HIGH-PERFORMANCE WIRELESS PERCEPTION USING DEEP LEARNING AND MEMS DEVICES

BY

JUNFENG GUAN

DISSERTATION

Submitted in partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois Urbana-Champaign, 2022

Urbana, Illinois

Doctoral Committee:

Assistant Professor Haitham Al-Hassanieh, Chair
Professor Romit Roy Choudhury
Professor Sanjay Patel
Dr. Alberto Valdes Garcia, IBM

Abstract

Recent years have witnessed much interest in expanding the use of wireless networks beyond their traditional use for communications to providing new perception solutions, such as sensing, imaging, and localization. The vision is, the wireless perception functionalities of the next generation wireless networks are going to create a digital twin of the physical world.

This thesis introduces new software and hardware primitives that advance wireless technologies towards achieving the vision of ubiquitous perception in next-generation wireless networks.

The software primitive we introduce is AI-enhanced wireless imaging, where we leverage recent advances in deep neural networks to extract the underlying perceptual and contextual information of the environment from raw wireless images. We demonstrate the applications of AI-enhanced wireless imaging in self-driving car perception, where we develop systems to achieve millimeter-wave radar-based high-resolution imaging and accurate object detection.

The hardware primitive we introduce is the first of its kind Micro-Electro-Mechanical System (MEMS) filter hardware, which we leverage to enable joint communication and high-performance sensing in next-generation wireless networks. Towards this end, we first present a spectrum sensing scheme that can efficiently sense wideband spectra with high time resolution. This system can be used to enable dynamic spectrum sharing between perception and communication services in future wireless networks for them to coexist. We also exploit reusing communication signals for perception. We develop an accurate Internet-of-Things (IoT) self-localization system that simply overhears ambient 5G communications signals without any coordination with the base stations in 5G cellular networks.

Dedicated to my parents and Han.

Acknowledgments

First and foremost, I want to express my deepest gratitude to my advisor Haitham Hassanieh, who has been and will always be my role model. He has enabled me to grow and achieve goals that were beyond my wildest dreams. I have enormous respect and gratefulness for the guidance and support he has provided to me.

I am very grateful to my Ph.D. committee members and mentors too - Prof. Romit Roy Chowdary, Dr. Alberto Valdes-Garcia, and Prof. Sanjay Patel. Romit was like a second advisor to me, especially in the first two years of my Ph.D journey. The discussions and brainstorming in the SyNRG group helped me develop and improve all-around research skills. I also truly admire Romit's life wisdom and his genuine eager to pursue knowledge. Working with Alberto during my internship at IBM Research was a great learning experience, because he offered me the opportunity to explore and conduct a project as an independent researcher. Obviously, however, I couldn't have completed much of that project without the guidance and help from my other mentors, Dr. Bodhisatwa Sadhu and Dr. Arun Paidimarri. Last but not least, I also want to thank Sanjay for all his suggestions and kind support.

I also owe my sincere gratitude to many other mentors I've had along my Ph.D. journey. They are Prof. Saurabh Gupta, Prof. Songbin Gong, Prof. Deepak Vashisht, Prof. Kirill Levchenkov, Prof. Jin Zhou, Prof. Sibin Mohan, Dr. Radhika Gowaikar, Dr. Saeed Khosravirad, Jakub Sapis, and Dr. Sheng Shen. I've learned a lot from every single one of them.

A lot of my thesis work could not be done without the help of a few close collaborators, in particular, Suraj Jog, Sohrab Madani, Waleed Ahmed, Ruochen Lu, and Steffen Link. I want to especially thank Prof. Songbin Gong and Prof. Ruochen Lu, because it is their amazing work on the overtone MEMS resonators and filters that enables the spectrum sensing and IoT self-localization projects.

I also must acknowledge the National Science Foundation for supporting my Ph.D. study and research. I would like to thank Qualcomm as well, for awarding me the Qualcomm Innovation Fellowship.

The next name on my gratitude list is my alma mater, University of Illinois at Urbana-Champaign. I enjoyed every bit of my nine years at UIUC, through which I've grown from an 18-year-old international student to a Ph.D. It's not simply a place where I received my education, but also the place I would call my second home, and where my outlook on life and core values were formed. I am grateful to everyone I met here, especially my many friends and colleagues in SyNRG, ILIRM, and CSL: Jiaming, Jitian, Hailan, Ishani, Thomas, Karthik, Mahanth, Nirupam, Ashutosh, Zhijian, Walley, Hyungjoo, Yansong, Ali, Anming, Ashish, Megan, etc. I also feel so lucky to meet all of my classmates and friends in my undergraduate here.

I also want to thank all the obstacles, rejections, and failures in my life. They made me the stronger and better me.

Lastly but not least, I can never express sufficient gratefulness to my parents Shaobo and Yajun, my wife Han, for their endless love and support. I owe all my achievements to them.

Table of Contents

Chapter 1

Introduction

Wireless networks have traditionally been used for communications. However, as wireless technologies evolve, the applications of wireless networks are expanding beyond communications to perceiving the physical world. In fact, a central vision of next-generation wireless networks like 6G is to create digital twins of the physical and biological worlds using wireless perception functionalities integrated into wireless networks. Towards this vision, there has been significant research investigating the use of wireless signals for localization [1, 2, 3, 4, 5], health monitoring including measuring vital signs like breathing and heart rate [6, 7, 8], as well as wireless imaging for applications like behavioral sensing and autonomous driving [9, 10, 11, 12, 13].

Despite significant advances in the past decade, wireless perception systems remain limited in terms of resolution and accuracy. For example, autonomous radars used on self-driving cars, which are the top-notch wireless imaging technology in terms of performance and resolution, are still not ranked among the key sensing modalities in such applications. Self-driving cars nowadays mainly rely on LiDARs and cameras for perception. Radars only play supportive roles such as collision avoidance and parking assistance despite the fact the radar can work well in bad weather like fog and snow where LiDARs and cameras fail. However, the current imaging resolution of radar is nowhere near the resolution provided by optical sensors. Wireless signals also suffer from problems like multipath and specular reflections which lead to artifacts and missing parts of the scene in a wireless imaging system. Another example is in localization systems where the resolution is limited by the bandwidth of the wireless network. While new technologies like 5G have increased the available bandwidth, using large bandwidth consumes a lot of power and remains infeasible for many IoT applications. Moreover, a lot of the current proposals for localization require deploying extensive infrastructure or modifying the existing infrastructure which significantly hinders their

deployment in practice.

In addition to their poor performance, current wireless perception technologies cannot be easily integrated in next-generation wireless networks. In many cases, they use specialized waveforms, require a dedicated wireless spectrum, and utilize different processing hardware than the ones used in wireless networks. These gaps between prevent us from seamlessly integrating wireless perception into next-generation wireless communication networks.

The research presented in this thesis aims address the above problems by answering the following key question:

How can we enhance the resolution and accuracy of wireless perception systems & facilitate their integration into next generation wireless networks?

This thesis answers the above question by introducing new software and hardware primitives that advance wireless technologies towards fulfilling the vision of ubiquitous perception in next-generation wireless networks. In particular, we present:

- **AI-Enhanced Wireless Imaging:** where we introduce new deep neural network architectures along with novel signal processing algorithms to enable high resolution wireless imaging that does not suffer from artifacts or specularities. We demonstrate the effectiveness of our approach for the application of self-driving cars where we show through fog high resolution wireless imaging and accurate detection.

- **MEMS-Enhanced Wireless Sensing and Localization:** where we introduce the first of their kind Micro-Electro-Mechanical-Systems (MEMS) filters that look like spike trains in the frequency domain into the analog radio front-end. We show how combining these filters with new sparse recovery algorithms can enable joint communication and sensing in 5G. We demonstrate the effectiveness of our approach for the applications of spectrum sensing and IoT localization at low power and without the need to modify the 5G infrastructure.

Below, we describe the systems we built around these primitives to enhance wireless perception and demonstrate the ability for joint communication and sensing in 5G networks and beyond.

1.1 AI-Enhanced Wireless Imaging

1.1.1 High Resolution Imaging of Autonomous Vehicles Using Millimeter Wave Radar

There has been a lot of interest in achieving the vision of full autonomy for self-driving cars, which requires autonomous navigation in all weather conditions, including low visibility conditions like dense fog. However, one of the biggest roadblocks to this vision is the inability of optical perception sensors like cameras and LiDARs in harsh weather.

Millimeter-wave radars have more robust propagation characteristics and, hence, can operate reliably in low visibility condition. However, mmWave radars have not yet played central roles in the perception module of self-driving cars, because mmWave radars suffer from very low resolution, specularity, and noise artifacts.

To resolve the limitations of mmWave radar imaging, we develop *HawkEye*, a system that leverages a conditional GAN architecture to recover high-frequency shapes from low-resolution radar heatmaps. We propose a novel design that addresses challenges specific to the structure and nature of the radar signals involved. We also develop a data synthesizer to aid with large-scale dataset generation for training.

1.1.2 Accurate Detection of Autonomous Vehicles Using MIMO Millimeter Wave Radar

Although *HawkEye* has taken an initial step towards high-resolution mmWave radar imaging, it is far from a practical system that can image highly mobile scenes on board self-driving cars. Therefore, in the second system, we take a step further and use commercial off-the-shelf (COTS) mmWave radars known as the cascaded Multiple Input Multiple Output (MIMO) radar.

Furthermore, we also take the mmWave radar perception functionality to one level higher, from imaging to semantic scene understanding. We design *Radatron*, a system capable of accurate object detection using mmWave radar as a stand-alone sensor.

Radatron is trained with a real-world high-resolution automotive radar dataset that we collect in practical self-driving scenarios. We also develop

a novel hybrid radar processing and deep learning approach to achieve high vehicle detection accuracy.

1.2 MEMS-Enhanced Wireless Sensing and Localization

1.2.1 Efficient Wideband Spectrum Sensing

Dynamic spectrum sensing is becoming a new spectrum access paradigm in next-generation wireless networks. In this new paradigm, wireless perception applications can opportunistically access idle channels that are not being occupied by communication services. In order to enable dynamic spectrum sharing, one needs to first sense the wideband spectrum, which is power consuming and expensive due to the requirement for high-speed analog-to-digital converters (ADC). Although sub-Nyquist sampling techniques exist, they are built on the assumption of sparse spectrum and cannot work in dynamically shared spectrum with dense occupancy.

To overcome this challenge, we present S^3, an sensing system that can reconstruct the wideband spectrum occupancy while sampling the spectrum below the Nyquist rate using cheap and low-power ADCs. Besides, S^3 is able to operate in densely occupied spectrum and achieve high time resolution to detect instantaneous vacancies.

To do so, S^3 leverages a novel RF filtering solution - MEMS spike-train filters. These filters sparsify the spectrum while at the same time allowing S^3 to monitor a small fraction of bandwidth in every band. Using a new structured sparse recovery algorithm we design, S^3 is able to accurately detect the occupancy of multiple bands across a wide spectrum.

1.2.2 IoT Self-Localization Using Ambient 5G Signals

5G cellular networks, especially those in the mmWave bands, provide unique opportunities for localization, because of their wide signal bandwidth and small cell deployment of base stations. In this project, we try to utilize these unique features to enable a high-accuracy and scalable IoT self-localization

technique, which we name as *ISLA*.

Our approach is to have IoT devices simply overhearing ambient 5G signals and reuse them for localization without any coordination with the base stations, so that the IoT localization system does not occupy any base station resources and can hence scale. The key challenges of this system are: 1) Low-power IoT nodes are incapable of capturing wideband 5G signals, so that they cannot leverage the full bandwidth to achieve high localization accuracy. 2) The localization algorithm running on the IoT nodes must not require coordination with the base stations or modifications to the 5G standards.

To address these challenges, *ISLA* utilizes MEMS spike-train filters to stretch the effective localization bandwidth to 100 MHz while using 6.25 MHz receivers, improving ranging resolution by 16×. We also design new coordination-free localization algorithms, that leverages unique 5G-NR waveform structures.

1.3 Organization

The rest of this thesis is organized as follows. In the first part, we demonstrate AI-empowered high-resolution wireless imaging that can recover the underlying perceptual and contextual information of the physical world. We take the example of mmWave radar perception on self-driving cars in next-generation vehicular networks. In particular, chapter 2 presents *HawkEye* and how it recreates high-resolution 3D images of cars from mmWave radar heatmaps. Chapter 3 describes *Radatron* and how it accurately detect vehicles from raw radar heatmaps in practical highly mobile self-driving scenarios.

Then, before moving on to the other two systems presented in the second part of the thesis, we first give a primer on MEMS Spike-Train filters in chapter 4, which are the key enablers of S^3 and *ISLA*.

The second part of the thesis presents how we leverage the MEMS Spike-Train filters to enable joint communication and sensing. Chapter 5 presents S^3 and how it efficiently senses wideband spectra with high time resolution. Note that spectrum sensing can used to enable dynamic spectrum sharing and the coexistence between perception and communication services in next-generation networks. Finally, chapter 6 presents *ISLA* and how it co-designs

the accurate localization functionality for low-power IoT devices with the 5G cellular network by reusing the ambient 5G-NR communication signals.

We conclude the thesis and list some possible future directions in chapter 7.

Part I

AI-Enhanced Wireless Imaging

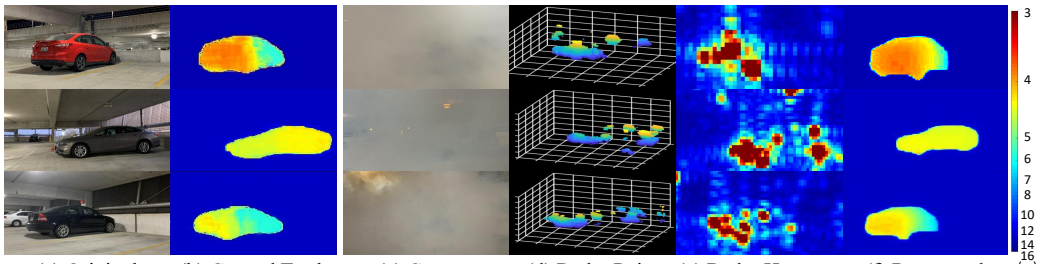
Chapter 2

Through Fog High Resolution Imaging Using Millimeter Wave Radar

This chapter demonstrates high-resolution imaging using millimeter-wave (mmWave) radars that can function even in dense fog. We leverage the fact that mmWave signals have favorable propagation characteristics in low visibility conditions, unlike optical sensors like cameras and LiDARs which cannot penetrate through dense fog. Millimeter-wave radars, however, suffer from very low resolution, specularity, and noise artifacts. We introduce *HawkEye*, a system that leverages a conditional GAN architecture to recover high-frequency shapes from raw low-resolution mmWave radar heatmaps. We propose a novel design that addresses challenges specific to the structure and nature of the radar signals involved. We also develop a data synthesizer to aid with large-scale dataset generation for training. We implement our system on a custom-built mmWave radar platform and demonstrate performance improvement over both standard mmWave radars and other competitive baselines.

2.1 Introduction

Achieving fully autonomous vehicles, referred to as *Level 5* in the standards for driving automation, has gained significant interest from major companies like Tesla, Ford, Honda, Waymo, Toyota, Uber, and NVIDIA [14, 15, 16, 17, 18, 19]. *Level 5* autonomy requires the ability to operate in severe weather conditions such as dense fog, smog, snowstorms, and sandstorms [16, 20]. Autonomous vehicles, however, mainly use cameras or LiDARs, to obtain an accurate and reliable view of the environment, which suffer in low visibility conditions and bad weather [21, 22, 23, 24, 25, 26]. Cameras also suffer at night in low light conditions. This is problematic as many manufacturers including Tesla avoid using LiDAR altogether, making cameras their primary



(a) Original Scene (b) Ground Truth (c) Camera in Fog (d) Radar Point Cloud in Fog (e) Radar Heatmap in Fog (f) Recovered Image

Figure 2.1: The figure above shows (a) the original scene, (b) ground truth depth map captured with stereo camera, (c) the scene in fog, (d) the mmWave radar point cloud generated in the presence of fog, (e) the corresponding radar heatmap, and (f) the recovered output of our system.

sensory module [27].[1]

Millimeter-wave (mmWave) radars offer more favorable characteristics due to their ability to work at night and penetrate through fog, snow and dust [29, 30]. However, car manufacturers today, still use mmWave radar for the sole purpose of unidirectional ranging, i.e., to determine the distance to other vehicles [31, 32]. Imaging using mmWave radar is challenging for several reasons. First, the resolution of mmWave radar is extremely low compared to LiDARs or cameras. Figure 2.1 (d,e) show an example where the radar image appears as blobs of radar reflections and carries little to no contextual and perceptual information as compared to the corresponding camera shown in Fig. 2.1 (a). Second, unlike optical signals, wireless mmWave signals are highly specular, i.e., the signals exhibit mirror-like reflections from the car [33]. As a result, not all reflections from the car propagate back to the radar receiver and major parts of the car do not appear in the image, making it impossible to detect its shape, as can be seen in Fig. 2.1 (d,e). Finally, wireless reflections from the car can also bounce off the road and other cars and travel along multiple paths to the radar receiver creating shadow reflections and artifacts in various locations in the scene as shown in Fig. 2.1 (d,e).

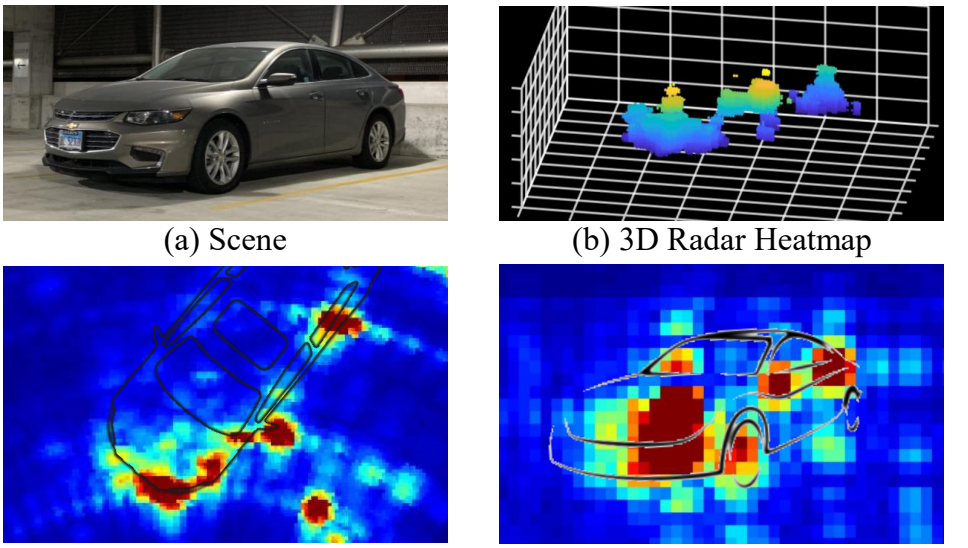
Today's commercial mmWave imaging systems, like airport scanners, use human-sized mechanically steerable arrays to improve the resolution. They also isolate the object being imaged in the near field to eliminate multipath reflections and rotate the arrays around the object to address specularity [34,

[1]Other modalities such as thermal imaging also fail in dense fog [28].

35]. However, such a design would be extremely bulky and not practical for self-driving cars as we have no control over the cars being imaged.

In this chapter, we present *HawkEye*, a system that consolidates advances to enable practical use of mmWave imaging radars in realistic applications. Using a mmWave radar, we can sense and predict shape for cars in the presence of dense fog. Figure 2.1 column (e) shows our system's predicted depth (as predicted from the sensed radar signal visualized in column (d)). Column (b) shows the ground truth depth map (recovered using a stereo camera in the absence of fog). While, obviously, we are far from a practical system that can use imaging radars on board self-driving cars, we have made huge advances towards this. This chapter describes the different aspects that have enabled this advance.

Our central contribution is to cast the problem of predicting high-frequency shape from raw radar heatmaps as a learning problem. Use of learning provides robustness to hard-to-model radar reflections and sources of noise like specularity and multipath reflections. At the same time, learning can effectively leverage priors on shapes of cars to make reasonable predictions from coarse radar heatmaps. However, use of learning for this task was non-trivial. We had to innovate on the design of the neural network, loss functions for training, and development of large-scale realistic datasets for training. We employed Generative Adversarial Network (GAN) [36] based architectures that consume mmWave radar heatmaps and predict high-resolution depth maps. The specific nature of the signal required the design of custom neural network architectures (that map 3D input heatmaps to 2D depth maps, and skip connections that project 3D information to 2D) and custom loss functions (combination of perceptual, L_1, and adversarial loss). We built a realistic radar data synthesizer that captures unique characteristics of radar. We used this module to create a synthesized dataset which was used to train our expressive neural networks. Finally, we built a real-world data collection platform to collect real data for fine-tuning and benchmarking. These all collectively enable the end-to-end system of *HawkEye*, which to the best of our knowledge, is the first system that can deliver the results shown in Fig. 2.1(f). Our results show that *HawkEye* is able to generate high-resolution depth maps from raw 3D radar heatmaps and accurately reconstruct the car in the real scene even in low visibility conditions like fog.



(a) Scene (b) 3D Radar Heatmap

(c) 2D Top View of Radar Heatmap (d) 2D Front View of Radar Heatmap

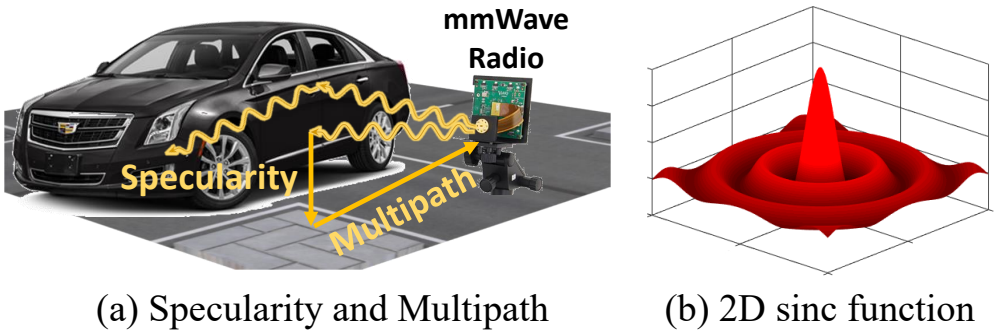
Figure 2.2: Output of the Millimeter Wave Imaging Radar

2.2 Background

Millimeter wave radar works by transmitting a wireless signal and receiving back the reflections from various objects in the scene. It operates in the high frequency bands such as 24 GHz, 60 GHz, and 77 GHz and uses techniques like Frequency Modulated Continuous Wave (FMCW) and antenna arrays to separate the received reflections.[2] The antenna arrays are electronically steered to capture and separate reflections from the scene based on their spatial direction (ϕ, θ) whereas FMCW is used to separate reflections based on the range (ρ) from the reflecting object. This allows us to compute a 3D heatmap $x(\phi, \theta, \rho)$ where each point represents the energy reflected from a voxel in space.

Figure 2.2 shows an example of the output of our radar. The 3D heatmap corresponding to the car in (a) is shown as a point cloud in (b). The point cloud is generated by thresholding out the voxels where the reflected signal energy is very weak. We also show projections of the 3D heatmap in the 2D top view in (c), and the 2D front view in (d). The figure also overlays the car's silhouette on the 2D heatmaps to better demonstrate where the reflections are coming from.

[2]At such high frequencies, there is abundant bandwidth available for FMCW signals. The signal wavelength is also small (millimeters) which enables the design of large compact antenna arrays [37, 38].



(a) Specularity and Multipath (b) 2D sinc function

Figure 2.3: Challenges in Millimeter Wave Imaging

As can be seen from the figure, mmWave radar imaging has fundamentally different challenges compared to camera and LiDAR data. First, the imaging resolution is significantly lower than vision. For example, in our system, the range resolution is 10 cm which is 3.3× worse than that of the commercial LiDAR [39]. The azimuth and elevation resolution is 5° which is 50× worse than LiDAR [39]. Range resolution depends on the FMCW signal bandwidth and can potentially be improved using more expensive hardware. However, angular resolution depends on the aperture of the antenna array. To achieve sub-degree angular resolution, similar to LiDAR, we would need a 9 m long antenna array which is impractical both in terms of cost and form factor.[3] For practical aperture sizes (few centimeters), the output mmWave radar image gets convolved with a very wide 2D *sinc* function along the azimuth and elevation dimensions similar to the one shown in Fig. 2.3 (b). The 2D *sinc* function eliminates almost all high frequency perceptual content such as object boundaries. That is why the mmWave image in Fig. 2.2 looks like blobs. The *sinc* sidelobes also create artifacts and noise in the image as can be seen in the 2D projections in Fig. 2.2.

Resolution, however, is not the only challenge. Unlike light, mmWave signals do not scatter as much and mainly reflect off surfaces. Hence, the car is highly specular and acts as a mirror reflector of radar signals. As a result, most reflections never trace back to the mmWave receiver. This leads to specularity as shown in Fig. 2.3 (a), making certain portions of the car impossible to image as can be seen in Fig. 2.2, where a large portion of the car's surface is missing. Moreover, due to multipath propagation, some

[3]Note that for systems like the airport security scanners, the target being imaged is in short range and hence, human sized arrays are sufficient.

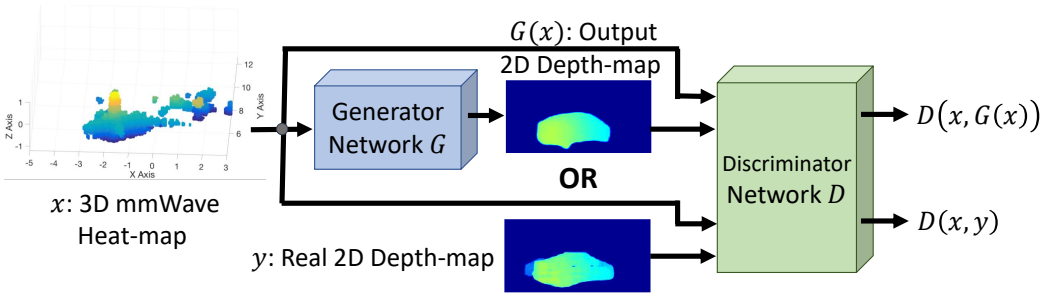


Figure 2.4: Conditional GAN Architecture

reflections bounce off the street and other obstacles and trace back to the receiver as shown in Fig. 2.3 (a) creating many artifacts in the image as can be seen in Fig. 2.2 (c,d). Finally, radar data has a different representation and perspective as compared to cameras. We must accommodate the above challenges in designing a neural network framework that is able to recover the shape, size, and location of the car being imaged.

2.3 Primer on GANs

GANs are generative models that have proven to be very successful since they are able to generate data samples that closely follow the distribution of data without explicitly learning the distribution. To do so, GANs adopt an adversarial learning paradigm where two players compete against each other in a *minimax* game [36]. The first player, the Generator G, attempts to generate data samples that mimic the true distribution, e.g. generate realistic 2D depth-maps of cars. The second player, the Discriminator D, attempts to differentiate between samples generated by G from real data samples e.g. differentiate output of G from ground-truth 2D depth-maps from stereo cameras. G and D keep learning until G can generate samples that D can no longer distinguish from real samples of the true distribution. At this stage, D is no longer needed and G can be used during inference to generate new samples.

HawkEye uses a variant called conditional GANs (or cGANs) where G attempts to mimic a distribution of the data conditioned on some input [40], i.e. $P(y|x)$ where x is the low resolution mmWave heat-map and y is the high resolution 2D depth-map. x is given as input to both the generator G and

the discriminator D. Figure 2.4 shows the architecture of a conditional GAN. Intuitively, D serves two purposes. First, it helps generalize G by eliminating dependency on the environment i.e., G can create realistic images of cars by learning features that are independent of the background and location of the car in the training scenes. Second, it teaches G to fill in the missing parts of the car due to specularity and eliminate artifacts causes by multipath, i.e., unless artifacts are removed and specularity is addressed, D will be able to tell that the output was generated by G.

Mathematically, G learns a mapping from the input 3D mmWave heatmap x to the output 2D depth-map $G(x)$. D, on the other hand, attempts to learn a mapping from the input x and a 2D depth-map to a probability $\in [0, 1]$ of the 2D depth-map being real y, or generated $G(x)$. A perfect discriminator would give $D(x, y) = 1$ and $D(x, G(x)) = 0$. Hence, to win the game against a given G, D tries to maximize the following objective function:

$$\mathcal{L}(G) = \max_D \Big(\mathbf{E}_y \big[\log D(x, y) \big] + \mathbf{E}_x \big[\log \left(1 - D(x, G(x))\right) \big] \Big),$$

where the first term is maximized when $D(x, y) = 1$ and the second term is maximized when $D(x, G(x)) = 0$, i.e. when D correctly classifies the images as real or generated.

Generator G on the other hand tries to minimize the above objective function (which is referred to as its loss function $\mathcal{L}(G)$), since its goal is to fool the Discriminator into classifying its output data samples as being real. Therefore, the GAN optimization is a *minimax* problem given by:

$$\min_G \Big(\max_D \Big(\mathbf{E}_y \big[\log D(x, y) \big] + \mathbf{E}_x \big[\log \left(1 - D(x, G(x))\right) \big] \Big) \Big).$$

Since the mapping functions in G and D can be very complex, G and D are implemented and optimized using deep convolutional neural networks. The final output of the above GAN optimization is a G^* that minimizes the loss function $\mathcal{L}(G)$ and can be used to generate 2D depth-maps from new unseen 3D mmWave heatmaps. A few points are worth noting:

- The generator G never actually sees any real ground-truth data. Instead, it must learn to create realistic images based only feedback it receives from the discriminator D.

- The GAN never explicitly learns the distribution of the data. Prior to

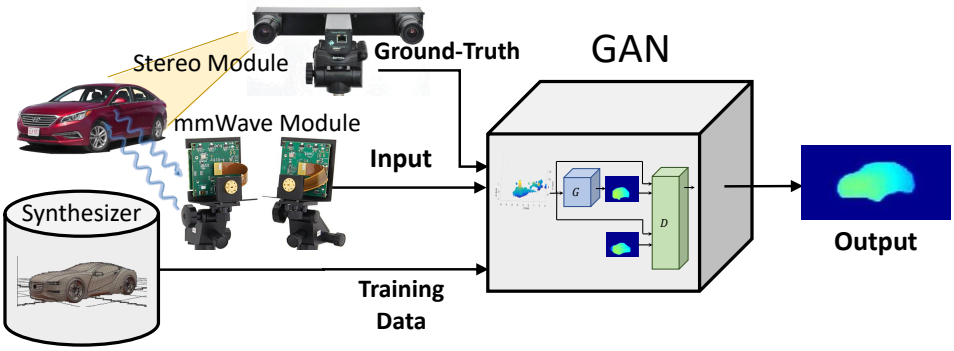


Figure 2.5: *HawkEye*'s **System Overview**

GANs, generative models would attempt to explicitly learn the distribution by approximating it as mixture of Gaussians or other simplified models that typically fail to capture the real distribution of data. GANs are very powerful since they can generate real looking samples without having to learn complex data distributions.

- The GAN adaptively learns its own loss function. Any machine learning model is trained by optimizing a given loss function that provides a quantitative measure of the model's performance e.g. ℓ_1 or ℓ_2 distance. Choosing the right loss function is a very difficult task [41]. GANs are powerful since they do not require a fixed hand-tuned loss function and rather can adapt the loss function in the above equation as they learn.

2.4 System Overview

HawkEye is a mmWave imaging system designed for autonomous vehicles. *HawkEye* can generate high resolution perceptually interpretable 2D depth-maps from 3D mmWave heat-maps. To do so, *HawkEye* has five modules shown in Fig. 2.5: a mmWave imaging module, a stereo camera, a simulator, a GAN, and a perception module. The stereo camera and simulator are used only during training and evaluation. The mmWave and GAN modules are used during training and testing. We summarize these components below:

- *GAN Model* (section 2.5): We design a new GAN architecture customized for mmWave imaging in self-driving cars. The GAN uses an encoder-

decoder paradigm, a modified loss function, and a skip connection to produce perceptually interpretable and accurate reconstructions of cars.

- *mmWave Radar Module* (section 2.6): We custom-build a mmWave imaging module using off-the-shelf 60 GHz radio and RF circuit components. It also uses a linear slider platform to emulate a large antenna array leveraging synthetic aperture radar, which provides us with more flexibility to test and experiment various parameters and setups. This module produces 3D mmWave heat-maps. However, on its own, it is intrinsically limited in resolution and suffers from specularity and artifacts caused by multipath.

- *Simulator* (section 2.7): This module augments the training dataset with synthesized data obtained from 3D CAD models of cars and mmWave ray tracing algorithms. It produces both the ground truth 2D depth-map and the synthesized 3D mmWave depth-maps.

- *Stereo Camera* (section 2.9): We custom-build a long-range stereo camera that can generate high-resolution 2D depth-maps of cars that serve as ground-truth for *HawkEye*.

Next, we will discuss these modules in more detail.

2.5 *HawkEye*'s GAN Architecture

2.5.1 Overview

We propose a conditional GAN [40] based architecture. Given an input mmWave RF heatmap x, we learn a conditional generator G. This conditional generator employs an encoder decoder architecture. Although mmWave heatmaps have low spatial resolution, they can achieve high resolution in the depth dimension due to their large sensing bandwidth. In order to retain these high-frequency details in depth, we use skip-connections [42] in our design. Our discriminator D takes in (x, y) or $(x, G(x))$ pairs, and learns to discriminate between them. The generator and discriminator are trained jointly, with the discriminator trying to distinguish generated output from ground truth, and the generator trying to fool the discriminator. We additionally use L_1 and perceptual losses to make the output of the generator consistent

with the instance being input to it. Figure 2.6 shows our architecture. The peculiarities of the raw mmWave signal requires us to carefully consider the design choices involved. We next provide necessary technical details and emphasize the important design choices. Further details of *HawkEye*'s neural network architecture can be found in the supplementary material.

2.5.2 Input and Output Representation

We use the per-voxel energy in the sensed mmWave heatmap as our input representation in the 3D spherical coordinate system (ϕ, θ, ρ). The output from the GAN is the high-frequency shape of the object. We represent the predicted shape in the form of a 2D depth map in the stereo camera frame, where the GAN predicts the depth for each pixel in the image. The generator learns a mapping from $\mathbb{R}^{64\times32\times96}$ to $\mathbb{R}^{256\times128}$.

Most past works that employ conditional GANs either study 2D to 2D or 3D to 3D transformations. However, our problem requires design of a hybrid 3D to 2D transformation. The sensed mmWave signal is very low resolution, and we do not want to introduce further aliasing by projecting the sensed 3D heatmap to 2D. At the same time, we desire the high-frequency shape as output. It is challenging to predict high-resolution 3D heatmap for computational and optimization reasons [43, 44]. Thus, we chose to represent our outputs as 2D depth maps. Further, note that mmWave signals only provide reflections from metal surfaces in the line-of-sight, since mmWave is shielded by metal surfaces. Thus, the 2D depth map representation of the car serves as a meaningful intermediate representation that can be post-processed to construct full 3D predictions.

2.5.3 Generator Architecture

We follow standard encoder-decoder architecture [45] for representing the generator. The generator is implemented using a deep neural network that maps the input 3D heatmap to a low-dimensional representations z using the encoder. This low-dimensional representation is used by a decoder to produce the 2D depth prediction. We use a 2048 dimensional z-vector. The encoder starts with one channel of 3D input (ϕ, θ, ρ) of size $1 \times 64 \times 32 \times 96$. There

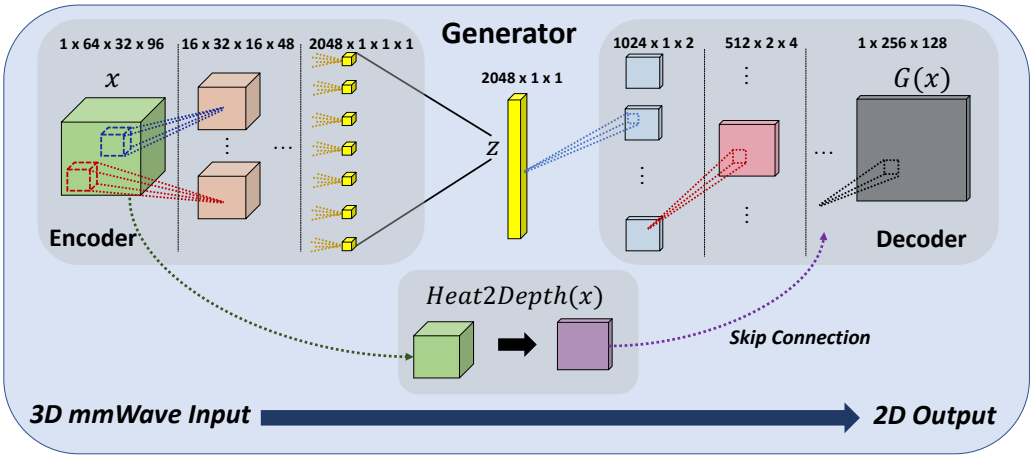


Figure 2.6: Network Architecture of the Generator G

are 6 3D convolution layers with kernel sizes=6 and strides=2 on all three dimensions. With every convolution layer, the number of channels increases, while the 3D feature map size decreases by half in every dimension. We use a BatchNorm layer followed by a Leaky-ReLU layer after every 3D convolution layer. The encoder outputs a $2048 \times 1 \times 1 \times 1$ dimensional z-vector, which is then squeezed to $2048 \times 1 \times 1$. At each layer we appropriately zero-pad the features in order to get the desired input and output sizes.

The decoder begins with the $2048 \times 1 \times 1$ dimensional z-vector, and it contains 8 2D deconvolution layers produce the 2D depth map. The first deconvolution layer has a kernel size=$(4, 3)$, and stride=$(2, 1)$ in order to get an output of size $(2, 1)$. The subsequent deconvolution layers have kernel sizes=4 and strides=2 on both dimensions. Hence, each layer doubles the feature map sizes, and we also decrease the number of channels. We use a BatchNorm layer followed by a ReLU layer after every deconvolution layer. Again, at each layer we appropriately zero-pad the features in order to get the desired input and output sizes. This results in an output feature map of size $1 \times 256 \times 128$.

Finally, we apply the hyperbolic tangent function alongside a linear transformation that maps the final output between 0 and 1, which corresponds to the absolute depth in the scene. This resulting generator output $G(x)$ has a dimension of 256×128.

2.5.4 Skip Connections

We also use skip connections [42] in the generator. Skip connections provide higher layers in the decoder with high-frequency information from the input / early layers of the encoder. While this is simple in networks that map 2D to 2D or 3D to 3D, our network learns a 3D to 2D projection. Thus, our skip connection design projects the input 3D heatmap into a 2D image, which is concatenated with the higher layers of the decoder. This projection is done as follows. We compute the following 2D image by recording the location that corresponds to the highest value along a ray that projects at that location:

$$x_{2D}(\phi, \theta) = \arg\max_{r} \; x_{3D}(\phi, \theta, r). \tag{2.1}$$

Simply choosing the depth corresponding to the largest value is unstable. Thus, we choose the m largest values and create $m = 8$ channels of $2D$ feature maps of size 64×32, ordered from highest to lowest power. These 2D feature maps are concatenated with the features maps at the 6^{th} layer in the decoder along the channel axis, so that the high-resolution depth information from the radar heatmap is directly extracted and passed to the output in order to retain the high-frequency details in depth. The concatenated feature map then goes through the last deconvolution layer and a fully connected layer. Note that, this projection is done in a spherical coordinate frame, while our output is in the camera coordinate frame. As the field of view of the camera is not very large, the two images are still reasonably well aligned. This is a non-differentiable operation and is only done with the input.

2.5.5 Discriminator Architecture

The discriminator takes two inputs: the 3D heatmap x and a 2D depth map that either was the ground-truth y or was generated $G(x)$. It outputs the probability of the input being real. Typically, the input and output to the generator are of the same type (both 2D or both 3D). However, in our case input is 3D and output is 2D. Thus, we adopt a two-stream architecture that uses two separate networks to map x and y to 1D feature vectors, and then fuses them to classify real vs generated samples, as shown in Fig. 2.6. Heatmap x is processed through a 3D CNN with the same architecture as used in the generator but with different weights, resulting in 512 dimensional

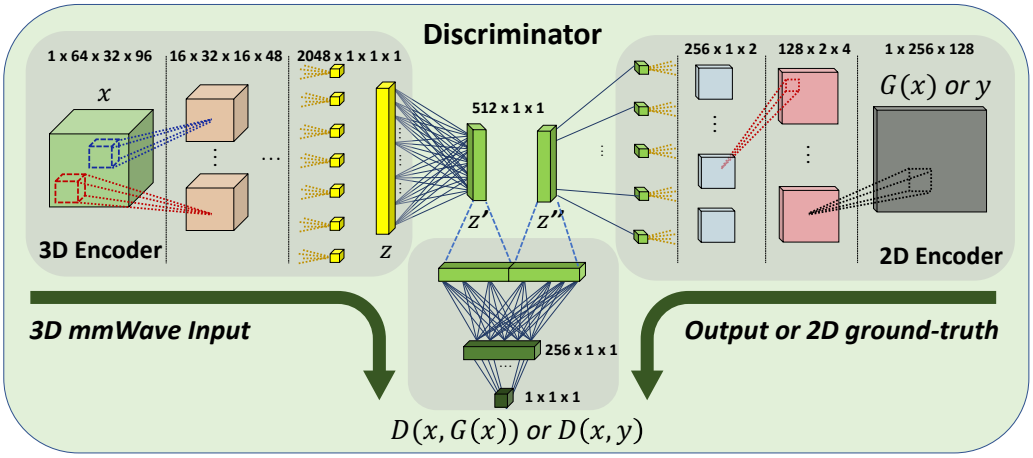


Figure 2.7: Network Architecture of the Discriminator D

output vector z'. The encoder for 2D depth-maps takes the ground-truth y or the generator output $G(x)$ as input and also outputs a 512 dimensional vector z''. It has eight 2D convolution layers with kernel sizes = 4 and strides = 2 on both dimensions with appropriate zero-padding. Each convolution layer is followed by a BatchNorm layer and a Leaky-ReLu layer. z' and z'' are then concatenated and fed into two fully connected layers with ReLu and Dropout layers in between. Finally, we use the sigmoid activation function to get the output probability of the discriminator as the final classification.

