (the yellow star), and the walls (the black lines). (b) Peak-based ray tracing: it first finds the peak in the original RF-Snapshot, then performs ray tracing on the peak pixel to estimate the true location of the reflector. (c) All-pixels ray tracing: it first performs ray tracing on every pixel in the original RF-Snapshot, then adds up the RF power (shown in green). Note that (b) and (c) show the the same RF-Snapshot as (a) but with a different color scheme, for better visibility.

of reflection i'. After reflecting off a wall, the ray is no longer a straight line. Instead, it is now zigzagged, as shown in Figure 3-2b. Its total length however remains the same as that of the original straight ray. The process is repeated until the reflected ray stops crossing walls. The final end of the ray corresponds to the location of the target as shown in Figure 3-2b.

The above description is the most common form for applying ray tracing to RF signals, which we denote as *Peak-Based Ray Tracing*. It is used by multiple past papers on RF-based around-corner localization, such as Zhao et al. [65]. Alternatively, one could perform ray tracing on every pixel in the RF-Snapshot, not just the peak pixel, and sum up the power after ray tracing. This approach is less sensitive to the choice of peak pixel. We call this approach *All-Pixels Ray Tracing.* It is illustrated in Figure 3-2c where the green color represents the RF power after ray tracing all pixels in the original RF-Snapshot (the darker the green, the higher the power is). This approach is used by Scheiner et al. [47] to localize people and vehicles around corners.

Chapter 4

Why Ray Tracing Fails


(a) Mirroring

(b) Splitting

Figure 4-1: Ray Tracing Failure Cases. The yellow star indicates the person's true location. The purple ellipse shows the location of strong power in the original RF-Snapshot. The yellow and green regions show where ray tracing maps the blue spot. (a) shows a scenario where ray tracing maps the person to the corridor though he is behind the wall. (b) shows that ray tracing can split a single target to two targets.

As mentioned earlier, past papers[20, 47, 65, 40] on around-corner RF-based localization use ray tracing as it applies to visible light. While this approach works in simple environments, it fails in general indoor scenarios. We highlight two failure modes: mirroring and splitting. Mirroring refers to ray tracing confusing a person behind a wall with being in front of the wall because it ignores that RF signals can traverse walls. Figure 4-1a illustrates such a scenario where the person, denoted by a yellow star, is walking inside the room, but ray tracing maps him to a different location on the other side of the wall (the yellow patch). This is because the simple ray tracing model in past work ignores that RF signals can traverse walls.

The splitting problem is more complex. It is due to the fact that RF receivers have poor spacial resolution. RF-based localization systems use antenna arrays to distinguish between signals arriving from different spatial angles. However, unlike cameras which have millions of photo diodes, radios have a relatively small number of antennas, and hence a coarse angular resolution (e.g., 10 degrees) [37, 60]. As a result, even a point reflector appears as a wide path in the RF-Snapshot, as shown in Figure 3-2a. This causes a problem when applying ray tracing to such a patch. Specifically, different parts of the patch can end up back traced to different locations, creating confusion about the actual location of the person, as illustrated in Figure 4-1b.

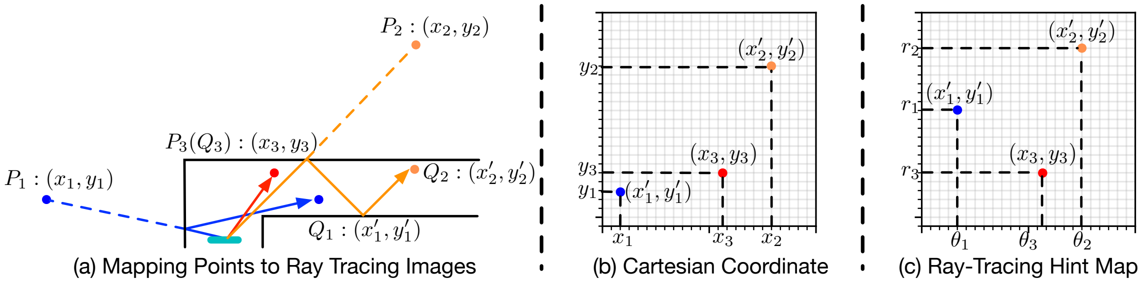
We note that the mirroring and splitting problems affect both peak-based ray tracing and all-pixels ray tracing. In particular, for peak-based ray tracing the splitting problem appears as ambiguity, since depending on which pixels is chosen as the peak (which can easily change with a small amount of noise), the ray-tracing algorithm can pick a completely different location.

Chapter 5

CornerRadar

CornerRadar is an automatic system for localizing people around corners by analyzing the combination of received RF reflections and the environment layout. CornerRadar adopts a data-driven approach. Instead of directly back tracing the RF signal to the target's location, CornerRadar trains a neural network that takes both the radio signal along with information about potential direct and indirect paths, and predicts the location of the person. Thus, CornerRadar has two main components that together localize a hidden target: 1) a *Hint Map*, which is a novel encoding that describes how RF signals propagate in an environment while taking into account both ray tracing and propagation along the direct path; and 2) a *Localization CNN* which takes as input the Hint Map and the received RF-Snapshots, and infers whether there is someone around the corner and their location.

Before delving into the design details, we note that all past works on around-corner localization (i.e., both optical and RF-based approaches) have so far focused on scenarios in which the robot/car/radio is static [65, 8, 31, 51, 48, 15, 20, 40, 47]. Similarly, in this thesis we focus on scenarios in which the radio is static as the natural first step to addressing the problem.



(a) Mapping Points to Ray Tracing Images

(b) Cartesian Coordinate

(c) Ray-Tracing Hint Map

Figure 5-1: Encoding the ray-tracing rules into the hint map. (a) shows each point is mapped to its ray-tracing image. (b) shows the encoding in the Cartesian coordinates. (c) shows the final ray-racing hint map which is in the polar coordinates.

5.1 Hint Map

The Hint Map is an encoding that describes how ray tracing and direct path propagation work in a particular environment, but without applying these rules to the received RF signals. Making the encoding independent of the received RF signals serves two goals. First, the encoding can be used by the neural network as coarse hints, which alone do not provide the full picture. This allows the neural network to learn from the data how to apply these rules and when to favor ray tracing over direct path propagation. The second purpose of the Hint Map is to help the neural network better generalize to new test environments that it did not see during training. By capturing much of the environment-dependent information, and expressing it in a standardized way via a Hint Map, the neural network learns a more general function, hence improving its chances of working in new environments that are different from the environments in the training set.

The next few subsections describe how we generate such encoding. For simplicity, our illustrations use L-shaped corners. CornerRadar however does not make assumptions on corner shape or angle, and works well with corners that do not have L-shape or 90-degree angles, as shown in Section 6.3.

5.1.1 Generating the Hint Map

The hint map has two components: a ray-tracing hint map, and a direct-path hint map. We refer to these two components as the R-Map and D-Map, respectively.

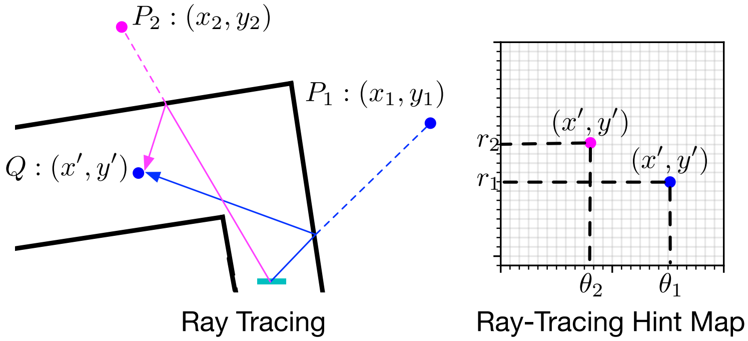


Figure 5-2: Multipath. The figure shows that ray-tracing hint map encoding naturally supports multipath. Here, (θ_1, r_1) and (θ_2, r_2) are both encoded with the same coordinates of the target location Q.

We first describe how to obtain the R-Map. To do so, we encode each point in the spatial layout with the coordinates of its ray-tracing image. Figure 5-1 illustrates this encoding. Specifically, Figure 5-1a shows three example points P_1, P_2 and P_3, which are mapped to their ray-tracing images, i.e., the corresponding points after ray tracing, Q_1, Q_2 and Q_3. Note that P_3 is mapped to itself since it is directly visible. Figure 5-1b shows the resulting encoding in the Cartesian coordinate system. Specifically, the coordinates of Q_1, (x'_1, y'_1), are recorded in the pixel (x_1, y_1) in the R-Map. Since the encoding will be consumed by the neural network along RF-Snapshots, we transform it to the polar coordinates to better align it with the RF-Snapshots.

Since there is a significant energy loss each time the RF signal bounces off a wall, when calculating the R-Map, we allow each ray to bounce off a maximum of three walls. Points whose ray-tracing images require more than three off-wall reflections are considered invalid, and their corresponding coordinates are set to 0.