2.5.6 Loss Function

The output of the discriminator D and generator G are used to calculate the vanilla GAN loss function $\mathcal{L}(G)$ [36]. As with past work [41, 46], we also include $\mathcal{L}_1$ loss (between the ground truth and the prediction), and a perceptual loss term $\mathcal{L}_p$ [46, 47] (on activations of a pre-trained neural network, VGG [48] in our case, on y and $G(x)$). During training, D and G are optimized to minimize the $\mathcal{L}_H(G)$ loss as below:

$$\mathcal{L}_1(G) = \mathbf{E}\|y - G(x)\|_1 \tag{2.2}$$

$$\mathcal{L}_p(G) = \mathbf{E}\|VGG(y) - VGG(G(x))\|_1 \tag{2.3}$$

$$\mathcal{L}_H(G) = \mathcal{L}(G) + \lambda_1\mathcal{L}_1 + \lambda_p\mathcal{L}_p \tag{2.4}$$

While $\mathcal{L}_1$ losses aren't effective for pixel prediction, our outputs are depth values and thus $\mathcal{L}_1$ makes sense. We use the feature space of a VGG net-

work [48] to compute the perceptual loss. We feed *HawkEye*'s output $G(x)$ and the corresponding ground-truth y into a pretrained VGG16 [48] model, by replicating $G(x)$ and y to three channels. We obtain outputs of the VGG model at the 3^{rd}, 8^{th}, 15^{th}, and 22^{nd} layers for $G(x)$ and y. Then we compute the L1 difference of the outputs at each layer and normalize them to get the perceptual loss $\mathcal{L}_p(G)$. *HawkEye* employs a combination of three losses (Eq. 2.4), λ_1 and λ_p are hand-tuned relative weights of the loss functions. Using this loss function enables *HawkEye* to accurately capture both the low and high frequency components in the image. This results in perceptually interpretable high-resolution images that faithfully represent the scene.

2.6 Data Collection Platform

Since there is no publicly available 3D mmWave imaging radar dataset, we custom-built a data collection platform and collected our own dataset, which includes 3D mmWave radar heatmaps of cars and the corresponding stereo camera depthmaps. We custom-built the mmWave imaging radar system using 60 GHz radios and a SAR (Synthetic Aperture Radar) platform. We send standard FMCW radar waveform generated by our custom-build FMCW circuit.

2.6.1 2D Synthetic Aperture Radar (SAR)

We leverage SAR to emulate a 2D antenna array by mechanically scanning a single mmWave radio. We opt for SAR-based implementation because of the limited availability of 2D phased arrays with hundreds of antennas like [37, 38] in commercial systems and the high flexibility of SAR. SAR provides us with a reconfigurable antenna array for a wide range of frequencies and aperture sizes, which allows us to generate radar heatmaps with different resolutions. We build a 2D SAR platform shown in Fig. 2.8 using three FUYU FSL40 linear sliders [49] with sub-millimeter accuracy. We mount a Pasternack 60 GHz radio front-end [50] on the SAR platform as the receiver, and another radio on the side as the transmitter. We use omni-directional antennas for both the transmitter and receiver to have a maximum field-of-view of 180° in azimuth and 35° in elevation. We also place RF absorbers and shields around

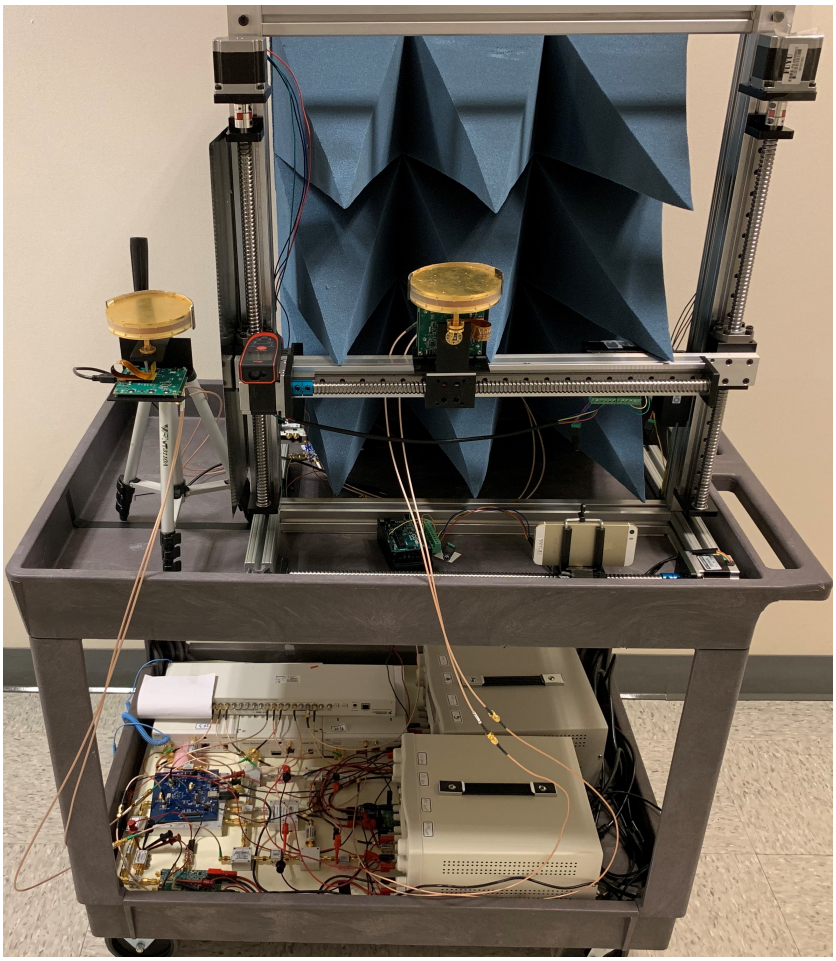


Figure 2.8: *HawkEye*'**s Data Collection Platform**

the antennas to eliminate the direct path leakage and unwanted reflections from the backside. The horizontal slider scans the mounted receiver radio along the X-axis, while two vertical sliders scan along the Z-axis. In *HawkEye*, only a fraction of $10cm \times 10cm$ area is scanned to emulate a 40×40 array at 60 GHz, which provides $\sim 8^{\circ}$ angular resolution along azimuth and elevation axes. The scanning time is 5 minutes and reduces to 90 seconds for a 20×20 array.

2.6.2 Frequency Modulation Continuous Wave (FMCW) Radar Circuit

For the mmWave radar circuit, we implement a heterodyne architecture, as shown in Fig. 2.9. We first generate the same FMCW waveform at baseband

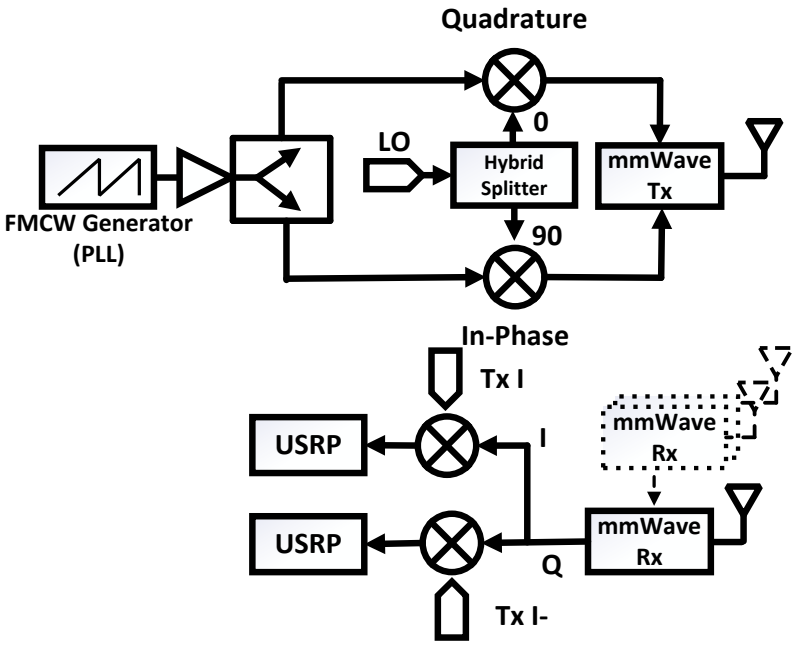


Figure 2.9: *HawkEye*'**s mmWave Imaging Radar Circuit Diagram**

using ADF4159 PLL (Phased Locked Loop) [51], with a bandwidth of 1.5 GHz sweeping from 0.1 GHz to 1.6 GHz. Then we up-convert it to have a center frequency of 60.15 GHz using quadrature modulation. The resulting signal sweeps from 59.4 GHz to 60.9 GHz with the other unwanted sideband suppressed. The FMCW receiver has a reciprocal architecture. The reflected signals at 60 GHz are first down-converted to the baseband through quadrature demodulation to get 90° phase-shifted I and Q channels. Then we feed them separately into RF mixers along with the original baseband FMCW waveforms to extract the beat signal, whose frequency is proportional to the time-of-flight of the radar waveform in the air. We sample the I and Q components of the complex beat signal with two N210 USRP software-defined radios [52] for direct phase measurement. We use a common clock to enforce frequency and phase synchronization in the radar circuit.

We then align the continuously sampled complex beat signal to the antenna positions in the array. In this process, we track the SAR trajectory by leveraging the phase shift in the residual direct path leakage. We then apply Fast Fourier Transform and conventional beamforming in sequence to estimate the reflected signal power from every voxel $x(\phi, \theta, \rho)$ to generate the 3D mmWave radar heatmap.

2.6.3 Stereo Camera Module

To capture the corresponding high resolution 2D depth maps for ground truth, we build a custom wide baseline stereo camera system. We mount an iPhone camera on a linear slider with sub-mm accuracy to capture multiple images of the scene, and apply a standard stereo image processing algorithm [53] to extract 2D depth maps. The linear slider setup is stable, which allows us to calibrate once and apply the same rectification to all experiments. In the ground truth, we filter out pixels that do not belong to the vehicles of interest using labeled object masks generated from Mask R-CNN [54].

2.7 *HawkEye*'s Data Synthesizer

Collecting real-world mmWave data using our custom-built mmWave module is very time-consuming. Hence, training with real data would take a prohibitively long time. To address this, we build a synthesizer to generate paired 3D radar heatmaps and 2D depth maps of cars from 3D CAD models. Our synthesizer is designed to create 3D point reflector models of cars and then simulate mmWave radar signals using ray tracing. It takes into account multipath reflections as well as specularity based on reflection angles to generate realistic mmWave radar 3D heatmaps. Simulation has 3 stages:

1. *Scene Generation*: We first generate scenes of cars based on two types of datasets: 3D CAD models for autonomous driving [55] and Cityscapes [56], a street view video recordings dataset. The 3D CAD models provide us with precise 3D meshes of a wide variety of vehicles, while the street view photos offer references for car placement in the camera frame. We apply Mask R-CNN [54] on the street views to detect objects of interest.

2. *Ray Tracing*: Here we model the mmWave reflectors in the scene. First, we remove occluded bodies through spherical projection. Then, we model the remaining parts as clusters of point reflectors, where the number of points represents the size of the radar cross section. We classify the specularity of each cluster as scattering corners or mostly specular surfaces by referring to the known car outline. Finally, we perform standard ray tracing [57] on the point reflectors with their specularity taken into account.

3. *mmWave Heatmap and Ground-truth Generation*: We simulate the received signal based on the point reflector model with background noise introduced. We add thermal noise and phase noise to the mmWave signals. Additionally, to avoid the nontrivial extrinsic calibration for the field point and point of view between the mmWave and stereo camera modules in our experimental setup, we import the same displacement into our synthesizer to make predictions at the stereo camera view point, and to accurately train and test *HawkEye*'s GAN architecture. Similarly, our model can be re-trained to make predictions from any other viewpoint as well (with appropriately modified skip connections). By applying mmWave processing as described in section 2.2, we get the 3D mmWave heatmap. The ground-truth 2D depth map is generated to match the stereo camera frame.

2.8 Experiments

2.8.1 Dataset

We imaged 327 scenes of cars in 3 types of backgrounds: indoor parking garage, outdoor lot, and outdoor house drive-through. The dataset includes 9 categories of cars spanning 60 different models: 2 Sub-compact, 12 Compact, 16 Mid-sized, 7 Full-sized, 5 Sports, 11 SUVs, 1 Jeep, 2 Vans, and 4 Trucks. We tested all 360° orientations of the car with respect to the radar. The distance from the radar to the car is between 3.3 to 11.9 meters, with a mean of 6.2 m and a standard deviation of 1.66 m. We then created a dataset of paired 3D mmWave heatmaps, RGB camera images, and stereo camera depth maps. In addition to real data, we also have 4000 synthesized scenes generated from *HawkEye*'s data synthesizor for 120 car models.

2.8.2 Controlled Experiments in Fog/Rain

Out of the 327 real scenes we imaged, there are 101 experiments in fog and rain to test *HawkEye*'s performance in poor visibility conditions where today's optical sensors fail. Due to practical limitations such as the risk of water damage to our setup, we conduct controlled experiments where we

emulate real fog and rain. We use a fog machine along with a high-density water-based fog fluid to emulate severe and realistic fog conditions, similar to previous studies [58, 59]. We emulate rain using a water hose in a confined region around the object of interest (the car).

2.8.3 Training

HawkEye's GAN is trained with the Adam optimizer in two stages. In the first stage, we train for 170 epochs using a synthesized dataset of 3000 images with batch size 4. We use a constant learning rate of 10^{-4} for the first 100 epochs, and then linearly decay the learning rate to zero for the next 70 epochs. In the second stage, we fine-tune the model for 60 additional epochs with 100 real mmWave images captured in clear weather. In this stage, we use a learning rate of 10^{-5}. It is important to note that *HawkEye*'s GAN model is never trained on examples collected in fog or rain. The training takes 12 hours on an Nvidia Titan RTX GPU. We test *HawkEye*'s performance on 1000 synthesized images, and the remaining 227 real images including the fog and rain experiments. For testing, we follow standard k-fold cross-validation with $k = 5$ to test all 327 scenes while ensuring examples in the test dataset are not used during training.

2.9 Results

2.9.1 Baselines

We compare *HawkEye* to three baselines:

1. *mmWave Radar:* We compare against raw mmWave radar heatmaps to evaluate *HawkEye*'s improvement over the low resolution and artifact-ridden radar images.

2. L_1 *Based Loss:* To determine the utility of the GAN and discriminator in *HawkEye*, we compare against an identical neural network trained only with the L_1 based loss function, $\mathcal{L} = \mathcal{L}_1 + \lambda_p \mathcal{L}_p$, as defined in Eq. 2.2 and Eq. 2.3.

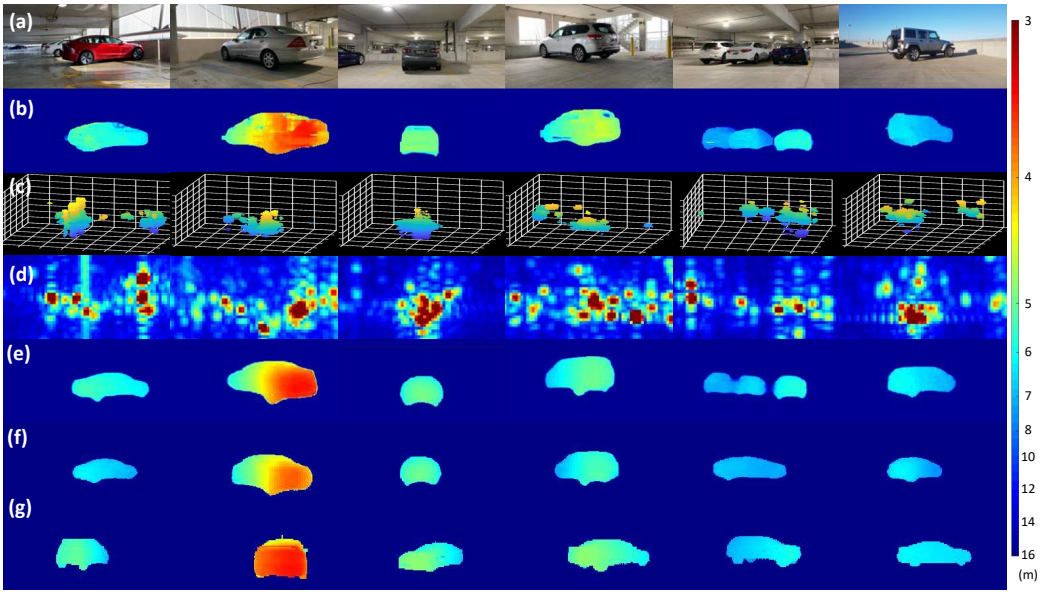


Figure 2.10: ***HawkEye*'s qualitative performance on real test data in clear weather conditions.** Row (a) and (b) show the original clear scene with the car and the corresponding stereo depth map. Row (c) and (d) show the radar heatmap as 3D point cloud and 2D front-view projection. Row (e) shows the output from *HawkEye*, while rows (f) and (g) show the output from L1 and Nearest Neighbor baselines. The scale bar shows the absolute depth metric in the depth map.

3. *Nearest Neighbor:* One could argue that our method overfits and simply memorizes sample points from the training dataset. To understand this, we compare against a Nearest Neighbor scheme, which retrieves samples in the input feature space of 3D radar heatmaps with the minimum Euclidean distance.

2.9.2 Qualitative Results

We show *HawkEye*'s performance in clear weather and fog in Fig. 2.10 and Fig. 2.11 respectively.[4] In both visibility conditions, *HawkEye* accurately reconstructs the shape and size of the car in the scene, and captures key defining features such as its wheels and orientation comparably better than the other baselines. *HawkEye* can also accurately determine the distance to the car in 3D space, as can be seen from the intensity in the depth maps.

[4]We show additional qualitative results for the synthetic test dataset and for experiments in rain in the supplementary material.

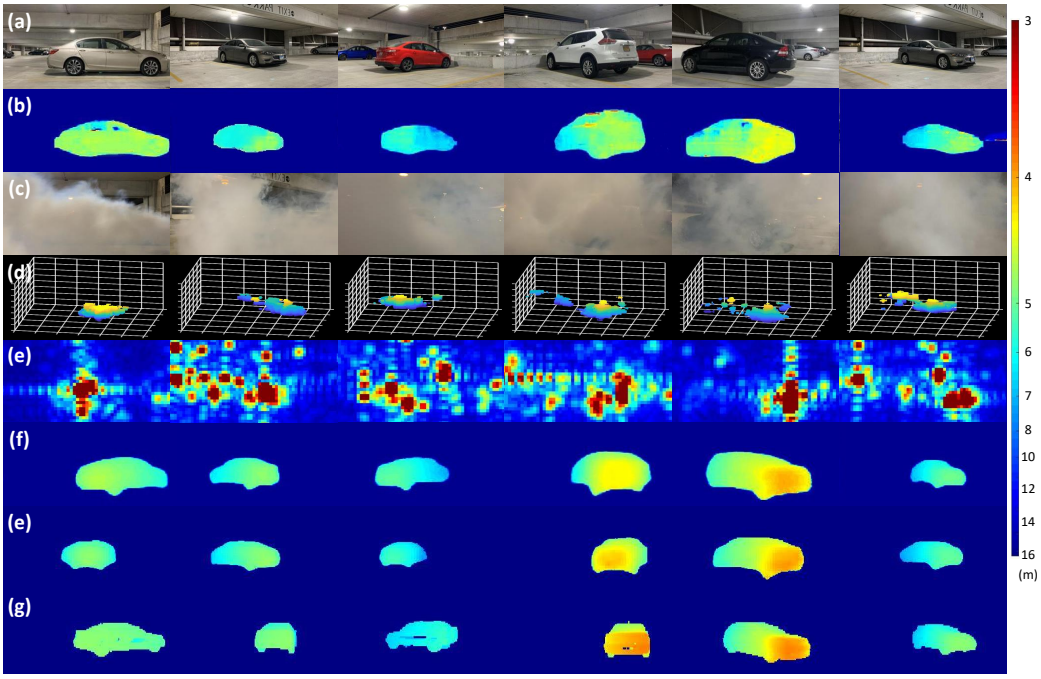


Figure 2.11: *HawkEye*'**s performance with fog in scene.** Row (a) and (b) show the original scene and corresponding stereo depth map. Row (c) shows the scene filled with fog. Row (d) and (e) show the radar heatmap in the fog scene as 3D point cloud and 2D front-view projection. Row (f) shows the output from *HawkEye*, while rows (g) and (h) show the output from L1 and Nearest Neighbor baselines. The scale bar shows the absolute depth metric in the depth map.

HawkEye's ability to accurately image in fog and rain[5], despite not being trained with such examples, demonstrates that our model can generalize well in different weather conditions due to the favorable propagation characteristics of mmWave signals. Further, note that although *HawkEye* is trained primarily on synthesized data, it could generalize well to real scenes with different backgrounds and visibility conditions with only a small amount of fine-tuning. Hence, the simulator faithfully emulates the real mmWave heatmaps.

Failure Examples: Figure 2.12 shows some typical failure cases for *HawkEye*. (i) and (ii) are from the fog experiments. In (i), although *HawkEye* estimates the correct bounding box, it misjudges the front and back of the car. In (ii), although *HawkEye* successfully detects the corner of the car, due to both strong fictitious reflections and specularity in the heatmap, it incorrectly estimates the orientation of the car. Lastly, a current limitation of

[5]See supplementary material for results in rain.

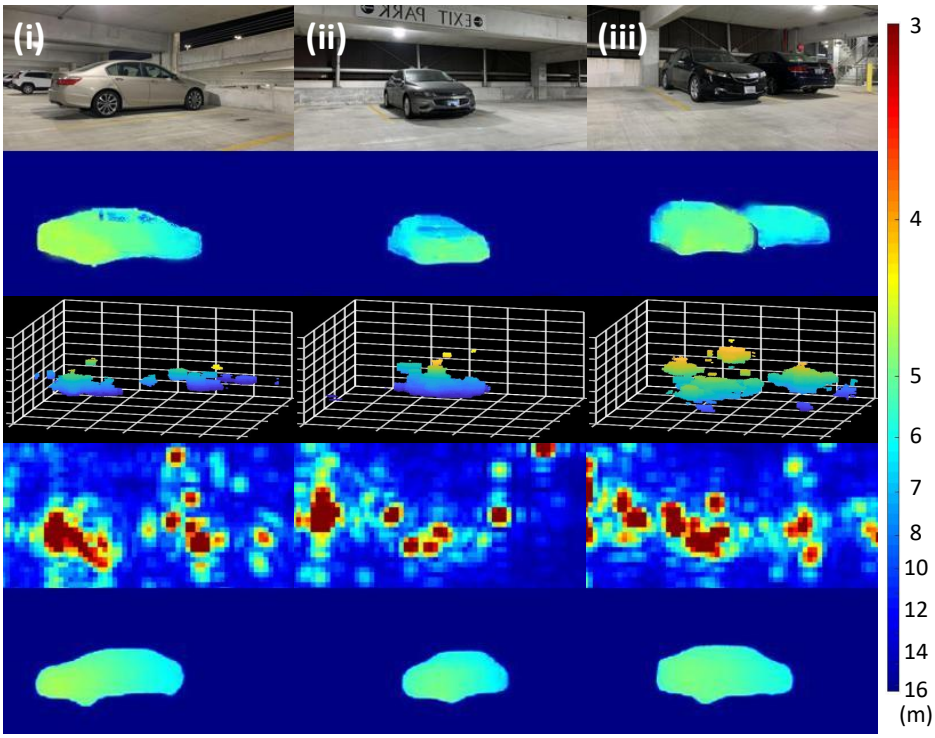


Figure 2.12: Examples where *HawkEye* fails. The first and second rows show the original scene and corresponding stereo depth map. Third and fourth row show the radar heatmap as point cloud and front view projection. The fifth row shows *HawkEye*'s output.

our system is that its performance deteriorates when the scene has multiple cars (Fig. 2.12(iii)). To address this, a potential future direction is to adopt a *Region Proposal Network* [11], where *HawkEye* can first isolate the reflections from the cars in the scene, and then reconstruct each car individually.

2.9.3 Quantitative Metrics

We evaluate on range, size (length, width, height), and orientation of the car, as they represent the contextual information of the car in the scene (shown in Fig. 2.13(i)). We define the distance to the closest corner of the car as the range, and orientation as the angle between the longer edge of the car and the 0° azimuth of the mmWave heatmap. First, we convert depth maps into 3D point clouds in the camera frame based on the mapping from pixel values to metric depth as shown in the scale bars. Then we estimate the bounding boxes of cars by projecting the point clouds onto the horizontal plane and fitting the points into either a 90-degree corner or a straight line.

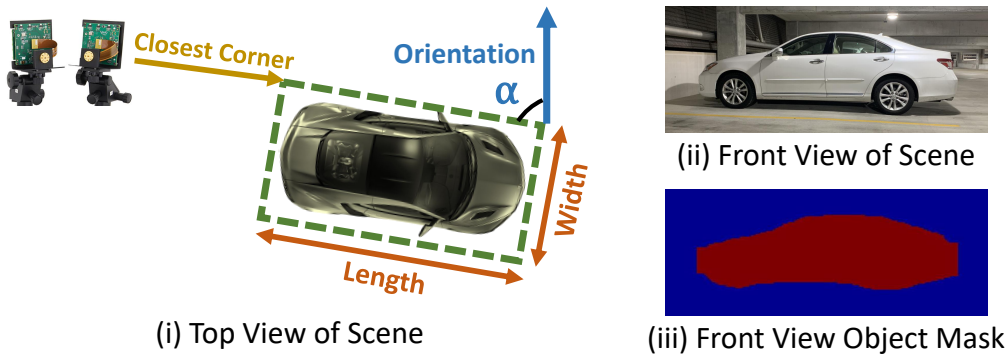


Figure 2.13: Quantitative Metrics used in the Evaluation of *HawkEye*

For radar heatmaps, we manually annotate the bounding boxes in the 2D top-view projections, similar to Fig. 2.2(c). Finally, we fill up the occluded parts of the bounding boxes and extract the metrics. Range, length, width, and orientation of the cars are computable from the corners and edges of the top-view bounding boxes, while heights are estimated from the 3D points inside the bounding boxes.

We also evaluate accuracy in shape prediction by comparing *(a) % of Car's Surface Missed* (False Negatives) and *(b) % of Fictitious Reflections* (False Positives) in the object masks of *HawkEye*'s output along the front view of the scene as shown in Fig. 2.13(ii,iii). Note that (a) is indicative of the specularity effects whereas (b) is indicative of artifacts such as multipath and ambient reflections in the image. We extract the object masks from the mmWave heatmap and the outputs of *HawkEye* and baseline methods separately, and then compute the False Positive Rates (FPR) and False Negative Rates (FNR) against the ground truth object masks. We obtain the ground truth object mask by applying Mack-RCNN on the camera image. For the depth map outputs of *HawkEye* and the baseline methods of L_1 and Nearest Neightbor, we eliminate noise that are far away from the object body using a distance threshold, so that the remaining pixels in the region of the object form the mask. For mmWave heatmaps, we project the 3D heatmaps onto the front-view plane and then select pixels exceeding a power threshold as the object mask. We choose the distance and power thresholds from the ROC (Receiver operating characteristic) curves.

Table 2.1: Quantitative Results. See text for more details.

Experiment	System	Error in Ranging	Error in Length	Error in Width	Error in Height	Error in Angle	Fictitious Reflections	Surface Missed
Clean Air	HawkEye	**30 cm**	**47 cm**	**29 cm**	**9 cm**	**27°**	**1.5%**	**12.9%**
	mmWave	53 cm	179 cm	89 cm	45 cm	64°	15.6%	30.5%
	L_1 Based Loss	40 cm	97 cm	76 cm	13 cm	37°	2.5%	13.1%
	Nearest Neighbor	90 cm	114 cm	70 cm	17 cm	68°	3.5%	16.0%
Fog	HawkEye	**50 cm**	**83 cm**	**44 cm**	**11 cm**	**29°**	**2.5%**	15.4%
	mmWave	67 cm	222 cm	99 cm	53 cm	72°	20.9%	31.9%
	L_1 Based Loss	60 cm	108 cm	80 cm	12 cm	38°	3.5%	**13.8%**
	Nearest Neighbor	121 cm	117 cm	76 cm	18 cm	45°	3.6%	22.3%
Synthetic Data	HawkEye	23 cm	**64 cm**	**37 cm**	**8 cm**	**30°**	**1.3%**	10.2%
	mmWave	29 cm	182 cm	77 cm	31 cm	62°	10.8%	19.2%
	L_1 Based Loss	**20 cm**	113 cm	73 cm	14 cm	47°	3.4%	**9.3%**
	Nearest Neighbor	81 cm	81 cm	57 cm	13 cm	64°	5.2%	17.5%

2.9.4 Quantitative Results

Table 2.1 shows median errors comparing *HawkEye* to the baseline schemes. These results are extracted from 168 scenes in clean air, 59 scenes in fog, and 510 synthesized scenes. We summarize *HawkEye*'s performance compared to each baseline below.

- *mmWave radar: HawkEye* achieves an improvement in ranging accuracy of 1.35× in fog and 2× in clear weather. Although mmWave radars can achieve high ranging resolution, the artifacts in the radar heatmaps lead to high ranging error. The skip connections in *HawkEye*'s design allow for the direct transfer of the high ranging resolution from the mmWave radar input to *HawkEye*'s output, while additionally *HawkEye*'s GAN model corrects for the sinc artifact to achieve lower median ranging error. However, note that *HawkEye*'s gains over mmWave radar become more apparent for the other metrics, spanning from 2× to 12× gain for percentage of fictitious reflectors. This is because the other metrics are a lot more sensitive to the specularity and multipath artifacts, and *HawkEye* can significantly improve these metrics by correcting for these noise sources.

- L_1 *based Loss:* The L_1 loss baseline achieves good performance in terms of ranging error compared to *HawkEye*. This is expected since optimizing for L_1 loss over 2D depth maps would directly optimize for ranging error. However, L_1 loss cannot capture the high frequency components of the output shape, resulting in blurring of boundaries. As a result, the errors in estimated size, orientation and fictitious reflectors are high for L_1 loss,

with *HawkEye* achieving approximately 2× performance gains across these metrics. These results demonstrate the importance of the GAN architecture in *HawkEye*.

- *Nearest Neighbor: HawkEye* outperforms the Nearest Neighbor baseline, achieving an improvement of 1.3× to 3× in clear weather, and 1.4× to 2.4× in fog across various metrics. This demonstrates that our model is not overfitting and can generalize well to new data points in the test set.

For the synthesized dataset, the performance trends are similar. The above results show that *HawkEye* can faithfully reconstruct an accurate and high resolution image of the car in the scene in both clear weather and in low visibility conditions. One should note that *HawkEye*'s performance in fog degrades slightly compared to clear weather. This can be attributed to the poor propagation characteristics of 60 GHz RF signals in the presence of water particles in fog. It is worth noting that due to FCC regulations, we are constrained to build our experimental setup at the 60 GHz unlicensed spectrum, which suffers from higher attenuation from water particles compared to other frequencies in the mmWave band. We believe that implementing *HawkEye* with commercial grade mmWave radars built at the 77 GHz frequency band, which is allocated specifically for automotive radar applications, would resolve the performance degradation observed here.

2.10 Related Work

2.10.1 Super-Resolution

Neural networks have been used to increase the resolution of camera images and near-Infrared images [60, 61, 46, 62]. Such techniques rely on the correspondence of image patches between low and high resolution images and can achieve an upscaling factor of 4×. The closest to our work are techniques for upsamplng sparse 3D LiDAR data to create dense 2D depth maps [63, 64, 65, 66, 67, 68]. However, these works require an RGB camera in addition to LiDAR [63, 64, 65, 66] and, hence, do not work in low visibility conditions, or rely on high frequency visual features like edges to cluster and upsample objects [67, 68]. Millimeter wave images, however, have sig-

nificantly lower spatial resolution where high frequency visual features like boundaries and edges are not apparent. Millimeter wave also suffers from artifacts and specularities that cannot be addressed with traditional super-resolution and upsampling techniques.

2.10.2 LiDAR in Fog

Recent work aims to improve the performance of LiDAR in fog [21, 22, 23]. However, even state-of-the-art research systems either require knowing a depth map of the scene a priori [21] or work only when the object is static by estimating the statistical distribution of the photon reflected off the object [22, 23]. These systems also work only up to 54 cm and have limited resolution (32×32 pixels) and field of view. Millimeter wave radar, on the other hand, can penetrate through dense fog and does not require the object to be static [58, 69].

2.10.3 Radar Imaging Systems

There exist mmWave radar imaging systems that can achieve high resolution [35, 70, 71, 72, 34]. However, these systems can only work at very near distances ($< 50cm$) and use very bulky human-sized arrays similar to airport security scanners [35]. Other radar systems that can achieve high resolution at longer distances, are integrated with optical components like a large focusing lens and a mechanically scanning raster [73, 72, 74]. Hence, they are bulky and perform poorly on mobile platforms like self driving cars [70].

Past works also leverage deep learning in the context of mmWave radar data. Danzer et al. [75] extend the PointNet architecture from PointNet [76] to perform 2D object detection from radar data but cannot perform high resolution depth imaging. Fang et al. [77] and Armanious et al. [78] apply neural networks to radar acquisitions to enhance their resolution. Both Fang et al. [77] and Armanious et al. [78], however, work only at short distances and use radar data both as input and ground-truth to their system, making them inherently incapable of dealing with challenges like specularity and multipath. *HawkEye*, on the other hand, achieves much better results by training using high resolution depth maps to recover the visual representation of the cars

and learn to cope with specularity and multipath.

Recent work showed significant progress in using low frequency wireless radar (below 6 GHz) to estimate the 3D pose of humans and track them through walls and occlusions [9, 11, 10, 12]. The work leverages human motion to combat specularity by combining reflections from different body parts over time and stitching them to form the full human body. The work also uses deep convolutional neural networks to label limbs and joints and map them to 3D models of the human skeleton. However, unlike humans in indoor settings, cars move as one single rigid body and only a single viewpoint of the car is typically observed in practice. Therefore, even during motion, most portions of the car will remain invisible due to specularity. Our system adopts a conditional GAN [40] architecture that is able to address specularity without relying on the object's mobility and, hence, can also image static objects like parked cars and cars stopped at traffic lights.

2.11 Conclusion

In this chapter, we show that *HawkEye* is a promising approach for achieving high resolution imaging with mmWave wireless systems, through the novel design of neural network architectures for processing mmWave data. We evaluate *HawkEye* in low visibility conditions such as heavy fog and show that it can significantly improve performance over mmWave radars today. While significant future work is required before *HawkEye* becomes a practical system that can be used on board self-driving cars, we have made huge advances toward this goal.

Chapter 3

Accurate Detection Using Multi-Resolution Cascaded MIMO Radar

Millimeter wave (mmWave) radars are becoming a more popular sensing modality in self-driving cars due to their favorable characteristics in adverse weather. Yet, they currently lack sufficient spatial resolution for semantic scene understanding. In this paper, we present *Radatron*, a system capable of accurate object detection using mmWave radar as a stand-alone sensor. To enable *Radatron*, we introduce a first-of-its-kind, high-resolution automotive radar dataset collected with a cascaded Multiple Input Multiple Output (MIMO) radar. Our radar achieves 5 cm range resolution and 1.2° angular resolution, 10× finer than other publicly available datasets. We also develop a novel hybrid radar processing and deep learning approach to achieve high vehicle detection accuracy. We train and extensively evaluate *Radatron* to show it achieves 92.6% AP_{50} and 56.3% AP_{75} accuracy in 2D bounding box detection, an 8% and 15.9% improvement over prior art respectively.

3.1 Introduction

Recently, there has been a significant amount of work, from both academia [79, 80, 81, 82] and industry [83, 84, 85, 86], on leveraging millimeter wave (mmWave) radars for imaging and object detection in autonomous vehicles. Millimeter wave radars are relatively cheap and can operate in adverse weather conditions such as fog, smog, snowstorms, and sandstorms where today's sensory modalities like cameras and LiDAR fail [23, 25]. Despite that, today's commercial use of mmWave automotive radars remains limited to unidirectional ranging in tasks like adaptive cruise control and parking assistance. This is mainly due to the fact that radar's angular resolution is extremely low, 100× lower than LiDAR as shown in Fig. 3.1(b, c), making it difficult to use radar for object detection. As a result, prior work aiming

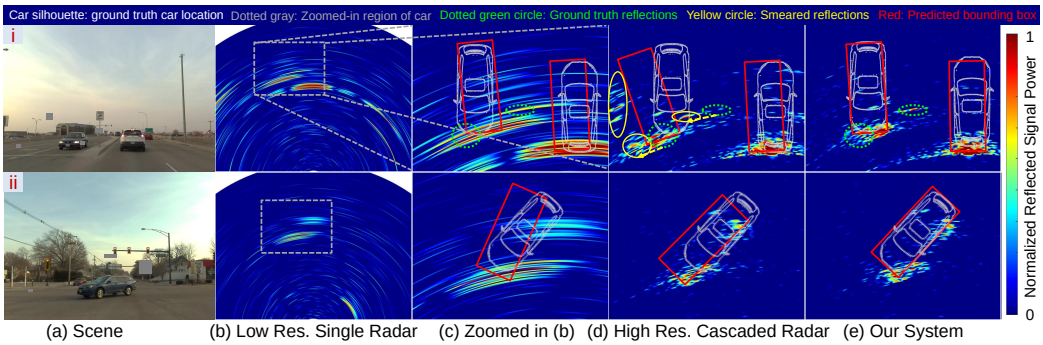


Figure 3.1: The low resolution of millimeter wave radar makes it difficult to perform accurate bounding box detection in (c). High resolution cascaded MIMO radars can improve the resolution but suffer from motion smearing in (d). *Radatron* delivers accurate detection in (e) by combining motion compensation with a two stream deep learning architecture that takes low and high resolution radar images as input.

to gain semantic understanding directly from low resolution radar heatmaps is only able to coarsely localize objects [87, 88, 89] or must fuse radar with LiDAR or cameras to enable object detection [90, 86]. In this chapter, we focus on exploring how well radar performs in object detection tasks and devise techniques to improve its performance.

Improving the angular resolution of conventional radar sensors is challenging. This is because in principle, radar's angular resolution is inversely proportional to the size of the radar antenna aperture [89]. For example, in order to achieve 0.1° angular resolution similar to LiDAR [91], we require a 10 meter-long aperture consisting of an array of 3000 antennas. The cost, power, and large form factor make such a design prohibitively expensive. An alternative cheaper solution is to use a cascaded MIMO (Multiple Input Multiple Output) radar in which multiple radars are combined to emulate a much larger radar aperture [92, 93]. The radars take turns transmitting to avoid interference between the transmitters. Signals from multiple transmitters and receivers are then combined coherently to generate a high resolution image as shown in Fig. 3.1(d) (for primer on radar, see section 3.3). This design, however, cannot work well for dynamic scenes like self-driving cars where the different radar transmitters capture snapshots of the scene at slight timing offsets. In vision, such a problem leads to motion blur which can be addressed using a higher frame rate or deblurring techniques [94, 95]. Radar, on the other hand, uses mmWave RF signals that travel as sine/cosine waves

with millimeter scale wavelength. As a result, even a slight motion of few millimeters can completely change the sign of signal across transmitters which can destructively combine to smear, defocus and even eliminate the object especially as the number of radar transmitters increases. Figure 3.1(d.i) shows this effect: reflections in the moving scene get smeared and appear in different locations than where they really are, which leads to inaccurate bounding boxes prediction.

In this chapter, we present *Radatron*, a mmWave radar-based object detection system that can detect precise bounding boxes of vehicles using a cascaded MIMO radar. *Radatron* overcomes the above challenge by combining a novel radar data pre-processing method with a new deep learning framework. First, we show how to compensate for motion induced errors in pre-processing the raw radar data from a large cascaded MIMO radar. This alleviates most errors, as can be seen by comparing the smeared versions in Fig. 3.1(d) with ones after pre-processing in Fig. 3.1(e). The remaining errors stem from scenarios where the relative speed of the cars is high (e.g. incoming cars, see section 3.6). To address these cases, we design a two stream neural network that takes as input both high and low resolution versions of the radar image. Since the low resolution image uses a single radar transmitter, it does not suffer from motion induced errors which allows the network to correct for faulty information like smeared or missed cars that might be mistaken as noise and artifacts.

The chapter also introduces a first-of-its-kind high resolution radar data set collected using a commercial cascaded MIMO radar in urban streets. The data set features radar heatmaps with 10x higher angular resolution than those used in prior work [88, 89, 96], resulting in rich geometric information of objects in the scene, i.e. boundaries and sizes. The data set also includes stereo-camera images which are used for extracting the ground truth and annotating the data. The data set includes 152k frames representing 4.2 hours of driving over 12 days. We also leverage data augmentation to generate significantly more data especially for less common cases (e.g. oriented cars).

We train and extensively evaluate *Radatron* using our self-collected dataset. Our results show that *Radatron* improves overall detection accuracy by 8% for AP_{50} and 15.9% for AP_{75} compared to low resolution radars used in prior work [88, 96, 89]. For hard cases like oriented and incoming cars, *Radatron* improves overall detection accuracy by upto 14.8% for AP_{50} and 33.1% for

AP_{75} compared to low resolution radars, and by upto 13.8% for AP_{50} and 25.2% for AP_{75} compared to a cascaded MIMO Radar without *Radatron*'s pre-processing and two stream network. We also conducted controlled experiments to qualitatively evaluate *Radatron*'s performance in fog.

Finally, this chapter makes the following contributions. First, we demonstrate the ability of achieving accurate vehicle detection using radar by leveraging the high resolution heatmaps captured by cascaded MIMO radars. Second, we propose a network architecture leveraging multi-resolution radar data along with a motion compensation pre-processing algorithm. Third, we collect a high resolution automotive radar dataset with real-world driving scenarios on urban streets using cascaded MIMO radar, which we plan to release once the chapter is accepted.