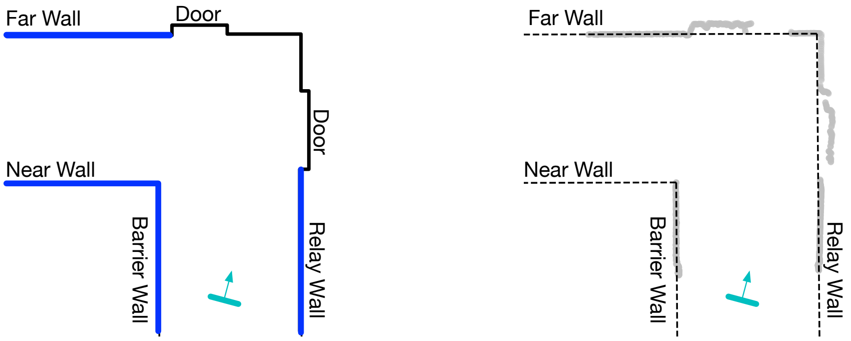
But how about multipath? In indoor environments, it is common for RF signals to propagate along multiple paths. Can this encoding support multipath propagation? Indeed, the encoding is designed to handle indoor multi-path effects and benefit from them. Figure 5-2 illustrates this property. It shows two points in the layout, P_1 and P_2, are mapped to the same ray-tracing image, Q. This means that if the localized target happens to be at Q, it would reflect the RF signal along two paths.

In fact there might be many other paths that end at Q, resulting in many other points in the layout being mapped to Q. All of those paths are encoded in the R-Map, since all of those points are encoded with the coordinates of Q (e.g. the coordinates of Q, (x', y'), appears in both pixel (θ_1, r_1) and pixel (θ_2, r_2)), using the encoding scheme described above.

We note that the multi-path effects can help CornerRadar differentiate through-wall targets versus around-corner targets, For example, in Figure 5-2, if the moving person is indeed located at P_1 instead of Q, then there will be only one possible path to reach it, the direct through-wall path, plotted in blue. In contrast, when the target is at Q behind the corner, there will be multiple indirect-paths for the RF signal to reach it (e.g., the path plotted in pink). As a result, the RF-Snapshot will have more peaks, thus differentiated from the through-wall target.

It is worth noting that the R-Map is different from all-pixels ray tracing, introduced in Section 3.2. The content of each pixel in the R-Map is the coordinates of its ray-tracing image. In contrast, the content of each pixel in all-pixels ray tracing is the RF power of its ray-tracing image(s) in the corresponding RF-snapshot. Thus, the R-Map is an encoding of the environment and would change only if the environment changes. In contrast, all-pixels ray tracing is an encoding of a particular RF-snapshot in a particular environment. Hence, it would change if either the RF-snapshot or the environment changes. This difference is essential as it allows CornerRadar to convey to the neural network the rules of ray tracing but without applying them directly to the RF signals. This allows the neural network to learn from the data how to apply those rules, while avoiding splitting and mirroring.

Next, to accommodate the fact that signals can directly propagate through walls, we augment the hint map with an encoding of the direct path, i.e.,the *Direct-path Hint Map* or D-Map. The D-Map is simple. It maps each point to itself, i.e., each point is encoded with its own Cartesian coordinates. By giving it both the R-Map and D-Map, we allow the neural network to use the observed data to make its own decisions about whether the RF signal propagated directly or by bouncing off sidewalls.



(a) Actual Layout

(b) Detected Layout

Figure 5-3: In Figure 5-3a, we show the actual layout. The actual layout has some details that CornerRadar does not need for accurate localization. It is sufficient to focus on four major walls highlighted in blue. In Figure 5-3b, LiDAR measurements are plotted in grey, and our detection result is plotted as dashed lines.

5.1.2 Layout Detection

To generate the Hint Map, we need the layout of the environment. Since the floor plan may be unavailable, we present in this section a simple algorithm that automatically extracts an approximate layout sufficient for the operation of CornerRadar. The algorithm uses a simple LiDAR, mounted on the radio. The design is independent of the details of the LiDAR, but for our experiments we use RealSense LiDAR Camera L515 [18], which is small and low cost, and weighs only 100g (see Section 6.1 for more details).

It is computationally expensive to recover every detail of the layout. Instead, as shown in Fig. 5-3a, we only focus on recovering two sets of parallel walls that have the most impact on around-corner localization: a *Near Wall* and *Far Wall* that confine the hidden corridor where the target is located, and a *Barrier Wall* and *Relay Wall* that confine the corridor where the radio is located.

Our algorithm works as follows. The LiDAR mounted on our radio device scans the corner and generates a 3D point cloud of the environment. We first project these 3D points on the 2D floor plane, as illustrated by the grey points in Figure 5-3b. Next, we cluster the points into small clusters such that the maximum distance between any two points in a cluster does not exceed 10cm. We fit a line to all points in a

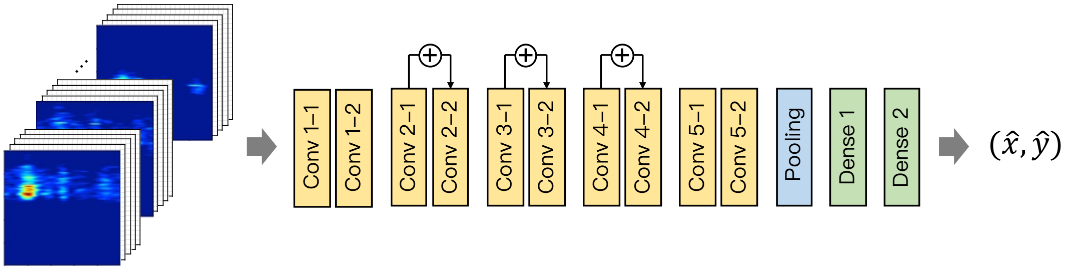


Figure 5-4: The pipeline of the Localization CNN. The input consists of RF-Snapshot and corresponding hint maps. CNN extracts features and predicts the location of the person in the corridor.

cluster using least squares regression [13]. If the fit quality exceeds a threshold, the line is kept; otherwise the points are released. We then iterate on these small line segments to merge them into larger segments. Specifically, for any two segments whose edges are less than 2cm apart, we compare the slope of the segments; if the two slopes are close, the points in the two segments are merged together, and a new line is fitted using least square. If the fit quality is acceptable, the newly generated segment is kept; otherwise the original two line segments are restored. We iterate on this process until it is not possible to create longer line segments. Next, for each segment longer than 0.5m, we consider only its two end points. We order these end points with respect to the angle they create with the LiDAR. We then process these end points sequentially according to their order to fit the minimum number of lines with acceptable fit quality. Finally, we pick two lines closest to the LiDAR, as the Barrier and Relay Walls. We then look for a line segment that has a gap of at least 1m with the Barrier Wall. This would be the Far Wall. The Near Wall is then assumed to be parallel to the Far Wall.

5.2 Localization CNN

5.2.1 Network Architecture

In this section, we introduce our Localization Convolutional Neural Network, which takes as input the 10 most recent RF-Snapshots and the corresponding hint maps, and outputs the current coordinates of the person. If the scene is empty, the coordinates

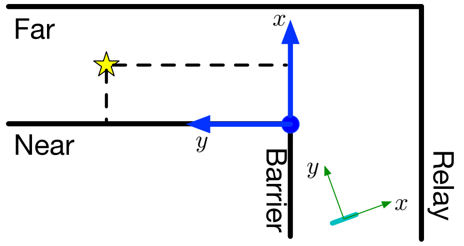


Figure 5-5: Illustration of Coordinate Systems. Radio Coordinate System (RCS) is plotted in green, having an origin at the position of the radio. Corridor Coordinate System (CCS) is plotted in blue. The yellow star is the person in the corridor.

will be at infinity.

Both the RF-Snapshots and the hint maps are represented as polar grids, which intrinsically are matrices. For the RF-Snapshot, each element in the matrix is the value of the RF power received at that location. And for the R-Map/D-Map, each element in the matrix is the coordinate after/without ray tracing. Since the coordinate is a tuple of x and y, we separate each hint map into two matrices: one for x coordinate and one for y coordinate. We then treat those matrices as different channels: one channel for RF-Snapshot, two channels for R-Map and two channels for D-Map.

As illustrated in Figure 5-4, our network has ten 3D convolution layers; after each convolution layer, there is a layer of batch normalization and a layer of ReLU activation. On top of the convolution layers, we have two fully connected layers. At the end, the network outputs the x and y coordinates of the target.

The CNN is trained with the following loss function:

$$\mathcal{L} = |x - \hat{x}| + |y - \hat{y}|,$$

where (x, y) is the ground truth coordinates of the target and $(\hat{x}, \hat{y})$ is its predicted location.

5.2.2 Aligning the Prediction Space across Environments

A key challenge in designing any neural network model is to ensure the network generalizes well to new environments that it did not train with. One way to address this challenge is to train with a very large and diverse dataset such that any new environment is likely very similar to some environment in the training set. In practice, however, the training set is always limited. So, in this section, we ask whether we can improve the generalizability of our CNN given that we train it with a limited number of layouts.

At a high level our idea is simple. The network will generalize better to new layouts if all layouts look similar, even those that the network has not seen. Of course, we cannot change the layout of a particular corner, but we can change how the layout is represented to make different layouts look as similar as possible to the neural network.

This idea motivates us to perform a change of coordinates from the Radio Coordinate System (RCS) to the *Corridor Coordinate System* (CCS). Specifically, so far we have kept the standard coordinates used in past work on RF-based localization, where the radio is the origin, and the x-axis is aligned with the radio's antenna array. While this coordinate system is the most common and natural choice for RF localization systems [1, 2, 47], it does not help in making various layouts look similar.

We propose the *Corridor Coordinate System* (CCS), a novel coordinate transform that aligns the corners across all training and testing environments, and hence enables the CNN to generalize better to new unseen environments. As indicated by blue arrows in Figure 5-5, the Corridor Coordinate System has its origin at the corner, and its y-axis along the corridor.

Since the CCS representation sees each layout from the perspective of its corner, all around-corner scenarios look fairly similar, with people mainly moving along the y-axis. This representation reduces disparity across environments, including those unseen during training. We show empirically in Section 6.11 that this coordinate transform significantly improves performance.

Finally, we emphasize that this transform is not simply a fixed transform between two Cartesian coordinates, which would be just a translation and rotation. It is a data-dependent transform because the actual corner location w.r.t the radio changes from one layout to another.

Chapter 6

Evaluation

We implement CornerRadar and evaluate it empirically.

6.1 Implementation Details

We have implemented CornerRadar, as described in Chapter 5. As in past work, we collect RF signals using an FMCW radio equipped with an antenna array. The radio sweeps the frequencies from 5.4 GHz to 7.2 GHz and transmits at sub-milliwatt power. The antenna array has 12 antennas, with about 150 degrees directionality. Due to its low transmission power, the radio has a maximum range of about 11 meters. We process the received RF signal using standard FMCW and antenna array equations [1, 2] to produce RF-Snapshots, where each snapshot has 100×100 pixels. The radio can generate 180 snapshots per second. To reduce computational complexity and boost the SNR, we average every 36 RF-Snapshots together, producing 5 RF-Snapshots per second.

To extract information about the layout, we use the Intel RealSense LiDAR Camera L515 [18]. It is low cost, light, and small (it has a diameter of only 6cm and weighs 100g). Its depth error is less than 2cm. It is configured to stream measurements at 5 frames per second. The LiDAR camera produces both RGB images and 3D point clouds with depth information. The LiDAR camera is co-located with the radio. The 3D point cloud is used to generate an estimate of the layout as described

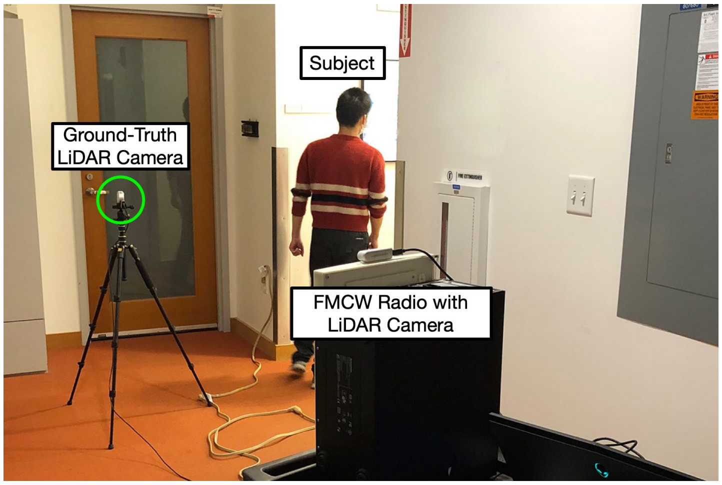


Figure 6-1: Data Collection Setup. To extract the ground-truth location of the person at every point in time, we place a LiDAR camera facing the corner and looking into the corridor. The subject is asked to walk in the hidden corridor while the radio is placed on the other side of the corner to record the RF signal. A camera LiDAR is mounted on the radio to capture the environment.

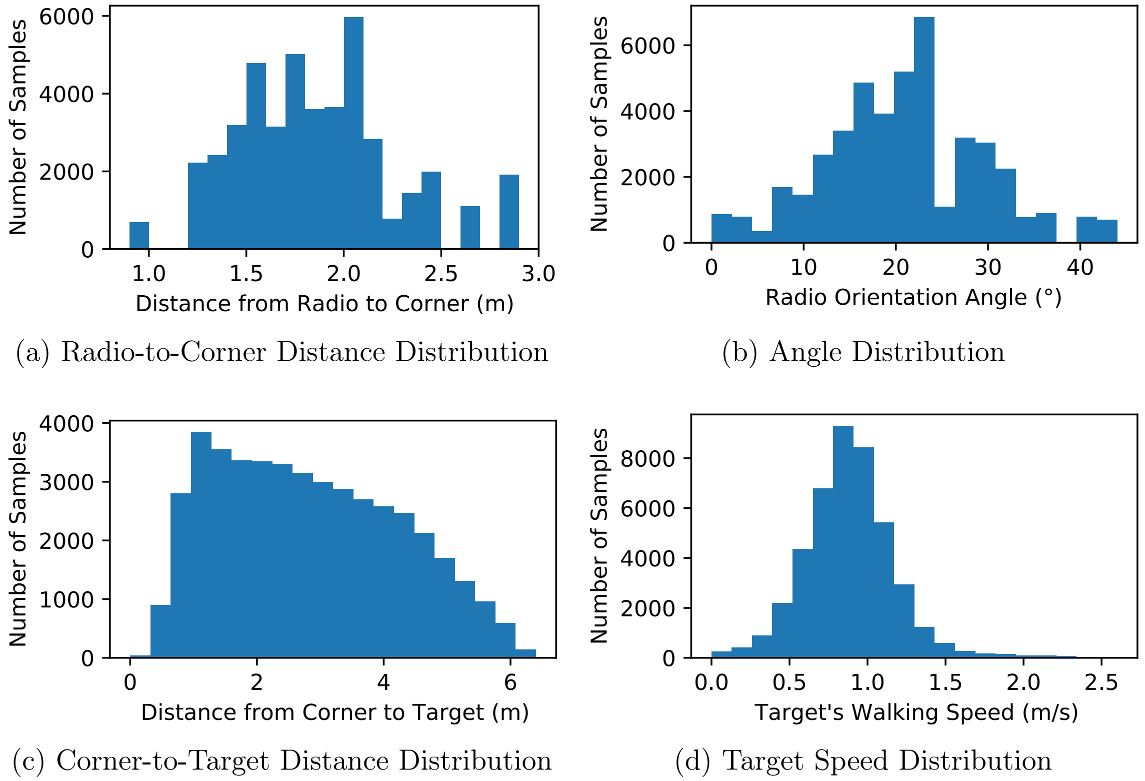
in Section 5.1.2.

The localization CNN is implemented in PyTorch [38]. As illustrated in Figure 5-4, our CNN has 10 3D convolution layers with the following hyper-parameters. The kernel size of the convolution layers is 3 for the temporal dimension, 5 for the two spacial dimensions. The stride is set to be 2 in odd layers and 1 in even layers. The model is trained for 60 epochs. The learning rate is initially set to 0.01, and decayed by an order of magnitude every 20 epochs. Data used for training is not used for testing. Specifically, we divide the 56 environments into 4 folds. We perform cross-validation testing, where we train a model on three folds and test it on the remaining fold. We repeat this process 4 times, each time changing the test fold.

Data processing is done offline on a Titan Xp GPU to generate the evaluation results. However, in Section 6.12 we measure CornerRadar's running time on a Jetson Nano and show that it can run in real-time on such a small low-power platform.

6.2 Experimental Setup

All experiments were approved by our IRB. We run our experiments in different indoor environments around our campus. In every environment, we place the radio



(a) Radio-to-Corner Distance Distribution

(b) Angle Distribution

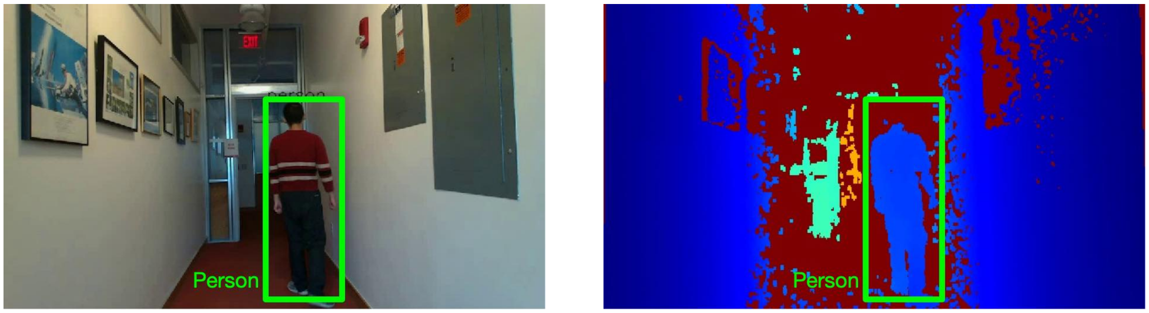
(c) Corner-to-Target Distance Distribution

(d) Target Speed Distribution

Figure 6-2: Experimental Setup Diversity. The figure reports the histograms of the distance from radio to corner, radio orientation, the distance from corner to target and target's walking speed.

randomly 1 to 3 meters away from the corner to emulate a robot that may run into a person as it turns a corner. The distribution of the radio's distance from the corner is plotted in Figure 6-2a. Since the radio antennas receive signals only within an angle of 150 degrees, we turn the radio to the right or left to allow it to receive signals that bounce off side walls. We randomly choose the orientation angle and whether to orient the radio left or right. Figure 6-2b plots the distribution of the orientation angle.

A key challenge in evaluating such a system is to extract the ground-truth location of the person at each time. To do so, we use a second LiDAR camera that faces the corridor and can directly see the person, as shown in Figure 6-1. We call this camera the ground-truth camera. Note that the ground-truth LiDAR camera is not part of our CornerRadar system. It is used only to collect the ground-truth location for the training of our model and is not required during operation.



(a) RGB Frame

(b) Depth Frame

Figure 6-3: Target localization with the ground-truth LiDAR camera. To extract the ground truth, we apply a computer vision person detector to the RGB frames and use the depth of the pixel in the detected region (the green rectangle) to compute the ground-truth location of the target.

The ground-truth LiDAR camera is capable of generating both an RGB video and a depth stream, with pixels aligned, as shown in Fig. 6-3. We apply an RGB person detector[43] to the RGB video, which generates for each frame a bounding box around the person. We then use the generated bounding boxes as a filter to select the depth pixels for the target/person. The selected depth pixels are projected to the 3D space, forming a point cloud. The ground-truth location of the person is calculated as the median of the selected point cloud in each frame. We calibrate the ground-truth LiDAR camera with the camera mounted on the radio, so that we can calculate the ground-truth coordinate of the subject with respect to the radio.