3.2 Related Work

3.2.1 Radar-based Datasets

Several radar datasets have recently been introduced using single TI chips [85, 97, 98, 80, 99], the Navtech CTS350-X radar device [82, 87, 81], or other low resolution and 1D radar device [84, 100]. Unlike these datasets, *Radatron* uses the cascaded MIMO TI radar which provides an angular resolution of 1.18° in azimuth, 18° in elevation and a range resolution of 5 cm enabling accurate object detection. Additional details of our dataset can be found in section 3.7. We summarize and compare our data set to other publicly available datasets in Table 3.1. [82, 81] are the closest in terms of resolution but use a mechanically rotating horn antenna which results in a low frame rate of 4 Hz, motion smearing that cannot be corrected in pre-processing, and inability to compute velocity from Doppler information in the radar signals.

3.2.2 Learning with Radar Data

Low-cost radar has been used with deep learning in applications such as hand-gesture recognition [101], imaging and tracking of the human body [10, 11, 12, 102], as well as indoor mapping [103]. Our work focuses on using radar for autonomous driving where prior work can be divided into two groups:

Table 3.1: Publicly available radar datasets. We only include publicly available data sets with more than 500 frames that provide 2D and 3D radar heatmaps. Hence, data sets like [86, 79, 83, 87, 104] are not included. N/A: Not Applicable. N/R: Not Reported.

Dataset	Dim.	Resolution			#Total Frames	#Labeled Frames	Frame Rate	Size	Ground Truth	Radar
		Azi.	Ele.	Range						
Nuscenes [105]	1D/2D	N/R	N/A	N/R	1.3 M	40 K	13 fps	5.5 hrs	LiDAR	N/R
CARRADA [85]	2D[1]	15°	N/A	20cm	12.7 K	7.2 K	10 fps	21 mins	Camera	AWR1642
CRUW [80]	2D[1]	15°	N/A	23cm	400 K	N/R[4]	30 fps	3.5 hrs	Camera	AWR1843
OXFORD [81]	2D	1.8°	N/A	17cm	240 K	0	4 fps	280 km^3	N/A	CTS350-X
RADIATE [82]	2D	1.8°	N/A	17cm	200 K	44 K	4 fps	3 hrs	Camera	CTS350-X
Zendar [84]	2D	30°	N/A	18cm	400 K	11 K	10 fps	11 hrs	LiDAR	N/R
SCORP [98]	3D	15°	30°	12cm	4 K	4 K	10 fps	6.6 mins	Camera	AWR1843
RADDet [99]	3D	15°	30°	20cm	10 K	10 K	N/R	Static[2]	Camera	AWR1843
Radatron	3D	1.2°	18°	5cm	152 K	16 K	10 fps	4.22 hrs	Camera	MMWCAS

1. Radar Point Clouds: Point clouds are a common interface of commercial automotive radars. Therefore, learning radar data in the format of point clouds is widely studied [106, 107, 75, 96]. Schumann et al. [106, 107] demonstrate a semantic segmentation network on radar point clouds while Danzer et al. [75] adjust PointNet [76] for radar data to perform 2D object detection. Pointillism [96] performs 3D bounding box by combining point clouds from multiple spatially separated radars. However, to get point clouds, filtering and thresholding are performed to remove sensor leakage, background clutter, and noise. These hard-coded filtering algorithms lead to the loss of useful information and result in point clouds that are 10 to 100 times sparser than LiDARs [108].

2. Radar Heatmaps: To avoid loss of information, radar data can be processed as heatmaps with range-angle-Doppler tensors [89, 109, 83, 108, 99]. In order to learn the 3D radar tensors, past methods collapse the 3D radar tensor onto each dimension separately to extract features, and then concatenate the resulting multi-view feature maps for semantic segmentation [109], object classification and center point detection [89], as well as 2D bounding box detection [83]. Other work feeds the 2D BEV range-angle heatmap into the network as an image [87]. Note that while Major et al. [83] and Dong et al. [87] achieve relatively accurate 2D bounding box detection results, their datasets were collected on highways and are not publicly available. Compared to highway driving scenarios, where cars are all moving in the same direction and with similar speeds, our dataset is on urban and suburban streets with more complicated traffic intersections, parked cars on the curbside, and various clutters. Zhang et al. [99] provides urban street radar data, but it places the radar on the side of the street for infrastructure-based traffic monitoring. In addition to CNN-based networks, Meyer et al. [108] use graph neural network to achieve a 69% AP_{50} but their data and code are not available.

[1]The radar in [85, 80] can provide 3D data with 30° resolution in elevation. However, the data sets provided are 2D.

[2]The radar is mounted on the side of the road rather than on a moving car.

[3]Driving for 280 km which can correspond to 3 to 10 hrs.

[4]Report 260 K objects but only the center is annotated, not the bounding box.

3.2.3 Radar-optical sensor fusion

Complementary features of multi-sensor data along with the added redundancy has encouraged previous work to combine different sensors. In particular, Radar and LiDAR fusion has been studied [110, 90, 111], while radar and monocular camera fusion has also been studied [112, 113, 114, 115, 116, 117]. In this work, we focus on radar as a stand-alone sensor and aim to show the capabilities of high resolution radar in detecting objects with high accuracy, even in urban and dynamic scenarios.

3.3 Background

3.3.1 mmWave MIMO Radar

Millimeter wave radars transmit Frequency Modulated Continuous Wave (FMCW) signals to sense the environment. The chirps emitted from the transmitter antenna (TX) reflect off objects in the scene which are then captured by the receiver antenna (RX). By comparing the transmitted and received chirp, we can estimate the round-trip Time-of-Flight (ToF) τ, and hence the ranges of the reflectors $\rho = \tau c/2$ (c denotes the speed of light) in the scene. Ranging alone, however, is not sufficient to localize objects. One step further is to use a radar with multiple RX antennas that all receive the reflected chirp. The minute ToF differences $\Delta\tau_{ij} = \tau_i - \tau_j$ between these received version leads to different phase shifts of the electromagnetic waves:

$$\Delta\theta_{ij} = 2\pi f_0 \Delta\tau_{ij} = 2\pi f_0(\tau_j - \tau_i) = 2\pi \frac{l \sin(\phi)}{\lambda}(j - i), \tag{3.1}$$

where l is the spacing between adjacent elements, λ is the radar signal wavelength, and ϕ is the angle from which the reflections arrive, also known as the Angle-of-Arrival (AoA). Therefore, one can exploit the phase shift differences across the antenna array $\Delta\theta_{ij}$ to estimate ϕ [118]. The pair (ρ, ϕ) creates a radar heatmap that localizes objects in the 2D polar coordinate.

For this technique to be viable for applications such as semantic scene understanding and object detection, we need to consider the resolution of the radar, which is closely tied to hardware configuration: the range resolution

is proportional to the bandwidth of the FMCW chirp, while the angular resolution is proportional to the number of RX antennas. Thanks to the high bandwidth in the mmWave band, mmWave radars achieve cm-level ranging resolution, which is sufficient for most applications. However, reaching an acceptable angular resolution is much more difficult. For instance, to achieve the same angular resolution as a commercial LiDAR, we would need to build a radar with hundreds of RX antennas, which is simply impractical due to the hardware complexity, cost, and power consumption.

A much more scalable solution is to use multiple TX as well as multiple RX antennas, a technique referred to as MIMO radar. In MIMO, each of the N TX antennas take turns to transmit one FMCW chirp, which is then received by all M RX antennas, thereby emulating N×M total *virtual* antennas, while using only N+M *physical* antennas [92]. The received chirps from all N·M virtual antennas are then combined to create the (ρ, ϕ) heatmap of the scene.

3.3.2 Motion-Induced Distortion

While MIMO enables higher angular resolution, it comes at the cost of unique challenges. To understand these challenges, we reiterate that in MIMO, TX antennas each transmit one chirp, and all these chirps jointly contribute to the radar heatmap. As TX antennas need to take turns transmitting, there will be a slight time offset δt_{ij} between when the i^{th} and j^{th} chirp are transmitted. For stationary scenes ($v \approx 0$), such time offsets are harmless since they will not affect the ToF difference $\Delta\tau_{ij}$ and phase difference $\Delta\theta_{ij}$ between different virtual antennas.

However, if the scene moves even by as much as 1 mm ($\sim \frac{\lambda}{4}$ at 77 GHz) during the transmitting interval δt_{ij}, the phase different can be significantly off because of $f_0 = 77$ GHz. As a result, the angle estimation and overall radar heatmap can be significantly distorted, especially in sensing highly dynamic environment like self-driving cars. This is because the movement of reflections within δt_{ij} contaminates the phase differences $\Delta\theta_{ij}$ between different virtual antennas, making Eq. 3.1 as follows:

$$\Delta\theta'_{ij} = 2\pi\frac{lsin(\phi)}{\lambda}(j-i) + 2\pi f_0 \delta t_{ij}\frac{2v}{c} \tag{3.2}$$

where v is the relative speed of the object in the scene, and c is the speed of

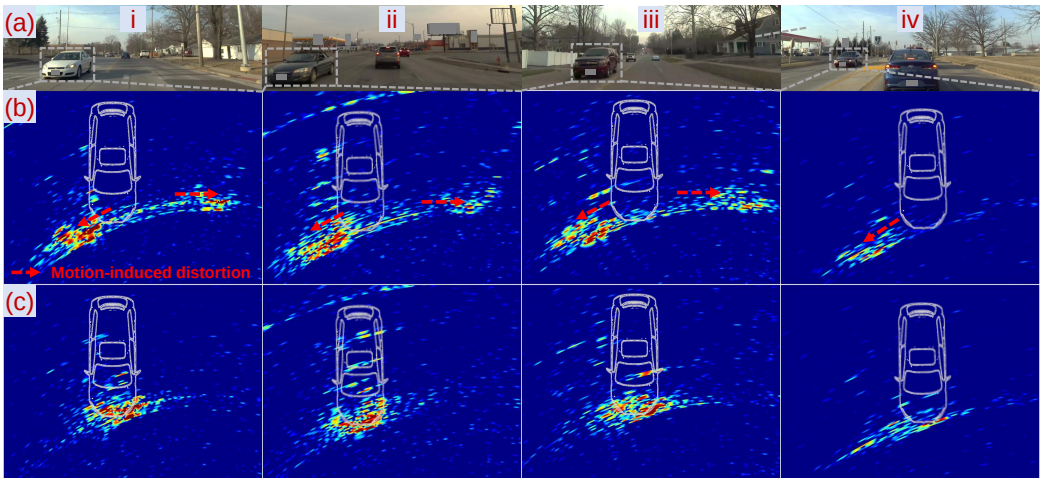


Figure 3.2: Motion-induced distortion and *Radatron*'s compensation algorithm. (a) Original scene. (b) Bird's-eye view radar heatmap under motion-induced distortion. (c) Processed heatmap after applying *Radatron*'s motion compensation algorithm.

light. Note that the motion induced phase $2\pi f_0 \delta t_{ij} \frac{2v}{c}$ cannot be isolated from the AoA dependent phase difference $2\pi \frac{lsin(\phi)}{\lambda}(j-i)$. Therefore, object reflections can get smeared in the radar heatmap, moved into another location, or split into multiple less prominent reflections at different angles. We note that the effect of the error term increases with the speed of the object v, making the problem even more severe for high speed objects. We call this effect the *motion-induced distortion* of the MIMO radar. Figure 3.2(b) shows the impact of *motion-induced distortion* in selected range-azimuth radar heatmaps where there is a car moving towards the radar, and we zoom into the region of the incoming car. As one can see, reflections of the car got smeared along ϕ axis, and even split into multiple less prominent reflections appearing at wrong locations away from the car.

3.4 Overview

Our goal is to design a system that can leverage the high resolution cascaded radar as a stand-alone sensor and perform accurate object detection. While the radar heatmaps created using cascaded radar benefit from high angular and range resolution, they come with a set of unique challenges as laid out in section 3.1 and 3.3. On the one hand, if we cascade multiple TX antennas

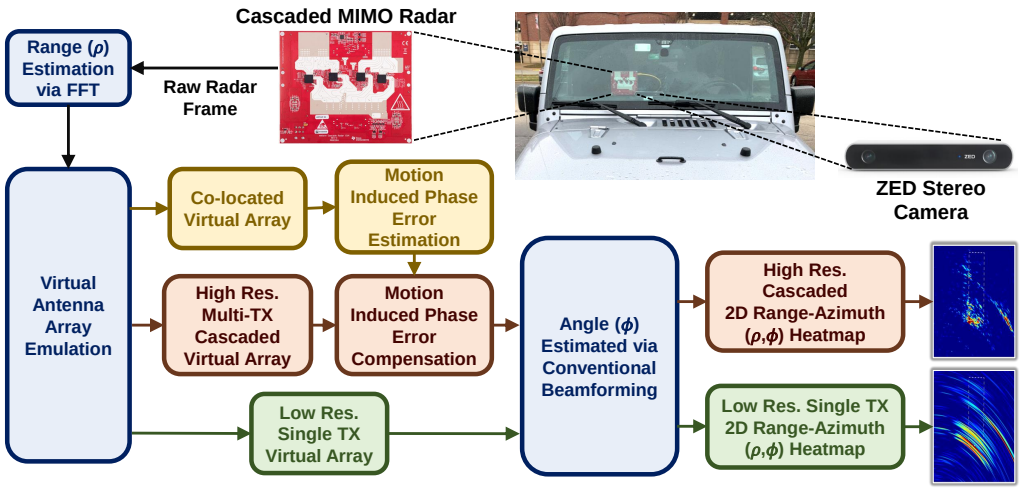


Figure 3.3: *Radatron*'s **Data Pre-Processing Pipeline**

to emulate a virtual array with more antenna elements, we can maximize the angular resolution and minimize leakages due to sparsity in the antenna array. However, the transmit time offsets between different TX antennas can cause *motion-induced distortion* (section 3.3.1), and the resulting radar heatmap will be smeared. This issue is particularly severe for automotive radars since both the radar and the scene are moving at high speeds. *Radatron* overcomes this challenge via a hybrid signal processing and deep learning approach. We will start by explaining our radar processing solution and then proceed to describe our network design to tackle this problem.

3.5 Radar Signal Processing

On the signal processing end, we design a *motion compensation* algorithm and integrate it into our radar processing pipeline as shown in Figure 3.3. It takes the raw radar signal samples as input, and first applies a standard fast Fourier transform to the time-domain signal, which estimates the reflected power from different ranges. Then, before estimating the angles of reflections to localize the objects, we first compensate for the motion-induced distortion.

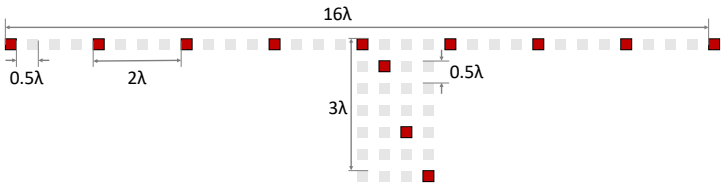


Figure 3.4: Physical TX antenna Array of *Radatron*'s Cascaded Radar

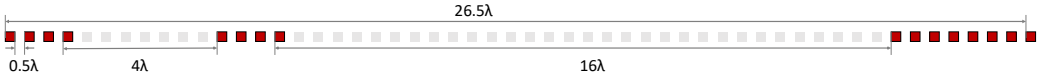


Figure 3.5: Physical RX Antenna Array of *Radatron*'s Cascaded Radar

3.5.1 Virtual Antenna Array Emulation

We collect our own mmWave radar data featuring high angular resolution using TI MMWCAS mmWave cascaded MIMO radar [92]. By cascading four radar system on chips (SoCs), we form a 12 TX and 16 RX MIMO radar system, which can emulate a very large antenna array with up to $16 \times 12 = 192$ elements.

The emulation of virtual antenna array is a process of converting the multistatic TX and RX antennas to a monostatic virtual antenna array, where the effective phase center theory is commonly employed [119]: Under the far field assumption, a pair of TX and RX antennas can be approximated by a TX/RX collocated phase center at their midpoint. This equivalent can also be explained using the space convolution of the TX and RX antenna array [120].

Figure 3.4 shows the physical positions of the 12 TX antennas, while Fig 3.5 shows the physical positions of the 16 RX antennas. Note that, out of the 12 TX antennas, there are nine TX antennas in the same row (height), whereas the other three antennas located on different rows (heights). These three TX antennas can be used to estimate the elevation angle of the reflections. Although we provide data from these three TX antennas in *Radatron*'s dataset, we do not use them to generate the 2D range-azimuth input heatmap to *Radatron*'s network. According to the effective phase center theory, we use the other 9 TX antennas along with all 16 RX antennas to emulate an 86×1 uniform 1D virtual antenna array as shown in Fig 3.6.

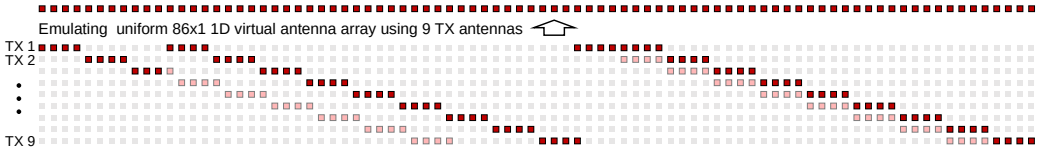


Figure 3.6: Emulating large 1D virtual antenna array using *Radatron*'s cascaded radar. Each row shows the horizontal locations of virtual antennas emulated using one of the TX antennas. Virtual antenna elements used to emulate large 1D array are marked in red, whereas the unused ones due to overlapping are marked in pink.

3.5.2 Motion-Induced Distortion Compensation Algorithm

We design a *motion compensation* algorithm as the first step to mitigate the motion induced distortion problem. To do so, we leverage the fact that the emulated virtual antenna array has some *redundancies*; that is, there are 32 pairs of co-located virtual antennas in the 192 emulated virtual antennas, that are emulated using adjacent physical TX. Therefore, the time interval between each co-located virtual antenna pair i and i' is one chirp interval ΔT. Besides, since virtual antennas i and i' are co-located, there will be no AoA dependent phase differences, and Eq. 3.2 becomes:

$$\Delta\theta^{\dagger}_{ii'} = 2\pi f_0 \delta t_{ii'} \frac{2v}{c}. \tag{3.3}$$

Since the only phase difference between these two co-located virtual antennas is the motion-induced phase variance, we can estimate the motion-induced phase variance by measuring $\Delta\theta^{\dagger}_{ii'}$. Therefore, in our radar signal pre-processing pipeline, in addition to the two virtual antenna array formulations, we also group together the 32 pairs of co-located virtual antennas, as shown in Fig. 3.7. We measure the phase differences between each co-located antenna pairs for each range bin and take an median between the 32 measurements as our final motion-induced phase variance estimation. We then scale the estimated motion-induced phase variance according to the transmitting interval δt for all TX antennas. Finally, we compensate for the motion-induced phase variances for all virtual antennas by multiplying with phasors with opposite phases.

Figure 3.2(c) shows the intermediate motion compensation results, where the smearing artifacts are mostly corrected, and the reflections overlap well with the ground truth location of the car.

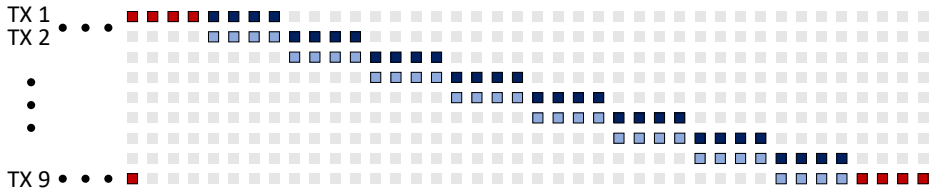


Figure 3.7: Emulated co-located virtual antennas used for motion-induced phase variance estimation. Co-located virtual antenna pairs that are emulated using adjacent TX antennas (time gap equals single chirp interval) are marked in navy and light blue.

Note that, although prior works have also noticed the similar motion-induced distortion problem and tried to compensate for it [89, 121], because of their smaller single chip MIMO radar with only two TX antennas, their motion-induced distortion is much less severe. Their compensation technique using multiple chirps from the same TX antenna also cannot work well for our cascaded MIMO radar due to the 6× longer time gap between when the same TX antenna transmits.

3.5.3 Radar Heatmap Generation

After compensating for the motion-induced phase variances, we then utilize the non-overlapping virtual antennas to extract the angular information of the reflections. We use the Conventional Beamforming algorithm [122] that outputs a 2D range-azimuth (RA) radar heatmap of the scene in the polar coordinates, where the pixel values represent the reflected signal power. We use this radar signal processing pipeline to create two types of inputs for the network:

High resolution cascaded radar: The high-resolution radar heatmap is created using the uniform 86×1 virtual antenna array, emulated with multiple TX antennas. It features the high azimuth resolution achieved using our cascaded MIMO radar.

Low resolution single radar: Instead of using multiple TX antennas, here we only use one TX antenna with all the RX antennas to emulate a non-uniform 16×1 virtual antenna array, so motion compensation is not needed and hence skipped. This processing pipeline approximately reduces the angle resolution by half and introduces leakage artifacts.

3.6 *Radatron*'s Network Design

Although our motion compensation algorithm can alleviate the motion-induced distortions to some extent, it is not perfect. Specifically, the algorithm fails in cases of high speed incoming cars, and there will be residual distortions even after applying the motion compensation algorithm. For example, in Fig. 3.2(c.iv), although after compensation the reflection is centered at the location of the car, it is still smeared across a wider range of angles. To deal with these residual distortions, one potential solution would be to cascade M RX antennas with a single TX antenna. As we use only one TX here, the radar heatmap does not suffer from any motion-induced distortion. However, the virtual antennas in the low resolution version are a sparse subset of the complete N·M virtual array. This results in a heatmap with lower resolution and more leakages, as shown in Fig. 3.3. Using this heatmap alone as a solution is therefore not sufficient.

In order to get the best of both worlds, *Radatron* combines the high resolution with the low resolution solution. Specifically, we leverage the high angular resolution nature of former and the distortion-free nature of latter, by fusing these two versions of radar heatmaps in *Radatron*'s network model. We adapt the Faster R-CNN FPN architecture [123] which has been shown effective previously [90, 108] for radar data. Figure. 3.8 shows *Radatron*'s network architecture. It takes the two versions of radar heatmaps as input into two parallel branches: The first branch uses the low resolution single radar heatmap, which is free of motion-smearing and hence effective in detecting highly dynamic objects such as incoming vehicles; the second branch uses the high resolution cascaded radar heatmap and excels in accurately capturing vehicle outlines. *Radatron* processes these two parallel branches to bring them into a common feature space and then deep-fuses them at an intermediate layer of the backbone network as shown in Fig. 3.8. At the end of the backbone, the feature maps are then converted from the polar to Cartesian coordinates before being fed to the Region Proposal Network and the ROI heads. The output of the network will be 2D vehicle bounding boxes. We will now explain each part of *Radatron*'s network in more detail.

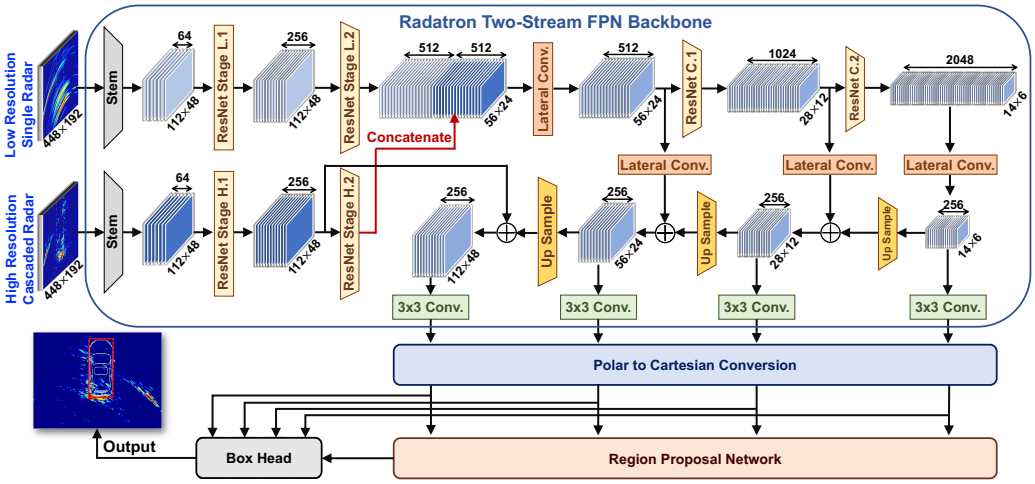


Figure 3.8: *Radatron*'s network architecture. We combine two branches of high resolution and low resolution radar data in an intermediate layer. For each feature map the number of channels and dimensions is indicated above and below it respectively.

3.6.1 *Radatron*'s backbone

For the backbone, we adapt an FPN-based architecture. We process the two input heatmaps to have the same dimension, and feed them into two identical branches. Each of the two branches first goes through a stem layer which consists of a 7×7 Conv. layer, ReLU non-linearity [124] and BatchNorm [125]. Each branch then goes through two ResNet stages, which are the same ones used as the building blocks of ResNet50 [126]. We then combine the two branches by concatenating their feature maps of the same dimension across channels, and fuse them by applying a 3×3 Conv. layer. We further encode the feature maps by passing them through ResNet stages, and combine them to create the feature maps similar to [123].

3.6.2 Coordinate conversion

Compared to the Cartesian coordinate, the polar coordinate is more natural to radar data as radar has uniform resolution across range and angle. It is also easier for a convolutional network to learn radar artifacts like side lobe leakages in the polar coordinates as they appear parallel to the range and angle coordinates, but extend in a circular fashion in the Cartesian coordinates. On the other hand, bounding boxes work naturally with Cartesian

coordinates. We therefore feed in the radar data in the polar coordinates to *Radatron*'s backbone network, and at the end of the backbone explicitly map the features from polar to Cartesian coordinates using bilinear interpolation and before feeding it to the RPN and ROI heads.

3.6.3 RPN and ROI head

As described earlier, the output feature maps of the backbone are converted from polar to Cartesian coordinates before being fed into the network. We adopt the RPN and ROI architecture [123] and add oriented boxes. Implementation details can be found in section 3.8.

3.6.4 Data augmentation

We applied two forms of data augmentations in training:
A. Flipping in Angle. The input heatmap is flipped along the angle axis. In normal driving scenarios, most incoming cars appear on only one side of the ego vehicle, and flipping azimuth angles eliminates such inherent bias in the dataset.
B. Translation in Angle. We translate the input heatmap along the angle axis. This transformation is similar to one demonstrated by Gao et al. [89], with the difference that we perform circular shift in angle; i.e., the angles outside the field of view wrap around and fill in the resulting blank space after translation. As most other vehicles appear straight with respect to the ego vehicle, this helps create more oriented cars.

3.7 *Radatron* Dataset

3.7.1 Data Collection Platform

Our data collection platform consists of a TI-MMWCAS cascaded MIMO radar [92] and a ZED stereo camera [127] as shown in Fig. 3.3. Our radar data features high resolution in both range and angle. Our hardware cascades four TI radar chips, with 3 TX and 4 RX antennas each similar to the ones

used in prior work [88, 96, 89], into a 12 TX and 16 RX MIMO radar system. This cascaded MIMO radar can emulate a large virtual antenna array with up to 192 antenna elements, which provides us with 1.2° azimuth resolution and and 18° elevation resolution. We transmit FMCW radar signals at 77 GHz with 3 GHz bandwidth, yielding a range resolution of 5 cm. We show more details on our radar hardware in the supplementary material.

3.7.2 Data Collection Experiments

We drove with our data collection platform in diverse scenarios including campus roads, our local urban streets, and the downtown area of Chicago, US over 12 days. Each day, we conducted four 20-minute data collection sessions, during which we streamed data with a frame rate of 10 FPS. Then we further refined the data and filtered out empty frames with no objects. Our final dataset consists of 152K frames translating into a duration of 4.2 hrs. Note that although *Radatron*'s network only takes 2D range-azimuth heatmap as the input, the raw radar data in our dataset also contains elevation and Doppler information. For operator safety and numerical evaluation need, our dataset was collected in clear weather, but we expect the results to hold in tough weather, as vast prior works have shown that radar works well in fog, rain, and snow [58, 81, 128]. As a initial verification, we conducted controlled fog experiments to qualitatively evaluate *Radatron*'s performance in fog.

3.7.3 Annotation

We manually annotated 2D bird's-eye view (BEV) bounding boxes on our radar data using stereo camera point clouds and RGB camera images as references. We synchronized the radar and stereo camera frames using their own time stamps after aligning the starting time of both sensors. We also calibrated for the coordinate system offsets between the two sensors by applying a rigid motion transformation on anchor points [129].

Table 3.2: Parameters and experimental configurations of *Radatron*'s mmWave cascaded MIMO radar.

Center Frequency	78.5 GHz	Chirp Duration	34.13 us
Bandwidth	3 GHz	# Chirp Loops	64
Range Resolution	5 cm	Chirp Interval	45.62 us
Chirp Slope	88 GHz/ms	Frame Periodicity	40 ms
ADC Sampling Rate	15 MHz	Velocity Resolution	0.054 m/s
# ADC Samples	512	Max Unambiguous Velocity	$\pm$20.85 m/s
Max Range	25.59 m		
Azimuth Aperture	43λ	Elevation Aperture	3.5λ
Azimuth Resolution	$\sim 1.2^\circ$	Elevation Resolution	$\sim 18^\circ$

3.8 Implementation

3.8.1 MIMO Radar Configuration

We report our cascaded radar parameters as well as its configuration in our data collection experiments in table 3.2.

3.8.2 Training Details

We summarize our training details:

- *Input:* The input dimensions to our network are both 448×192 in the polar (ρ, ϕ) coordinates, with range going from 2m to 22.4m and 5cm resolution, and the azimuth angle in $[0^\circ, 180^\circ]$, with 0.94° resolution. The output after conversion to Cartesian (section 3.6) is of size 256×320, with the x-axis from -16 to 16m and y-axis from 0 to 25.6m, both with 0.1m resolution. We zero-pad the unmatched areas between the two representations.

- *Anchor Boxes:* We choose two anchor sizes of 28 and 35 pixels (geometric mean of dimensions) according to the average sizes of the cars in our dataset and our output grid resolution. We choose the aspect ratio of the anchors to be 2.5 which is typical for most vehicles, and anchor orientation angles of $-90^\circ, \pm 45^\circ$, and 0°.

- *Training Parameters:* We train for 25K iterations with SGD Optimizer. The learning rate starts at 0.01, decays by 0.2 after 15K and again after 20K iterations.

3.9 Evaluation and Results

3.9.1 Evaluation Metrics

We use Average Precision (AP) as our main metric to evaluate *Radatron*'s detection performance, following recent work [90, 108] in radar object detection, using Intersection over Union (IoU) thresholds values of 0.5, and 0.75. We also use the mean AP (mAP) of IOU values from 0.5 to 0.95 with 0.05 steps. We follow the COCO framework [130] to evaluate *Radatron*.

3.9.2 Baselines

We compare with the following baselines:

- *Radar used in prior work:* We implement a virtual array equivalent to the radar used in recent radar datasets [85, 97, 80, 98, 82, 131, 89].
- *Stand-alone single radar TX:* We trim *Radatron*'s network to parse one TX antenna only, which is equivalent to having stand-alone top stream in Fig. 3.8.
- *Stand-alone cascaded radar:* We process the Cascaded radar data with high resolution but bypass our motion compensation algorithm, and feed it into stand-alone bottom stream in Fig. 3.8.

3.9.3 *Radatron* Variants

We also implement three different variants of *Radatron*:

- *Radatron (No Compensation):* We remove the motion compensation algorithm (3.5) from the signal processing pipeline.
- *Radatron (High-res Only):* We remove the *top branch* from Fig. 3.8 and only feed in the high-resolution processed radar data through the bottom branch.
- *Radatron(Multi-res):* We perform the motion compensation algorithm and use both branches with high- and low-resolution processed radar data in Fig. 3.8.

3.9.4 Dataset Split

Out of 152K overall frames, we manually annotate 16K frames following section 3.7. We split the dataset into train and test sets by a 3 to 1 ratio. The set of days from which train and test frames were chosen were disjoint.

In addition, for the test test, we further split the vehicles of interest into the following three different categories, as these scenarios introduce different challenges to *Radatron* as we have discussed in section 3.5 and section 3.6.

1. *Straight*: Any vehicle on the same lane with an orientation within $\pm 5°$.
2. *Oriented*: Any vehicle whose orientation is out of the $\pm 5°$ range.
3. *Incoming*: Any vehicle on the opposite lane, moving towards the ego vehicle.

The *straight* vehicles are relatively easy to detect even using low resolution radars. However, for *oriented* vehicles, high resolution radar is required to accurately detect their angle with respect to the ego vehicle. Finally, *incoming* vehicles tend to get missed by the high resolution heatmap due to the motion induced distortions, as explained in section 3.4. Instead, our partial cascade radar will pick up the incoming cars when the high resolution heatmap fails. Our test set includes 2854 straight, 327 oriented, and 512 incoming cars.

3.9.5 Performance Against Baselines

We first compare *Radatron* with the prior work radar baseline which uses radar heatmaps used by previous art. As seen in table 3.3 , *Radatron* outperforms the prior work radar baseline consistently across all evaluation metrics. This proves empirically that the higher angular resolution of our radar data indeed improves the vehicle detection task. We highlight that while their difference in the overall AP_{50} is around 8%, for the harder cases of oriented cars, *Radatron* outperforms the baseline by as much as 14.8% in the AP_{50} metric. The gap in performance becomes even more prominent for AP_{75}, where *Radatron* outperforms the prior work radar baseline by as much as 15.9% overall and 33.1% for oriented cars. The same trend is also seen using the mAP metric. We attribute this performance gap to our motion compensation algorithm, multi-resolution network, and high angular resolution of

Table 3.3: Performance against baselines. Best performing model is boldfaced. Str. stands for straight. Ori. stands for oriented. Inc. stands for incoming.

Eval Metric	Split	Radar in Prior work	Stand-alone single-TX	Stand-alone cascaded	*Radatron* (multi-res)
AP 50	str.	88.6%	92.4%	87.7%	**95.6%**
	ori.	73.9%	77.6%	80.9%	**88.7%**
	inc.	69.4%	74.3%	65.9%	79.7%
	overall	84.6%	88.9%	84.6%	**92.6%**
AP 75	str.	45.0%	50.2%	42.9%	**56.3%**
	ori.	24.0%	31.6%	31.9%	**57.1%**
	inc.	24.6%	33.6%	26.2%	38.2%
	overall	40.4%	46.4%	39.8%	**56.3%**
mAP	str.	47.3%	51.4%	45.5%	**53.8%**
	ori.	34.4%	36.6%	38.1%	**53.1%**
	inc.	31.2%	37.6%	30.9%	41.4%
	overall	44.2%	48.4%	43.2%	**53.8%**

our dataset. For example, as shown in Fig. 3.1, one can visually make out the outline of a vehicle by only looking at the radar heatmaps of *Radatron*, while the prior work radar baseline only roughly localizes the car. This also explains increased performance gaps for the harder cases of oriented cars, and for the higher IoU thresholds.

We next compare *Radatron* with the other two baselines to show the impact of the our compensation algorithm (section 3.5) as well as our fusion network (section 3.6) on *Radatron*'s performance. We state few points. First, in AP_{50}, *Radatron* outperforms the single-TX and cascaded baseline baselines by 3.7% and 8% respectively. For AP_{75}, the margin jumps to 9.9% and 16.5% respectively. This indicates that *Radatron* is better able to capture the harder cases compared to the two baselines. Second, *Radatron* outperforms the single-TX baseline in the oriented cars significantly, by 11.1% and 25.5% in AP_{50} and AP_{75} respectively. This is in line with our expectation from section 3.6, as the low-resolution and high leakage of single-TX makes it difficult to find the vehicle orientation. Finally, for the incoming cars, *Radatron* outperforms the cascaded baseline by large margins of 13.8% and 12% for AP_{50} and AP_{75} respectively. This confirms our hypothesis in section 3.5 and 3.6, as the lack of motion compensation algorithm severely distorts the cascaded baseline, as shown in Fig. 3.2(b).

Table 3.4: Performance of *Radatron*'s variants. Best performing model is boldfaced. Str. stands for straight. Ori. stands for oriented. Inc. stands for incoming.

Eval Metric	Model / Split	*Radatron* (no comp.)	*Radatron* (high-res only)	*Radatron* (multi-res)
AP 50	str.	93.3%	94.7%	**95.6%**
	ori.	84.6%	**90.7%**	88.7%
	inc.	78.9%	73.1%	**79.7%**
	overall	91.1%	92.4%	**92.6%**
AP 75	str.	49.9%	**61.4%**	56.3%
	ori.	40.4%	56.3%	**57.1%**
	inc.	37.3%	34.6%	**38.2%**
	overall	46.9%	**57.1%**	56.3%
mAP	str.	51.3%	**56.6%**	53.8%
	ori.	43.9%	52.3%	**53.1%**
	inc.	40.6%	37.6%	**41.4%**
	overall	49.1%	**53.9%**	53.8%

3.9.6 *Radatron*'s Performance

We now analyze the performance of three different variants of *Radatron* defined earlier in this section. The results are shown in table 3.4. The *multi-resolution* model outperforms the *no compensation* model by 1.5% and 9.4% in AP_{50} and AP_{75} respectively, which means that the multi-res architecture alone without the motion compensation algorithm will not perform well enough, especially for the harder cases, like high-speed incoming cars. On the other hand, the *multi-resolution* model also outperforms *high-resolution only* for incoming cars by 6.6% and 3.6% respectively, which further shows that the motion compensation algorithm alone is not sufficient and can be improved upon using the multi-res network. We note, however, that *multi-resolution*'s performance improvement for the high speed incoming vehicles comes with a slight decrease in performance for oriented cars compared to the *high-resolution only* network. We envision that one could come up with smart combination of high-res and multi-res variants of *Radatron*to improve the results on all metrics.

Table 3.5: Ablation studies on data augmentation and coordinate system conversion. Best performing model is boldfaced.

Eval Metric		AP 50			AP 75		
Ablation	Split	str.	ori.	inc.	str.	ori.	inc.
Cartesian input		91.8%	86.3%	66.5%	49.1%	53.5%	23.8%
Learned conversion		86.5%	55.4%	45.4%	42.7%	9.0%	8.7%
No augmentation		90.6%	77.7%	65.9%	53.2%	29.6%	21.3%
Radatron		**95.6%**	**88.7%**	**79.7%**	**56.3%**	**57.1%**	**38.2%**

3.9.7 Ablation Study - Data Augmentation

To study the impact of the two forms of data augmentations applied (discussed in section 3.4) on *Radatron*'s performance, we remove the data augmentations while keeping the rest of *Radatron*'s pipeline the same. As the results in Table 3.5 show, the augmentations consistently improve the performance across all metrics. The 16.9% AP_{75} improvement over incoming cars confirms our assumption on the horizontal flipping augmentation (section 3.4), while the 27.5% AP_{75} improvement for oriented cars shows affirms that angular shift can help with oriented vehicle predictions.

3.9.8 Ablation Study - Coordinate System

Here we wish to study the impact of different possible choices for input coordinates. To do so, we consider two alternatives to our design. In the first version, *Cartesian input*, we feed in Cartesian coordinates to the network from the beginning by converting the input radar tensors from polar to Cartesian. In the second version, *learned conversion*, we remove the conversion and let the network implicitly learn to convert from the polar input to Cartesian bounding boxes at the output. As the results in Table 3.5 show, *Radatron*'s original coordinate conversion outperforms *Cartesian input* by 3.8% in AP_{50} and 7.2% in AP_{75} for straight cases. A similar trend is seen for oriented and incoming cars. This confirms our hypothesis in section 3.4 that it is easier for the network to learn the radar artifacts and suppress them in polar coordinates compared to Cartesian. *Radatron* also outperforms *learned conversion* by 9.1% in AP_{50} and 13.6% in AP_{75} for straight cars and even larger margins for other cases. Hence, explicit conversion of the coordinates rather than letting the network learn the conversion improves the performance.

Table 3.6: Ablation study on fusion at different stages. Best performing model is boldfaced.

Eval Metric	Model Split	*Radatron* (Early Fusion)	*Radatron* (Late Fusion)	*Radatron*
AP 50	str.	91.0%	**93.5%**	93.0%
	ori.	88.0%	**88.1%**	86.8%
	inc.	69.0%	62.5%	**74.1%**
	overall	87.9%	89.4%	**90.1%**
AP 75	str.	51.5%	**55.7%**	54.0%
	ori.	53.6%	**55.7%**	53.7%
	inc.	28.4%	27.2%	**31.9%**
	overall	48.8%	**51.6%**	51.4%
mAP	str.	52.0%	**54.0%**	53.2%
	ori.	**52.1%**	51.8%	50.0%
	inc.	32.4%	29.5%	**36.6%**
	overall	49.5%	**50.7%**	**50.7%**

3.9.9 Ablation Study - Fusion at Different Stages

In section 3.6, we proposed a fusion based approach for *Radatron* to leverage the high resolution of the cascaded radar input and the distortion-free nature of the single radar input. We pass the two inputs through identical streams and concatenate them after the second ResNet block. The decision of where to fuse the two input streams is a key design choice that affects the performance of *Radatron*. We show this ablation study in Table 3.6 where we compare *Radatron* with its two other implementations: one where we fuse the two inputs at the beginning and pass them through a single stream network, and, second, where we fuse the two streams after passing them individually through all the ResNet blocks.

Looking at the results, it's evident that fusing the low resolution and high resolution inputs before feeding them into the network gives the worse performance. While *Radatron* is outperformed by its late fusion implementation for straight and oriented cars in all metrics, it still holds significant advantage over the late fusion implementation for incoming cars with improvements of 11.6%, 4.7% and 7.1% in the AP_{50}, AP_{75} and mAP metrics respectively. One possible reason for this improvement is that the number of learnable parameters increase exponentially for the late fusion implementation and the network does not see enough of these rare hard examples to learn so many parameters optimally.