To collect data, we ask subjects to walk in the hidden area around a corner, in any way they want. That is, the subjects can either walk slowly or quickly, in a straight line or in a zigzagged way. In this way we can test the performance of our system in all possible scenarios. The distribution of the subjects' distances from the corner is plotted in Figure 6-2c, and the distribution of the subjects' walking speeds is plotted in Figure 6-2d.

6.3 Dataset

Overall, we collect data in 56 different environments with diverse layouts. The dataset contains $47,355$ samples of RF-Snapshots and the corresponding person location. As

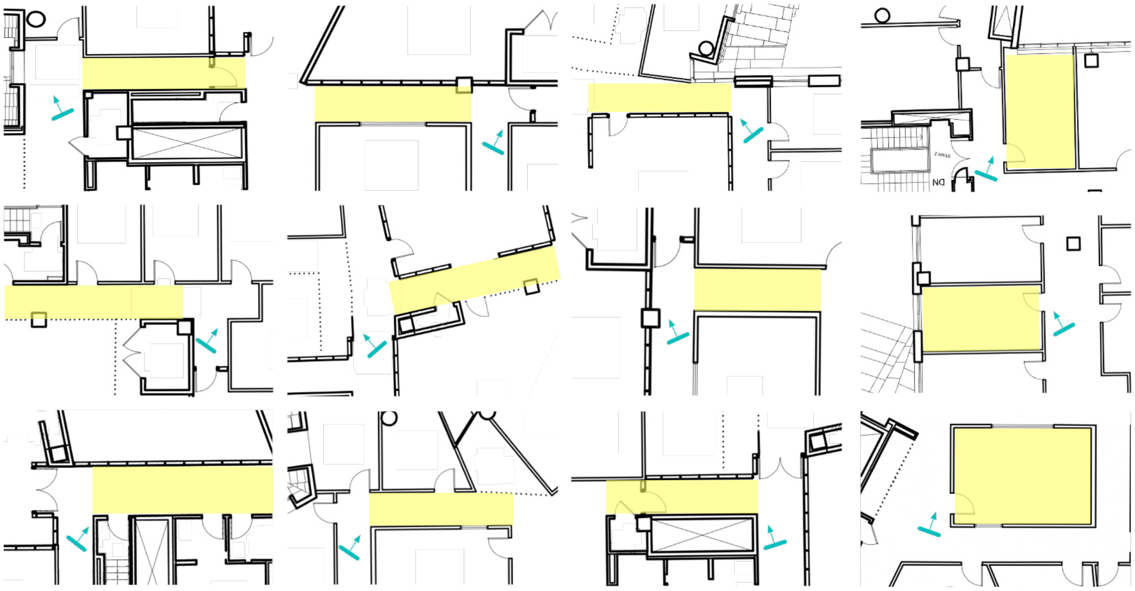
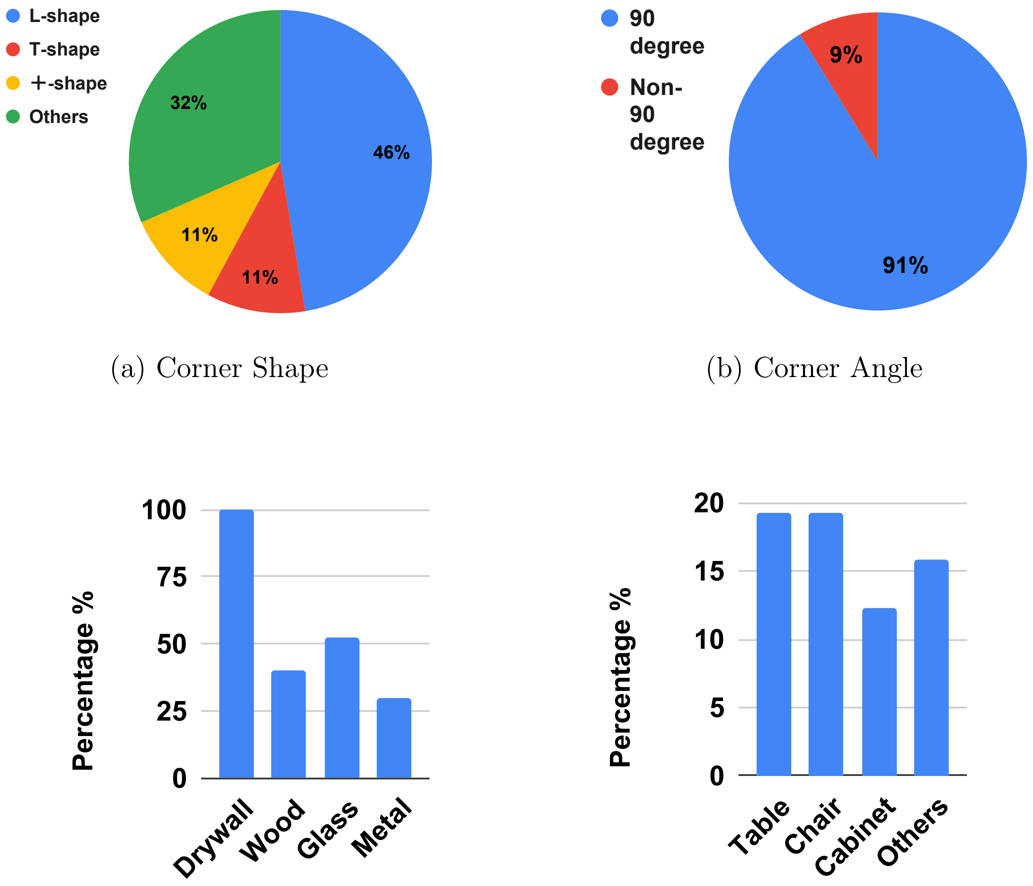


Figure 6-4: Example Layouts. The areas where the subjects walk are highlighted in yellow. The cyan boxes show the radio positions. The left three columns are environments where the direct path is blocked, while the last column are environments where there exists a direct path.

shown in Figure 6-4, we consider two scenarios where people may suddenly appear from a side corridor or as they exist an adjacent room. Given the ground-truth locations and the corresponding RF-Snapshots, we can identify the cases where the signal propagates along a direct path from the person to the radio, and the cases where the direct path is blocked. Overall, 27% of our RF-Snapshots have a direct path, whereas the rest have only indirect paths.

We classify the corners in our dataset into: (1) **L-shape**: a junction with only two exits; (2) **T-shape**: a junction with 3 exits; (3) +-**shape**: a junction with 4 exits; (4) **Other**: a corner that involves open spaces, stairs, pillars or other structures. The pie chart in Figure 6-5a shows that our dataset is composed of corners with various shapes: 46% L-shape, 32% T-shape, 11% +-shape and 11% others.

Further, the corners in our experiments do not necessarily have a ninety degree angle. The two walls intersecting at the corner could also form acute or obtuse angles. As shown in the pie chart in Figure 6-5b, 9% of the corners in our dataset have a non-90-degree angle, which is representative of real-world scenarios where a small



(a) Corner Shape

(b) Corner Angle

(c) Wall Materials

(d) Furniture

Figure 6-5: Dataset profile: (a) shape of the corner; (b) angle of the corner; (c) material of the walls around the corner; (d) furniture around the corner.

percent of corner angles are not 90 degrees.

In our dataset, the walls in different environments exhibit different materials including plaster, wood, glass and metal. As shown in Figure 6-5c, 100% of the corners have some drywall, 37% of them contain wood walls, 53% contain glass windows or glass walls, and 30% contain large structures of metal.

In addition, the environments in the dataset have typical furniture. As summarized in Figure 6-5d, 20% have tables, 20% have chairs, 12% have cabinets, 16% have other small objects including trash bins, kitchen equipment like coffee-machine and microwave, screens and computers, etc.

6.4 Metric

Let (x, y) and $(\hat{x}, \hat{y})$ denote the coordinates of the ground-truth location and the prediction, expressed in the corridor coordinate system. We consider the following metrics:

1. **Absolute Error** $e_{xy} = \sqrt{(x - \hat{x})^2 + (y - \hat{y})^2}$ is the Euclidean distance between the ground-truth and the prediction.

2. **Cross-Corridor Error** $e_x = |x - \hat{x}|$ is the error perpendicular to the corridor.

3. **Along-Corridor Error** $e_y = |y - \hat{y}|$ is error along the direction of the corridor.

6.5 Baselines

We compare our system with the following baselines: [1]

1. **Zhao et al. [65].** This work uses peak-based ray tracing to localize people around corners. It first detects the highest power pixel in an RF-Snapshot, then performs ray tracing from that pixel to recover the location of the target. The method requires the floor layout. So we feed it the ground-truth layout (manually labeled using the floor plan) for ray tracing purpose.

2. **Zhao et al. [65] (Multi-bounce).** The original paper by Zhao et al. [65] allows RF signals to bounce off only one wall. This is too restrictive since RF signals in a corridor tend to bounce multiple times before reaching the receiver. Thus, for fair comparison with CornerRadar, we also compare with an updated version of this work where the RF signal can bounce off side walls multiple times if the environment allows for such bouncing. We denote the augmented method Zhao et al. [65] (Multi-bounce).

3. **Scheiner et al. [47]** This method was designed for outdoor environments to allow a car to detect other cars and pedestrians around street corners. Instead

[1] None of the baselines is open sourced; therefore we implement them with our best effort.