3.9.10 Qualitative Results

We show example qualitative results from our test set in Fig. 3.9, by overlaying the predictions (in solid red line) and ground truth bounding boxes (dotted green line) on top of *Radatron*'s high-resolution input radar heatmaps in column (b). We also compare *Radatron*'s performance against other baselines, and summarize our observations as follows. As the resolution of the radar heatmap improves, the predictions also become more accurate especially for oriented cars. However, even with the same resolution as *Radatron*'s heatmap, the cascaded baseline suffers when the targets are moving with a high relative speed to the radar, e.g. the incoming cars in Fig. 3.9(c.iii-vi), due to motion-induced distortion as we described in section 3.3. Through distortion compensation and fusion network, *Radatron* is able to overcome this challenge and accurate predict incoming cars. We also noticed some typical failure cases for *Radatron*, which we show in Fig. 3.9(b.vi-vii). These cases are likely caused by the fusion network falsely trusting the low-resolution branch and trying to resolve non-existing motion distortion. We show more results and failure mode analysis in supplementary material.

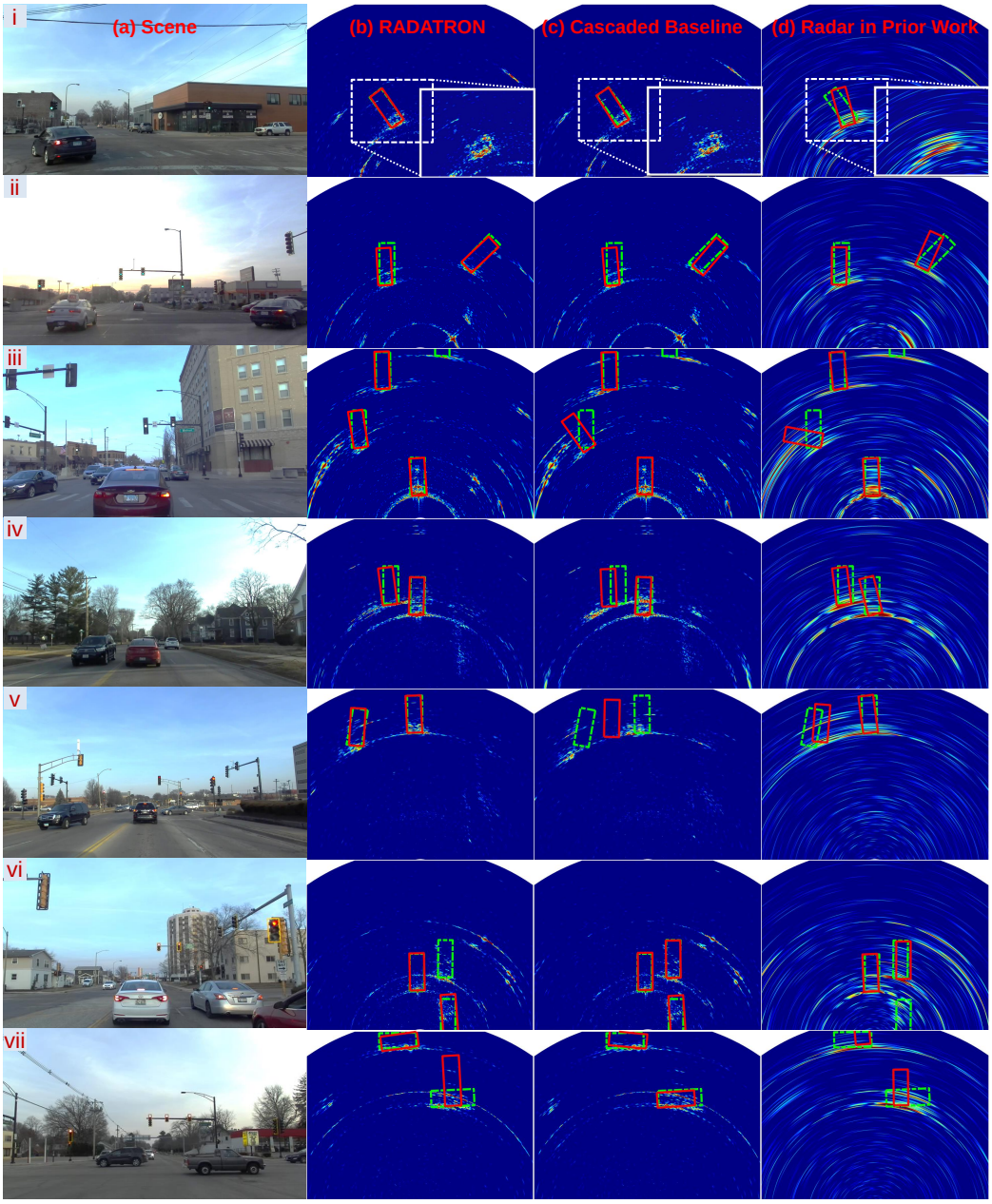


Figure 3.9: Examples from our test set. Ground truth marked in green and predictions in red. (a) Original scene. Column (b) shows *Radatron*'s performance overlaid on distortion compensated radar heatmaps. Columns (c) and (d) show the performances of stand-alone cascaded and radar in prior work baselines along with their input heatmaps respectively.

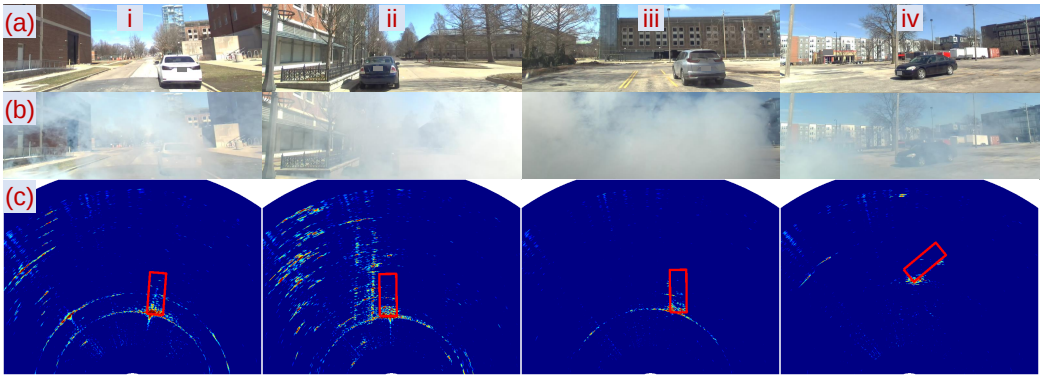


Figure 3.10: Controlled Fog Experiment. (a) Original scene. (b) Scene in fog. (c) Prediction overlaid on radar heatmap captured in fog.

3.9.11 Controlled Fog Experiment

Figure 3.10 shows *Radatron*'s performance in realistic fog emulated using a fog machine with high-density water-based fog fluid, following past work [58, 59]. As depicted in the figure, while the cars are not be visible in the RGB image, *Radatron* can accurately detect cars in the scene.

3.9.12 Failure Cases Analysis

Here we summarize a few typical failure examples of *Radatron*, and we analyze the possible reason for the prediction errors.

1. *Occlusion.* The first type of failure cases we notice is when the line of sight path to a car is partially blocked by another car. In these scenarios, *Radatron* can either miss the occluded car, e.g. Fig. 3.11(1), or predict misplaced bounding boxes, e.g. Fig. 3.11(2). This is because the metallic bodies of vehicles block mmWave signals, such that the radar signals cannot reach the occluded parts of cars. Therefore, these parts become invisible in the radar heatmap, and, in some cases, the incomplete reflections provide too little information for *Radatron* to detect the partially occluded cars.

2. *Specular reflection.* We also noticed that some predicted bounding boxes suffer from low intersection over union (IoU), either because of incorrect car size, e.g. Fig. 3.11(3,4), or inaccurate orientation, e.g. Fig. 3.11(5). Such errors are likely caused by the specular nature of mmWave radar

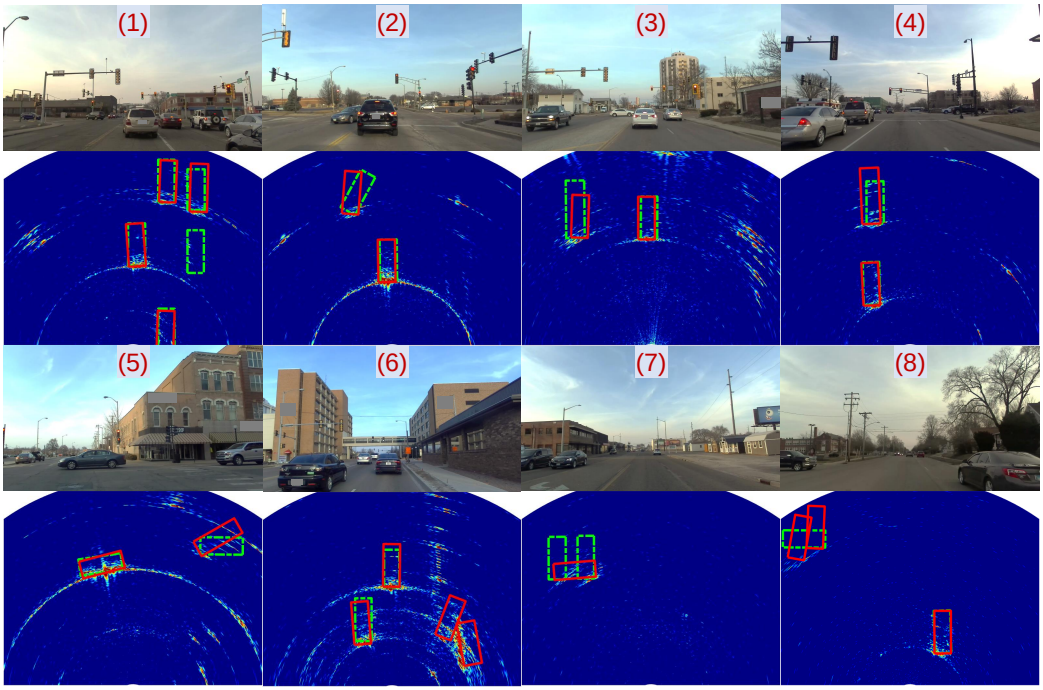


Figure 3.11: Typical prediction errors in our test set. Ground truth is marked in green and predictions are marked in red. Top row of each example shows the original scene and the bottom row shows *Radatron*'s predictions and ground truth bounding boxes overlaid on the input radar heatmaps.

reflections. Millimeter-Wave signals exhibit mirror-like reflections on the smooth metallic surfaces of cars [33], as a result, even if the car is not occluded, reflections from some parts of the car cannot propagate back to the radar receiver, rendering these parts invisible in the heatmap. *Radatron* tries to learn the specularity effect in radar reflections and infer the complete car bounding boxes. However, due to severe specularity in some scenarios, e.g. the side of the incoming pickup truck in Fig. 3.11(3), predictions can be off in size and orientation.

3. *False alarm due to background reflections.* Although in most cases *Radatron* correctly identifies foreground objects from the background, it sometimes confuses background reflections for cars. For example, in Fig. 3.11(6), the strong reflections from the building structures very close to the road is incorrectly detected as cars.

4. *Two adjacent cars.* Another tricky scenario for *Radatron* is when two cars are very close to each other as shown in Fig. 3.11(7). *Radatron* sometimes

mistakes the two clusters of reflections from the two nearby cars as the specular reflection from a horizontal car, so it draws a single bounding box across the two cars. Interestingly, we have also seen the reverse case where *Radatron* predicts two vertical bounding boxes for a single horizontal car as shown in Fig. 3.11(8). Fortunately, as we will discuss in section 3.10, we can leverage Doppler information to better distinguish two cars very close to each other versus a single horizontal car.

5. *Lower spatial resolution on the edges of the field of view.* Finally, compared to the center of the scene, *Radatron* tends to make more mistakes on the edges of the radar field of view, e.g. Fig. 3.11(5,8). This is potentially due to the lower spatial resolution on the edges compared to the center. Note that radar heatmaps do not have uniform spatial resolution across the entire field of view. The radar angular resolution decreases towards the left and right boundaries of the field of view. Besides, for the farther away distances, the same angular resolution translates into a lower spatial resolution. Finally, the transmitter and receiver antennas of the radar also have lower gain away from the center. As a result, prediction errors caused by the above mentioned sources are more commonly seen on the edges of the heatmap due to relatively lower spatial resolution. On the other hand, the reduced detection accuracy in the lower resolution regions also proves the importance of improving the spatial resolution of radar in achieving accurate object detection.

3.10 Doppler

3.10.1 Doppler Pre-Processing Algorithm

As we have described in 3.5.1, we combine 9 TX chirps to create a range-azimuth (RA) radar heatmap. However, a radar frame further include 64 such chirp loops, which we leverage to extract Doppler information. Similar to how the we estimate the motion-induced phase variances, we can calculate the phase differences of the same virtual antenna over time ($\Delta\theta^{\dagger}$) to estimate

the velocity-induce Doppler shift, and hence the velocity:

$$v = \frac{c}{4\pi f_0 \cdot 9T}\Delta\theta^{\dagger} = \frac{\lambda}{36\pi T}\Delta\theta^{\dagger} \tag{3.4}$$

where T is the time interval between consecutive chirps.

A standard algorithm applies another fast Fourier transform along the 64 chirp loops that outputs a 3D range-azimuth-Doppler (RAD) radar tensors. Objects with different velocities are grouped into different bins along the Doppler dimension in the 3D radar tensors. Prior works [89, 109, 83, 108, 99] take this 3D radar tensor and collapse it into three different 2D radar feature maps for processing and then recombined the encoded latent vectors. Considering the sparse 3D RAD radar tensor, this multi-view network design also reduces the sparsity in each 2D feature maps, making it easier to learn.

However, simply applying the 3D radar tensor processing using Doppler FFT to our cascaded radar is also problematic. This is because the much long time gap between two chirps used for Doppler processing leads to aliasing in the Doppler/velocity domain. For example, in our experimental radar configuration, the time gap between when the same TX antenna transmits in adjacent chirp loops is $45.62\mu s \times 12 = 547\mu s$. This results in a the maximum unambiguous velocity of only ± 1.73 m/s. As a result, all objects whose velocities differ by $n \times 3.47$ m/s will end up in the same velocity/Doppler bin.

To resolve the velocity/Doppler ambiguity, we leverage the fact that the minimum time gap between chirps transmitted by our cascaded MIMO radar is only one chirp interval (T). This very short time gap can be leveraged to resolve a lot of aliasing. Therefore, we try to combine the 12 TX chirps in a chirp loop and the 64 chirp loops to achieve high-resolution and less aliased Doppler estimation. Unfortunately, chirps transmitted by adjacent TX antennas are not co-located, so that in addition to the phase variance introduced by motion, they also experience AoA dependent phase differences. Earlier, when we tried to accurately estimate AoA, we tried to disentangle these two sources of phase variances by compensating for the motion-induced phase. Here, in order to accurately estimate Doppler/velocity, we need to compensate for the AoA dependent phase differences instead.

To do so, we processed low-resolution range-azimuth (RA) heatmaps with every single TX antennas in every chirp loop separately, which provides us

with $12 * 64 = 768$ 2D RA heatmaps. Every RA heatmaps is created using only one TX chirp with one chirp interval time gap in between. For each azimuth angle in these heatmaps, we compensate for the AoA dependent phase differences by multiplying with the complex conjugate of our TX antenna array steering vector. Then we take a fast Fourier transform along the 768 RA heatmaps, which outputs a 3D range-azimuth-Doppler (RAD) radar heatmap, whose azimuth resolution is the same as the low-resolution input RA heatmap to *Radatron*'s network. This 3D RAD radar tensor has very high velocity resolution of 0.05 m/s, and a maximum unambiguous velocity of ± 20.85 m/s.

Although there are still residual aliasing along the Doppler dimension due to imperfect AoA phase compensation, the dominant velocity of each object always correspond to the highest power bin the the Doppler dimension. Therefore, we further take a $argmax$ operation along the Doppler dimension to extra the dominant velocity for each range-azimuth bin. In this way, the aliases in Doppler are neglected due to their lower power, and we can obtain a 2D Range-Azimuth Doppler index feature map, whose pixel values represent the dominant velocity of the corresponding range-azimuth bin. Moreover, the sparsity of the 3D RAD radar tensor also significantly reduced, making it much easier for a relatively smaller neural network model to learn. We concatenate this 2D RA Doppler index feature map as a second channel to the single-TX input of our network.

3.10.2 Results

Table 3.7 shows the comparison of *Radatron* with and without Doppler. It can be observed that concatenating Doppler information as a second channel to the single radar input improves the overall performance by 0.5% overall, by 3.2% for oriented and by 1.5% for incoming cars in the AP_{50} metric. In the AP_{75} metric, Doppler improves the performance by 3.1% overall, by 3.6% for incoming and by 4% for straight cars. A similar improvement with Doppler also follows for the mAP metric.

We note that the network can leverage the Doppler information to separate out closely spaced cars based on their different velocities. Similarly, it can also use the Doppler information to distinguish a moving car from static

Table 3.7: Ablation study on Doppler input. Best performing model is boldfaced.

Eval Metric	Model Split	*Radatron* (With Doppler)	*Radatron*
AP 50	str.	**93.5%**	93.0%
	ori.	**90.0%**	86.8%
	inc.	72.6%	**74.1%**
	overall	**90.6%**	90.1%
AP 75	str.	**58.0%**	54.0%
	ori.	53.5%	**53.7%**
	inc.	**35.5%**	31.9%
	overall	**54.5%**	51.4%
mAP	str.	**55.3%**	53.2%
	ori.	**51.9%**	50.0%
	inc.	**37.5%**	36.6%
	overall	**52.8%**	50.7%

background clutters. We believe that these are the key reasons behind the improvement in performances with Doppler.

3.11 Limitations

First, the maximum range of *Radatron*'s radar was configured to 25m to match that of our stereo camera [127]. Hence, our dataset does not include cars beyond 25m. Second, *Radatron* does not leverage the 3D nature of its high resolution datasets, which could potentially be used to detect 3D bounding boxes. Third, *Radatron* was trained and tested using data collected in the same country and may not work as well in other locations. Finally, *Radatron* currently only detecs vehicles but could be expanded to more objects like pedestrians and bikes by annotating these classes. Addressing these limitations is left for future work.

Part II

MEMS-Enhanced Wireless Sensing & Localization

Chapter 4

MEMS Spike-Train Filter: A Primer

4.1 Micro-Electromechanical System Resonator

Part II of this thesis builds on recent advances in Micro-Electromechanical System (MEMS) Radio Frequency (RF) resonators and filters [132, 133]. MEMS technology has become popular in the design of RF filters because MEMS RF filters built with MEMS resonators feature chip-scale form-factors and a wide range of operating frequencies from a few MHz to 30 GHz. Besides, MEMS filters are also fully integratable with Integrated Circuits (IC) to form a single-chip RF solution for mobile and IoT devices.

MEMS resonators leverage the piezoelectric effect to convert RF signals between electromagnetic domain signal and mechanical domain waves, so that the signal can be processed in the mechanical domain leveraging the unique structures of the MEMS devices. Specifically, the input RF electrical signal is first converted into acoustic vibrations through inverse piezoelectric effect. The mechanical waves than pass through the structure for processing in the acoustic domain. In this process, frequency selectivity is achieved because not all frequencies can be efficiently converted between RF and acoustic domains. Finally, the acoustic waves are converted back to RF signals through piezoelectric effect. As a result, only frequencies that match the resonance frequencies of the piezoelectric structure can go through the conversions with little loss, while other frequencies are filtered out.

To better understand how the MEMS devices work, consider the diagram of a MEMS acoustic resonator shown in Fig. 4.1. This resonator is commonly referred to as a Lateral Overtone Bulk Acoustic Resonator (LOBAR). The device consists of three electrodes on the top of a thin film made of the piezoelectric material $LiNbO_3$. RF signals come through the middle electrode and can be efficiently converted into acoustic waves through the piezoelectric

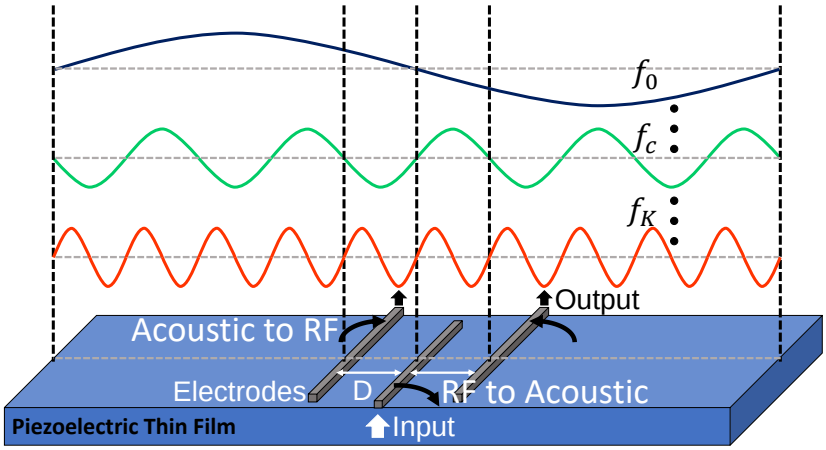


Figure 4.1: MEMS Resonator Architecture and Diagram of Operation

effect, as long as their frequencies match the resonances of the film and are supported by the electrode design. Otherwise, the signals are reflected back and the frequencies are filtered out.

4.2 MEMS Spike-Train Filters

Past work on MEMS RF filters optimize for filters with a single passband [134, 135]. In contrast, this thesis uses some of the very first MEMS Spike-Train filters designed and fabricated at University of Illinois Urbana-Champaign [136, 137], which has a number of sharp and narrow passbands, creating a spike train in the frequency domain as shown in Fig. 4.2. Therefore, the filtered spectrum becomes very sparse similar to being sampled on the frequency axis, which enables sparse recovery after below Nyquist sampling, which we will cover in more details in the following chapters.

To create such filter frequency response, the MEMS spike-train filter leverages the periodic resonance frequencies of LOBAR to create overtone resonators that have an assortment of equally spaced resonance frequencies. Note that the spike frequencies are determined by the LOBAR resonator architecture, so they can be specifically designed.

The filter design actually involves a 4-way trade-off [137] between (1) the frequency span, (2) the spacing between adjacent spikes, (3) the insertion loss in the spikes, and (4) the out-of-band rejection. In this thesis, we focus on the design of the spike-train frequencies and the spike width.

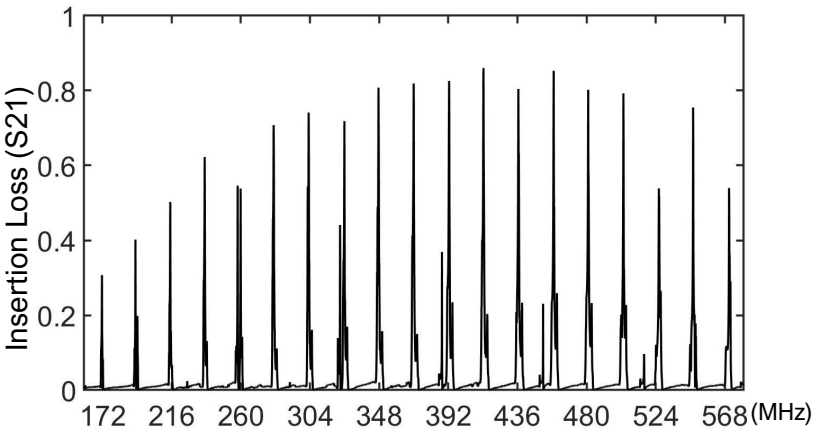


Figure 4.2: Spike-Train Filter Frequency Response

The resonance frequencies are determined by:

- The width of the film: the film supports resonance frequencies for which acoustic wave vanishes at the edges of the film [138] i.e., the sine wave crosses zero at the edges as shown in Fig. 4.1. This condition is satisfied when the width of the film W is an integer (k) multiple of half a wavelength ($W = k\lambda/2$). Since $f = v/\lambda$, where v is the acoustic velocity in the piezoelectric material, which is $\sim 4\ km/s$ in our design. The MEMS resonator will resonate at frequencies: $f_k = kv/2W$, and the spacing between spikes $\Delta f = v/W$.

- The placement of electrodes: the filter will operate at center frequency f_c determined by the distance D between the electrodes: $f_c = v/2D$. Furthermore, for an odd number of electrodes, only acoustic waves that cross zero at the middle electrode, as shown in Fig. 4.1, will resonate.

- The bandwidth or frequency span of the filter around f_c is determined by the electrodes where their RF-to-acoustic conversion efficiency degrades for resonance frequencies far from f_c, resulting in higher loss in spikes far from f_c. Adding more electrodes reduces the loss in spikes near f_c but narrows down the frequency span. We found that a three electrodes give the widest span with minimal loss of at most 2 dB.

Thus, the resonance frequencies will be the f_ks around f_c where k is even. This leads to a filter with center frequency f_c and a spike train where the spacing between the spikes is $\Delta f = v/W$. By modifying the width of the film and the position of the electrodes, we can modify Δf and f_c to control

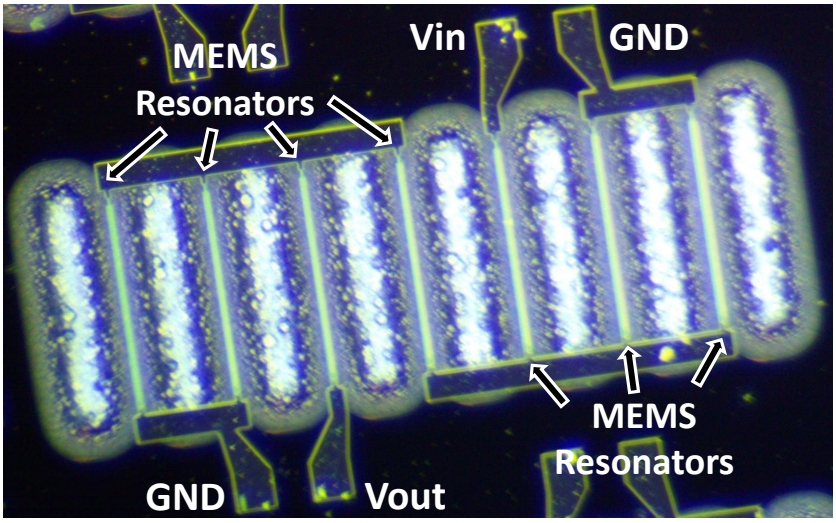


Figure 4.3: Microscope Image of Our Fabricated MEMS Filter

the frequency of the spikes in the filter.

In order to create filter passbands, a number of shunt and series resonators are combined into a ladder filter circuit [139] as shown under the microscope in Fig. 4.3. The resonance frequencies of the shunt and series resonators are set slightly differently, and this resonant frequency difference determines the spike width ΔF. Therefore, spike width ΔF of the spike-train filter is determined by the film width difference between ΔW between shunt and series resonators: $\Delta F = fc\frac{\Delta W}{W}$.

The frequency response of our fabricated filter is shown in Fig. 4.2. It has 19 periodic spikes with lowest loss between 161 and 579 MHz spanning a wide bandwidth of 418 MHz. The spacing between spikes is $\Delta f = 22$ MHz and the width of each spike is around $1 \sim 1.5$ MHz.

Chapter 5

Efficient Wideband Spectrum Sensing Using MEMS Acoustic Resonators

This chapter presents S^3, an efficient wideband spectrum sensing system that can detect the real-time occupancy of bands in large spectrum. S^3 samples the wireless spectrum below the Nyquist rate using cheap, commodity, low power analog-to-digital converters (ADC). In contrast to existing sub-Nyquist sampling techniques, which can only work for sparsely occupied spectrum, S^3 can operate correctly even in dense spectra. This makes it ideal for practical environments with dense spectrum occupancy, which is where spectrum sensing is most useful. To do so, S^3 leverages MEMS acoustic resonators that enable spike-train like filters in the RF frequency domain. These filters sparsify the spectrum while at the same time allow S^3 to monitor a small fraction of bandwidth in every band.

We introduce a new structured sparse recovery algorithm that enables S^3 to accurately detect the occupancy of multiple bands across a wide spectrum. We use our fabricated chip-scale MEMS spike-train filter to build a prototype of an S^3 spectrum sensor using low power off-the-shelf components. Results from a testbed of 19 radios show that S^3 can accurately detect the channel occupancies over a 418 MHz spectrum while sampling 8.5× below the Nyquist rate even if the spectrum is densely occupied.

5.1 Introduction

The past decade has witnessed significant changes in the wireless spectrum as the FCC (Federal Communications Committee) has repurposed many frequency bands for dynamic spectrum sharing. This includes the 6 GHz band, released in April 2020, to be shared between Wi-Fi 6E and the incumbent users in this band like microwave backhaul [140]. Another example is the 3.5 GHz Citizens Broadband Radio Service (CBRS) band, which was recently

approved for commercial deployments in September 2019. To leverage the CBRS band, unlicensed devices must sense a 200 MHz spectrum and avoid causing interference to primary and licensed users like military radars [141]. Of course, an earlier and more well-known example of spectrum sharing is the TV White Spaces which were released in 2010 [142]. Moreover, there are lots of opportunities for spectrum sharing in the millimeter-wave frequencies. In particular, the FCC released 14 GHz of unlicensed spectrum in the 60 GHz band that can be shared among Wi-Fi and IoT technologies [143]. These changes have been driven by the ever-increasing demand for wireless connectivity and aim to exploit previously underutilized frequency bands to accommodate new unlicensed applications and achieve highly efficient usage of the spectrum.

Efficient and truly dynamic spectrum sharing, however, requires unlicensed devices to sense wideband spectrum (hundreds of MHz to GHz) in real-time to spot and access momentarily idle channels. Unfortunately, real-time wideband spectrum sensing is challenging since it requires high-speed analog-to-digital converters (ADCs) that can sample the signal at the Nyquist sampling rate. Such high-speed ADCs are expensive, have low bit resolution, and can consume several watts of power [144, 145, 146, 147, 148].[1] To avoid using high-speed ADCs, today's systems sequentially scan the spectrum, monitoring each narrow band for a short period of time [149, 150]. As a result, they cannot continuously sense all bands in real-time and can easily miss highly dynamic and fleeting signals such as radar waveforms in the CBRS band [151].

Past work has proposed using compressive sensing or sparse Fourier transforms to sense wideband spectrum without sampling at the Nyquist rate [152, 153, 154, 155]. However, these approaches inherently rely on the assumption that the frequency spectrum is sparsely occupied. Hence, they only work in the case of underutilized spectrum where at most 5% to 10% of the frequency bands are occupied [155, 148]. The goal of dynamic spectrum sharing, however, is to efficiently utilize the spectrum. Hence, wideband spectrum sensing must work even in a densely occupied spectrum in order to scale usage to many users and achieve high utilization.

[1] In fact, the power consumption of spectrum sensors is dictated by the ADC sampling rate as shown in [148]. Hence, we can significantly improve the energy efficiency by reducing the sampling rate.

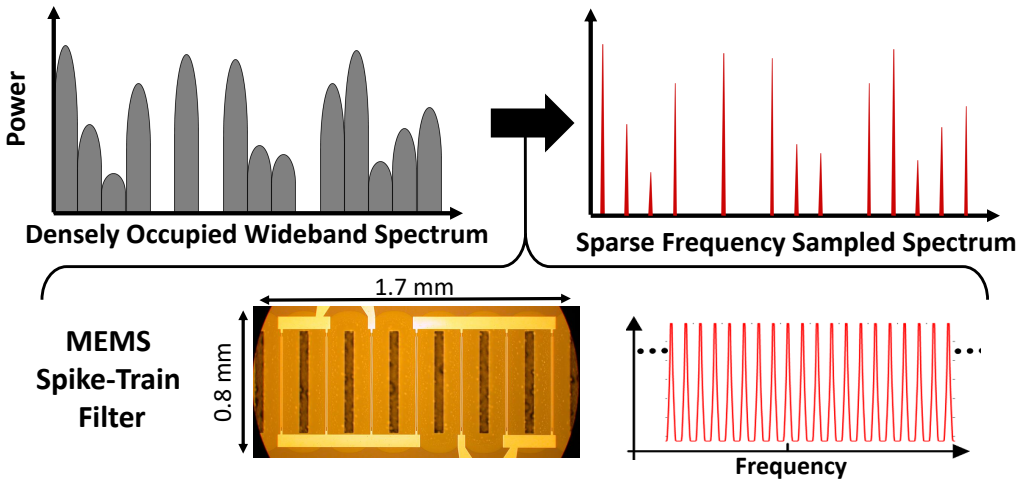


Figure 5.1: Filtering using MEMS Spike-Train Resonators

In this chapter, we introduce S^3 (Spectrum Sensing Spike-train), an efficient low power spectrum sensing system that can monitor the real-time occupancy of multiple frequency bands in a wide spectrum. S^3 samples the wireless spectrum below the Nyquist sampling rate using cheap, commodity, low power ADCs but does not assume that the spectrum is sparsely occupied. A key enabler of S^3 is the use of MEMS acoustic resonators that can create a spike-train like filter in frequency as shown in Fig 5.1. The MEMS filter processes the signal in the acoustic domain using carefully designed piezoelectric resonators with an assortment of equally spaced resonance frequencies. The resonators will pass the signals in these resonance frequencies and filter out the rest before converting the signal back to the RF domain. This creates an RF filter with very narrow, sharp, and periodic passbands across a wideband spectrum.

The spike-train filter enables S^3 to sample the spectrum in the frequency domain and monitor a small fraction of bandwidth in every band. S^3 can then tell if a band is occupied or idle by examining the sampled bandwidth in it, without the need to recover the entire band. This is like finding an available spot in a parking lot; we can tell if a spot is taken by peeking at some part of the car in it, and we don't need to get close to every spot and see the entire car. Moreover, even if the wideband spectrum is densely occupied, the filter makes the spectrum significantly sparser as shown in Fig. 5.1. This enables S^3 to sample the signal below the Nyquist sampling rate and still recover the channel occupancies.

Translating S^3 into a practical system, however, requires addressing two

key challenges. First, we need an algorithm that can accurately and efficiently reconstruct the spectrum occupancy. To address this, S^3 builds on past work in sparse recovery theory but differs from it in key aspects. In particular, compressive sensing algorithms require randomly sampling the time signal and cannot simply be implemented using low-speed ADCs [153, 156]. Sparse Fourier transform algorithms, on the other hand, can be implemented using low-speed ADCs, but they assume that the sparsely occupied bands are randomly distributed in the frequency spectrum [157, 155]. The MEMS filter creates a sparse spectrum that is highly periodic and far from random. For such sparsity patterns, sparse Fourier transform algorithms are highly sub-optimal.

S^3 aims to achieve the best of both worlds, i.e. no random sampling in time and no assumption of random distribution of occupied frequencies. To this end, S^3 leverages the uniquely structured sparsity pattern created by the filter to overcome the above challenges. The filter restricts the occupied frequencies to known locations in the spectrum, which significantly reduces the search space. It also allows us to optimize the sub-Nyquist sampling rate. In particular, optimal recovery can be achieved by choosing a sub-sampling factor that is co-prime to the number of spikes in the filter, as we show in section 5.4.

The second challenge is that in practice the MEMS resonators do not create an ideal spike-train. The spikes are not extremely narrow and have a small passband bandwidth which reduces the sparsity. Moreover, the separation between the spikes is not perfectly equal, and the spikes themselves are not identical. To address this, S^3 leverages the fact that different filters that are manufactured using the same process exhibit a very similar non-ideal spike-train, as we show in section 5.5. Hence, the filter frequency response can be measured once and incorporated into the design of S^3. Specifically, we co-design the hardware and recovery algorithm of S^3 to account for the filter non-idealities and optimize its performance.

Evaluation: We had fabricated a chip-scale MEMS filter, shown in Fig. 5.1, which we leveraged to build a working prototype of S^3. The prototype can sense channel occupancies over a 418 MHz spectrum in real-time while sampling 8.5× below the Nyquist rate. The prototype uses two cheap, low power, off-the-shelf ADCs that sample around 50 MS/s ($\approx 1/17$ of the Nyquist rate).

We extensively evaluate the performance of S^3 using a wireless testbed with 20 software defined radios that can occupy the entire 418 MHz spectrum at various power levels. Our results show that S^3 can accurately detect occupied channels. Even when the spectrum is as crowded as 90% occupied, S^3 achieves a false positive rate of 0.02 and a false negative rate of 0.0047. We also compare S^3 to state-of-the-art prior work like BigBand [155] and SweepSense [158] and demonstrate $5 - 10\times$ lower error rate for non-sparse spectrum. Furthermore, we show that S^3 can recover the wireless spectrum by performing outdoor and indoor measurements at various frequencies using a spectrum analyzer as the ground truth. Finally, we extend S^3 to not only detect the occupancy of the bands but also capture the power spectral density of the spectrum by quickly sweeping the center frequency for 22 MHz to cover the separation between the spikes.

Contributions: This chapter has the following contributions:

- The chapter bridges the latest advances in overtone MEMS acoustic resonators to RF spectrum sensing by leveraging spike-train filters to enable cheap and low power real-time wideband sensing of a densely occupied spectrum.
- The chapter presents a novel sparse recovery algorithm that leverages the uniquely structured spectrum sparsity to efficiently recover a spectrum sampled significantly below the Nyquist sampling rate.
- The chapter builds a prototype using commodity low-power components and evaluates its performance in a real testbed.

5.2 Background

In this section, we provide a brief background on wideband spectrum sensing using sub-Nyquist sampling. Further related work and background on spectrum sensing can be found in section 5.9.

This chapter builds on past work that senses wideband spectrum without sampling at the Nyquist rate using compressive sensing [153, 159, 156, 160, 161, 154, 162, 148] or sparse Fourier transform algorithms [155, 163]. However, these approaches only work when the spectrum is underutilized and

sparsely occupied, which defeats the purpose of efficiently utilizing the spectrum. Furthermore, compressive sensing needs random sampling [153, 156, 148], and as a result, requires custom hardware designs that can consume as much power as an ADC that samples at the Nyquist rate [164, 165]. Sparse Fourier transform algorithms do not necessarily require random sampling but must assume that the sparsely occupied bands are randomly distributed in the frequency spectrum to accurately recover the frequencies [155, 157].

BigBand [155] leverages sparse Fourier transform and uses co-prime sampling to acquire a sparse bandwidth while sampling 6× below the Nyquist rate. However, it only works up to 10% spectrum occupancy at which point it cannot recover the status of more than 14% of the spectrum. An extension, D-BigBand [163] can sense dense spectrum by considering the differential changes in occupancy. However, it assumes that the spectrum occupancy is mostly static with very few changes over time. Hence, it would not work for dynamic spectrum sharing where users sense and opportunistically transmit whenever they find an idle channel. Ma et al. [166] also attempt to extend BigBand to dense spectrum but requires sampling the signal first at the Nyquist rate in order to permute the samples and filter the signal before further sub-sampling it below Nyquist. S^3, on the other hand, can sense dense spectrum without the need for Nyquist sampling or random sampling. It also makes no assumptions on the changes in occupancy or the distribution of occupied bands across the spectrum.

The possibility of using MEMS RF filters in wideband spectrum sensing has been raised up with both parallelized single passband filter array [167] and overtone filters [136, 137]. Nguyen el.all proposed a frequency gating spectrum analyzer consist of a RF channel-select filter bank implemented a micromechanical circuit [167]. Using spike-train filters as narrowband channelizers in spectrum sensors have been proposed [136, 137], along with feature-base algorithm and sparse-Fourier transform based algorithm respectively. However, to the best of our knowledge, no proof of concept wideband spectrum sensing systems using spike-train filters have been demonstrated. In this paper, we present a full energy-efficient wideband sensing system enabled by spike-train filter. Besides, instead of directly applying existed algorihtm, we co-design the sensor hardware and the occupancy detection to achieve minimum sampling rate and computational complexity.

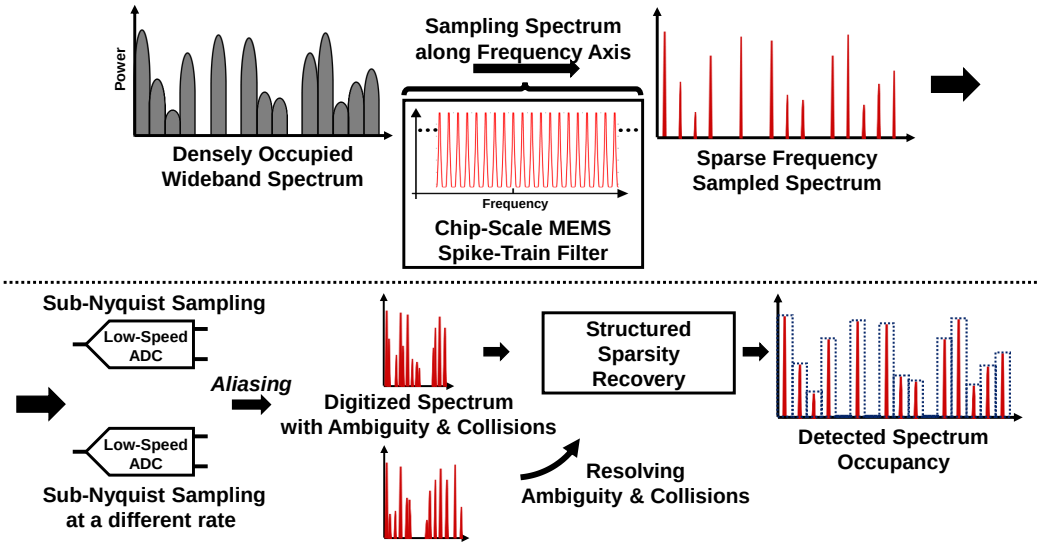


Figure 5.2: S^3 System Pipeline: S^3 samples the wideband spectrum along the frequency axis using the spike-train filter. The output spectrum is sub-sampled with low-speed ADC. S^3 leverages the structured sparsity in the filtered spectrum to resolve ambiguity and collisions due to aliasing and recovers spectrum occupancy.

5.3 S^3 Overview

S^3 leverages the MEMS spike-train filters we introduced in chapter 4 to sense wideband spectrum while sampling below the Nyquist rate. Figure 5.2 illustrates an overview of the system pipeline. The received wideband spectrum is passed through the MEMS spike-train filter which samples the bands in the spectrum along the frequency axis. Specifically, the filter passes signals in frequencies aligned with the spikes and suppresses all the rest of the frequency components in the spectrum as shown in Fig. 5.2. The output of the filter is a sparse spectrum that preserves a small fraction of each band which we can use to monitor the occupancy of the band. Since the output spectrum is sparse, we can sample it below the Nyquist rate and still recover the occupancy information efficiently.

S^3 uses low-speed ADCs to sub-sample the signal. However, sampling below the Nyquist rate results in *"aliasing"* in the frequency domain i.e., multiple frequencies across the wide spectrum will alias (map) to the same frequency. Aliasing can lead to ambiguity and collisions, which prevent us from distinguishing frequencies that are occupied from those that are not. S^3 leverages the uniquely structured sparsity at the output of the spike-train filter to resolve such ambiguity and collisions and recover the spectrum oc-

cupancy. Ideally, one ADC is sufficient as we prove in section 5.4. However, due to practical limitations and imperfections in the spike-train filter, S^3 must use two ADCs sampling at different rates to accurately resolve ambiguity and collisions. We co-design the hardware and recovery algorithm to optimize the ADC sampling rates while accounting for the non-idealities of the spike-train filter as we describe in detail in section 5.5.

5.4 S^3 Recovery Algorithm

In this section, we describe S^3 recovery algorithm assuming an ideal spike-train filter. In later sections, we extend S^3 to deal with practical limitations.

Ideally, the spike-train filter will have equally spaced, very narrow and sharp spikes that can be approximated as an impulse train.[2] The frequency response of such a filter can be modeled as:

$$G(f) = \sum_{k}^{K} \delta(f - k\Delta f - f_0) \tag{5.1}$$

where K is the number of spikes, Δf is the spacing between spikes, and f_0 is the frequency of the first spike as shown in Fig. 5.2. Hence, the filter covers a spectrum bandwidth of $BW = \Delta f \times K$.

Let $x(t)$ be the input wideband signal in time domain and $X(f)$ be its non-sparse frequency representation whose bandwidth is also BW. After passing $x(t)$ through the spike-train filter, we get the signal $\tilde{x}(t)$ whose frequency spectrum is:

$$\tilde{X}(f) = X(f)G(f) = \sum_{k}^{K} A_k \delta(f - k\Delta f - f_0) \tag{5.2}$$

where $A_k = X(k\Delta f + f_0)$. $\tilde{X}(f)$ is at most K sparse i.e., it has at most K large frequency coefficients. Our goal is to recover these K coefficients A_k and estimate their power to detect the occupancy of the band around the frequency $f_0 + k\Delta f$.

S^3 samples the signal $\tilde{x}(t)$ using a low-speed ADC that samples at a rate

[2]We can approximate the spikes as impulses if the width of the spike $\ll 1/T$ where T is the time window over which we sample the signal.

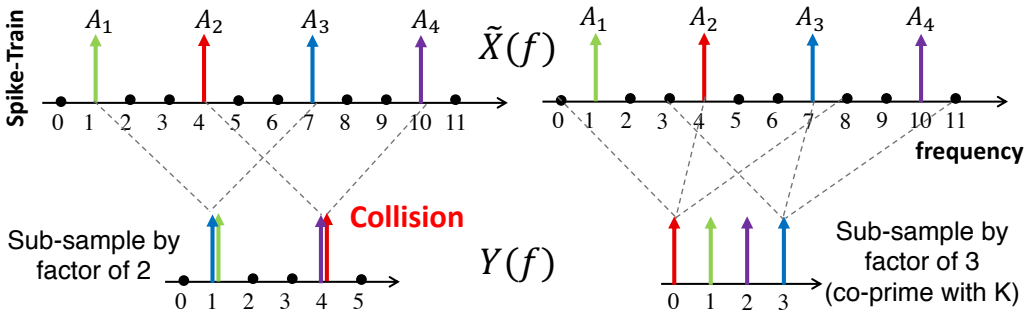


Figure 5.3: Co-prime sub-sampling factor avoids frequency collisions: Sub-sampling the spike train by 2 results in collisions between the spikes, but sub-sampling by 3 can avoid collisions because it is co-prime with K=4.

$R = BW/P$ where P is an integer corresponding to the subsampling factor.[3] The sampling rate R is chosen such that $K \leq R \ll BW$. Let $y(t)$ be the sampled signal i.e., $y(t) = \tilde{x}(P \times t)$, and let $Y(f)$ be the Fourier transform of $y(t)$. Then, $Y(f)$ is an aliased version of $\tilde{X}(f)$:

$$Y(f) = \sum_{i=0}^{P-1} \tilde{X}(f + iR) \tag{5.3}$$

$Y(f)$ will cover a narrow bandwidth equal to R where frequencies in $\tilde{X}(f)$ that are equally spaced by R alias and sum together in the same frequency bin in $Y(f)$. Hence, once S^3 detects power in a frequency bin $Y(f)$, it knows that this power could have come from P different candidate frequencies in $\tilde{X}(f)$. Figure 5.3 shows an example where if we sub-sample the signal by a factor $P = 2$, then every two equally spaced frequencies in $\tilde{X}(f)$ map to one value in $Y(f)$. Since $\tilde{X}(f)$ only has power in K coefficients A_k corresponding to the spikes of the filter, S^3 can easily eliminate a lot of candidates. Ideally, we want these coefficients to map to different bins. In this case, the bin value will be the same as the coefficient A_k which we can immediately estimate. However, if two coefficients A_{k_1} and A_{k_2} collide in the same bin as shown in Fig. 5.3, it will not be possible for S^3 to distinguish and estimate them.

S^3 can choose the sampling rate R in a manner that guarantees that no two coefficients collide. In particular, if the sub-sampling factor P and the number of spikes K are co-prime, then we can guarantee that none of K

[3]Note that for simplicity, we have assumed that the ADC takes complex samples of the signal i.e., there are two ADCs sampling the I and Q of the wireless signal. We will relax this assumption in the following section.

coefficients collide in the same bin and become indistinguishable. To see this, consider the example shown in Fig. 5.3 where we have $K = 4$ spikes with coefficients A_1 to A_4 in the filtered spectrum $\tilde{X}(f)$. When we sub-sample by a factor of 2 below Nyquist, there will be collisions between A_1 and A_3, as well as A_2 and A_4. However, when we sub-sample by a factor of 3 below Nyquist, none of the coefficients collide, because the sub-sampling factor $P = 3$ and the number of spikes $K = 4$ are co-prime.

It is worth noting here that even though $P = 3$ uses a lower sampling rate than $P = 2$, increasing the sampling rate in this case results in more collisions. This is in contrast to past work on sub-Nyquist sampling [155, 157] where higher sampling rates reduce collisions as the coefficients are assumed to be randomly distributed in the spectrum. Unlike past work, the structured sparsity of our spectrum requires carefully selecting the sampling rate to ensure that all coefficients can easily and immediately be recovered.

The below lemma theoretically proves that if P and K are co-prime, then none of the coefficients will collide.

Lemma 5.1. *Given K, P are co-prime integers, let f_i and f_j be the frequencies of any two spikes in the spike train filter i.e. $f_i = k_i\Delta f + f_0$ and $f_j = k_j\Delta f + f_0$ such that $0 \leq k_i, k_j < K$. Then, for all $f_i \neq f_j$, we have $f_i \neq f_j$ mod R.*

Proof. Assume there exist an $f_i \neq f_j$ such that the coefficients collide i.e., $f_i = f_j \bmod R$. Note that by definition of the spike train, we also have $f_i = f_j \bmod \Delta f$. Consequently, f_i and f_j are equal modulo the least common multiple: $\text{LCM}(R, \Delta f) = \text{LCM}(BW/P, BW/K) = BW$, since K and P are co-prime. Thus, $f_i = f_j \bmod BW$ which is a contradiction since BW is the entire bandwidth and we are given that $f_i \neq f_j$. Hence, by contradiction, for all $f_i \neq f_j$, we have $f_i \neq f_j \bmod R$ and none of the K coefficients collide. □

Given that we can choose a sampling rate that results in no collisions, we can easily recover the coefficients A_k as follows. We can compute $Y(f)$ by taking an FFT of $y(t)$ and for $0 \leq k < K$, we directly set $A_k = Y((k\Delta f + f_0) \bmod R)$. We then apply an energy detector on A_k to obtain the occupancy of the band around the frequency $k\Delta f + f_0$. If $|A_k|^2$ is above the noise floor, then the band is occupied, otherwise, it is empty. A pseudocode for the overall sensing of S^3 with an ideal spike-train filter is shown in Alg. 1.

Algorithm 1 S^3 Sensing with an Ideal Spike-Train Filter

Input: $x(t)$
$B_k \leftarrow$ Band around frequency $f = k\Delta f + f_0$
$\tilde{x}(t) = g(t) \circledast x(t)$ ▷ Filter $\tilde{X}(f) = X(f)G(f)$
$y(t) = \tilde{x}(P \times t)$ ▷ Sub-Nyquist Sample
$Y(f) = \text{FFT}(y(t))$
$A_k = Y\big((k\Delta f + f_0) \bmod R\big)$
if $E\big[|A_k|^2\big] > \sigma^2$ **then**
    B_k is occupied
else
    B_k is empty

We can also prove the following theorem about the correctness and the computational complexity of the algorithm.

Theorem 5.1. *Assuming a signal SNR > 0 dB for each occupied band, the system correctly recovers the occupancy of the bands using $O(K)$ samples and $O(K \log K)$ computations which is optimal.*

Proof. We will prove the above statement for the case where the entire spectrum is occupied. We can compute the the signal power of the filtered and sub-sampled signal as:

$$\begin{aligned} E\left[\|\tilde{Y}(f)\|_2^2\right] = E\left[\sum_{f=0}^{R-1} |Y(f)|^2\right] = E\left[\sum_{f=0}^{R-1}\sum_{i=0}^{P-1} |\tilde{X}(f+iR)|^2\right] \\ = E\left[\sum_{k=1}^{K} |\tilde{X}(k\Delta f + f_0)|^2\right] = E\left[\sum_{k=1}^{K} |A_k|^2\right] \geq KE\left[\min_k |A_k|^2\right] \end{aligned} \tag{5.4}$$

Let σ^2 be the noise power per frequency. Since the spike-train filter suppresses the noise outside the spikes, the remaining noise in the signal is $K\sigma^2$. Hence, the SNR of the filtered and sub-sampled signal is:

$$SNR = \frac{E\left[\|\tilde{Y}(f)\|_2^2\right]}{K\sigma^2} \geq \frac{E\left[\min_k |A_k|^2\right]}{\sigma^2} > 1 \tag{5.5}$$

Thus, as long as the received signal is above the noise floor i.e. $SNR > 1$ (0 dB), filtering and sub-sampling will not increase the noise floor and the occupancy of the band can be detected correctly. Now, the algorithm samples at rate $R = O(K)$, takes an FFT of size $O(R)$ and then performs $O(R)$ computations. Hence, it requires $O(K \log K)$ computations and $O(K)$ sam-

ple, which is optimal. The algorithm is also deterministic, unlike compressive sensing and sparse Fourier transform algorithms which are randomized. □

5.5 S^3 with Practical Limitations

As one can see from the frequency response of our fabricated filter shown in Fig. 4.2, the MEMS spike-train filter is non-ideal i.e., the spikes have some width as can be seen from Fig. 5.4. Although the ~ 1.5 MHz bandwidth is narrow compared to the channel bandwidth, it is still significant. Moreover, the spikes are neither identical nor perfectly equally spaced. In fact, they differ in magnitudes, bandwidths, and shapes. As a result, if we simply pick a sub-sampling factor P that is co-prime to the number of spikes K, there could be many collisions among the wide spikes. Figure 5.5 shows how the 19 spikes of our spike-train filter alias after sub-sampling. Figure 5.5(a) shows the spikes in the original wideband spectrum, while Fig. 5.5(b-d) show the aliasing of the spikes when sub-sampled at three different sampling rates. First, we choose the sampling rate to be 38 MS/s, because the resulting sub-sampling factor $P = 11$ is co-prime to $K = 19$. However, the aliased spectrum ends up with many collisions, as shown in Fig. 5.5(b). This suggests that the derived optimum no longer holds due to the practical limitations of the filter.

Fortunately, different filters that are manufactured through the same process exhibit a very similar spike train. Figure 5.4 compares the measured frequency responses of three spike-train filters we fabricated. We zoom into two spikes; otherwise, the differences are very hard to spot. As one can see, the filters are almost identical. Hence, we can measure the frequency response of one spike-train filter and use it for the others.

Knowing the filter frequency response, we run an optimization problem to find a sampling rate that has as little collisions as possible. Ideally, this sampling rate should separate all the wide spikes after aliasing and prevent them from overlapping with one another. If a collision is unavoidable, we want it to only occur at the boundaries of the spikes, rather than having two wide spikes fully overlap. For example, as shown in Fig. 5.5(b), the collisions marked in red are unacceptable, because most of a spike's frequencies experience collision. In contrast, the collisions marked in green in Fig. 5.5(c) are

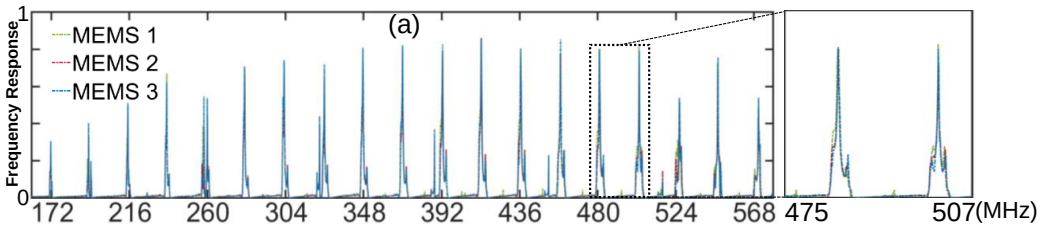


Figure 5.4: Measured Frequency Responses of Three Fabricated MEMS Spike-Train Filters using the Same Process

tolerable, because only the boundaries of two spikes collide. Because we can simulate and compare the aliasing at different sampling rates offline, the optimization problem, in fact, can exhaustively search for all possible sampling rates.

Another practical aspect is that in the real system, we only sample real signals and not complex in order to reduce complexity. Since the signal is real, the frequency representation is symmetric around the y-axis. Hence, with a sampling rate of $R = 38$ MS/s, the wideband spectrum actually aliases to a bandwidth of $R/2 = 19\ MHz$. Formally, if the original frequency of a spike is $f_{spike} = k\frac{R}{2} + b$, where $b < R/2$, then the aliasing frequency f_{alias} of the spike can be found through the following equation:

$$f_{alias} = \begin{cases} b & \text{if k is even} \\ \frac{R}{2} - b & \text{if k is odd} \end{cases} \tag{5.6}$$

In our specific case, we find 45.5 MS/s to be a really good sampling rate. As can be seen from Fig. 5.5(c), it spreads out the aliased frequencies of the wide spikes to 1.5, 2.6, 3.8, 4.6, 5.7, 7.3 MHz, etc. Therefore, most collisions only occur at the boundaries of the wide spikes. However, it still cannot avoid all unacceptable collisions. In fact, it is likely that no sampling rate can. For example, our spike-train filter has a unique 17^{th} spike that is composed of two very close spikes. When sampling at 45.5 MS/s, these two small spikes completely overlap with the 15^{th} and 19^{th} spikes respectively. Therefore, when spike 15 and 19 are both occupied, we might falsely classify the spike 17 as occupied.

To resolve such unavoidable collisions, we leverage another sampling rate that provides us with a different set of aliasing frequencies for the spikes. We pick the second sampling rate in a way that any two spikes colliding at

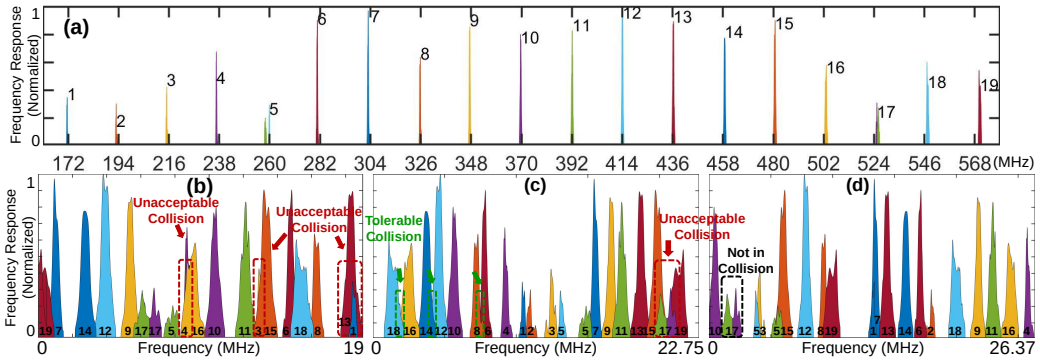


Figure 5.5: Aliasing of the spike-train filter at different sub-Nyquist sampling rates: (a) Locations of the 19 spikes on the frequency axis (b)Aliasing of the spikes at 38 MS/s (c) Aliasing of the spikes at 45.5 MS/s (b) Aliasing of the spikes at 52.74 MS/s.

45.5 MS/s do not collide again. To this end, we find a good sampling rate of 52.74 MS/s, and the resulting aliased frequencies of the spike train is shown in Fig. 5.5(d). One can see that the two parts of spike 17 do not collide with any other spikes at 52.74 MS/s. Thus, as long as we observe no power on frequencies corresponding to spike 17 at 52.74 MS/s, we will classify spike 17 as empty. Hence, by leveraging such incoherence between the two sampling rates, we can further resolve unavoidable frequency collisions and correctly identify the empty bands.

Using the two sub-Nyquist sampled spectra, S^3 recovers signal power in each spike, and then identifies the occupancy of the corresponding band. We leverage the two sampling rates through a soft voting scheme. The idea is that given an aliased spectrum and the sampling rate, we know all the possible original frequencies that correspond to the aliased frequencies. Hence, each aliased spectrum provides a vote for the source frequencies of the non-empty spectral components. Moreover, the non-empty frequencies on the original spectrum are also constrained to the spike-train frequencies. Therefore, when the two sampling rates vote for the same frequency that also falls in a spike, the frequency is very likely to be the true source frequency on the wideband spectrum.

Consider the two aliased versions shown in Fig. 5.6(a,b), where 11 out of the 19 bands are occupied and the other 8 bands are empty. Now we use them to vote where the non-empty frequency components come from. According to Eq. 5.6, aliasing folds the wideband spectrum on to the bandwidth of $\frac{R}{2}$. Therefore, we can vote on all the possible source frequencies by unfolding

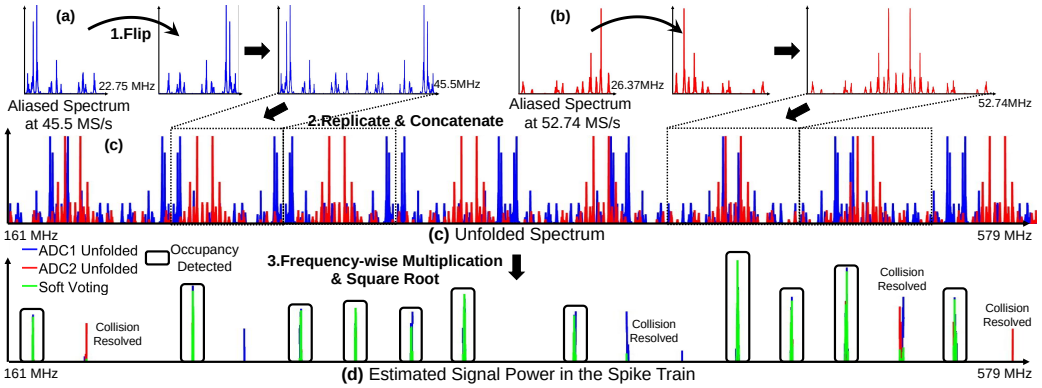


Figure 5.6: Practical S^3 Recovery Algorithm Through Voting: S^3 unfolds two aliased versions of the filtered spectrum to get a vote for the frequency components in the spike train. It then combines votes from the two different sampling rates to estimate the signal power in the spikes and the spectrum occupancy.

the aliased spectrum. We accomplish this goal in the following three steps:

- **1. Unfold - Flip**: First, we flip the aliased spectrum $Y(f)$ with a bandwidth of $\frac{R}{2}$ to get the bandwidth between $\frac{R}{2}$ and R, as it equals to $Y(\frac{R}{2}-f)$ according to Eq. 5.6.

- **2. Unfold - Replicate**: Then we replicate and concatenate the resulting spectrum from 0 Hz to R, and we get a vote for all frequencies in the frequency range of the spike train as shown in Fig. 5.6(c).

- **3. Soft Voting**: Finally, we combine the votes of the two sampling rates, where we only consider the frequencies within the spikes. This is done by multiplying the two votes on every frequency and taking a square-root. As a result, the non-empty frequencies that are voted by both sampling rates are amplified. In contrast, the frequencies where the two sampling rates vote differently will be attenuated as shown in Fig. 5.6(d).

After unfolding the aliased spectra and recovering the filtered spectrum through voting, we calculate the average signal power in each spike by summing up the voting results and divide it by the spike width. Additionally, we also estimate the average signal power in the spikes using the unfolded spectrum at each sampling rate separately. We classify a band as occupied if all three power estimations in the corresponding spike exceed a pre-selected power threshold. This power threshold is selected based on the noise floor, which is measured when all bands are empty. By using all three estimates,

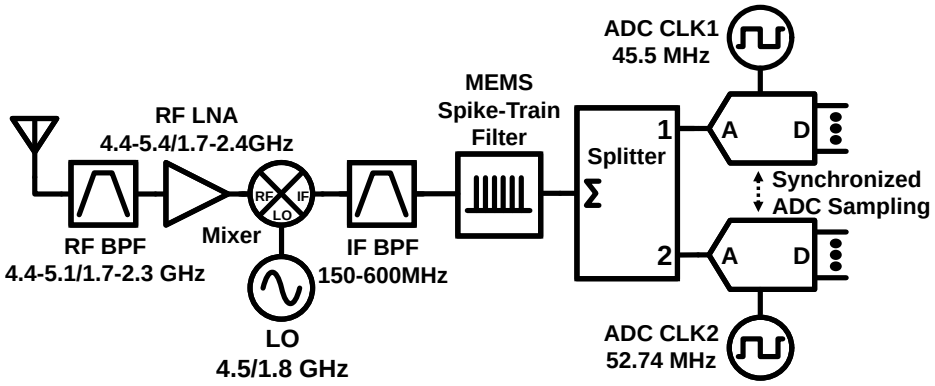


Figure 5.7: Circuit Diagram of S^3 Basic Prototype

we add hard voting on top of the soft voting which adds more robustness to the occupancy detection.

5.6 Implementation

We have built a basic prototype of an S^3 spectrum sensor by combining our MEMS spike-train filter with commodity, off-the-shelf, low-power components. Figure 5.7 shows the circuit diagram of this basic prototype, and the actual prototype is demonstrated in Fig. 5.8. The signal is received through a broadband receiver. It is bandpass filtered and amplified before down-conversion to an intermediate frequency (IF) between 150 and 600 MHz. The IF signal is bandpass filtered and passed through the spike-train filter. It is then split and sampled by the two synchronized ADCs.

5.6.1 Basic Prototype

We wire-bond the MEMS spike-train filter onto a gold-plated PCB (printed circuit board) as shown in Fig. 5.8. We use K&S 4523A Wedge Bonder and 25 μm Aluminum wire. We use two Anolog Devices LTC2261-14 14-bit ADCs to sample the output of the spike-train filter. This ADC features an 800 MHz wideband input analog bandwidth and low power consumption of 89 mW. The ADC sampling is timed through an external square-wave clock signal. We use the DC1370A ADC evaluation board and the DC890 data acquisition controller to control the ADC sampling through the open-sourced LinearLab

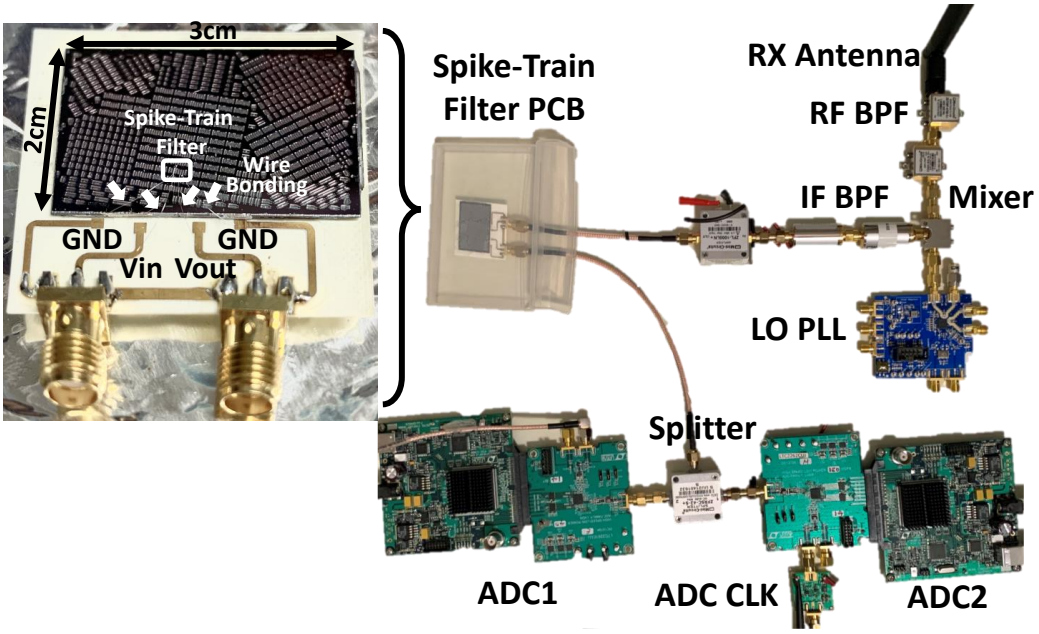


Figure 5.8: S^3 Basic Prototype Circuit

Tools Python API. We bypass the input low pass anti-aliasing filter on the ADC evaluation board to maintain the wide analog bandwidth.

5.7 Evaluation

In this section, we present our main evaluation results along with a few microbenchmarks that provide insights into the performance of S^3 in various spectra.

5.7.1 Testbed

We evaluate the performance on S^3 both through controlled experiments in an indoor wireless testbed as well as through measurements of ambient transmissions outdoors and indoors. The wireless testbed allows us to control the spectrum sparsity, how fast the occupancy changes for different bands, the type of signals transmitted, and the power of various transmissions. It also allows us to know the groundtruth band occupancy in order to evaluate the performance of S^3.

The testbed, shown in Fig. 5.9, can create a 418 MHz spectrum with var-

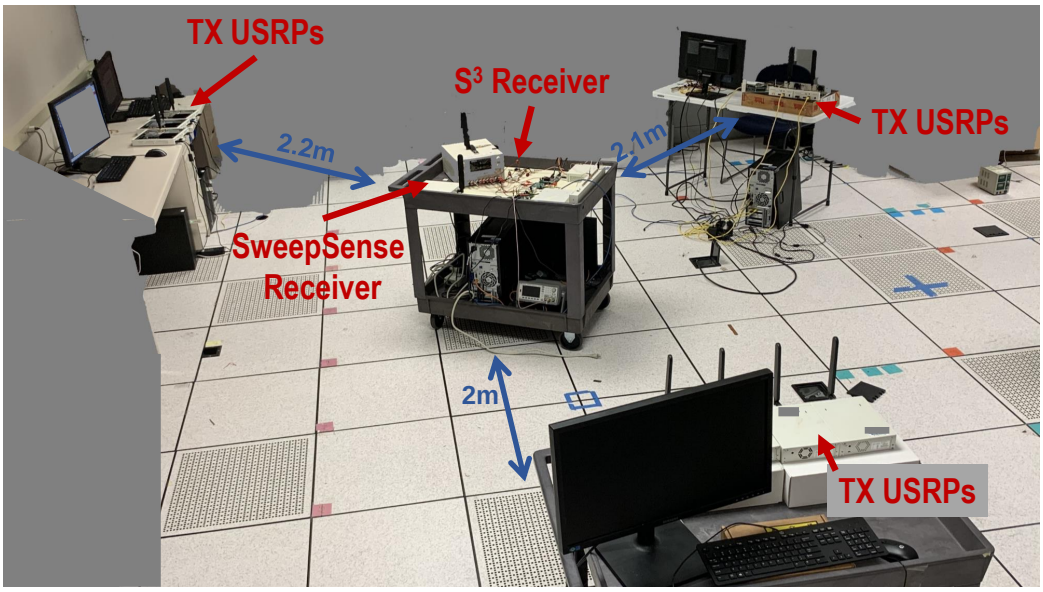


Figure 5.9: Evaluation Testbed of S^3

ious occupancy status at different frequencies. It consists of 19 N210 USRP software-defined radios, each transmitting on a 25 MHz bandwidth. While the USRPs are not very far from each other, we vary their transmission power randomly by up to 10 dB and observe received signal SNR that varies by up to 20 dB between different USRP transmitters. To avoid interference from ambient 2.4 and 5 GHz ISM band signals, we conducted experiments in two 418 MHz-wide spectra: 4.73 to 5.15 GHz and 1.93 to 2.35 GHz, each divided into nineteen 22 MHz bands. We vary the spectrum occupancy from 10% to 90%. We also vary the type of modulation being used. We test with single carrier BPSK and QAM as well as OFDM signals. Note that single carrier modulation has a non-flat power spectral density and significantly more leakage, so it results in higher false positive rates as we show in section 5.7. We also leverage the testbed to compare S^3 with state-of-the-art sensing systems as our baselines. We ran over 5000 experiments with different configuration of occupancy, power, modulation, etc.

5.7.2 Sensing Densely Occupied Spectrum

We evaluate S^3 using following metrics:

- *False Positive Rate (FPR)*: Percentage of empty bands that S^3 incorrectly reports as occupied.

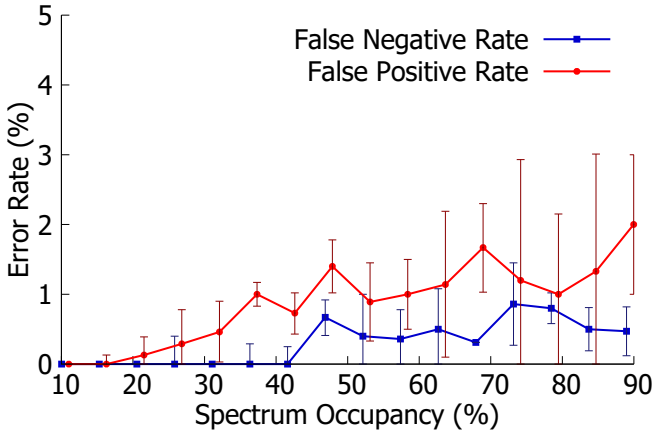


Figure 5.10: False Positives and Negatives as a Function of Spectrum Occupancy (Mixed Modulation Schemes): The figure shows the false positive rate (FPR) and false negative rate (FNR) of S^3 as the spectrum occupancy increases when modulation schemes are randomly picked by transmitters.

- *False Negative Rate (FNR)*: Percentage of occupied bands that S^3 incorrectly reports as empty.
- *True Positive Rate (TNR)*: Percentage of occupied bands that S^3 correctly reports as occupied.

Figure 5.10 and Fig. 5.11 shows S^3's error rate in detecting occupied bands as we vary the total occupancy of the spectrum between 10% and 90%. Figure 5.10 shows the results when the transmitters randomly pick a modulation scheme (e.g. single carrier BPSK, QAM, or OFDM). In this case, when the total occupancy of the spectrum is less than 30%, S^3 achieves a median false positive rate (FPR) less than 0.5% and a median false negative rate (FNR) of 0%. As the total occupancy increases and the spectrum becomes more crowded, the FPR and FNR gradually increase. However, even when the spectrum is extremely crowded ($\sim$ 90% occupied), S^3 can still achieve 2% median FPR and 0.47% median FNR.

Figure 5.11 shows the same results when the transmitters only use OFDM modulation. In this case, the FPR and FNR become even smaller at all levels of occupancy with a maximum median FPR of 0.6% and a maximum median FNR of 0.25%. This result can be attributed to two factors: (1) OFDM signals have flat power spectral densities. Therefore, signal power detected in the spike train can more accurately reflect the signal presence in the corresponding channels. (2) Single carrier modulation schemes have

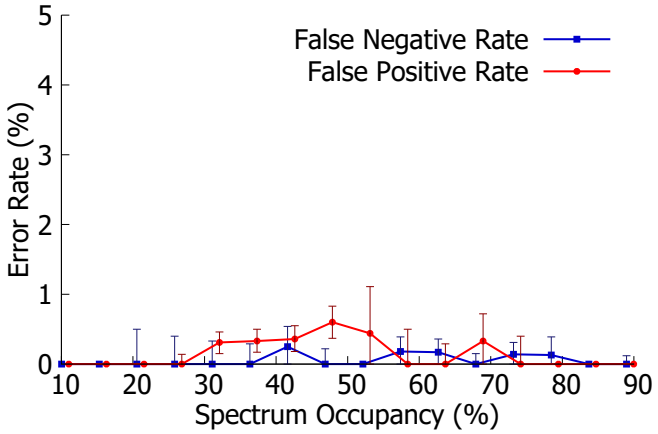


Figure 5.11: False Positives and Negatives as a Function of Spectrum Occupancy (OFDM Modulation): The figure shows the false positive rate (FPR) and false negative rate (FNR) of S^3 as the spectrum occupancy increases when transmitters only use OFDM modulation.

lower spectral efficiency and leak power outside their bands, which leads to a higher FPR as can be seen from Fig. 5.10. Finally, Fig. 5.12 shows the receiver operating characteristic (ROC) curve, which demonstrates the trade-off between false positives and false negatives as we vary the threshold for detecting occupied band.

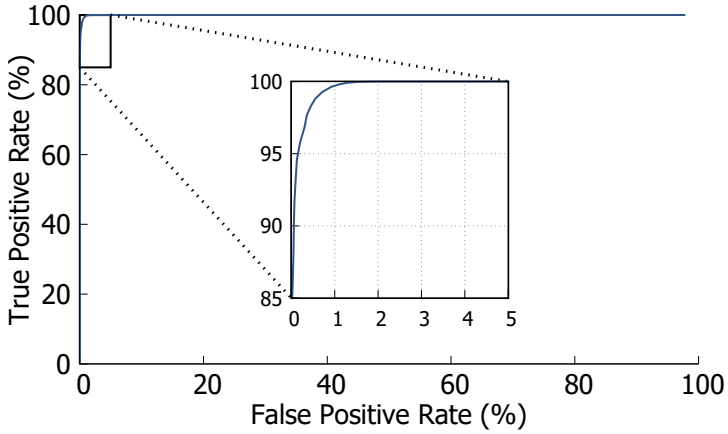


Figure 5.12: ROC Curve for Selecting Power Threshold: The figure shows the Receiver Operating Characteristic (ROC) curve we used to select the power threshold for deciding occupied and idle bands in the experiments where the modulation schemes are randomly picked by the transmitters.

Table 5.1: Sum of false positives and negatives for S^3 and State-of-the-Art prior work: The table compares the sum of FPR and FNR of S^3, BigBand, D-BigBand, and SweepSense at different spectrum occupancies.

	BigBand	D-BigBand	SweepSense	S^3
10%	0.38% + 14%(unresolved)	$\sim$ 0.95%	4.88%	0.00%
50%	N/A	$\sim$ 1.75%	13.09%	1.29%
90%	N/A	$\sim$ 3%	13.76%	2.47%

5.7.3 Comparison with State-of-the-Art

We compare S^3 with three baselines from prior work:

- **BigBand:** [155] leverages sparse Fourier transform and uses co-prime sampling to acquire sparse spectrum. It achieves 6× sub-sampling below the Nyquist rate, but only works when the spectrum is sparse.

- **D-BigBand:** [163] extends BigBand to sense dense spectrum by considering the differential changes in occupancy. It also achieves 6× sub-sampling, but assumes the changes in the spectrum occupancy over time are sparse.

- **SweepSense:** [158] enhances USRP software-defined radio's ability to quickly scan and sense wideband spectrum. It is able to scan 5 GHz bandwidth in 5 ms with 2×25 MS/s ADC sampling rate.

Table 5.1 shows the sum of FPR and FNR when the total spectrum occupancy is 10%, 50%, and 90%. We compare S^3 directly to the results reported in [155] and [163], because they used custom hardware but were evaluated using the same metrics as ours. One can see that in sparse spectrum (¡10% occupied) where BigBand works, BigBand has a total error rate of 0.38% but still cannot recover the status of 14% of the spectrum. In contrast, S^3 accomplishes a 0% error rate at such low spectrum occupancy and samples 8.5× below the Nyquist rate, which exhibits a 1.4× gain over BigBand.

D-BigBand is able to work in densely occupied spectrum. It has a total error rate of 0.95% and 3% when the spectrum is 50% and 90% occupied respectively. However, S^3 is able to outperform D-BigBand at all occupancy levels with a 1.2× to 1.35× gain in accuracy. Moreover, S^3 also achieves 1.4× gain in sampling rate reduction and makes no assumptions on the changes in spectrum occupancy. Therefore, unlike D-BigBand, S^3 can monitor highly dynamic spectrum, which we will demonstrate later in this section.

To compare S^3 to SweepSense, we reproduce SweepSense on a N210 USRP

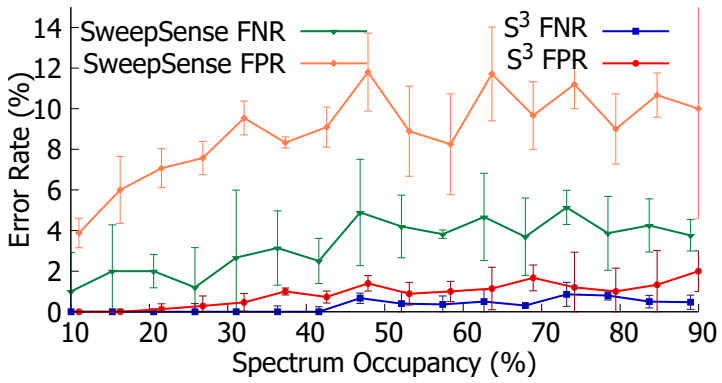


Figure 5.13: False positives and negatives comparison between S^3 and SweepSense

with a CBX daughterboard using the codes and FPGA images released by the authors. We use SweepSense to sense the spectrum generated by our testbed along with S^3. In Fig. 5.13, we show SweepSense's error rate in detecting occupied bands as we vary the spectrum occupancy between 10% and 90% and when the modulation scheme is randomly picked by the transmitters. This result shows that SweepSense can work in densely occupied spectrum. When the spectrum is 10% occupied, SweepSense achieves an FPR of 3.88% and an FNR of 1%. As the spectrum becomes more crowded, the FPR and FNR of SweepSense increase, but they remain below 10% and 3.76% respectively even if the spectrum is 90% occupied. SweepSense's higher error rates are likely due to the fact that fast LO sweeping can smear non-empty frequency components, resulting in more leakage from the occupied bands to the adjacent bands, which increases its false positive rates. We also note that, SweepSense requires accurate phase information for the digital chirp demodulation, so it is sensitive to the IQ imbalance in the hardware, which is likely why it underperforms S^3. However, SweepSense is highly valuable as we can combine it with S^3 to capture the power spectral density as we show later in this section.

Next, we present some microbenchmarks that provide more insights into the working of S^3 and its performance.

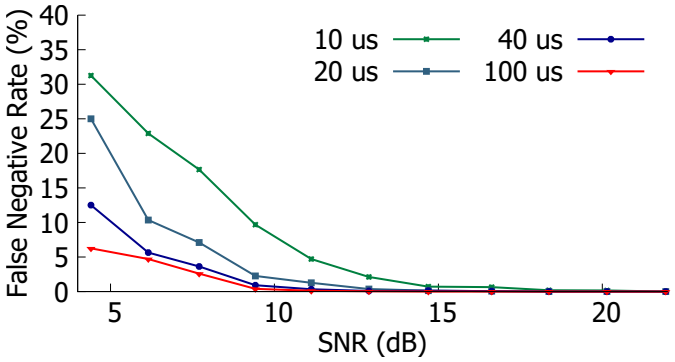


Figure 5.14: FNR vs SNR and Sampling Time

5.7.4 Microbenchmark - Sensitivity

To understand the ability of S^3 to detect low signal-to-noise ratio (SNR) signals, we examine the FNR of bands with different SNRs. The SNR we show is the average signal power per Hz of RX signal / noise floor. We compare four different sampling duration: 10, 20, 40, and 100 μs. The FNR is high when the SNR is low; however, this can be addressed by increasing the sampling duration. In fact, we can reduce the FNR by 5× at 3dB SNR. As the SNR gets higher, the FNR goes down and down, and eventually even for shoft sampling windows, the FNR is very low ($\approx 0\%$). Note that $40 \sim 100$ μs is a short enough window to detect short transient packets and fleeting signals, as it is comparable to the DIFS duration for Wi-Fi carrier sensing (e.g. 34 or 50 μs).