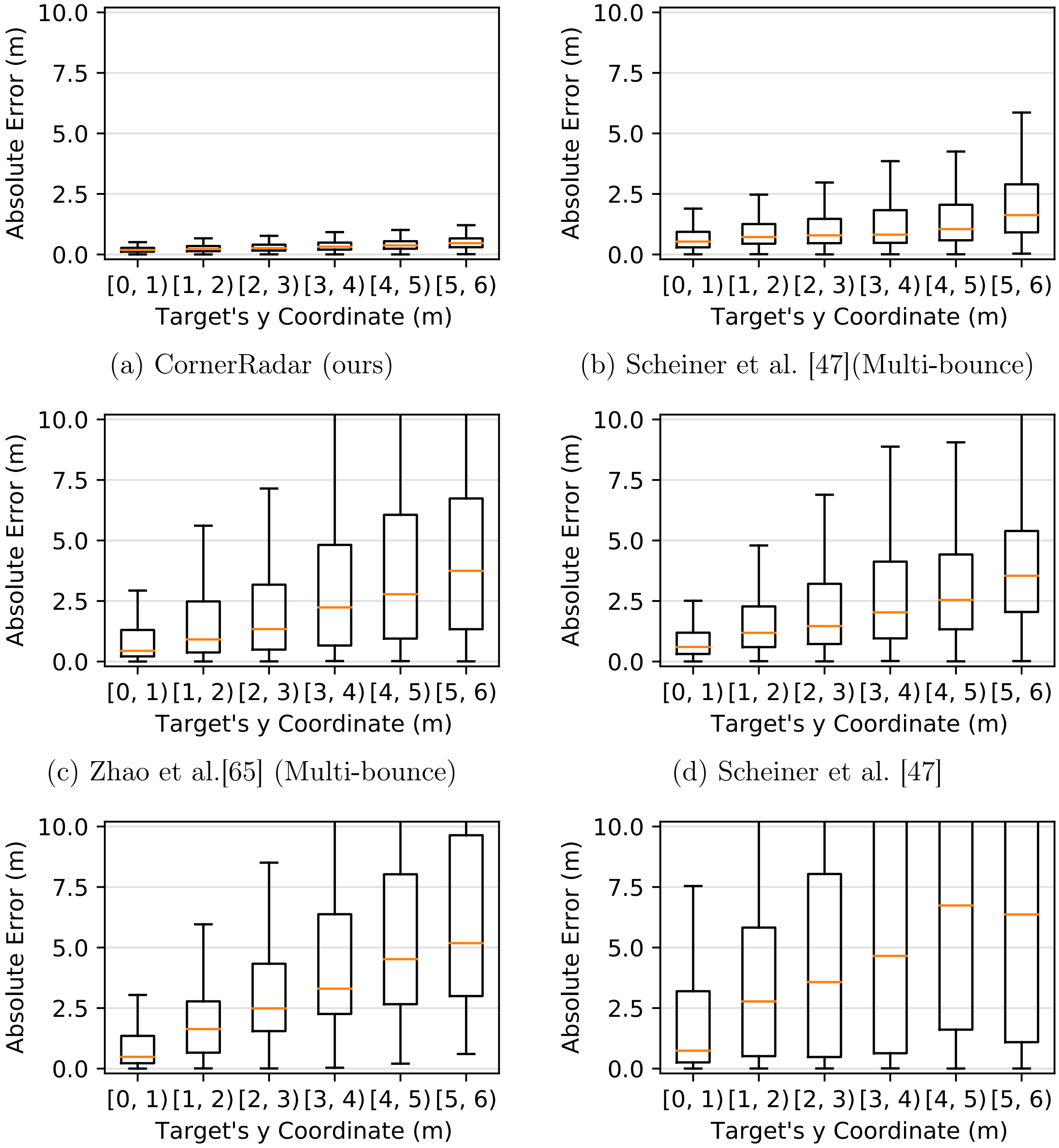
of just ray tracing the main peak in the RF-Snapshot as in Zhao et al. [65], this method uses all-pixels ray tracing, i.e., it applies ray tracing to every pixel in the RF-Snapshot and projects the results on the spatial layout. This projection is then fed to a neural network to extract the exact location of the hidden target. Further, while the paper provides an algorithm for detecting the layout using the car's LiDAR, their algorithm assumes all wall segments are at least one meter long, which is not true for indoor scenarios. Thus, provide this baseline with ground-truth layout for ray tracing purpose.

4. **Scheiner et al. [47] (Multi-bounce)** Like Zhao et al. [65], Scheiner et al. [47] also assumes the signal bounces only once before reaching the target. Thus, we consider a stronger baseline, Scheiner et al. [47](Multi-bounce), which allows the signal to bounce multiple times when backtracking each pixel.

5. **Adib et al. [2]:** Finally, we compare CornerRadar with a standard technique for device-free indoor localization based on the work of Adib et al. [2]. It localizes the target by detecting the highest power peak in the RF-Snapshot. As stated earlier, this traditional line of work assumes the presence of a direct propagation path, which is not the case in around-corner scenarios. However, we compare with it to show that this assumption leads to very large errors when applied to around-corner scenarios.

6.6 Localization Accuracy

In this section, we evaluate CornerRadar's accuracy in localizing hidden people and compare it with the baselines. We apply CornerRadar and all baselines to the same dataset described above, and report the localization results. Figure 6-6 shows the distribution of errors for each scheme, as a function of the target's distance to the corner. The results are presented as a boxplot, where each box extends from the 25-th percentile to the 75-th percentile, and the line in the middle shows the median.



(a) CornerRadar (ours)

(b) Scheiner et al. [47](Multi-bounce)

(c) Zhao et al.[65] (Multi-bounce)

(d) Scheiner et al. [47]

(e) Zhao et al.[65]

(f) Adib et al. [2]

Figure 6-6: Box plots of absolute error of CornerRadar and baselines as a function of the person's distance from the entry to the corridor (i.e., the target's y coordinate). Each box is drawn from 25th percentile to 75th percentile with an orange horizontal line drawn in the middle to denote the median. The whiskers extending from the boxes denote the minimum and maximum.

Figure 6-6 shows that CornerRadar is highly accurate and its median localization error is 3x to 12x smaller than the baselines. Figure 6-6 also shows that the deeper the person is inside the corridor, the larger the error for all schemes. This is due to two reasons. First, the radio uses an antenna array to detect the signal's angle of arrival. An angle error of 1 degree translates to a larger Euclidean error at distance. Second, the farther away the person is, the longer the RF signal travels, and the more likely it bounces off multiple walls, which causes attenuation and reduces the signal to noise ratio (SNR). Nonetheless, CornerRadar significantly outperforms the baselines. Furthermore, CornerRadar's errors fall in a tight range, whereas the baselines produce a wide range of errors indicating low robustness.

Figure 6-6 also shows a performance gap between the baselines [65, 47] and their multi-bounce versions. For example, comparing Figure 6-6d and Figure 6-6b, we find that allowing the RF signals to bounce off walls multiple times significantly improves the performance of Scheiner et al. [47]. The median errors are halved almost across all target's distances. Similarly, Figure 6-6e and Figure 6-6c show that Zhao et al. [65]'s error is reduced when allowing for multiple bounces. Also, the farther away the person is, the more the error decreases. This is because when the target is far from the corner, the RF signal cannot reach the target by bouncing only once.

6.7 Presence or Absence of Direct Path

Table 6.1 summarizes the median localization error across all experiments for all schemes as well as their performance in the presence and absence of the direct path. The table shows that CornerRadar's performance is the same with and without a direct path. In contrast the performance of the baselines varies significantly depending on whether a direct path exists. In particular, Adib et al. [2] assumes that RF signals traverse walls and occlusions. Hence, it has a very good performance when the direct path is available, yet its performance is very poor when the direct path is blocked. In contrast, past schemes for localizing people around corners, i.e., Zhao et al. [65], Scheiner et al. [47], and their variants perform well in the absence of a direct path,

Table 6.1: Comparison of localization error (in meters) of CornerRadar and baselines. The table reports the median error for all environments, and the median errors for environments with and without a direct path.

(a) all environments

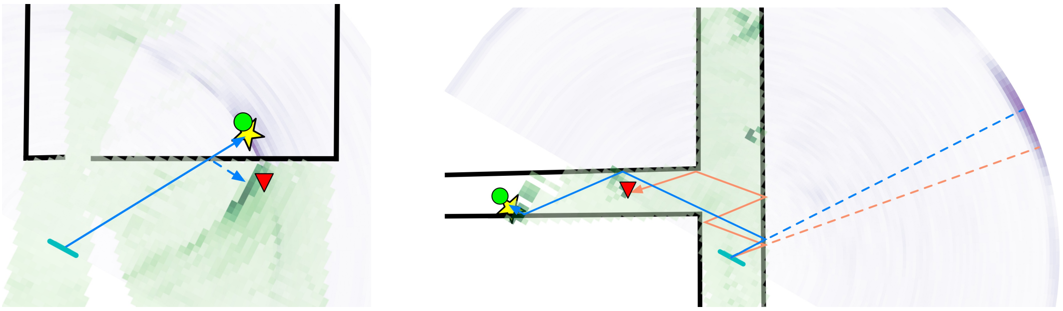
Method	e_x	e_y	e_{xy}
Adib et al. [2] (Direct-path)	1.166	0.662	3.368
Zhao et al. [65]	1.469	0.447	2.396
Zhao et al. [65] (Multi-bounce)	0.428	0.459	1.319
Scheiner et al. [47]	0.428	1.366	1.597
Scheiner et al. [47] (Multi-bounce)	0.341	0.618	0.832
CornerRadar (ours)	**0.168**	**0.138**	**0.266**

(b) without direct-path

Method	e_x	e_y	e_{xy}
Adib et al. [2] (Direct-path)	2.056	3.905	4.855
Zhao et al. [65]	0.463	1.351	2.082
Zhao et al. [65] (Multi-bounce)	0.372	0.434	0.696
Scheiner et al. [47]	0.414	1.276	1.497
Scheiner et al. [47] (Multi-bounce)	0.308	0.582	0.760
CornerRadar (ours)	**0.169**	**0.137**	**0.263**

(c) with direct-path

Method	e_x	e_y	e_{xy}
Adib et al. [2] (Direct-path)	0.175	0.152	0.290
Zhao et al. [65]	1.821	0.385	2.715
Zhao et al. [65] (Multi-bounce)	1.829	0.402	2.720
Scheiner et al. [47]	0.477	1.606	1.869
Scheiner et al. [47] (Multi-bounce)	0.491	0.768	1.126
CornerRadar (ours)	**0.165**	**0.139**	**0.272**



(a) Example of Mirroring

(b) Example of Splitting

Figure 6-7: Empirical examples of splitting and mirroring. The received RF-Snapshot is plotted in shaded purple and the ray-traced RF-Snapshot is plotted in shaded green. The target's true location is represented by a yellow star, the prediction of CornerRadar is represented by a green circle and the prediction of Scheiner et al. [47] is represented by a red triangle. The blue and orange lines in (b) show that the same blob of RF power (in shaded purple) is split after ray tracing between two locations. The figure shows that the approach of Scheiner et al. suffers from splitting and mirroring errors, whereas CornerRadar avoids such problems.

but degrade badly when the direct path is available. This is because those schemes ignore that RF signals can traverse walls, and hence when such event occurs, they lead to large errors.

6.8 Empirical Examples of Mirroring and Splitting

In Chapter 4, we described two limitations of ray tracing that affect past work: mirroring and splitting. In this section, we show empirical data that exhibit those problems. Figure 6-7 shows data from two environments in our dataset. Figure 6-7a shows an example of mirroring. The target (indicated by the yellow star) is actually located inside the room, but Scheiner et al. [47] positions him at its mirror location outside the room (as indicated by the red triangle). Figure 6-7b shows an example of splitting. All-pixels-ray-tracing (shown in shaded green) creates multiple bright spots, one near the entrance of the corridor, and one near the true location. The neural network in Scheiner et al. [47] gets confused between these two spots and picks the wrong location (as indicated by the red triangle).

In comparison, CornerRadar predicts the correct location in both examples and

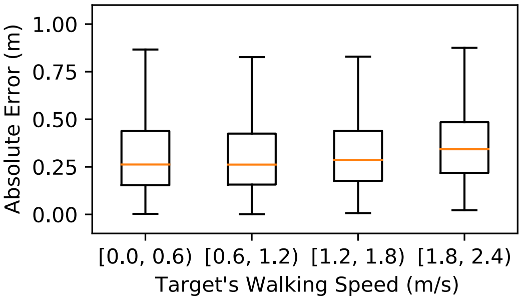
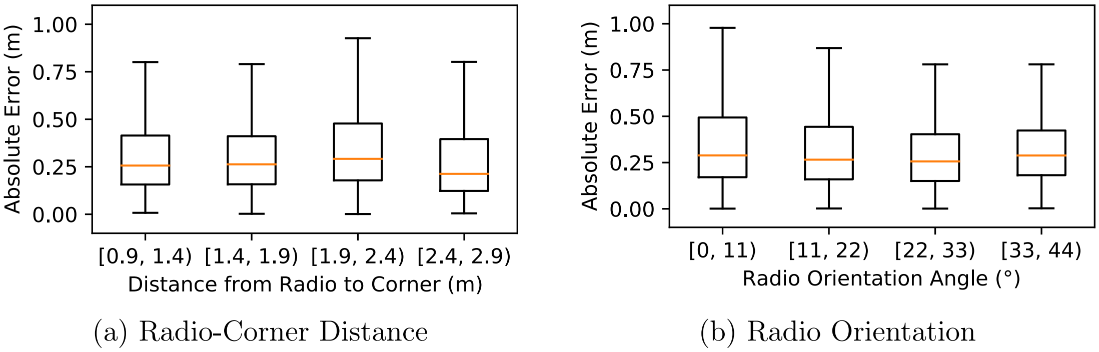


Figure 6-8: CornerRadar's robustness to target's speed.



(a) Radio-Corner Distance

(b) Radio Orientation

Figure 6-9: CornerRadar's robustness to radio placement and orientation.

avoids splitting and mirroring (as indicated by the green circles). This is because our neural network does not operate on all-pixels-ray tracing. Our neural network takes the RF-snapshots, the R-Map and the D-Map. The combination of the RF-snapshots and D-Map incorporates the fact that RF signals can propagate through walls, and allows the network to avoid mirroring. The combination of the RF-snapshots and R-Map shows the neural network that there is only one reflector in the RF-snapshots and hence it should not be split.

6.9 Robustness to Target's Speed

In real-world scenarios, people can walk slowly or move quickly. We would like to ensure that CornerRadar stays accurate regardless of the target's speed. Thus, in Figure 6-8, we plot CornerRadar's localization error as a function of the person's walking speed. The results show that CornerRadar's errors do not change with the person's speed. In all cases, the median error stays around 0.27m.

Table 6.2: Impact of CornerRadar's Components. The table reports the median error of potential variants of CornerRadar to highlight the importance of the ideas underlying the design.

Method	e_x	e_y	e_{xy}
CornerRadar (Full System)	**0.168**	**0.138**	**0.266**
No Hint Map	0.254	0.312	0.500
Replace Hint Map with Floor Map	0.237	0.273	0.441
Replace CCS with RCS	0.219	0.277	0.426

6.10 Robustness to Radio Placement and Orientation

We investigate CornerRadar's robustness to radio placement and radio orientation. For radio placement, we consider scenarios in which the radio is 1 to 3 meters away from the corner, since at such distances a moving robot may run into a person who suddenly turns the corner and appears in the robot's path. As for radio orientation, it is important to ensure that the radio can receive the RF signals that bounce off sidewalls. Hence, when fixing the radio on a robot, one should turn the radio slightly to the left or right. We explore angles that vary between 0 and 45 degrees.

Figure 6-9a and Figure 6-9b plot CornerRadar's error as a function of radio placement and radio orientation. They show that our system maintains low errors across a wide variety of placements and orientations.

6.11 Performance Analysis of CornerRadar's Components

The design of CornerRadar involves multiple components and sub-components. In this section, we evaluate how some of these components impact localization accuracy. Specifically, we evaluate the following variants of our system:

1. **No Hint Map**: To show the importance of the hint map, we evaluate a variant of CornerRadar that feeds the RF-Snapshots to the neural network without the

hint map.

2. **Replace Hint Map with Floor Map**: One may wonder whether simply providing the neural network with the floor map is sufficient for good performance. Thus we evaluate a variant of CornerRadar where the hint map is replaced by the floor map. The floor plan is provided to the neural network as a binary image, where the pixels that belong to a wall are assigned 1 and the others are assigned 0.

3. **Replace CCS with RCS**: In this case, the neural network is given the hint map as input, but the coordinates in the hint map and the coordinates of the label are expressed in the radio coordinate system (RCS) instead of the corridor coordinate system (CCS).

Table 6.2 reports the median localization error of all of the above CornerRadar variants. First, we can see that the hint map is crucial to our system. Without the hint map, CornerRadar's error along the corridor is doubled (from 0.266m to 0.500m). The results also show that the floor map is not as effective as the hint map. In particular, replacing the hint map with the floor map causes the error along the corridor to become 1.6 times larger (0.441m vs. 0.266m). This is because, without the hint map, the neural network has to learn the ray tracing rules on its own from the data. Finally, the table also demonstrates the benefit of using the Corner Coordinate System (CCS) instead of the Radar Coordinate System (RCS). It shows that this change of coordinates leads to a localization error 1.6 times smaller.

6.12 Evaluation of Computational Requirements

While our data processing was done on a Titan Xp GPU, in this section, we show that CornerRadar can run in real-time on a small low-power platform like the NVIDIA Jetson Nano Developer Kit (Nano) [35]. There are three steps for CornerRadar to generate a prediction. 1) **RF-Snapshots Generation**: the system collects raw RF samples from the antenna and uses the antenna array equation and Fast Fourier

Table 6.3: CornerRadar's average running time for computing a localization reading on NVIDIA Jetson Nano. For reference, we also report the running time on NVIDIA Titan Xp. Step 1: RF-Snapshots Generation. Step 2: Hint Map Generation. Step 3: Neural Inference.

Device \ Step	Step 1	Step 2	Step 3	Total
NVIDIA Jetson Nano	40ms	25ms	58ms	123ms
NVIDIA Titan Xp	11ms	8ms	8ms	27ms

Transform to obtain the RF-Snapshots. 2) **Hint Map Generation**: the system takes the point clouds measured by the LiDAR and runs the layout detection algorithm to get the current layout. Then it uses the layout to generate the corresponding hint maps as explained in Section 5.1. 3) **Neural Inference**: The system finally feeds the RF-Snapshots and the hint maps to the trained neural network and gets a location prediction.

As shown in Table 6.3, CornerRadar generates a position reading in 123 milliseconds on Jetson Nano (40ms for generating the RF-snapshots, 25ms to generate the hint-map, and 58ms for the neural network inference). This shows that CornerRadar can support real-time operation using an on-board small IoT computer.

Chapter 7

Discussion

In this chapter, we discuss the implications of our results for robot navigation, and the limitation of the system.