5.7.5 Microbenchmark - Dynamic Range

Here we evaluate the dynamic range of S^3, which is the ratio between the strongest and weakest signal powers S^3 can accurately detect at the same time. It reflects the ability of S^3 to detect low-power signals with the presence of much higher power signals that would cause interference and lower the signal-to-noise-pluse-interference ratio (SINR), making low-power signals harder to be detected. In Figure 5.15, we compare the FNRs in experiments with different dynamic ranges. One can see that S^3 achieves very low FNR ($< 0.63\%$) when the power difference between the occupied bands is up to 15 dB. As the signal power difference becomes even larger, the FNR of S^3 increases. When the spectrum dynamic range reaches ~ 21 dB, the FNR of

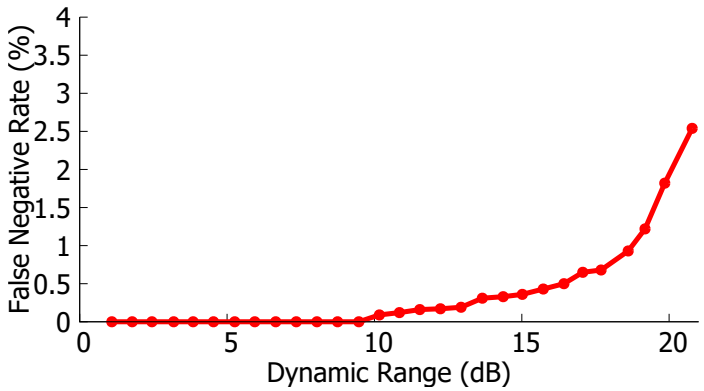


Figure 5.15: FNR vs Dynamic Range

S^3 is 2.54%. This result shows that S^3 can accurately detect (FPN$<$ 1%) the relatively weak signals under interference from signals 19 dB stronger. Note that, after wire-bonding, the spike-train filter experiences degradation in the out-of-band suppression due to the direct leakage from the input port to the output port of the PCB. Hence, the sensitivity and dynamic range of S^3 also degrade, but this issue can be resolved by better isolation in the PCB design.

5.7.6 Microbenchmark - Resolving Collisions with Voting

We want to verify that through voting using two different sampling rates, S^3 can effectively resolve frequency collisions. To this end, we compare S^3 to baselines where we detect the spectrum occupancy using only one ADC. In Fig. 5.16, we qualitatively compare the correctness of occupancy detection on each band in 20 randomly selected experiments. It shows that when using either ADC alone, we have many false positives due to frequency collisions. However, the two ADCs exhibit false positives in different bands, because they experience frequency collisions between different spikes. As a result, through voting S^3 is able to distinguish and resolve false positives where the two sampling rates disagree with each other. Furthermore, we also quantitatively show the FPR of S^3 and baselines. As can be seen from Fig. 5.17, the FPR of S^3 is much lower than those of baselines, which suggests that our voting scheme can effectively leverage the different sampling rates to resolve frequency collisions. Note that ADC1 outperforms ADC2. This is expected because, as we discussed in section 5.5, 45.5 MHz is an optimized sampling

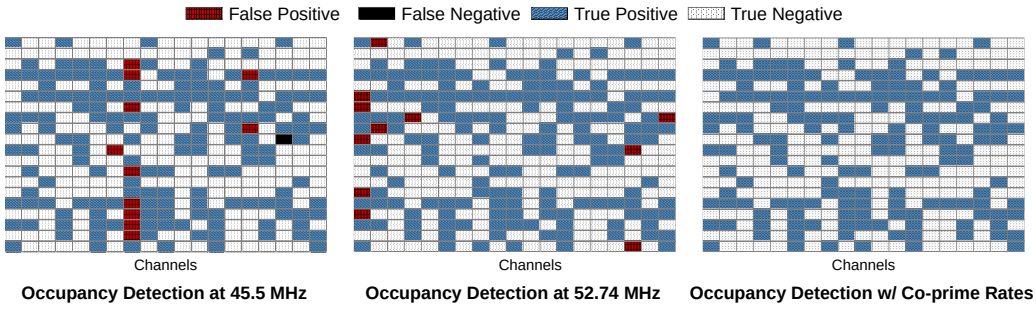


Figure 5.16: False positives comparison between S^3 and baselines that use single sampling rate: the spectrum occupancy detected by S^3 and the baselines in 20 randomly selected experiments.

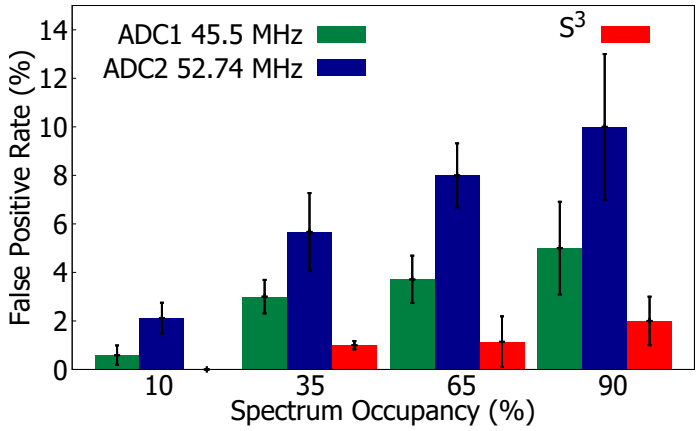


Figure 5.17: False positives comparison between S^3 and baselines that use single sampling rate: the false positive rate comparison as the spectrum occupancy increases.

rate that can spread out the spikes and minimize the frequency collisions. In contrast, the second sampling rate of 52.74 MHz is optimized to avoid having the same collisions as ADC1, so it does not work as well by itself.

5.7.7 Monitoring Dynamic Spectrum

S^3 senses all bands in the spectrum in real-time and makes no assumptions on the changes of spectrum occupancy, so it can monitor highly dynamic spectrum with rapidly-changing occupancied bands. To evaluate this ability of S^3 we create a rapidly-changing spectrum in our testbed whose occupied bands change every 327 μs, and as a result, the total spectrum occupancy varies between 0% and 63%. We use S^3 to continuously monitor the occu-

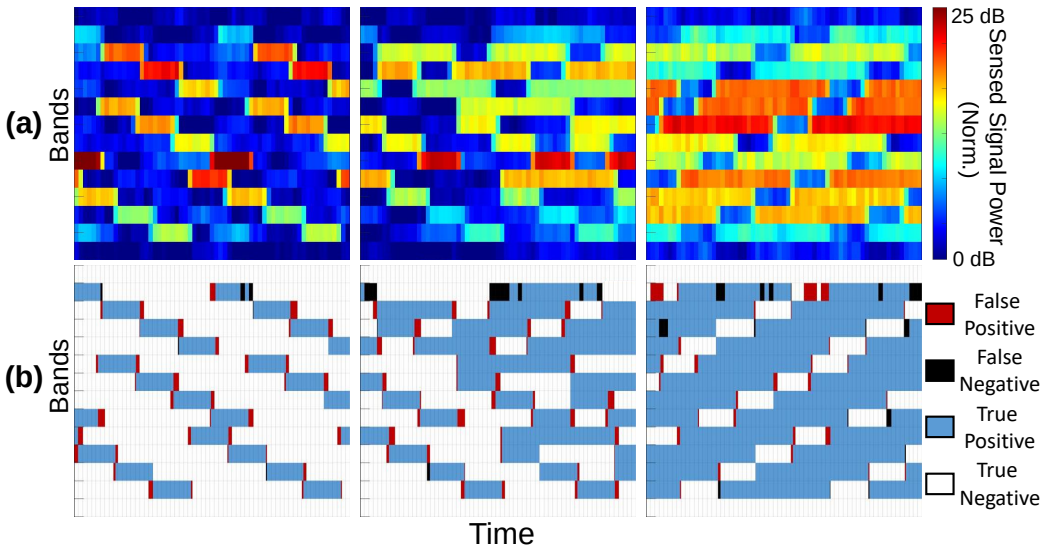


Figure 5.18: Monitoring rapidly-changing spectrum: The figure shows (a) spectrogram (b) spectrum occupancy captured by S^3 in real time.

pancy changes in the spectrum, and output a signal power estimation and occupancy detection for every 76 μs-long frame. We show a spectrogram captured by S^3 consisting the signal power detected in every band per frame in Fig. 5.18(a). Furthermore, we show the accuracy of the corresponding occupancy detection per frame in Fig. 5.18(b). It shows that S^3 is able to capture the occupancy of rapidly-changing spectrum with great accuracy and time precision.

5.7.8 Capturing Wideband Power Spectral Density

As we mentioned in section 5.6, we can sweep the LO frequency of S^3 over the 22 MHz spacing between spikes to sense all the frequencies in the spectrum. This enables S^3 to capture the power spectral density (PSD) of the entire wideband spectrum. At every LO frequency, S^3 captures signal power in the spike train and identifies the occupancy of each spike. For the occupied spikes, S^3 uses the signal power estimates in them to reconstruct the PSD at the corresponding RF frequencies. When LO sweeping finishes, all frequencies on the wideband spectrum will be reconstructed. Comparing to conventional spectrum scanners, this extended S^3 prototype only needs to sweep a much narrower frequency range. Therefore, the scanning time is much shorter.

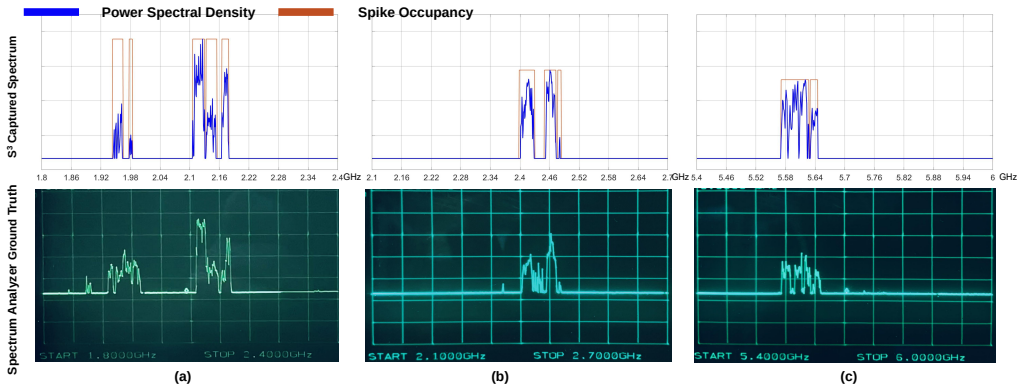


Figure 5.19: Wideband Power Spectral Density Capture (Testbed): The figure shows the wideband power spectral density captured by the extended S^3 prototype. Spectra are generated by our evaluation testbed.

Figure 5.19(a-c) shows the PSD captured by the extended S^3 prototype of spectra generated by our testbed, along with the detected spike occupancy. We use S^3 and an HP 8563E Spectrum Analyzer to monitor the 1.8 to 2.4 GHz spectrum simultaneously. As one can see, the PSDs captured by S^3 match the ground truth from the spectrum analyzer very well. Besides, we also measure PSDs of real-world spectra, both outdoors and indoors, which are shown in Figure 5.20(a-c). Figure 5.20(a) shows the spectrum between 1.8 and 2.4 GHz measured outdoor at our geographical location. It shows that S^3 is able to capture the PSD of 4G LTE signals in Band 2 and 66. In Fig. 5.20(b) and (c), we show the PSD of 2.4 GHz and 5 GHz Wi-Fi signals captured by S^3 respectively. One can see that Channel 1 and 11 in the 2.4 GHz band as well as four non-overlapping 20 MHz channels (Channel 116, 120, 124, and 128) from 5.57 to 5.65 GHz in the 5 GHz band are being used. Figure 5.20(a-c) demonstrate that the real-world PSDs captured by S^3 also closely match the spectrum analyzer ground truth. On some frequencies that S^3 classifies as empty, the spectrum analyzer shows some non-zero spectral components. However, this is expected because in our experiments, the spectrum analyzer takes the maximum over a lot more scans than S^3.

5.8 Extending the Prototype

The basic prototype we built using only one spike-train filter can be extended to sense spectra with different center frequency, bandwidth, and channel

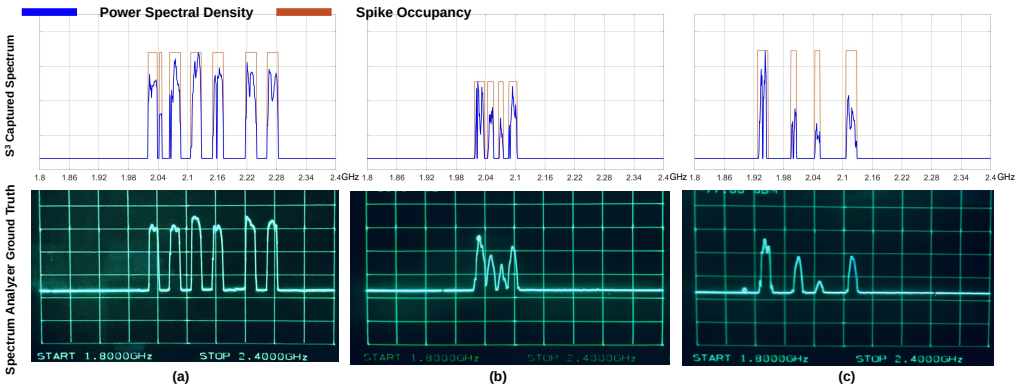


Figure 5.20: Wideband Power Spectral Density Capture (Real-World): The figure shows the wideband power spectral density captured by the extended S^3 prototype. Spectra are real-world spectra captured both outdoor and indoor.

allocation. Moreover, system level parallelism introduces another degree of freedom and allow us to break the fixed design trade-offs at the filter level.

- **Different Spectrum:** By changing the LO frequency as well as the RF bandpass filter and LNA, we can sense different frequency ranges. In our evaluation, we test at center frequencies of 2.1, 2.4, 4.9, and 5.7 GHz.

- **Larger Bandwidth:** The current spike-train filter supports a bandwidth of 418 MHz. We can extend S^3 to larger bandwidth by either using two sensors and configuring them to sense adjacent spectra or by using two MEMS filters in parallel channels before combining the signals and sampling it.

- **Narrower Bands:** The spikes in the spike-train filter are separated by 22 MHz. Hence, narrowband signals ($<$ 20 MHz) that are not aligned with the spikes might be filtered out. To address this, we can combine frequency domain sampling with LO frequency sweeping over 22 MHz to capture and sense all the frequencies in the spectrum as shown in our results in section 5.7. Alternatively, we can design a MEMS filter with narrower spacing or use two MEMS filters and set the center frequency to be slightly different.

5.8.1 Changing LO Center Frequencies

First, we can combine two identical spike-train filters that are fabricated using the same process. Hence, these two filters will have almost identical

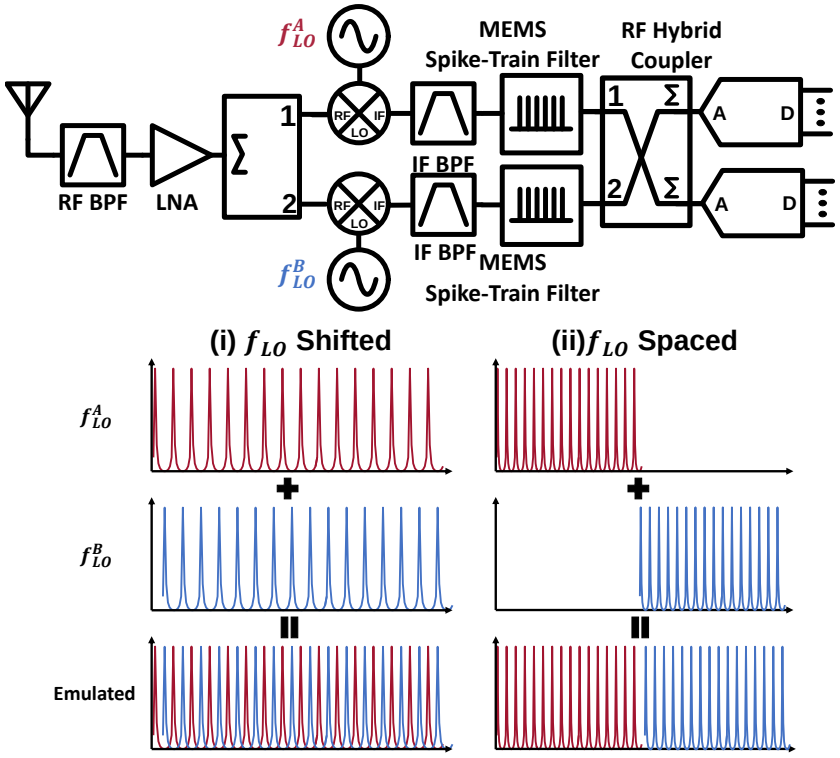


Figure 5.21: Circuit diagram and emulated spike-train filter for alternative S^3 architecture leveraging LO center frequency difference.

frequency responses with the same center frequency f_c and spacing between spikes Δf. In order to cover different frequencies in the RF spectrum, we use the two filters on separate receiver RF chains with different LO frequencies. We demonstrate the circuit diagram and the emulated spike trains in the RF spectrum in Fig. 5.21.

After bandpass filtering and amplifying the received signal, we split the RF signal into two channels, and use two LOs with center frequencies f_{LO}^{A} and f_{LO}^{B} to down-convert the signal to the IF frequencies. Then we pass each IF signal into a spike-train filter to sampling the spectrum along the frequency axis. Based on the LO frequency difference $df_{LO} = f_{LO}^{B} - f_{LO}^{A}$, we can emulate two types of spike trains, as shown in Fig. 5.21(i) and (ii). When $\Delta f_{LO} < \Delta f$, the two spike trains are slightly shifted on the frequency axis as shown in Fig. 5.21(i). As a result, we can emulate a spike-train filter with narrower spacing between the spikes. This increases the frequency domain sampling rate of the filter and enables §3 to sense narrower channel bandwidths. On the other hand, when $df_{LO} = K\Delta f$, the two spike trains are concatenated along the frequency axis as shown in Fig. 5.21(ii). In this way, a longer spike

train with more spikes covering wider bandwidth is emulated.

Although it is straight forward to sample the two IF signals separately, the number of ADCs required will increase linearly with the number of spike-train filters. Instead, after passing IF signals on the two channels through the spike-train filters, we combine the filtered signals and sample the combined signal using two low-speed ADCs. The analog combination and splitting can be achieved using an RF power combiner in series with an RF power splitter, but it can also be done using a RF hybrid coupler. Note that with more spikes in the emulated filter, there will be more aliasing in the sub-sampled spectrum. Therefore, a higher ADC sampling rate might be needed, but the sampling rate should be able to scale sublinearly with respect to the number of spikes. Besides, the ADC input cutoff frequency needs to be higher than the spike-train bandwidth.

The advantage of this architecture is that we can use the same MEMS spike-train filter on the two channels without needing to redesign a new filter. However, it requires two LOs and mixers which increases the cost and power consumption of the system. [4]

5.8.2 Changing Spike-Train Filter Structure

Instead of introducing a second local oscillator, we can use only one LO and two different spike-train filters to emulate spike trains with wider bandwidth as well as narrower or nonuniform spike spacing. As we mentioned in section 4.2, we can modify the width of the piezoelectric film and the position of the electrodes to obtain different Δf and f_c. When two spike-train filters with different Δf and/or f_c are combined in parallel, we can emulate a spike train with more sophisticated sparsity structures. We show the circuit diagram for this type of alternative architectures in Fig. 5.22, along with three emulated spike-train filter frequency responses.

In this architecture, the down-converted IF signal is split and filtered by two different MEMS spike-train filters, whose center frequencies and spike spacing are f_c^A, f_c^B, and Δf^A, Δf^B. The output spectra of the two filters are then combined and sampled by two low-speed ADCs. Using this architecture, we can emulate the same spike trains as the first alternative architecture. For

[4]Since the power consumption of spectrum sensors is dictated by the ADC [148], the additional power consumption of the second LO is not the primary concern.

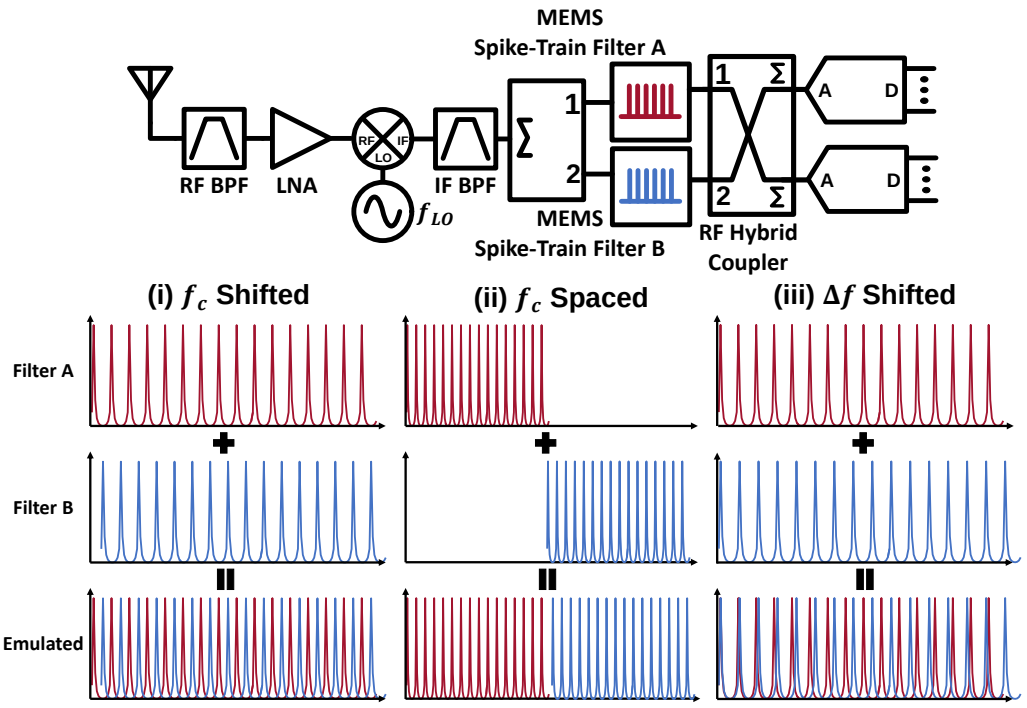


Figure 5.22: Circuit diagram and emulated spike-train filter for alternative S^3 architecture leveraging different MEMS spike-train filters.

instance, when the difference between the filter center frequencies $df_c = f_c^B - f_c^A < \Delta f$, as shown in Fig. 5.22(i), the two spike trains are slightly shifted on the frequency axis and emulate a spike-train filter with narrower spacing between the spikes. Besides, when $df_c = K\Delta f$, the two spike trains are spaced by the bandwidth of the spike train and emulate a wider bandwidth spike train as shown in Fig. 5.22(ii). However, in additional to enlarging the filter bandwidth and narrowing the spike spacing, we can even emulate a non-uniformly spike train as as shown in Fig. 5.22(iii). This is achieved by combining two spike-train filters with different Δf^A and Δf^B. Such spike train profile provides us with all sorts of frequency resolutions across the spectrum to accommodate the different channel bandwidth required by the secondary users in TV Whitespace and CBRS bands.

5.9 Related Work

In this section, we provide more related work. For further background, we refer the reader to section 5.2.

Spectrum sensing has been extensively studied in the past two decades [168, 169, 170]. However, most of this work focuses on narrowband sensing [171, 172, 173, 174, 175, 176, 177, 178]. In contrast, this chapter focuses on wideband spectrum sensing to enable dynamic spectrum sharing of many channels. Several systems attempt to sense wideband spectrum using narrowband sensors without sequentially scanning each band [179, 180, 181]. QuickSense [179] leverages analog filters and energy detectors to hierarchically sense wide bandwidth by detecting the total signal power in groups of consecutive channels. However, when the spectrum is densely occupied, QuickSense's approach reduces to sequentially scanning the spectrum. SpecInsight [181] leverages machine learning to predict spectrum occupancy based on learned utilization patterns and optimize which channels to sense. Similarly, Su and Wu. [180] use time-series analysis to predict which bands are occupied. However, these systems are sensitive to training data and assume that transmissions follow predictable patterns. Spectrum sharing is based on opportunistic access and as a result is highly dynamic and unpredictable [182].

Recent work aims to enhance USRP software-defined radio's ability to sense wideband spectrum [158, 183]. SweepSense [158] enables sensing wideband spectrum by quickly sweeping the center frequency of the USRP. It is able to sweep 5 GHz bandwidth in 5 ms, which offers great potential for sensing an extremely wideband spectrum on commercial software radios. However, SweepSense requires accurate phase information for the digital chirp demodulation and is sensitive to the IQ imbalance. As a result, our comparison with SweepSense in section 5.7 shows that it can suffer from a high error rate especially when the spectrum is not sparse. SparSDR [183] reduces the backhaul and computation requirements for sensing sparse spectrum on USRPs, which offers great utility for continuously monitoring underutilized spectra but cannot scale to densely occupied spectrum.

The use of single passband MEMS filters in spectrum sensing has been studied [184, 185]. However, these techniques require an array of channel-select MEMS filters to form a reconfigurable filter bank. In contrast, S^3 only

uses a single MEMS spike-train filter that consists of overtone resonators.

Our work is also related to theoretical work on co-prime sampling [186, 187, 188] and multicoset sampling [189, 190] of sparse wideband spectrum. These approaches also do not work for densely occupied spectrum. Moreover, Xia [186, 187] require using k ADCs where k is the number of occupied frequencies. Herley et al. [189] and Venkataramani et al. [190] aim to recover the signals in each occupied band and must assume prior knowledge of which bands are occupied. In contrast, S^3 aims to recover the occupancy of each band and uses 2 ADCs irrespective of the number of occupied frequencies. S^3 is further implemented and shown to work in practice.

Sub-Nyquist sampling has been used for test equipment to reconstruct wideband periodic signals [191, 192]. However, these techniques require the signal to be periodic and repeat for a long time in order to take on samples during each period until all samples are recovered. Hence, these techniques are not applicable to real communication signals where the signal is constantly changing and carries different modulated bits.

Finally, some works aim to capture spectrum usage at large geographical and time scales through crowdsourcing [193, 194, 195]. S^3 is complementary to these works, as it enables real-time wideband occupancy detection of every single sensor with minimum data size and computational complexity.

5.10 Limitations

In this section, we discuss some limitations of S^3.

- The frequency-domain sampling rate and maximum sensing bandwidth is limited by the filter design trade-offs. As a result, narrowband signals (< 20 MHz) and over GHz-wide spectrum cannot be sensed using a single spike-train filter. This can be resolved by hopping the LO frequency as shown in section 5.7. Alternatively, we can also use the extended architectures proposed in section 5.8 that combines multiple spike-train filters in parallel to break the fixed filter-level design trade-offs.

- Sub-Nyquist sampling leads to aliasing of both signals and noise, which typically lowers the signal SNR and degrades the spectrum sensor's sensitivity. To minimize the loss of SINR, we design the spike-train filters to

have low insertion loss and high out-of-band suppression i.e., most of the noise is filtered out before it aliases. Moreover, instead of detecting signal power, known signals like the preambles can be leveraged to improve the sensitivity [196]. However, directly applying this technique to S^3 would require further research as the preambles might become corrupted after applying the filter.

- While S^3 can detect the occupancy of the different bands and reconstruct the power spectral density of the spectrum, it cannot recover complex I and Q samples of the signal. As results, S^3 cannot reconstruct the signal itself or decode the data in the signal.

5.11 Conclusion

This chapter presents S^3, a new efficient real-time wideband spectrum sensing mechanism that can work in densely occupied spectrum. S^3 monitors only a small fraction of bandwidth in each band to accomplish significantly below-Nyquist sampling and, hence, great energy efficiency. It leverages recent advances in RF MEMS filtering solution that enables sampling the spectrum along the frequency axis. Empirical evaluation demonstrates that S^3 can accurately sense densely occupied spectrum and rapidly-changing spectrum; we also show that S^3 can be extended to capture the power spectral density of the entire spectrum. We believe S^3 can enable dynamic spectrum access and very high spectrum utilization.

Chapter 6

Enabling IoT Self-Localization Using Ambient 5G Signals

This chapter presents *ISLA*, a system that enables low power IoT nodes to self-localize using ambient 5G signals without any coordination with the base stations. *ISLA* operates by simply overhearing transmitted 5G signals and leverages the large bandwidth used in 5G to compute high-resolution time of flight of the signals. Capturing large 5G bandwidth consumes a lot of power. To address this, *ISLA* leverages recent advances in MEMS acoustic resonators to design a RF filter that can stretch the effective localization bandwidth to 100 MHz while using 6.25 MHz receivers, improving ranging resolution by 16×. We implement and evaluate *ISLA* in three large outdoors testbeds and show high localization accuracy that is comparable with having the full 100 MHz bandwidth.

6.1 Introduction

Recent years have witnessed a tremendous growth in the number of connected IoT devices, with surveys projecting up to 31 billion deployed IoT nodes by 2030 [197]. With such ubiquitous deployment of IoT nodes, the ability to localize and track these nodes with high accuracy is essential for many applications. For example, in data driven agriculture, it can enable real time micro-climate monitoring and livestock tracking [198]. In smart cities, IoT sensors are deployed throughout the city for tasks such as air quality monitoring, tracking buses, trains, and cars, and monitoring the structural health of infrastructure [199]. In the era of Industry 4.0, it can also enable wide area inventory tracking and facilitate factory automation [200].

Today, the most prevalent outdoor localization technology is GPS which is mainly used in cars and mobile phones. However, off-the-self GPS chips can consume about the same power as the entire IoT device, thus reducing the

battery life to half in addition to the extra hardware costs [3]. Due to this, past work has proposed the use of cellular networks or dedicated IoT base stations for localization [1, 201]. These solutions, however, either achieve very low resolution of 100s of meters [1, 202] or require active participation of the base stations to jointly compute the location or tightly synchronize the base stations [201, 5, 203]. Realizing such solutions in practice requires the cooperation of cellular providers to bear the additional cost of modifying the base stations and a back end server to support the localization feature.

In this chapter, *we ask whether an IoT device can accurately localize itself simply by listening to ambient 5G cellular signals, without any coordination with the 5G base stations?* Doing so would allow us to easily deploy self-localizing IoT nodes is wide areas without the need to modify the cellular base stations or deploy new base stations for localization.

5G cellular networks present unique opportunities for enabling accurate localization. First, the small cell architecture in 5G networks will lead to a very high density of 5G base stations, with up to 40 to 50 base stations deployed per square km [204], thereby allowing us to leverage more anchor points in the network for increased localization accuracy. Second, the 5G standard is designed to support very high data rates and can have OFDM signals spanning up to 100 MHz in bandwidth in the sub-6 GHz frequency range, and up to 400 MHz bandwidth in the mmWave frequency range [205]. Such large bandwidth can be used for accurate localization. To see how, consider the 5G OFDM signal shown in Fig. 6.1(a) where data bits are encoded in N frequency subcarriers. We can use the preamble which contains known bits to compute the channel impulse response (CIR) by taking an inverse FFT. The CIR in Fig. 6.1(a) shows the Time-of-Flight (ToF) of different signal paths. Estimating the ToF from few base stations allows us to localize the device. The larger the bandwidth of the signal, the higher the resolution. In fact, we can achieve a resolution of 3 meters for 100 MHz and 0.75 meters for 400 MHz signals.[1]

Leveraging these opportunities, however, is challenging since power-constrained and low-cost IoT nodes cannot capture the large bandwidth of the 5G signals. They are equipped with low-power and low-speed Analog-to-Digital Converters (ADCs) that can only capture a narrow bandwidth. In fact, while IoT

[1]The resolution is computed as c/B where c is the speed of light and B is the bandwidth of the signal.

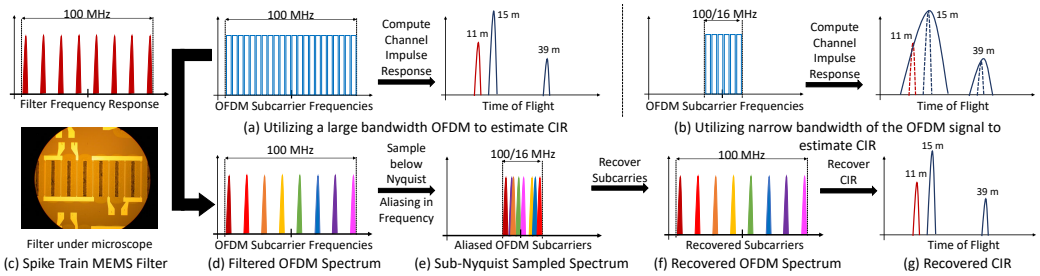


Figure 6.1: *ISLA*'**s pipeline:** (a) wideband OFDM signal and its corresponding CIR. (b) narrowband OFDM signal and its corresponding lower resolution CIR. (c) *ISLA*'s spike train MEMS filter that sparsifies the wideband signal. (d-f) follow the signal journey through *ISLA*'s pipeline that recovers the original CIR.

has been one of the cornerstone applications in the design of 5G, it is only supported in narrowband chunks for low data rate applications [206, 207]. Therefore, while the 5G standard does allocate higher bandwidth (up to 400 MHz) for mobile broadband and high data rate applications, IoT nodes can capture only a very small fraction of this bandwidth ($\sim 20\times$ smaller [205]). As a result, they significantly lose out on the ToF resolution that was made possible by the high bandwidth 5G signals as shown in Fig. 6.1(b). Moreover, it is infeasible to measure the absolute time-of-flight without any coordination or synchronization with the base stations.

In this chapter, we present *ISLA*, a system that enables **I**oT **S**elf-**L**ocalization using **A**mbient 5G signals. *ISLA* does not require any coordination with or modifications to the base stations. The key enabler of *ISLA* is the use of MEMS acoustic resonators. Past work [132] has demonstrated that we can use such MEMS resonators to design new kinds of RF filters that look like a spike-train in the frequency domain, as shown in Fig. 6.1(c). To understand how we can leverage such MEMS spike-train filters, consider the 5G OFDM signal shown in Fig. 6.1(a). Passing this signal through the filter allows us to keep a few subcarriers of the wideband OFDM symbol while suppressing all other subcarriers as shown in Fig. 6.1(d). There are two important features of the resulting signal: (1) Since the remaining subcarriers that are passed by the filter span the entire wideband, we should, in principle, be able to recover the channel impulse response at the same high resolution of the original signal. (2) Since the remaining subcarriers create a sparse signal in the frequency domain, it should be possible to recover these subcarriers by sampling the signal below the Nyquist sampling rate using the same low-power

low-speed ADCs on the IoT nodes.[2]

However, recovering the channel impulse response from a signal sampled with the low-speed ADCs is non-trivial. First, sampling the signal below the Nyquist rate leads to aliasing in the frequency domain as shown in Fig. 6.1(e). Some subcarriers might collide by aliasing on top of each other making it hard to recover these subcarriers. Past work in sparse recovery addresses this problem by using two co-prime subsampling rates [155]. Unfortunately, we do not have the flexibility to choose co-prime subsampling factors. In fact, since the number of OFDM subcarriers in the 5G standard is a power of 2 (e.g. 1024, 2048, 4096), we can only subsample the signal by powers of 2 otherwise the values of the subcarriers will be corrupted as we prove in section 6.5.[3] To address this, we carefully co-design the MEMS hardware with the recovery algorithm. In particular, we jointly optimize the filter shape (spacing between peaks, width of each peak, frequency span) with the subsampling rate to minimize the number of colliding OFDM subcarriers as we describe in detail in section 6.5.

Second, the recovered OFDM subcarriers are not uniformly distributed across the wideband bandwidth. This is because non-idealities in the MEMS filter make it hard to design a uniform spike train like the one shown in Fig. 6.1(c). As a result, we can no longer recover the CIR using standard super-resolution algorithms like MUSIC with spatial smoothing [208, 4] as they require uniform measurements. Instead, we formulate an inverse optimization problem that accounts for non-idealities and optimizes the CIR in the continuous time domain to achieve super resolution as described in section 6.5.

Finally, while the above can provide very precise ToF measurements, these ToF estimates are not going to capture the true time taken by the signal to travel between the base station and the IoT device. This is because the 5G base stations are not time-synchronized with each other or the IoT device. To localize the device without any synchronization with the base station, *ISLA* leverages a second antenna on the receiver to compute the differential ToF of the propagation paths. While the absolute ToF measurements are corrupted by synchronization offsets, these offsets are constant across the 2 antennas on

[2]Note that the MEMS filter is passive and does not consume any power.

[3]For example, for a 100 MHz OFDM signal, we can only sample at 50 MS/s (2×), 25 MS/s (4×), 12.5 MS/s (8×), 6.25 MS/s (16×), ...

the IoT node, and hence can be eliminated by subtracting the measurements from the 2 antennas. Using this differential ToF at the IoT receiver, we show in section 6.7 that with measurements from four or more base stations, the IoT device can localize itself regardless of its orientation. We integrate our approach into a full system that addresses additional system challenges such as figuring the base station ID and accounting for carrier frequency offsets.

Evaluation: We implemented and evaluated *ISLA* indoors for microbenchmarks and outdoors for overall localization performance. We ran experiments in three outdoor settings:(1) Between campus buildings (52 m×85 m), (2) a large parking lot (240 m×400 m), and (3) an agricultural farm (480 m×860 m). We use USRP X310 radios as base stations that can transmit high-bandwidth packets of 100 MHz. Our custom IoT nodes are equipped with 2 antennas and subsample the 5G signals at 6.25 MS/s which is 16× below the Nyquist rate. We fabricated a MEMS spike-train filter operating at a center frequency of 400 MHz and used it to demonstrate accurate reconstruction of the channel impulse response. However, due to significant interference at the 400 MHz band outdoors in our city, we ran experiments at 1 GHz and applied the filter response in digital. Our results reveal that with 5 base stations in range, *ISLA* can achieve a median accuracy of 1.58 m on campus, 17.6 m in the parking lot, and 37.8 m in the farm where the IoT node can be as much as 500 meters away from most base stations. For the parking lot testbed, the accuracy improves to 9.27 m with 15 base stations and 4.26 m with 25 base stations in range. We compare *ISLA*'s localization approach with several baselines [208, 209, 1] and show up to 4−11× higher localization accuracy. Finally, we show that *ISLA* achieves a comparable performance to having a full 100 MHz receiver while using a 16× lower sampling rate.

Contributions: We make the following contributions:

- We present, to the best of our knowledge, the first system that allows IoT nodes to localize themselves using ambient 5G signals without any coordination with the base stations.
- We demonstrate the ability to reduce the sampling rate by 16× while retaining the benefits of high bandwidth 5G signals by leveraging recent advances in MEMS RF filters.
- We implement and evaluate *ISLA* to demonstrate accurate localization in 3 outdoor settings.

6.2 Related Work

Localization has been extensively studied in cellular, WiFi, and IoT networks. Our work differs from past research in that it is the first to enable self-localization using ambient 5G signals without requiring coordination with the base stations.

6.2.1 Cellular Based Localization

Several studies [210, 202, 1, 2, 211] have proposed to use nearby cell tower information and statistics in order to localize a mobile device. These methods, however, have a median accuracy of around 100 to 500 meters and are mostly useful for very coarse localization. To improve localization accuracy, Aly et al. [212] and Shokry et al. [213] propose to combine WiFi APs with cellular base stations. Despite their relatively higher accuracy, these methods require fingerprinting the surroundings and as such require extensive training and do not generalize to new locations. More recent work exploits massive MIMO and millimeter wave for localization in 5G [214, 215, 216]. However, all of this work requires coordination with base stations and assumes the devices can capture the entire bandwidth of the 5G signals which does not work for IoT devices.

6.2.2 IoT Based Localization

Bansal et al. [3] leverage TV whitespace to achieve high localization accuracy for LoRA IoT devices. However, it requires all base stations to be tightly synchronized at the physical layer (time and phase) in order to measure TDoA (Time Difference of Arrival). Recent work [201] designs low power backscatter devices that leverage LoRa for localization to achieve high accuracy. However, the system mainly targets indoor applications where software radios can be deployed as base stations to sample the I/Q of the signal and localize the IoT node. Moreover, its current system design [201] supports only a single node. Sallouha et al. [217] propose an outdoors localization technique for SigFox IoT devices based on fingerprinting. However, as mentioned earlier, fingerprinting requires constant training and cannot scale to new environments. Finally, there is a lot of work on using UWB or RFID

nodes for localization [218, 219, 220]. However, these works focus on indoors and short range as the range of UWB and RFIDs is limited to 10-30 meters [221, 222].

6.2.3 IoT Self-Localization

LivingIoT [223] enables self-localization on IoT nodes. It designs a miniaturized device that can be carried by a bumblebee and uses backscatter for communication. The node localizes itself by extracting the angle to the Access Point from the amplitude measurements using an envelop detector. The technique, however, requires the APs to switch the phase across two antennas to change the received amplitude at the IoT node, and hence, cannot be applied to 5G without modifying the base stations. Naderiparizi et al. [224] enable self-localization by placing a camera on a WISP RFID but only operates within a range of 3.6 m from the RFID reader.

6.2.4 WiFi Based Localization

There has been a lot of work on indoor localization using WiFi [5, 208, 225, 226, 227, 228, 229, 4, 209]. The closest to our work are SpotFi [208], Chronos [5], and mD-Track [209], which estimate the channel impulse response (CIR) and time of flight (ToF) from the WiFi access point (AP). Chronos [5] hops between WiFi channels to compute the CIR at high resolution. However, it requires tight timing coordination with the AP to compensate for carrier frequency offset (CFO) and ensure phase coherence across the measurements. *ISLA*, on the other hand, captures measurements from many frequencies across a wideband without hopping by using the MEMS filter, and hence, does not require any coordination with the base stations. SpotFi [208] combines measurements across antennas with large WiFi bandwidth to separate Line of Sight (LoS) path from multipath reflections in the CIR using MUSIC along two dimensions: ToF and Angle of Arrival (AoA). mD-Track [209] also incorporates Doppler shifts and Angle of Departure (AoD) in addition to ToF and AoA and iteratively refines the CIR to achieve a better estimate of the LoS path. In section 6.10, we adapt SpotFi's and mD-Track's CIR estimation algorithms to our setting and demonstrate

that *ISLA*'s algorithm achieves $4-11\times$ higher accuracy. It is worth noting, however, that for our application, these past works cannot benefit from the doppler or AoA/AoD dimensions.

6.3 Background

6.3.1 Wireless Channel Impulse Response (CIR)

The wireless channel can be modeled as the superposition of the signal along all the different paths it takes to travel from the transmitter to the receiver. The channel at frequency f_i can be written as: $h_i = \sum_{l=1}^{L} a_l \exp^{-j2\pi f_i d_l/c}$, where L is the number of propagation paths between the transceivers, d_l is the distance traversed by path l, a_l is the complex path attenuation of path l, and c is the speed of light.