7.1 Implications for Robot Navigation

Research on robot navigation has shown a significant interest in strategies for avoiding robots colliding with humans [54, 3, 49, 16, 55, 10]. Simply having robots sounding an alarm as they move, and expecting humans to avoid them is not sufficient. Real-world accidents show that robots can run over young kids [45], who are unlikely to pay attention to such alarms. The robotic community has two general strategies for dealing with collisions. The robot may take a conservative strategy [54, 3, 49] and move only when it is certain it will not collide with a human. Such a strategy however can lead to the freezing robot problem [53] –i.e., in complex scenarios, the robot may have a high uncertainty about potential collisions, and as a result cannot make any movements. Alternatively, the the robot may use an aggressive strategy [57] and keep moving until it sees a human that may collide with it. Such a strategy is dangerous and requires the robot to have very accurate location estimates to avoid collisions.

CornerRadar's ability to deliver accurate around-corner localization has direct implications for robot navigation. Consider a robot that follows a conservative strategy. Say the robot needs to sustain a 0.5m safety margin from people. If the localization system has an x m error, then the robot needs to find a path that is $0.5 + x$ m away from the person to guarantee the desired safety margin. Assuming that a corridor has a width of 2m, the person and the robot both have a 0.25m radius. If the localization error can exceed 1 meter, it will be infeasible for the robot to find a safe path, and the robot ends up freezing. Our results show that CornerRadar's error are low and in no case in our experiments exceeded 1 meter. In contrast, the errors in past work are high, and with high probability, exceed 1m.

Now consider a robot that follows an aggressive strategy. In this scenario, it is critical for the robot to have a long sensing range. The longer the sensing range is, the earlier the robot can detect the person and adjust its path. The localization results show that CornerRadar can maintain low localization errors even if the person is 6 meters within the corridor. In contrast, the errors in past work increase quickly with distance, and for the same error level, the sensing range will be much smaller.

7.2 Limitations

While CornerRadar significantly improves around corner localization, it also has a few limitations.

7.2.1 Moving Radio

Like all past work on around-corner sensing (both optical and RF-based), this thesis mainly consider the scenarios with a static radio. We have conducted a preliminary study on applying CornerRadar to moving platforms. Initial results show CornerRadar can potentially be extended to moving radios.

A moving radio is challenging because it makes it hard to eliminate background reflections and focus on the moving person. Specifically, when the radio is static, one can apply consecutive subtraction to the RF-Snapshots to remove reflections from static objects, mainly walls, and focus only on reflections from moving people. However, when the radio is moving, consecutive subtractions cannot remove the walls since they move with respect to the radio. In this case, wall reflections can clutter the RF-snapshots and look to the model as if they were moving people. This is a general problem that affects all past work on localization from RF reflections.

We tried a simple strategy to address this challenge. Since our layout-detection method provides the location of the walls, we tried to mask out (i.e., zero out) the RF power that corresponds to wall locations in the RF-snapshots – i.e., remove any RF power that comes from the walls. We then run our system on these updated RF-snapshots with no additional modifications. We conducted experiments with a moving cart, and applied this strategy. The resulting localization errors in this case are $e_x = 0.155$m, $e_y = 0.204$m and $e_{xy} = 0.315$m. While the errors have increased in comparison to a static radio, the results show the feasibility of extending our method to moving robots. A detailed investigation of moving radios however is beyond the scope of this thesis and left to future work.

7.2.2 Multiple Targets

We have considered only scenarios with one person moving at any time. We do not believe this to be a strong limitation because applications of around-corner localization are typically focused on detecting the person closest to the corner (e.g., the person who may collide with the robot as it turns the corner). Further, we believe that extensions to multiple unspecified number of people can be achieved by changing the CNN to output people locations as a probability over a 2D space, instead of simply outputting the x-y coordinates of a single person. This however requires collecting training data with multiple people in the scene, which is left for future work.

7.2.3 Curved Walls

Our layout-detection algorithm assumes that walls are straight. However, in some environments the walls around the corner may be curved. In this case, our layout-detection algorithm may generate an incorrect layout. It is feasible to extend the layout-generation algorithm to deal with curved walls, albeit with increased computational complexity. No other component in CornerRadar assumes straight walls.

Thus, we believe with a modified layout detection and enough training data from curved-wall environments, CornerRadar can work well with curved walls. However, corners with curved walls are relatively uncommon, and we have not encountered such environment during data collection. In future work, we will consider extending CornerRadar to handle curved-wall scenarios.

Chapter 8

Conclusion

In this thesis, we present CornerRadar, a novel RF system that can localize people around corners. The combination of the hint map and the localization CNN enables CornerRadar to overcome the deficiency of ray tracing in indoor environments and leads to more robust performance in the presence and absence of a direct propagation path. Empirical evaluation from a large number of indoor environments demonstrates that CornerRadar is both accurate and robust. We believe that our system can serve as a fundamental building block that complements past work on indoor RF-based localization.

Bibliography

[1] Fadel Adib, Zach Kabelac, Dina Katabi, and Robert C Miller. 3d tracking via body radio reflections. In *11th USENIX Symposium on Networked Systems Design and Implementation (NSDI 14)*, pages 317–329, 2014.

[2] Fadel Adib, Zachary Kabelac, and Dina Katabi. Multi-person localization via rf body reflections. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 279–292, 2015.

[3] Daniel Althoff, James J Kuffner, Dirk Wollherr, and Martin Buss. Safety assessment of robot trajectories for navigation in uncertain and dynamic environments. *Autonomous Robots*, 32(3):285–302, 2012.

[4] Albin Antony and P Sivraj. Food delivery automation in restaurants using collaborative robotics. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 111–117. IEEE, 2018.

[5] Roshan Ayyalasomayajula, Aditya Arun, Chenfeng Wu, Sanatan Sharma, Abhishek Rajkumar Sethi, Deepak Vasisht, and Dinesh Bharadia. Deep learning based wireless localization for indoor navigation. In *Proceedings of the 26th Annual International Conference on Mobile Computing and Networking*, pages 1–14, 2020.

[6] Mufeed Batarseh, Sergey Sukhov, Zhiqin Shen, Heath Gemar, Reza Rezvani, and Aristide Dogariu. Passive sensing around the corner using spatial coherence. *Nature communications*, 9(1):1–6, 2018.

[7] Katherine L Bouman, Vickie Ye, Adam B Yedidia, Frédo Durand, Gregory W Wornell, Antonio Torralba, and William T Freeman. Turning corners into cameras: Principles and methods. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2270–2278, 2017.

[8] Carmen Cerasoli. The use of ray tracing models to predict mimo performance in urban environments. In *MILCOM 2006-2006 IEEE Military Communications conference*, pages 1–8. IEEE, 2006.

[9] Wenzheng Chen, Simon Daneau, Fahim Mannan, and Felix Heide. Steady-state non-line-of-sight imaging. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 6790–6799, 2019.

[10] Yuying Chen, Congcong Liu, Bertram E Shi, and Ming Liu. Robot navigation in crowds by graph convolutional networks with attention learned from human gaze. *IEEE Robotics and Automation Letters*, 5(2):2754–2761, 2020.

[11] Li-Xuan Chuo, Zhihong Luo, Dennis Sylvester, David Blaauw, and Hun-Seok Kim. Rf-echo: A non-line-of-sight indoor localization system using a low-power active rf reflector asic tag. In *Proceedings of the 23rd Annual International Conference on Mobile Computing and Networking*, pages 222–234, 2017.

[12] Robert L Cook, Thomas Porter, and Loren Carpenter. Distributed ray tracing. In *Proceedings of the 11th annual conference on Computer graphics and interactive techniques*, pages 137–145, 1984.

[13] Pieter PN de Groen. An introduction to total least squares. *arXiv preprint math/9805076*, 0(0):0, 1998.

[14] Andrew S Glassner. *An introduction to ray tracing.* Morgan Kaufmann, 1989.

[15] Magnus Gustafsson, Åsa Andersson, Tommy Johansson, Stefan Nilsson, Ain Sume, and Anders Örbom. Extraction of human micro-doppler signature in an urban environment using a "sensing-behind-the-corner" radar. *IEEE Geoscience and Remote Sensing Letters*, 13(2):187–191, 2015.

[16] Dirk Helbing and Peter Molnar. Social force model for pedestrian dynamics. *Physical review E*, 51(5):4282, 1995.

[17] Toshiro Higuchi, Koichi Oka, and Hiroshi Sugawara. Clean room robot with non-contact joints using magnetic bearings. *Advanced robotics*, 7(2):105–119, 1992.

[18] Intel. Lidar camera l515. `https://www.intelrealsense.com/lidar-camera-l515/`, 2020.

[19] Yifei Jiang, Xin Pan, Kun Li, Qin Lv, Robert P Dick, Michael Hannigan, and Li Shang. Ariel: Automatic wi-fi based room fingerprinting for indoor localization. In *Proceedings of the 2012 ACM conference on ubiquitous computing*, pages 441–450, 2012.

[20] Tommy Johansson, Åsa Andersson, Magnus Gustafsson, and Stefan Nilsson. Positioning of moving non-line-of-sight targets behind a corner. In *2016 European Radar Conference (EuRAD)*, pages 181–184. IEEE, 2016.

[21] Kiran Joshi, Dinesh Bharadia, Manikanta Kotaru, and Sachin Katti. Wideo: Fine-grained device-free motion tracing using rf backscatter. In *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, pages 189–204, 2015.

[22] Ori Katz, Pierre Heidmann, Mathias Fink, and Sylvain Gigan. Non-invasive single-shot imaging through scattering layers and around corners via speckle correlations. *Nature photonics*, 8(10):784–790, 2014.