In OFDM systems, data is transmitted over multiple frequency subcarriers $\{f_0, \dots f_{N-1}\}$. If the frequency spacing between these subcarriers is Δf, then the bandwidth spanned by the signal is $B = \Delta f \times (N-1)$. Now, given the channel measurements $\{h_0, \dots h_{N-1}\}$ across these frequencies, the Channel Impulse Response (CIR) can be computed as the inverse FFT of the channel measurements.

$$CIR(\tau) = \sum_{n=0}^{N-1} \Big(\sum_{l=1}^{L} a_l \exp^{-j2\pi \frac{d_l}{c} f_n} \Big) \exp^{j2\pi\tau f_n} \tag{6.1}$$

where $\tau = \{\frac{0}{B}, \dots \frac{(N-1)}{B}\}$ seconds. There are two important things to note here. First, the resolution in Time-of-Flight in the CIR is $1/B$ seconds, that is inversely proportional to the bandwidth B. Hence, larger bandwidth results in higher ToF resolution and more accurate ranging. Second, the maximum unambiguous ToF that can be measured from the CIR is $\frac{(N-1)}{B} = 1/\Delta f$ seconds. This means, if some physical propagation path in the environment has ToF $> 1/\Delta f$ then it would alias and appear at a different tap value in the estimated CIR in Eq. 6.1. For 5G OFDM signal with $B = 100$ MHz bandwidth and $\Delta f = 60$ kHz , we have a resolution of 10 ns (3 meters) and a range of 16.6 μs (5 km).

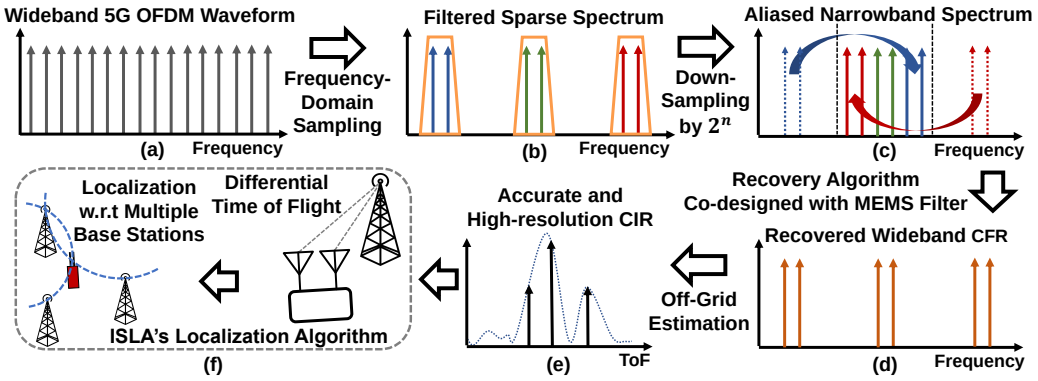


Figure 6.2: Overview Showing the Flow of *ISLA*'s System.

6.4 System Overview

ISLA enables self-localization on narrowband IoT devices by leveraging the MEMS spike-train filter to capture ambient wideband 5G signals. *ISLA*consists of 3 main components:

1. *Capturing the wideband 5G OFDM signal using the MEMS filter:* The received 5G signal is passed through the MEMS filter which samples the OFDM symbol in the frequency domain. Specifically, the MEMS filter passes the OFDM frequency bins that align with the filter passbands while suppressing all other frequency bins. The resulting output from the filter is a sparse spectrum as shown in Fig. 6.2(b). This sparse signal is then subsampled by the narrowband IoT device significantly below the Nyquist rate (16× lower) which results in aliasing the remaining subcarriers into the narrowband as shown in Fig. 6.2(c). We co-design the filter hardware with the recovery algorithm to easily reconstruct the wideband OFDM subcarriers as we describe in section 6.5.

2. *Super-Resolution CIR Estimation:* Using the recovered wideband channel measurements, *ISLA* then reconstructs a high resolution Channel Impulse Response (CIR) by leveraging its super-resolution algorithm which estimates the off-grid positions of the propagation paths as described in section 6.6. This high-resolution CIR allows *ISLA* to filter out the LoS path from the multipath in the channel for high resolution time-of-flight estimation as shown in Fig. 6.2(e).

3. *Localization Algorithm:* Since the IoT node is not synchronized with the

base station, the measured ToF will be corrupted by a timing offset. To address this, *ISLA* leverages two antennas on the IoT device and computes the differential CIR across the antennas to eliminate the synchronization offsets. This results in the locus of the IoT device to lie on a circle that is defined by the locations of the base stations and the angle subtended by the base stations at the IoT device's location, as we explain in section 6.7. Thus, by looking at the intersection of such circles, we can accurately infer the position of the IoT device as shown in Fig. 6.2(f). Finally, we show how to integrate *ISLA* with the 5G-NR standard by addressing additional system challenges in section 6.8.

6.5 Capturing 5G Signals Using Spike-Train Filter

In chapter 5, we have demonstrated using the MEMS spike-train filters for wideband spectrum sensing. However, S^3 can only detect signal power at different frequencies and cannot recover complex I and Q samples needed for estimating the CIR. Furthermore, S^3 deals with collisions resulting from aliasing by using co-prime sub-sampling rates. Such approach does not apply in the context of 5G OFDM signals, since, as we show in section 6.5 the sub-sampling factor can only be a power of 2. *ISLA* instead co-designs the hardware filter together with sampling rate to avoid collisions.

ISLA leverages the MEMS spike-train filters to capture the wideband channel measurements on a narrowband receiver. We explain this sensing process through Fig. 6.2. Consider a preamble OFDM symbol transmitted from the base station with N subcarrier frequencies at $\{f_0, \ldots, f_{N-1}\}$, shown in Fig. 6.2(a). Let the received time domain symbol be $x(t)$ and its frequency domain representation be $X(f)$. We have $X(f) = \sum_{n=0}^{N-1} c_n h_n \delta(f - f_n)$, where c_n are the data bits modulated onto the subcarriers and h_n are the channel values at f_n. We want to extract this channel information to compute the Channel Impulse Response $CIR(\tau)$. Since the preamble bits c_n are known, we can compensate for c_n and compute the $CIR(\tau)$ by taking an IFFT of the channel values h_n. However, this requires capturing the entire bandwidth of the 5G OFDM signal. Our goal is to recover the CIR using a narrowbandwith. To do so, we leverage the MEMS spike-train filter.

The spike-train filter response is made up of uniformly spaced passbands as

shown in Fig. 6.2(b). The spike-train filter serves to sparsify the OFDM symbol by selectively passing subcarriers that fall inside the MEMS passbands, while suppressing all other frequencies. Let the set of frequencies passed by the spike-train be indexed by M. Then, the frequency domain of the signal $\tilde{X}(f)$ ($\tilde{x}(t)$ in the time domain) after passing through the spike-train filter will be $\tilde{X}(f) = \sum_{i \in M} c_i h_i \delta(f - f_i)$.

This sparse spectrum is shown in Fig. 6.2(b). Next, the IoT receiver subsamples the signal $\tilde{x}(t)$ using a low-speed ADC that samples at a rate $R = B/P$, where B is the bandwidth of the transmitted symbol and P is an integer corresponding to the subsampling factor. Let $y(t)$ be the subsampled signal, that is, $y(t) = \tilde{x}(P \times t)$, and let $Y(f)$ be its frequency domain representation. Then $Y(f)$ is an aliased version of $\tilde{X}(f)$:

$$Y(f) = \sum_{i=0}^{P-1} \tilde{X}(f + iR) \tag{6.2}$$

$Y(f)$ will cover a narrow bandwidth equal to R MHz as depicted in Fig. 6.2(c). The process of aliasing is as follows. Any frequency f_j, $j \in M$, that falls outside the narrowband of the IoT device, will alias onto the frequency bin $\tilde{f}_j$ inside the narrowband after subsampling, such that $f_j - \tilde{f}_j = z \times R$, where z is some integer. Note that for every f_j, we have a unique $\tilde{f}_j$. So given the measurement at the aliased frequency $\tilde{f}_j$, we can potentially recover the channel value h_j at the corresponding unaliased frequency f_j.

However, recovering these channel values from the aliased spectrum is non-trivial because multiple of the frequency subcarriers passed by the spike-train filter may collide by aliasing on top of each other and summing up. This is unfavorable since now we are unable to extract the channel values for any of the colliding frequencies. Past work addresses this by leveraging multiple co-prime subsampling factors, which ensures that the same frequencies don't collide repeatedly.

Unfortunately, we do not have such flexibility to choose any sub-sampling factor here. This is because in order to recover the channel value h_j from the aliased frequency $\tilde{f}_j$, we need to ensure that the complex scaling factor $c_j \times h_j$ encoded on subcarrier f_j remains preserved upon aliasing. This is crucial because the wireless channel information is contained inside this scaling factor. The following lemma states the condition that ensures this:

Lemma 6.1. *For a sub-sampling factor P and N OFDM subcarriers, the complex valued scaling factors for each subcarrier will be preserved upon aliasing if $N = z \times P$, for some integer z, given the aliasing results in no collisions.*

Proof. Assume that $x[n]$ is a discrete signal from 0 to $N-1$, and we are sub-sampling (or *decimating*) it by a factor of P, meaning $y[n] = X[n \times P]$ for some integer P. Then the Discrete Fourier Transform of $y[n]$, denoted by $\hat{Y}[k]$ is

$$\begin{aligned}
\hat{Y}[k] &= \sum_{n=0}^{\lfloor N/P \rfloor - 1} x[nP] e^{-j2\frac{2\pi}{\lfloor N/P \rfloor} kn} \\
&= \frac{1}{P} \sum_{n=0}^{N-1} x[n] \sum_{m=0}^{P-1} e^{j\frac{2\pi}{P} mn} e^{-j2\frac{2\pi}{\lfloor N/P \rfloor} \frac{kn}{P}} \\
&= \frac{1}{P} \sum_{m=0}^{P-1} \big(\sum_{n=0}^{N-1} x[n] e^{-j(\frac{2\pi}{N} n)(k \frac{N/P}{\lfloor N/P \rfloor} - \frac{N}{P} m)} \big).
\end{aligned}$$

Now if P divides N, in other words $N = Pz$ for some integer z, the above simplifies to

$$\begin{aligned}
\hat{Y}[k] &= \frac{1}{P} \sum_{m=0}^{P-1} \big(\sum_{n=0}^{N-1} x[n] e^{-j(\frac{2\pi}{N} n)(k - zm)} \big) \\
&= \frac{1}{P} \sum_{m=0}^{P-1} \hat{X}[k - zm],
\end{aligned}$$

where $\hat{X}$ is the DFT of $x[n]$. This proves that, as long as there is no collision, meaning that there is at most one index m in the above equation for which $\hat{X}[k - zm] \neq 0$, then the complex values of $\hat{X}[k]$ will be fully preserved upon sub-sampling. This proves the lemma.

We also point out that if P does not divide N, then the complex values are *not* preserved. Specifically, if N/P is not a proper integer, $\hat{Y}[k]$ will be in terms of $\hat{X}[k \frac{N/P}{\lfloor N/P \rfloor} - \frac{N}{P} m]$ where inside the argument, $k \frac{N/P}{\lfloor N/P \rfloor} - \frac{N}{P} m$, is not necessarily an integer. As a result, the original information of $\hat{X}[k]$ is never repeated in any of the $\hat{Y}$ indices. In fact, $\hat{Y}$ would closely relate to an interpolated version of $\hat{X}$ with the Dirichlet kernel. □

Thus, to be able to recover channel values, we are restricted to subsample the signal by an integer factor of N. Further, since the OFDM subcarriers in the 5G standard are set to powers of 2, we can only subsample the wideband signal by powers of 2.

Due to this lack of choice in subsampling factors, we instead shift our focus on designing the spike-train filter such that the frequencies passed by the filter do not collide upon aliasing. We achieve this by leveraging the structured periodic sparsity of the spike-train, and design a filter that ensures no collisions for the given subsampling factor P.

Doing so significantly simplifies our recovery algorithm. In particular, given that (1) the frequency response of the spike-train filter and its collision-free aliasing patterns are known, and that (2) the scaling factors at the frequency subcarriers remain preserved upon aliasing, we can now simply rearrange the frequencies in Y(f) to their corresponding unaliased frequency positions as shown in Fig. 6.2(d). Further, we can extract the channel values at these unaliased frequencies by dividing the complex scaling factor $c_j \times h_j$ by the known preamble bit c_j. Thus, by leveraging the spike-train filter, *ISLA* is able to extract wideband channel values on a narrow band IoT device. Next, we discuss the design parameters of the spike-train filter that ensures no collisions.

6.5.1 Spike-Train Filter Hardware Co-Design

We explain the spike-train filter design with a specific example, shown in Fig. 6.3. Let the wideband transmitted OFDM signal (B MHz bandwidth) be comprised of 32 frequency subcarriers, indexed from -16 to 15, with 0 denoting the carrier frequency bin. From Lemma 6.1, we want the subsampling factor P to divide $N = 32$. So we choose $P = 4$, that is, the IoT receiver subsamples the signal by $4\times$. This implies that the IoT receiver is only able to capture $\frac{N}{P} = 8$ frequency bins centered around the carrier frequency as shown by the shaded region in Fig. 6.3. Let this narrow band set of frequencies be denoted as f_{NB}.

Recall that when you subsample a B MHz signal by $P\times$, then all frequency subcarriers spaced by $R = \frac{B}{P}$ MHz will alias onto the same frequency bin in the narrow band spectrum. Here, this translates into all frequencies spaced

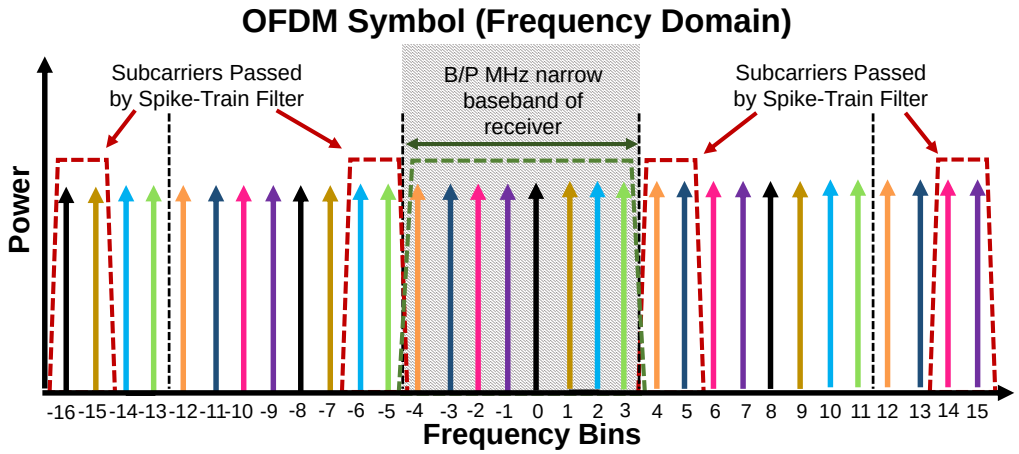


Figure 6.3: Desired Spike-Train Filter Parameters. MEMS Filter Parameters that ensure zero collisions while recovering maximum channel information.

by 8 subcarriers aliasing onto the same narrowband bin. This is depicted in Fig. 6.3 through the color coding scheme. For instance, the subcarriers at $\{-9, -1, 7, 15\}$ (represented as purple colored) would all appear at frequency bin -1 in the narrow band spectrum upon aliasing. For a given subcarrier k in the narrow band spectrum, that is, $k \in \{-4, \ldots, 3\}$, let us denote the set of subcarriers that would alias into k as I_k. So we have $I_{-1} = \{-9, -1, 7, 15\}$.

The spike-train filter will selectively pass frequency subcarriers in the wideband OFDM signal, which after aliasing can be recovered from the narrow band signal at the receiver. Let the set of frequency subcarriers passed by the spike-train filter be denoted by f_M, where $M \in [-15, \ldots, 16]$. We want the following conditions to hold:

1. *No Collisions:* To ensure that we can successfully recover the wideband channels, no two subcarriers in f_M should alias and collide in the same narrowband frequency bin upon subsampling. To achieve this, the spike-train filter must satisfy: *For any set I_k where $k \in \{-4, \ldots, 3\}$, f_M must contain at most one subcarrier from I_k.*

2. *Extract Maximum Possible Channel Values:* Given that the narrowband spectrum spans 8 frequency subcarriers, this means that the receiver can successfully recover at most 8 channel values after subsampling. In the presence of noise, we want to recover as many channel measurements as possible for robustness. Hence, every narrowband subcarrier in f_{NB} should yield one channel measurement from the wideband signal. This translates

to: *For any set I_k where $k \in \{-4, \ldots, 3\}$, f_M must contain at least one frequency subcarrier from I_k.* The first two conditions together dictate that the spike-train filter should pass *exactly one* frequency subcarrier from each I_k.

3. *Span the Wideband OFDM symbol:* To retain the high ToF resolution, we want the set of frequencies in f_M to span the entire wideband signal.

The above conditions can be met leveraging the structured sparsity in the spike-train filter response. Specifically, we can design three key parameters of the spike-train filter: (1) spacing between consecutive spikes ΔF, (2) width of the spikes ΔS, and (3) the starting frequency subcarrier f_M^0 in the spike-train, to follow Lemma 6.2. We prove that such a filter response satisfies the above conditions.

Lemma 6.2. *Consider an OFDM symbol with N frequency subcarriers, indexed as $\{f_{\frac{-N}{2}}, \ldots, 0, \ldots, f_{\frac{N}{2}-1}\}$ with inter-frequency spacing of Δf, and a narrowband receiver that subsamples by $P\times$. If P^2 divides N, then the ideal filter parameters that meet all three requirements are: (1) $f_M^0 = f_{\frac{-N}{2}}$, (2) $\left(\frac{N}{P^2} - 1\right) \times \Delta f < \Delta S < \frac{N}{P^2} \times \Delta f$, and (3) $\Delta F = \frac{N}{P}(1 + \frac{1}{P}) \times \Delta f$.*

Proof. First, we show that no two frequencies collide after aliasing. Let $q = \frac{N}{P}$, and assume that two frequencies f_α and f_β collide. Let f_α be k-th subcarrier (for $0 \leq k < P$) covered at the i-th passband ($0 \leq i < \left\lceil \frac{\Delta S}{\Delta f} \right\rceil$), and let f_β have k' and i' as corresponding indices. To collide after aliasing, $f_\alpha - f_\beta = (k - k')\Delta F + (i - i')\Delta f$ must be an integer multiple of $q\Delta f$. However, $|k - k'| \leq P - 1$ and $|i - i'| < \frac{N}{P^2}$. Thus $\frac{|f_\alpha - f_\beta|}{\Delta f} < (\frac{P-1}{P} + \frac{1}{P})q = q$, meaning we must have $f_\alpha - f_\beta = 0$, proving the first design requirement. Second, we note that P passbands that do not overlap (since $\Delta S < \Delta F$), and each passband covers exactly $\frac{N}{P^2}$ subcarriers. We therefore have a total of $P \times \frac{N}{P^2} = q$ subcarriers that, as we just showed, do not overlap after aliasing. Therefore, after aliasing, each of the q subcarriers is covered exactly once, ensuring the second design requirement. Finally, we note that the smallest bin index is covered by the filter is $\min f_M = \frac{-N}{2}$, and the largest bin index

is the last bin of the last passband, whose index can be computed as follows:

$$\begin{aligned}\max f_M &= \frac{-N}{2} + (P-1) \times \Delta F + \left\lceil \frac{\Delta S}{\Delta f} \right\rceil - 1 \\ &= \frac{-N}{2} + (P-1) \times \frac{N}{P}(1 + \frac{1}{P}) + (\frac{N}{P^2}) - 1 \\ &= -\frac{N}{2} + N - 1 = \frac{N}{2} - 1.\end{aligned}$$

Thus, the entire bandwidth (including $f_{\frac{-N}{2}}$ and $f_{\frac{N}{2}-1}$) is covered, ensuring the last design requirement. □

In Fig. 6.3, we show the ideal frequency response of the spike-train filter designed with the above parameters as the red dotted line. In theory, such a filter should allow us to leverage all f_{NB} subcarriers to recover the wideband channel measurements from the aliased signal. However, in practice, MEMS spike-train filters are non-ideal i.e., the roll-off of the passband boundaries are not as sharp as perfect rectangular functions, the spikes are not perfectly equally spaced, and the passband widths are not identical. These imperfections can be observed in the frequency response shown in Fig. 6.4. As a result of these non-idealities, there will still be collisions at the boundary regions of the spikes after aliasing, as shown in Fig. 6.5. To avoid collisions from polluting our CIR estimates, we only consider the subcarriers that do not collide as shown in Fig. 6.5. However, this results in non-uniform sampling of the OFDM subcarriers across the wideband channel. In section 6.6, we show how to leverage *ISLA*'s super-resolution algorithm to recover high resolution CIR estimates from these non-uniform channel measurements.

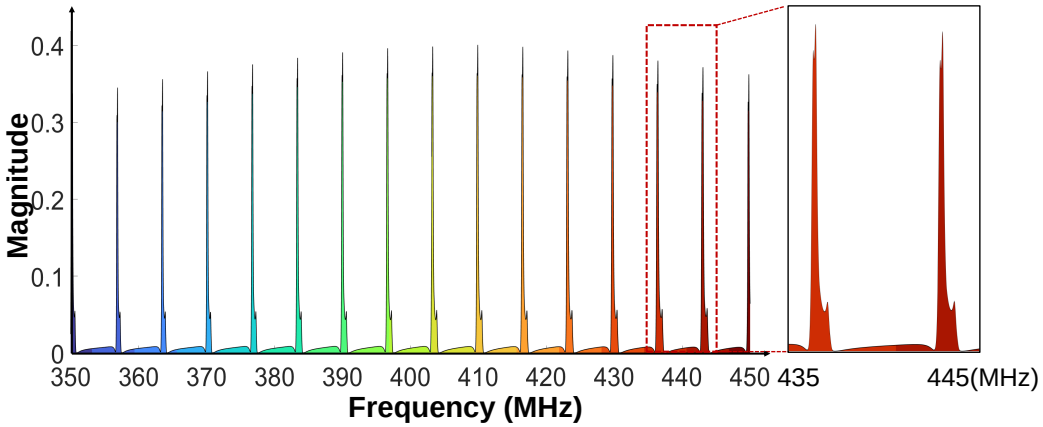


Figure 6.4: Spike-Train Filter Frequency Response

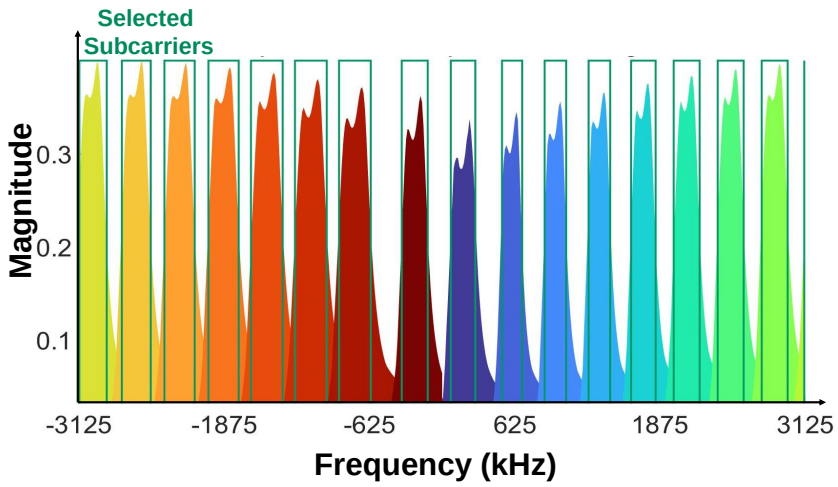


Figure 6.5: Aliasing Pattern of Spike-Train Filter Frequency Response

6.5.2 MEMS Spike-Train Filter Architecture

Following Lemma 6.2, we can derive the desired frequency response of the spike-train filter, and design MEMS resonators topology accordingly. For example, in our experiment, we used a 100 MHz 5G-like OFDM waveform with N=2048 subcarriers and a subcarrier spacing $\Delta f = 49\ kHz$, and we downsample the filtered waveform by a factor of P=16. According to Lemma 6.2, the desired filter should 16 spikes with a spike spacing of 6.64 MHz spanning the 100 MHz bandwidth and each spike should have a width around 400 kHz.

We can design a spike-train filter leveraging the periodic resonance frequencies of a type of MEMS LOBAR resonators. As shown in Fig. 6.6, the LOBAR resonator consists of 12 electrodes on the top of a thin film made of the piezoelectric material $LiNbO_3$. And we combine seven resonators in a ladder filter topology [139] to build a filter circuit. As a result, the LOBAR resonator architecture determines the spike frequencies, whereas the slight difference between different resonators determines the width of the spikes. For simplicity, here we only focus on these two key parameters of the spike-train filter response, since they are restricted by our channel recovery algorithm.

(1) The width of the film: the spacing between spikes Δf is determined by the width of the thin film W as $\Delta f = v/W$, where v is the acoustic velocity in the piezoelectric material, which is $\sim 4\ km/s$ in our design. Therefore, to achieve the 6.6 MHz spike spacing, we design the film width W to be $\sim 660\ \mu m$.

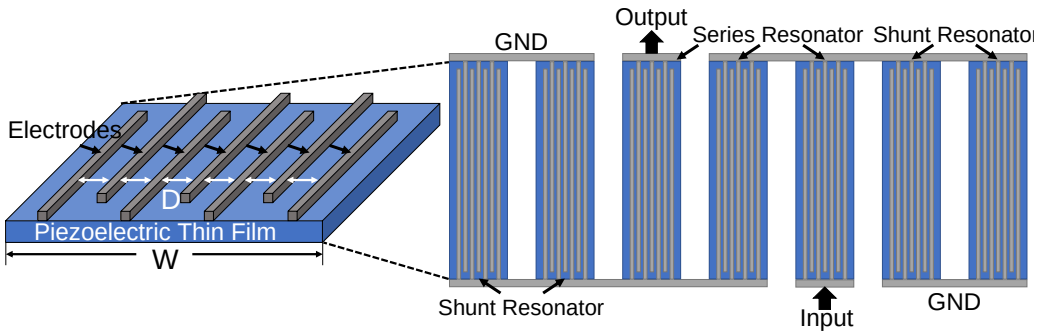


Figure 6.6: MEMS Spike-Train Filter Architecture

(2) The film width difference between different shunt and series resonators: the spike width ΔF of the spike-train filter equals to the resonant frequency difference between shunt and series resonators in the ladder filter, which is determined by the difference ΔW between shunt and series resonators: $\Delta F = fc\frac{\Delta W}{W}$. We design with piezoelectric film width to be 660 μm for series resonators and 660.26 μm for shunt resonators, which leads to $\Delta W = 0.26\mu m$, so that the widths of the spikes are around 400 kHz.

6.5.3 Tradeoff Between Range and Resolution

Recall from section 6.3 that the resolution in ToF depends on bandwidth, whereas the maximum unambiguous ToF (range) depends on the inter-frequency spacing between channel measurements. In the 5G OFDM signal with bandwidth $B = 100$ MHz and subcarrier spacing $\Delta f = 60kHz$, *ISLA* is able to retain the high ToF resolution of 10 ns (3 m) by collecting wideband channel measurements that span the entire 100 MHz. However, in doing so, the frequency spacing between the channel measurements in *ISLA* increases, thus reducing the maximum ToF range. Specifically, the frequency spacing increases by $P = 16\times$ in *ISLA*, thus reducing the maximum range from 5 km to 312 meters. This is an issue since now it becomes difficult to identify the LoS path from the CIR for localization. You could have the case where the LoS path is at 200 meters but a reflected path at 400 meters aliases and appears at the bin corresponding to 88 meters in the CIR. Thus, you cannot simply pick the first peak as LoS.

To address this, *ISLA* combines the wideband channel measurements from

the spike-train filter, h_M, with the narrowband channel measurements h_{NB} collected at the subcarriers f_{NB}, and formulates a joint optimization with both these channels to estimate the CIR. Since the narrowband channel measurements h_{NB} retain the same subcarrier spacing of $\Delta f = 60kHz$, it increases the effective maximum ToF range back to 5 km, thus resolving the LoS ambiguity in the CIR.

6.6 Super-Resolution CIR Estimation

Here we describe our super-resolution algorithm that can retrieve high resolution ToF estimates τ_l's along with the associated complex attenuations a_l for the L multipath components in the channel. As discussed in section 6.5, the IoT device can recover channel measurements $h_{tot} = h_M \cup h_{NB}$ at the subcarriers $f_{tot} = f_M \cup f_{NB}$ where f_M are recovered from the spike-train filter and f_{NB} without the filter. Since these channel values are sampled at non-uniformly spaced frequencies, we cannot apply standard super-resolution algorithms like MUSIC with spatial smoothing [208, 4] as they require uniform measurements. Instead, we optimize for the channel impulse response in the continuous time domain by leveraging an off-grid estimation technique that can estimate high resolution ToF values from the channel information.

We begin by framing this as an inverse problem. We start by modeling the forward operator $\mathcal{F}$: $h_{tot} = \mathcal{F}(\tau_1, \ldots, \tau_L, a_1, \ldots, a_L)$, which maps physical path parameters to the wireless channel. $\mathcal{F}$ comprises of the following distinct transformations, as illustrated in Fig. 6.7:

1. **CIR in Continuous Domain** (Fig. 6.7(a)): Given path parameters $\{\tau_1, \ldots, \tau_L, a_1, \ldots, a_L\}$, the continuous domain CIR can be written as: $CIR_{cont} = \sum_{l=1}^{L} a_l \delta(\tau - \tau_l)$, with each path represented as an impulse positioned at its respective ToF τ_l, and scaled by its complex attenuation a_l.

2. **Off-Grid Estimation** (Fig. 6.7(b)): The OFDM symbol spans a bandwidth B MHz and comprises of N subcarriers. Due to this discretization and truncation in the frequency domain, the observed CIR at the receiver will also be discretized, and computed on the grid defined by τ_g, where $\tau_g = \{\frac{0}{B}, \ldots, \frac{(N-1)}{B}\}$. However, as with most natural signals, the ToFs of

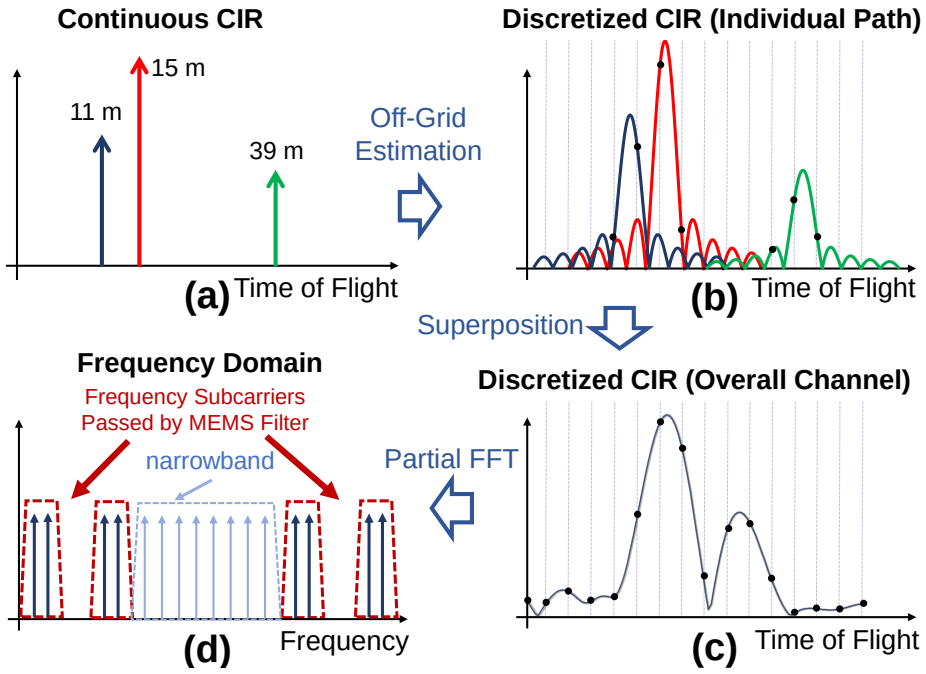


Figure 6.7: Signal Paths to Measured Channel Forward Function

the physical propagation paths τ_l will rarely align with this discretized τ_g grid, that is, the τ_l's will lie at an off-grid position. As a result, the leakage from the continuous off-grid CIR component from path l to the discrete CIR grid positions at τ_g can be computed as $CIR^l(\tau_g) = a_l\psi_N(\tau_g - \tau_l)$, where ψ_N is the discretized sinc function defined as:

$$\psi_N(\tau) = \frac{sin\,(\pi\tau)}{sin\left(\frac{\pi\tau}{N}\right)} \exp\left(-\pi j\left(\frac{N-1}{N}\right)\tau\right) \tag{6.3}$$

3. **Superposition** (Fig. 6.7(c)): With multiple propagation paths in the channel, the net observed CIR at the receiver is the sum of the CIR profiles contributed by each propagation path: $CIR^{net}(\tau_g) = \sum_{l=1}^{L} a_l\psi_N(\tau_g - \tau_l)$.

4. **Discrete Fourier Transform** (Fig. 6.7(d)): Finally, the channel h_{tot} can be computed by sampling the corresponding frequencies f_{tot} from the DFT of the superposed CIR. Let us denote the $N \times N$ Fourier matrix as $\mathbf{F}_N$, and let $\mathbf{V}$ be the matrix that chooses the rows corresponding to f_{tot} from $\mathbf{F}_N$. Then we have: $h_{tot} = \mathbf{V}\ \mathbf{F}_N\ \ CIR^{net}$ where CIR^{net} is a $N \times 1$ dimension vector.

Putting the above four transformations together, the forward operator $\mathcal{F}$

can be expressed as:

$$h_{tot} = \mathcal{F}(\{\tau_l, a_l\}_{l=1}^{L}) = \ \mathbf{V}\ \mathbf{F}_N \Psi \vec{a} \tag{6.4}$$

where Ψ is a $N \times L$ matrix with $\Psi_{i,j} = \psi_N(\tau_i - \tau_j)$, and $\vec{a}$ is a $L \times 1$ vector comprising the complex attenuations a_l for each path. Now that we have the forward operator, the inverse problem to retrieve the path parameters from observed channel vector h'_{tot} can be formulated as a L-2 minimization:

$$\{\tau_l^*, a_l^*\}_{l=1}^{L} = \underset{\tau_1,\ldots,\tau_L,a_1,\ldots,a_L}{\arg\min} \|h'_{tot} - \mathbf{V}\mathbf{F}_N \Psi \vec{a}\|^2 \tag{6.5}$$

6.6.1 Solving the Optimization

Note that if we are given Ψ, then Eq. 6.5 becomes a linear optimization problem in $\vec{a}$. Thus, given Ψ, the closed form solution for $\vec{a}$ that minimizes Eq. 6.5 is $\vec{a} = (\mathbf{V}\mathbf{F}_N\Psi)^{\dagger} h'_{tot}$, where $\dagger$ represents the pseudo-inverse. Thus the objective function in Eq. 6.5 can be rewritten as:

$$\begin{aligned} &\{\tau_l^*\}_{l=1}^{L} = \underset{\tau_1,\ldots,\tau_L}{\arg\min} \|h'_{tot} - \mathbf{V}\mathbf{F}_N \Psi (\mathbf{V}\mathbf{F}_N \Psi)^{\dagger} h'_{tot}\|^2 \\ &\text{s.t.} \qquad \tau_l \geq 0 \quad \forall\ l \in \{1, 2, \ldots, L\} \end{aligned} \tag{6.6}$$

The objective function is now reduced to just the ToF variables τ_l's. This optimization problem is non-convex and constrained, and we use the well-known interior-point method to solve this [230]. For the initialization point to the optimization algorithm, we use approximate ToF values from the CIR computed by taking the inverse FFT of the observed channel h'_{tot}. While these ToF estimates are distorted by the discretization and superpositioning artifacts described previously, it gives a good starting point for the optimization.

Also, note that the number of paths N in the wireless channel is not known a priori. As we keep increasing the number of paths N that the algorithm is initialized with, it keeps finding a better and better fit to the channel data, and after a point, starts overfitting to the noise. In order to avoid overfitting and yet yield accurate estimates for the path parameters, we run the optimization problem multiple times, each time increasing the number of paths it is initialized with by 1. We terminate the algorithm when the

decrease in the value of the objective function falls below some threshold ϵ, and set the current value of N to be the number of paths in the channel.

6.7 *ISLA*'s Localization Algorithm

The above off-grid estimation algorithm gives us highly precise ToF estimates for the propagation paths. However, since the 5G base stations are not time synchronized with the IoT device, there is going to be an offset between the sampling clocks in their RF chains. As a result, the measured ToF at the IoT node also includes delays from the sampling time offset (STO) between the different base stations and the IoT node, and hence cannot provide accurate distance estimates.

To address this, *ISLA* leverages two antennas on the IoT node to compute the differential ToF rather than the absolute. The key idea here is that while the absolute ToF measurements are corrupted by synchronization offsets, these offsets are constant across the two antennas on the IoT node. Hence, the offsets can be eliminated by differencing the two measurements. Let the ToF values to the two antennas be τ_1 and τ_2, and their corresponding distances be d_1 and d_2, as denoted in Fig. 6.8(a). Then the locus of the base station from the IoT device's frame of reference is a hyperbola with the two antennas being the foci, and the difference in distances to the two foci equaling $d_2 - d_1$. At large distances, this hyperbola can be approximated as two rays along the asymptotes of the hyperbola, depicted by the red dashed lines in Fig. 6.8(a).

By overhearing packets from different base stations, the IoT device can infer the locus of each base station to lie on approximated rays originating from the IoT device's location. This is shown in Fig. 6.8(b), where base station 1 can lie on the rays at angles θ_1 or $-\theta_1$, and similarly the base station 2 can lie on the rays at angles θ_2 or $-\theta_2$. Both θ and $-\theta$ are possible since there is the ambiguity that the signal might have arrived from the front or the back of the device. Given this, we can see that the angle subtended by the two base stations at the location of the IoT device will be $\|\theta_2 - \theta_1\|$, and this is going to be constant irrespective of the orientation of the IoT node (there is ambiguity in that the angle subtended can also be $\|\theta_2 + \theta_1\|$, and we will address this shortly).

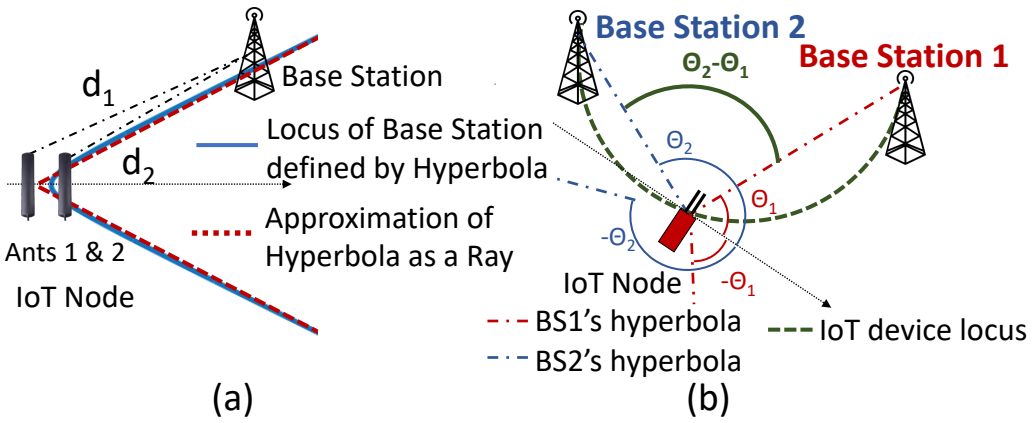


Figure 6.8: *ISLA*'s **Localization Algorithm**

Given the angle subtended by the base stations and the known locations of the base stations, according to the Inscribed Angle Theorem, we can determine the locus of the IoT device to lie on the arc of a circle, where the line segment connecting the two base stations is the chord and the corresponding inscribed angle is equal to the angle subtended by the base stations. This is illustrated in Fig. 6.8(b) as the green dashed arc. Leveraging different pairs of base stations, *ISLA* can draw multiple such arcs and the intersection points of these arcs will give us the IoT device's location.

6.7.1 Sources of Ambiguity

There are some sources of ambiguity that need to be resolved. First, the angle subtended by the two base stations in Fig. 6.8(b) could also be $\|\theta_2 + \theta_1\|$, and second, the arc drawn with the base stations at the end points could also be pointing towards the north rather than south, as depicted in Fig. 6.8(b). These ambiguities can be resolved easily by leveraging 4 base stations as anchor points. Keeping one base station common, we have three base station pairs which yields three unique arcs. Only the right configurations of angles subtended and arcs drawn will give us a common intersection point for all three arcs. *ISLA*'s localization algorithm tries all configurations and picks the one where all arcs coincide at the same point.

6.8 Integrating *ISLA* with 5G-NR Standard

The 5G-NR packet consists of 10 subframes, each of duration 1 ms [231]. To allow for coherent packet demodulation, the 5G frame appends known preamble bits on each subframe which enables channel estimation and correction across the entire bandwidth of the 5G channel. Additionally, in the first subframe of the packet, the base station also includes all information required by devices to associate with the network, which comprises of the synchronization signals (PSS and SSS frames) for CFO correction and frame timing, and the base station ID. To allow every device in the network to receive this critical information, it is always encoded in the narrowest supported bandwidth of the wideband packet, which is 4.32 MHz in the 5G standard [231].