[23] Ori Katz, Eran Small, and Yaron Silberberg. Looking around corners and through thin turbid layers in real time with scattered incoherent light. *Nature photonics*, 6(8):549–553, 2012.

[24] Jonathan Klein, Christoph Peters, Jaime Martín, Martin Laurenzis, and Matthias B Hullin. Tracking objects outside the line of sight using 2d intensity images. *Scientific reports*, 6(1):1–9, 2016.

[25] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, pages 269–282, 2015.

[26] Swarun Kumar, Stephanie Gil, Dina Katabi, and Daniela Rus. Accurate indoor localization with zero start-up cost. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 483–494, 2014.

[27] Shengjie Li, Zhaopeng Liu, Yue Zhang, Qin Lv, Xiaopeng Niu, Leye Wang, and Daqing Zhang. Wiborder: Precise wi-fi based boundary sensing via through-wall discrimination. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–30, 2020.

[28] Xiang Li, Shengjie Li, Daqing Zhang, Jie Xiong, Yasha Wang, and Hong Mei. Dynamic-music: accurate device-free indoor localization. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 196–207, 2016.

[29] Xiang Li, Daqing Zhang, Qin Lv, Jie Xiong, Shengjie Li, Yue Zhang, and Hong Mei. Indotrack: Device-free indoor human tracking with commodity wi-fi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–22, 2017.

[30] David B Lindell, Gordon Wetzstein, and Matthew O'Toole. Wave-based non-line-of-sight imaging using fast fk migration. *ACM Transactions on Graphics (TOG)*, 38(4):1–13, 2019.

[31] Robert Linnehan and John Schindler. Multistatic scattering from moving targets in multipath environments. In *2009 IEEE Radar Conference*, pages 1–6. IEEE, 2009.

[32] Christopher A Metzler, Felix Heide, Prasana Rangarajan, Muralidhar Madabhushi Balaji, Aparna Viswanath, Ashok Veeraraghavan, and Richard G Baraniuk. Deep-inverse correlography: towards real-time high-resolution non-line-of-sight imaging. *Optica*, 7(1):63–71, 2020.

[33] Felix Naser, Igor Gilitschenski, Alexander Amini, Christina Liao, Guy Rosman, Sertac Karaman, and Daniela Rus. Infrastructure-free nlos obstacle detection for autonomous cars. In *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 250–257. IEEE, 2019.

[34] Felix Naser, Igor Gilitschenski, Guy Rosman, Alexander Amini, Fredo Durand, Antonio Torralba, Gregory W Wornell, William T Freeman, Sertac Karaman, and Daniela Rus. Shadowcam: Real-time detection of moving obstacles behind a corner for autonomous vehicles. In *2018 21st International Conference on Intelligent Transportation Systems (ITSC)*, pages 560–567. IEEE, 2018.

[35] NVIDIA. Nvidia jetson nano developer kit, 2021.

[36] Alexander Orth, Patrick Kwiatkowski, and Nils Pohl. A novel approach for a mimo fmcw radar system with frequency steered antennas for 3d target localization. In *2019 16th European Radar Conference (EuRAD)*, pages 37–40. IEEE, 2019.

[37] Don Parker and David C Zimmermann. Phased arrays-part 1: theory and architectures. *IEEE transactions on microwave theory and techniques*, 50(3):678–687, 2002.

[38] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. In *Advances in neural information processing systems*, pages 8026–8037, 2019.

[39] Andrei Popleteev. Device-free indoor localization using ambient radio signals. In *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*, pages 549–552, 2013.

[40] Olivier Rabaste, Jonathan Bosse, Dominique Poullin, Israel Hinostroza, Thierry Letertre, Thierry Chonavel, et al. Around-the-corner radar: Detection and localization of a target in non-line of sight. In *2017 IEEE Radar Conference (RadarConf)*, pages 0842–0847. IEEE, 2017.

[41] Anshul Rai, Krishna Kant Chintalapudi, Venkata N Padmanabhan, and Rijurekha Sen. Zee: Zero-effort crowdsourcing for indoor localization. In *Proceedings of the 18th annual international conference on Mobile computing and networking*, pages 293–304, 2012.

[42] Juhi Ranjan, Yu Yao, and Kamin Whitehouse. An rf doormat for tracking people's room locations. In *Proceedings of the 2013 ACM international joint conference on Pervasive and ubiquitous computing*, pages 797–800, 2013.

[43] Joseph Redmon and Ali Farhadi. Yolov3: An incremental improvement. In *arXiv*, page 1, 2018.

[44] Giulia Sacco, Emanuele Piuzzi, Erika Pittella, and Stefano Pisa. An fmcw radar for localization and vital signs measurement for different chest orientations. *Sensors*, 20(12):3489, 2020.

[45] Saqib Shah. Amazon workers hospitalized after warehouse robot releases bear repellent, 2018.

[46] Charles Saunders, John Murray-Bruce, and Vivek K Goyal. Computational periscopy with an ordinary digital camera. *Nature*, 565(7740):472–475, 2019.

[47] Nicolas Scheiner, Florian Kraus, Fangyin Wei, Buu Phan, Fahim Mannan, Nils Appenrodt, Werner Ritter, Jurgen Dickmann, Klaus Dietmayer, Bernhard Sick, et al. Seeing around street corners: Non-line-of-sight detection and tracking in-the-wild using doppler radar. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2068–2077, 2020.

[48] Pawan Setlur, Tadahiro Negishi, Natasha Devroye, and Danilo Erricolo. Multipath exploitation in non-los urban synthetic aperture radar. *IEEE Journal of Selected Topics in Signal Processing*, 8(1):137–152, 2013.

[49] Derek Seward, Conrad Pace, and Rahee Agate. Safe and effective navigation of autonomous robots in hazardous environments. *Autonomous Robots*, 22(3):223–242, 2007.

[50] Andrew G Stove. Linear fmcw radar techniques. In *IEE Proceedings F (Radar and Signal Processing)*, volume 139, pages 343–350. IET, 1992.

[51] Ain Sume, Magnus Gustafsson, Magnus Herberthson, Anna Janis, Stefan Nilsson, Jonas Rahm, and Anders Orbom. Radar detection of moving targets behind corners. *IEEE Transactions on Geoscience and Remote Sensing*, 49(6):2259–2267, 2011.

[52] Tatsuro Terakawa, Masaharu Komori, Kippei Matsuda, and Shinji Mikami. A novel omnidirectional mobile robot with wheels connected by passive sliding joints. *IEEE/ASME Transactions on Mechatronics*, 23(4):1716–1727, 2018.

[53] Peter Trautman and Andreas Krause. Unfreezing the robot: Navigation in dense, interacting crowds. In *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 797–803. IEEE, 2010.

[54] Jur Van Den Berg, Pieter Abbeel, and Ken Goldberg. Lqg-mp: Optimized path planning for robots with motion uncertainty and imperfect state information. *The International Journal of Robotics Research*, 30(7):895–913, 2011.

[55] Jur Van Den Berg, Stephen J Guy, Ming Lin, and Dinesh Manocha. Reciprocal n-body collision avoidance. In *Robotics research*, pages 3–19. Springer, 2011.

[56] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-level localization with a single wifi access point. In *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, pages 165–178, 2016.

[57] Gorka Velez and Oihana Otaegui. Embedding vision-based advanced driver assistance systems: a survey. *IET Intelligent Transport Systems*, 11(3):103–112, 2017.

[58] Andreas Velten, Thomas Willwacher, Otkrist Gupta, Ashok Veeraraghavan, Moungi G Bawendi, and Ramesh Raskar. Recovering three-dimensional shape around a corner using ultrafast time-of-flight imaging. *Nature communications*, 3(1):1–8, 2012.

[59] Andreas Velten, Di Wu, Adrian Jarabo, Belen Masia, Christopher Barsi, Chinmaya Joshi, Everett Lawson, Moungi Bawendi, Diego Gutierrez, and Ramesh Raskar. Femto-photography: capturing and visualizing the propagation of light. *ACM Transactions on Graphics (ToG)*, 32(4):1–8, 2013.

[60] Wilhelm H Von Aulock. Properties of phased arrays. *Proceedings of the IRE*, 48(10):1715–1727, 1960.

[61] Jonathan Wittemeier, Aya Mostafa Ahmed, Thanh Nhat Tran, Aydin Sezgin, and Nils Pohl. 3d localization using a scalable fmcw mimo radar design. In *2020 German Microwave Conference (GeMiC)*, pages 100–103. IEEE, 2020.

[62] Yaxiong Xie, Jie Xiong, Mo Li, and Kyle Jamieson. md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking. In *The 25th Annual International Conference on Mobile Computing and Networking*, pages 1–16, 2019.

[63] Jie Xiong and Kyle Jamieson. Arraytrack: A fine-grained indoor location system. In *Presented as part of the 10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, pages 71–84, 2013.

[64] Dongheng Zhang, Yang Hu, and Yan Chen. Mtrack: Tracking multi-person moving trajectories and vital signs with radio signals. *IEEE Internet of Things Journal*, 0(0):0, 2020.

[65] Qingsong Zhao, Guolong Cui, Shisheng Guo, Wei Yi, Lingjiang Kong, and Xiaobo Yang. Millimeter wave radar detection of moving targets behind a corner. In *2018 21st International Conference on Information Fusion (FUSION)*, pages 2042–2046. IEEE, 2018.

[66] Xiuyuan Zheng, Jie Yang, Yingying Chen, and Yu Gan. Adaptive device-free passive localization coping with dynamic target speed. In *2013 Proceedings IEEE INFOCOM*, pages 485–489. IEEE, 2013.