ISLA's hardware circuit, discussed in section 6.9, is designed such that it can switch between capturing the 6.25 MHz narrowband spectrum, or the wideband spectrum via the spike-train filter. *ISLA* begins by capturing the first subframe of the 5G packet through its narrowband RF path, and extracts the synchronization frames and base station ID encoded in the narrowband subcarriers of the wideband packet. Using publicly available databases [232], *ISLA* can retrieve the location of the Base Station given its ID. The synchronization frames help eliminate coarse CFO and SFO. From the subsequent subframes, *ISLA* first estimates the narrowband channel, and then switches to the RF path with the spike-train filter to sense wideband channel. Note that *ISLA* does not need to meet tight timing constraints to switch since each subframe lasts 1 ms and there are multiple such subframes in each packet that can be leveraged for channel estimation. Thus, *ISLA* can simply skip a subframe while switching.

However, because *ISLA* captures the narrowband channel and wideband channel from different subframes, there is going to be an additional phase accumulation between the two measurements due to residual CFO. To address this, we slightly modify Eq. 6.6.

6.8.1 Updated Objective Function to Account for Residual CFO

ISLA captures the narrowband channel and wideband channel from different

subframes. Thus, there is going to be an additional phase accumulation between the two measurements due to residual CFO. To address this, we slightly modify Eq.6.6 where we split the objective function into two separate L-2 norm minimizations, with the first term containing only the wideband channel $h_M^{'}$, and the second term containing only the narrowband channel $h_{NB}^{'}$. This objective function is given below:

$$\{\tau_l^*\}_{l=1}^{L} = \underset{\tau_1,\ldots,\tau_L}{\arg\min}\Big(\|h_M^{'} - V_M F_N \Psi(V_M F_N \Psi)^{\dagger} h_M^{'}\|^2 + \|h_{NB}^{'} - V_{NB} F_N \Psi(V_{NB} F_N \Psi)^{\dagger} h_{NB}^{'}\|^2\Big) \tag{6.7}$$

$$s.t. \qquad \tau_l \geq 0 \quad \forall\, l \in \{1, 2, \ldots, L\}$$

The modified objective function is now invariant to phase offsets between the two channels, and *ISLA* can solve this updated optimization using the same technique described in section 6.6.

6.9 System Implementation

6.9.1 System Design

We have built a prototype *ISLA* device by combining our MEMS spike-train filter with commodity, off-the-shelf, low-power components. Figure 6.9 shows the circuit diagram, and Fig. 6.10 shows the actually prototype. It receives

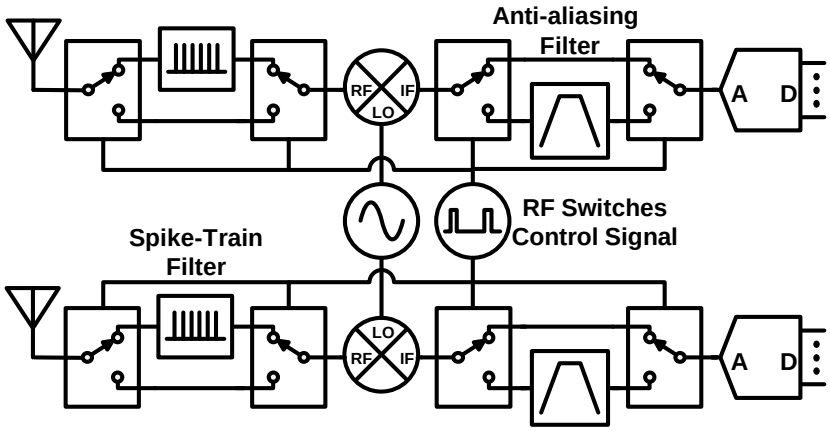


Figure 6.9: *ISLA* **Prototype Circuit Diagram**

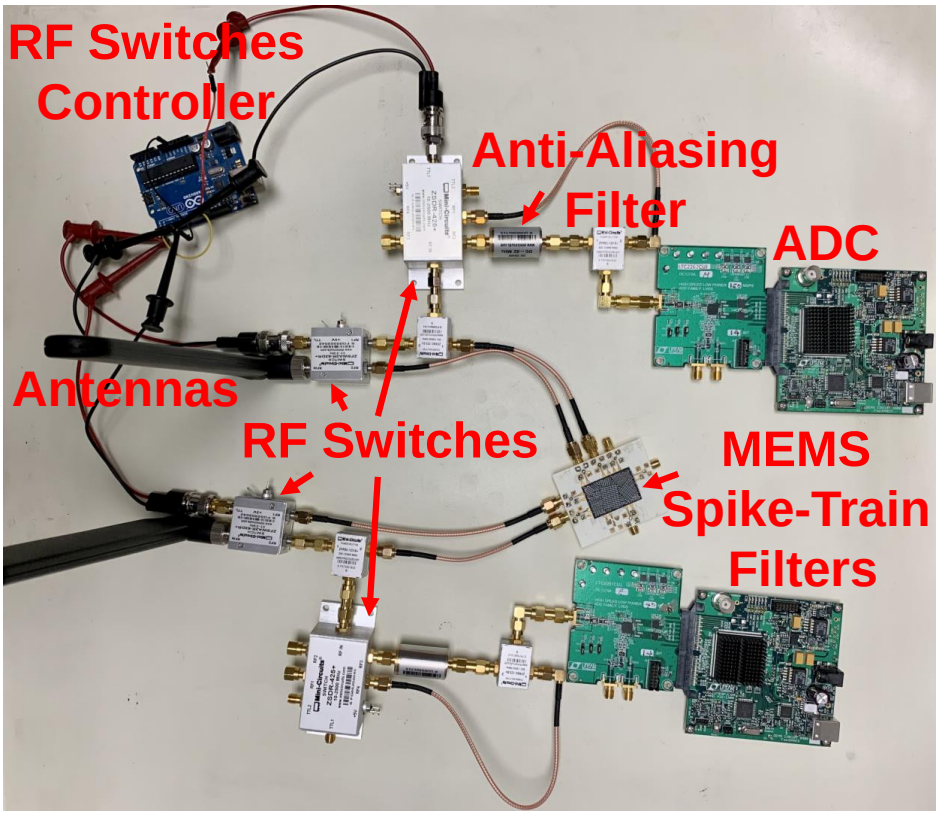


Figure 6.10: *ISLA* **Prototype Circuit**

ambient 5G transmissions with two antennas followed by identical RF chains. Depending on whether the IoT devices wants to receive the full 100 MHz spectrum using the spike-train filter or the narrowband spectrum, the RF chains can switch between two paths: (1) the received wideband spectrum first be filtered by the MEMS spike-train filter, and then down-converted and sampled without using the anti-aliasing filter. (2) the MEMS spike-train filter is bypassed but the down-converted signal will first go through an anti-aliasing filter before sampling. We select between the two paths using RF switches controlled by a single microcontroller.

6.9.2 Implementation

We fabricated a MEMS spike-train filter at 400 MHz center frequency. However, due to the strong interference from the amateur radios in this band, we were not able to run experiments outdoor using this filter. Hence, the above prototype was only used indoors. In the outdoor experiments, we transmit-

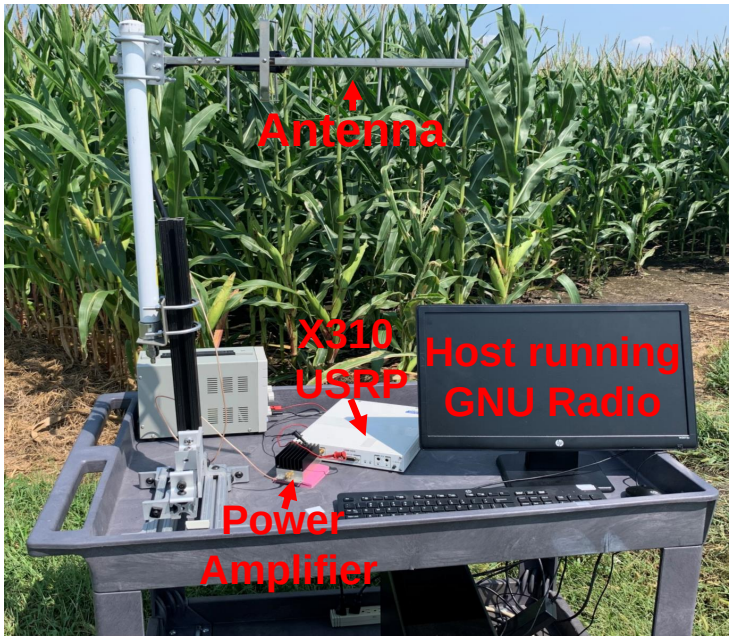


Figure 6.11: Prototype base station in the agricultural farm testbed.

ted in a vacant 100 MHz wide spectrum between 950 and 1050 MHz, and we emulate the IoT radio front-end described above with the MEMS spike-train filters in digital using an X310 USRP software-defined radio (SDR). We would like to note that in practical deployments we do not expect interference to play a major issue since *ISLA* will be deployed in the proprietary frequency bands licensed by cellular companies, which in turn will have limited interference.

The X310 SDR has two identical RF chains and can sample the full 100 MHz bandwidth with UBX160 daughterboards. To emulate the MEMS spike-train in digital, we first measure the spike-train filter frequency response once using a vector network analyzer (VNA), and we apply this filter frequency response to the received signals sampled at 100 MHz. Then, we downsample the filtered signal by simply keeping every 16th sample. This is equivalent to filtering the RF signal in analog and sample it below the Nyquist sampling rate. We also used a bandpass filters between the antenna and SDRs to remove out-of-band interferences and synchronized the two RF chains in time and phase through the GNU Radio Python API. In section 6.10.7, we present mircobenchmarks demonstrating the equivalence between applying the filter in digital and the above hardware prototype.

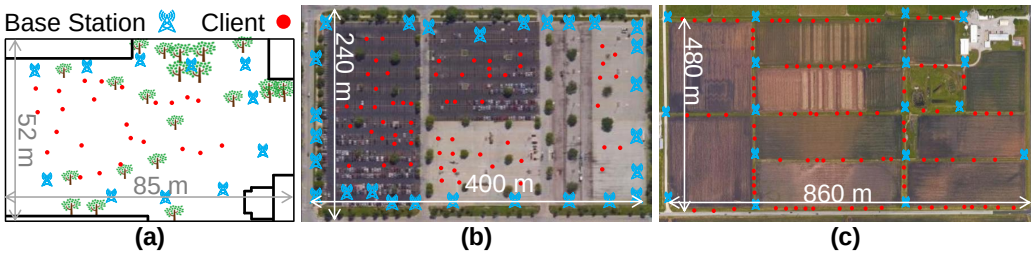


Figure 6.12: Ourdoor Experiment Testbeds: (a) Campus testbed surrounded by buildings. (b) Parking lot testbed. (c) Agricultural farm testbed.

6.9.3 Testbed

Additionally, we also built 5G base station TX prototypes to transmit ambient 5G communication signals. As shown in Fig 6.11, the base station prototype consists an X310 USRP SDR with a UBX160 daughterboard, a 9 dBi Yagi directional antenna, and an RF Bay MPA-22-30 30 dB power amplifier. The base stations transmit 100 MHz OFDM packets. Using five base station prototypes, we created three testbeds with different dimensions and at different locations to conduct our experiments. Figure 6.12 shows the satellite images of our testbeds with the base stations and clients locations marked. The first testbed is 85 m long and 52 m wide on a university campus, surrounded by buildings on all sides. We designated 11 basestation locations in this testbed and chose five of them for each experiment. The second testbed is a 400 m by 240 m parking lot with 27 base station locations. The third testbed is at a 102 acre farmland with 860 m length 480 m width. We selected five out of the 17 potential locations to place the base stations in each experiment. For ground truth locations, we used differential GPS RTK with real-time RTCM correction data, which provides centimeter-level positioning accuracy.

6.10 Evaluation

6.10.1 Baselines

1. *Spot-Fi* [208] proposes a 2D MUSIC algorithm with spatial smoothing, which can localize clients by separating the multipath components jointly along the ToF and AoA domains.

2. *mD-Track* [209] separates propagation paths by leveraging multiple dimensions of the wireless signal (ToF, AoA, AoD and Doppler), and proposes an iterative algorithm that goes through multiple rounds of error computation and path re-estimation. In our experimental setup, leveraging the AoD and Doppler dimensions provides little benefit since the base station is equipped with a single antenna and the IoT device does not have high mobility relative to the base station.

   Note that, systems like Spot-Fi and mD-Track were not designed for ambient localization, and thus need to be adapted here. Specifically, we leverage the ToF estimates provided by these baselines for the LoS path, and in turn self-localize the client by computing the relative ToF, as described in section 6.7.

3. *RSSI:* Past work leverages RSSI measurements to localize clients in outdoor cellular networks, by either using approximate path loss models for trilateration, or by using the known locations of nearby cells as coarse estimates. We implemented one recent RSSI baseline [1].

4. *Spike-train filter-adapted baselines:* To provide a fair comparison against *ISLA*, we modify Spot-Fi and mD-Track to leverage the spike-train filter and utilize the wideband channel measurements for localization. It is non-trivial to adapt Spot-Fi for the spike-train filter since the spatial smoothing technique used in Spot-Fi requires uniformly spaced channel measurements across frequency, whereas the spike-train filter samples the OFDM frequency bins non-uniformly. To address this, we restructure the spatial smoothing subarray from [208] that allows Spot-Fi to be applied across the non-uniform frequencies sampled by the spike-train filter.

6.10.2 Localization Accuracy Comparison Against Baselines

We compare *ISLA*'s localization against the baselines in Fig. 6.13. Note that, while *ISLA* is designed specifically to leverage the wideband channel sensed by the MEMS filter, the baselines are implemented without modification and thus utilize only the narrowband channel for localization. Unless otherwise specified, for all results, we utilize 5 randomly chosen base stations as the anchor points.

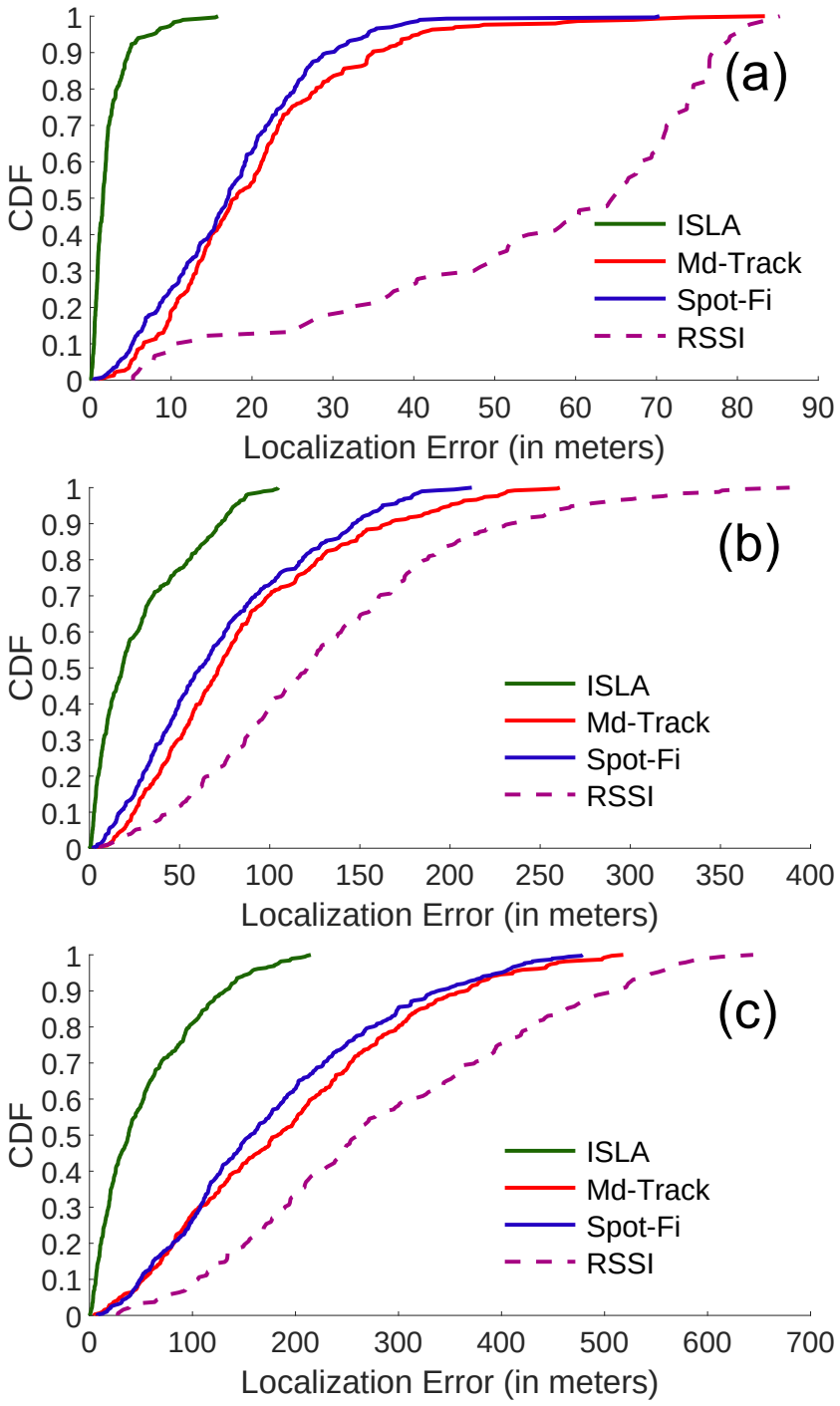


Figure 6.13: *ISLA*'**s localization accuracy compared against baselines at:** (a) Campus (b) Parking lot (c) Farm.

From Fig. 6.13, *ISLA* achieves a median localization accuracy of 1.58 meters in the campus testbed, 17.6 meters in the parking lot testbed, and 37.8 meters in the farm testbed. Across the same three testbeds, Spot-Fi achieves median accuracies of 17.05 meters, 61.2 meters and 156.6 meters, whereas mD-Track achieves 18.11 meters, 71.8 meters, and 183.1 meters respectively. Thus, *ISLA* improves the localization accuracy over Spot-Fi and mD-track by $\sim 11\times$ in the campus testbed, and by $\sim 4\times$ in the parking lot and farm. *ISLA* is able to achieve such high gains since it leverages the spike-train filter to sense wideband channel on the narrowband device, which allows for much higher resolution compared to the baselines operating solely in the narrowband. Further, the localization improvement over the narrowband baselines is most significant in the campus testbed, since it has the most multipath from surrounding buildings, and thus ToF resolution is critical to separate out the LoS path from reflections.

Lastly, the RSSI baseline achieves median accuracies of 64.54 meters, 120.7 meters, and 260.8 meters respectively across the three testbeds. RSSI based methods generally have poor performance, as they tend to oversimplify path loss models that map RSSI values to distance, which does not hold for real world multipath channels.

6.10.3 Comparison Against Spike-Train-Adapted Baselines

Next, we evaluate how leveraging the spike-train filter would benefit the performance of our narrowband baselines. Figure 6.14 shows the CDF of localization accuracy comparing *ISLA* against the modified baselines that utilize the wideband channel from the spike-train filter. The RSSI baseline is not included here since its localization performance does not depend on bandwidth. Compared to its narrowband implementation, Spot-Fi's median accuracy improves to 11.08 meters in the Campus testbed, 49.07 meters in the Parking Lot, and 137.76 meters in the farm. Similarly, mD-Track's median performance improves to 15.48 meters, 51.45 meters and 103.78 meters in the three testbeds respectively. Thus, Spot-Fi and mD-Track see improvements in localization accuracy by up to 54% and 76% respectively. This shows that other localization techniques can also benefit from the wide-band channel sensing capabilities enabled by the spike-train filter.

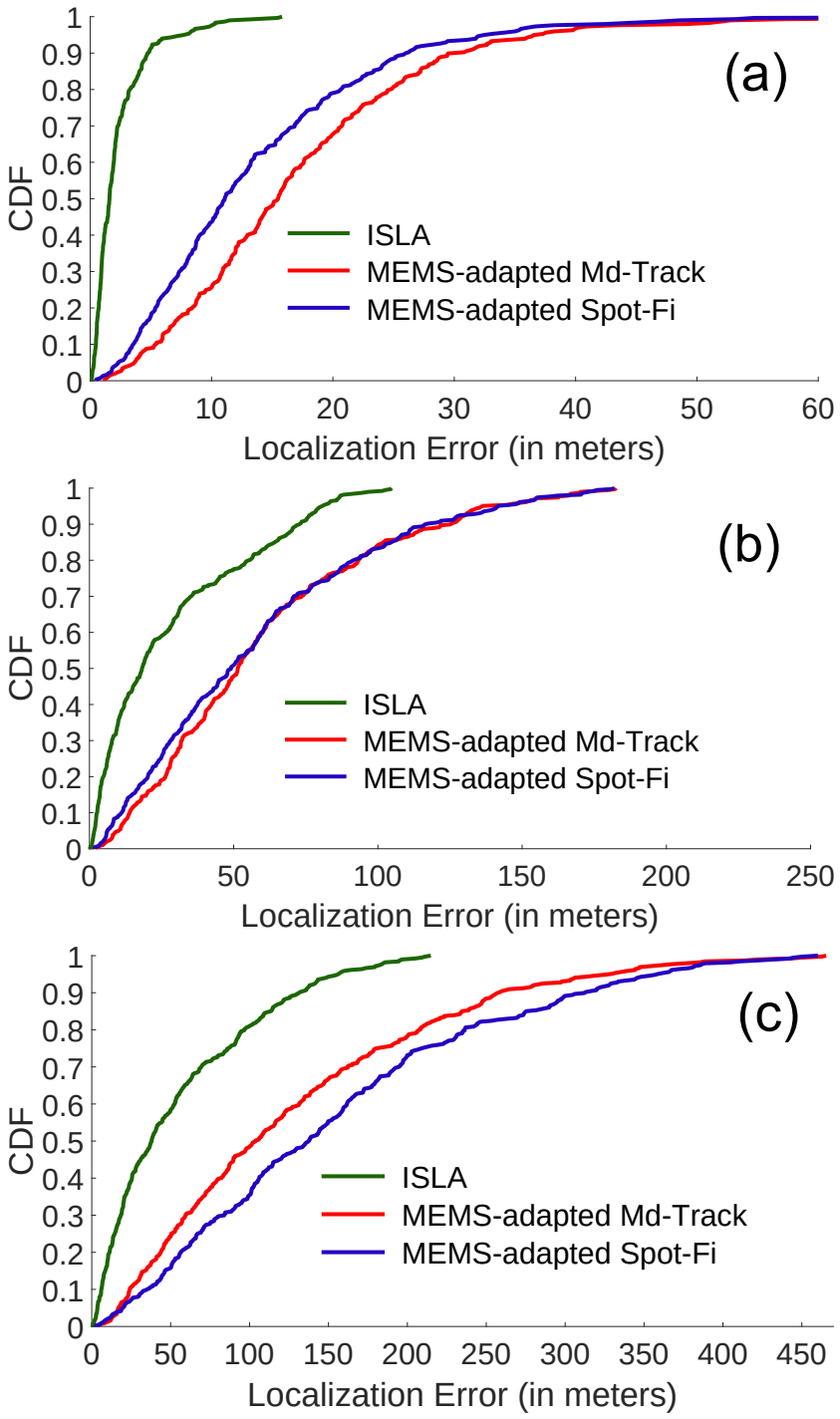


Figure 6.14: *ISLA*'**s localization accuracy compared against MEMS filter adapted baselines at:** (a) Campus (b) Parking lot (c) Farm.

Additionally, Fig. 6.14 shows that given the same channel information, *ISLA*'s off-grid CIR estimation algorithm is able to better resolve and estimate the relative ToF compared to Spot-Fi and mD-Track. This is because these baselines were designed to leverage multiple information dimensions to separate out the multipath components, with both baselines leveraging 3 or more antennas for separation in the AoA domain, and mD-Track further using the additional dimensions of Doppler and AoD as well. In contrast, here the IoT device has to separate out multipath in the ToF domain alone, and *ISLA* is able to achieve very accurate localization owing to its off-grid estimation algorithm.

6.10.4 *ISLA* Leveraging Different Amounts of Spectrum

In this experiment, we compare *ISLA*'s localization algorithm applied across three different amounts of spectrum utilization — (1) *ISLA* applied only to the wideband sparse channel sensed by the spike-train filter (without combining with narrowband channel), (2) *ISLA* applied only to the narrowband channel of IoT device, and (3) *ISLA* applied across the entire 100 MHz bandwidth of the received 5G signal. Figure 6.15 plots the CDF of localization accuracy achieved across the three testbeds.

ISLA applied on the narrowband channel performs the poorest, achieving median accuracies of 7.9 meters, 58.9 meters and 142.52 meters in the campus, parking lot and farm testbeds. In contrast, *ISLA* along with the spike-train filter can achieve corresponding median accuracies of 1.68 meters, 18.8 meters and 45.04 meters. Thus, *ISLA* along with spike-train achieves an improvement in localization accuracy of $3.16\times - 4.7\times$ compared to *ISLA* applied in the narrowband spectrum, despite both baselines capturing the same amount of channel measurements. The advantage of spike-train stems from the fact that it enables the narrowband receiver to capture channel measurements that span a much larger bandwidth, which results in much higher ToF resolution.

On the other hand, *ISLA*'s localization algorithm applied on the full 100 MHz spectrum achieves median accuracies of 1.38 meters, 11.44 meters and 25.8 meters respectively on the three testbeds. Thus, *ISLA* with the spike-train filter reduces the localization accuracy by only $1.21\times$, $1.64\times$, and $1.74\times$

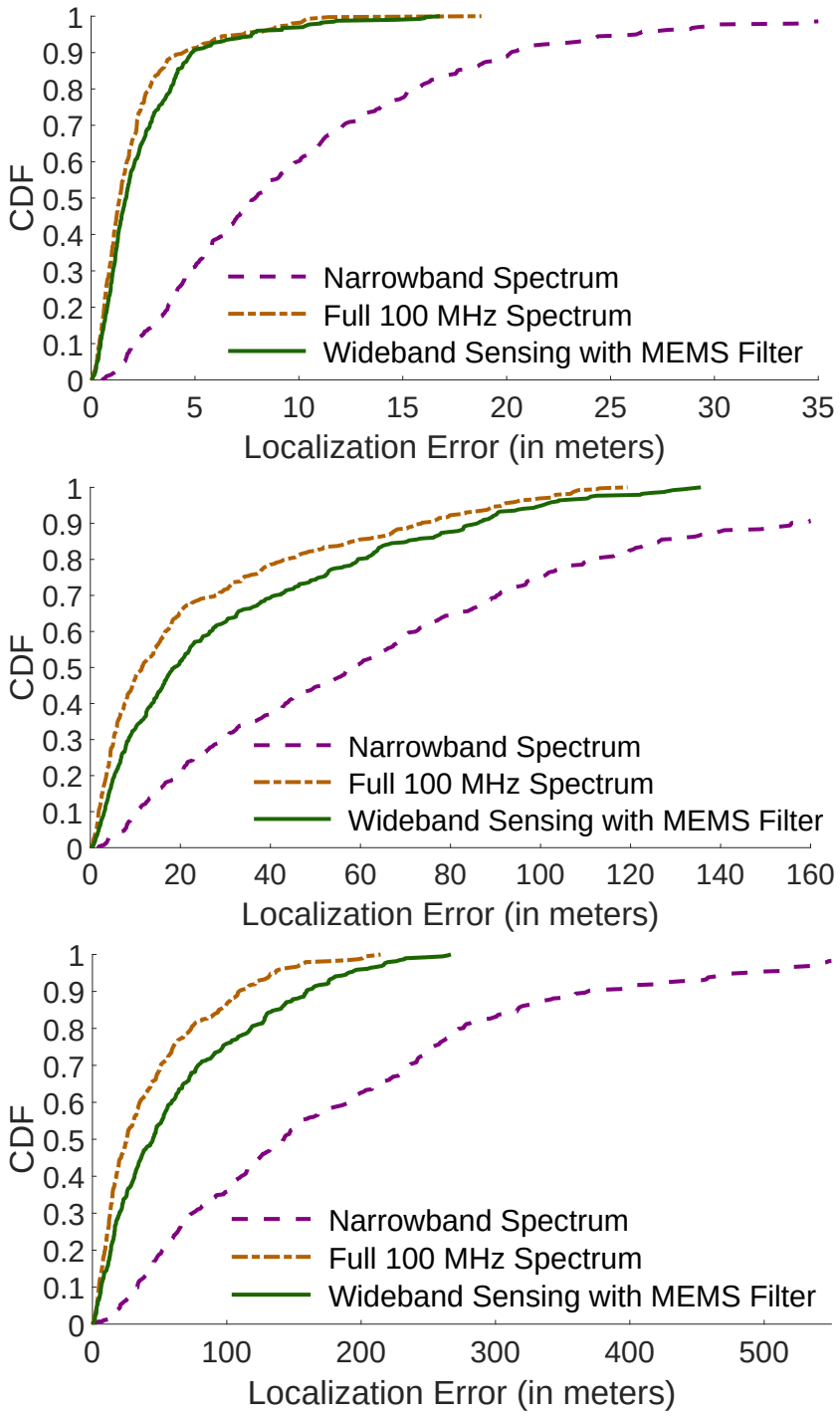


Figure 6.15: Comparison of *ISLA*'s localization accuracy when leveraging different amounts of spectrum at: (a) Campus (b) Parking lot (c) Farm.

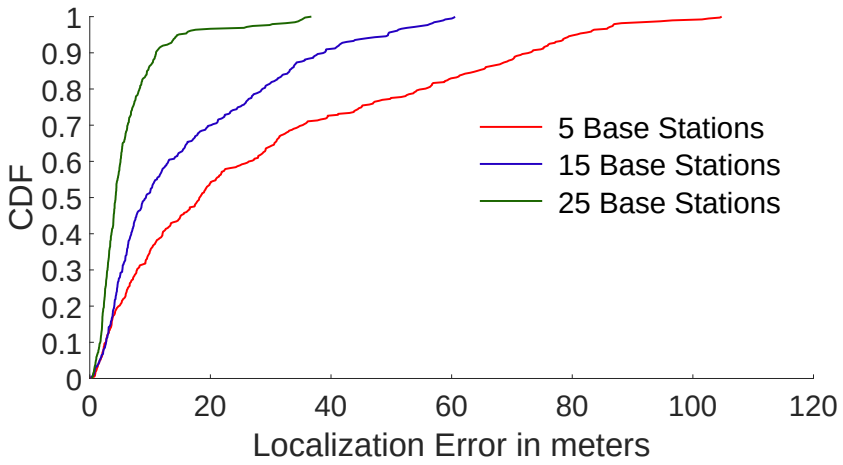


Figure 6.16: Localization error with different number of visible base stations.

respectively compared to this upper bound. This demonstrates that the spike-train filter can enable a narrowband device to achieve localization accuracy within a factor of 2× compared to a broadband receiver, despite the fact that it subsamples the signal by 16× below Nyquist.

6.10.5 Localization with Number of Anchor Base Stations

In Fig. 6.15(d), we compare *ISLA*'s localization performance with 5, 15 and 25 base stations used as anchor points respectively, in the parking lot testbed. With 5 base stations, *ISLA* achieves a median accuracy of 17.6 meters, which improves to 9.27 meters with 15 base stations, and 4.26 meters with 25 base stations. This improvement becomes even more significant at the tail, with *ISLA* achieving 90^{th} percentile accuracy of 73.16 meters with 5 base stations, which improves to 10.9 meters accuracy with 25 base stations at 90^{th} percentile. Thus, leveraging more base stations can significantly improve the localization accuracy achieved by *ISLA*.

6.10.6 Tracking Objects

We move the IoT device across an L-shaped trajectory (160 meters in length and 85 meters in width) in the parking lot testbed, and collect packet transmissions from the base stations at different points along this trajectory. In this experiment, we pick 7 fixed base stations to utilize as anchor points, and

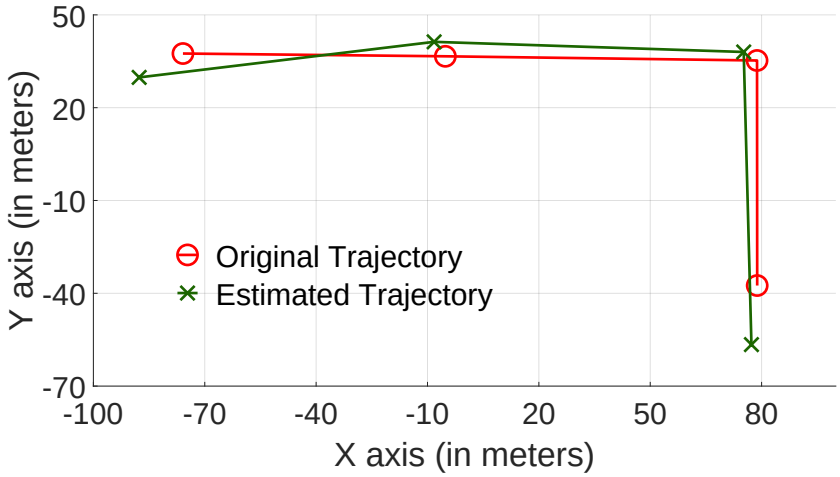


Figure 6.17: Object Tracking Trajectory Accuracy

we show the ground truth trajectory and corresponding estimated trajectory by *ISLA* in Fig. 6.17. As can be observed, *ISLA*'s high localization accuracy allows to faithfully capture the shape of the ground truth trajectory.

6.10.7 Microbenchmark - CIR Estimation using Fabricated MEMS Spike-train Filter

To verify the equivalence between our outdoor implementation and using the prototype with the fabricated MEMS spike-train filter at 400 MHz, we conduct indoor experiments at 400 MHz. Specifically, we evaluate the error in reconstructed CIR and estimated ToF values between the prototype with the fabricated filter and *ISLA* with the digital filter implementation. In Fig. 6.18, we show the CDF of the errors in ToF values (converted to distance (meters)) recovered by the two approaches, for both LoS and NLoS paths. We can see that the position of the LoS path in the CIR estimated from both approaches are very close, with the median error between their estimates being 0.075 meters. The error in the NLoS paths is higher, with a median error of 1.05 meters. However, this will not affect the localization performance between the two since localization only uses the LoS path. This microbenchmark demonstrates that *ISLA*'s approach of applying the filter and subsampling in digital is equivalent to using the fabricated filter from a localization perspective and that the results shown in this chapter are representative of a fully implemented system.

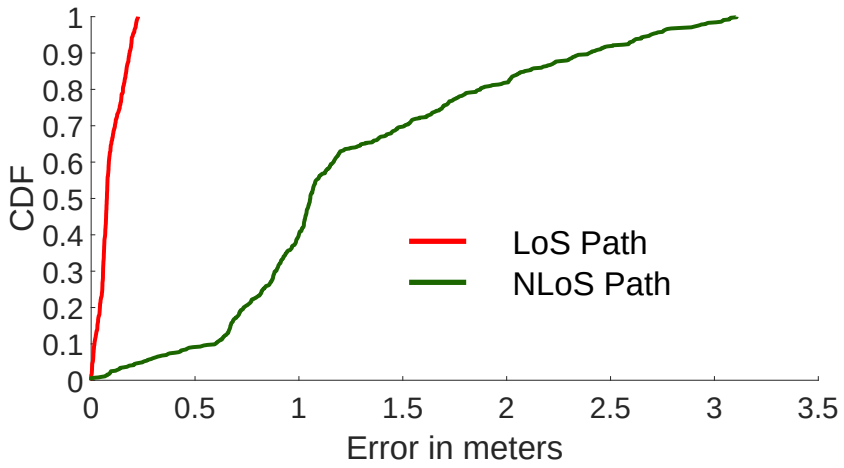


Figure 6.18: ToF difference between *ISLA*'s prototype with fabricated MEMS filter and digitally implemented MEMS filter.

6.10.8 Microbenchmark - Density of Deployed Base Stations

In section 6.10.2, we have shown that *ISLA*'s localization accuracy increases substantially as we use more anchor base stations. Here, we study the distribution of how many base stations can the client overhear at a given location. Using publicly available databases [232], we retrieved the locations of 4G LTE base stations belonging to 4 major carriers in the United States. We chose 4G LTE for this analysis since 5G deployment is still in its nascent stage in the USA, but we expect the target coverage for 5G networks to exceed the 4G deployment.

In Fig. 6.19, we show the scatter plot of the 4G base stations located in Chicago downtown area. Using the cell coverage information provided in [232] for the different base stations, in Fig. 6.20, we plot the CDF of the number of base stations that the client can overhear at different locations on the map. We can see that at the 10^{th} percentile, the number of visible base stations is 11, thus implying that less than 10% of client locations see less than 11 base stations. Further, the median number of base stations visible to the client is 29. This demonstrates that the cellular deployment is dense enough to allow many anchor points, which in turn can achieves high localization accuracy.

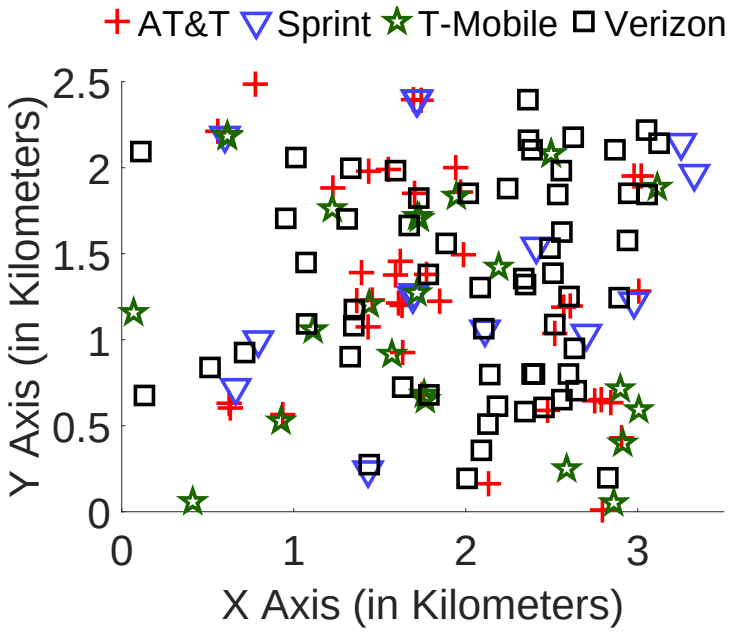


Figure 6.19: Deployment of 4G Base Stations in Chicago Downtown Area

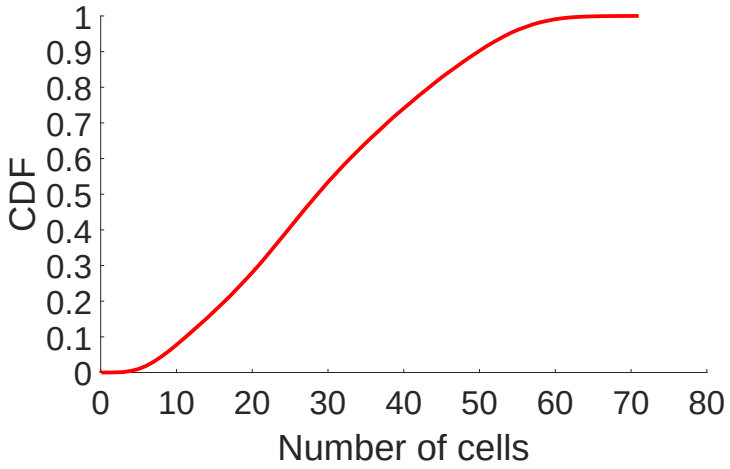


Figure 6.20: Number of Visible 4G Base Stations at Various Downtown Locations

6.10.9 Microbenchmark - Invariance to Orientation

Here, we demonstrate that the localization performance is independent of the orientation of the IoT device. This is because the arcs that define the locus of the IoT node, depend only on the angle subtended by the base stations at the IoT device's location, which is invariant to device rotation. At a given location in our campus testbed, we orient the IoT device along 4 different di-

Table 6.1: Invariance of Localization Error to Orientation

Direction	NW	NE	SE	SW
Median	1.3535 m	1.3544 m	1.3267 m	1.3681 m
Std Dev	0.4948 m	0.6026 m	0.4908 m	0.512 m

rections and perform 100 localization experiments at each orientation. From Table 6.1, we can see that the median and standard deviation in localization error is almost the same across the 4 orientations, thus demonstrating invariance to orientation.

6.11 Extending *ISLA* to mmWave

6.11.1 Motivation and Challenges

Our original implementation of *ISLA* is limited to sub-6GHz bands. However, leveraging the mmWave bands 5G signals for localizing IoT nodes is even more appealing, because of two characteristics of 5G mmWave networks: 1) The small cell sizes lead to very dense deployments of base stations, up to 40 to 50 BS per square km [204], resulting in more potential anchor points for accurate localization. 2) The unprecedentedly wide signal bandwidth, up to 400 MHz in mmWave eMBB channels, provides high-resolution Time of Flight (ToF) estimation and, hence, high localization accuracy.

However, adapting *ISLA*'s coordination-free localization protocol to mmWave bands would be impractical, because *ISLA* avoids coordination with the gNBs by measuring the Time Difference of Arrival (TDoA) between two antennas on the IoT node. Such IoT design requires two antenna front-ends with tightly synchronized RX chains, which is infeasible in mmWave frequencies because of the expensive and power-consuming mmWave front-ends. Therefore, *mm-ISLA* abandons the dual front-end IoT design and the TDoA-based localization algorithm of *ISLA*. Instead, *mm-ISLA* overcomes the coordination-free challenge by leveraging the additional degree of freedom provided by the MIMO antenna arrays at the 5G gNBs. *mm-ISLA* first resolves channels from multiple TX antennas at the gNBs leveraging a unique 5G-NR waveform – DeModulation Reference Signal (DMRS) in the Physical Downlink Shared Channel (PDSCH). The unique resource allocation pattern

in the DMRS waveforms allows *mm-ISLA* to distinguish the OFDM subcarriers allocated to each antenna in the gNB MIMO antenna array. Therefore, *mm-ISLA* can then leverage the channel differences across the antennas to estimate the Angle of Departure (AoD) of the Line-of-Sight (LoS) path from the gNB to the IoT node. Finally, with the AoD measurements of three gNBs, an *mm-ISLA* node can localize itself using the standard triangulation localization algorithm.

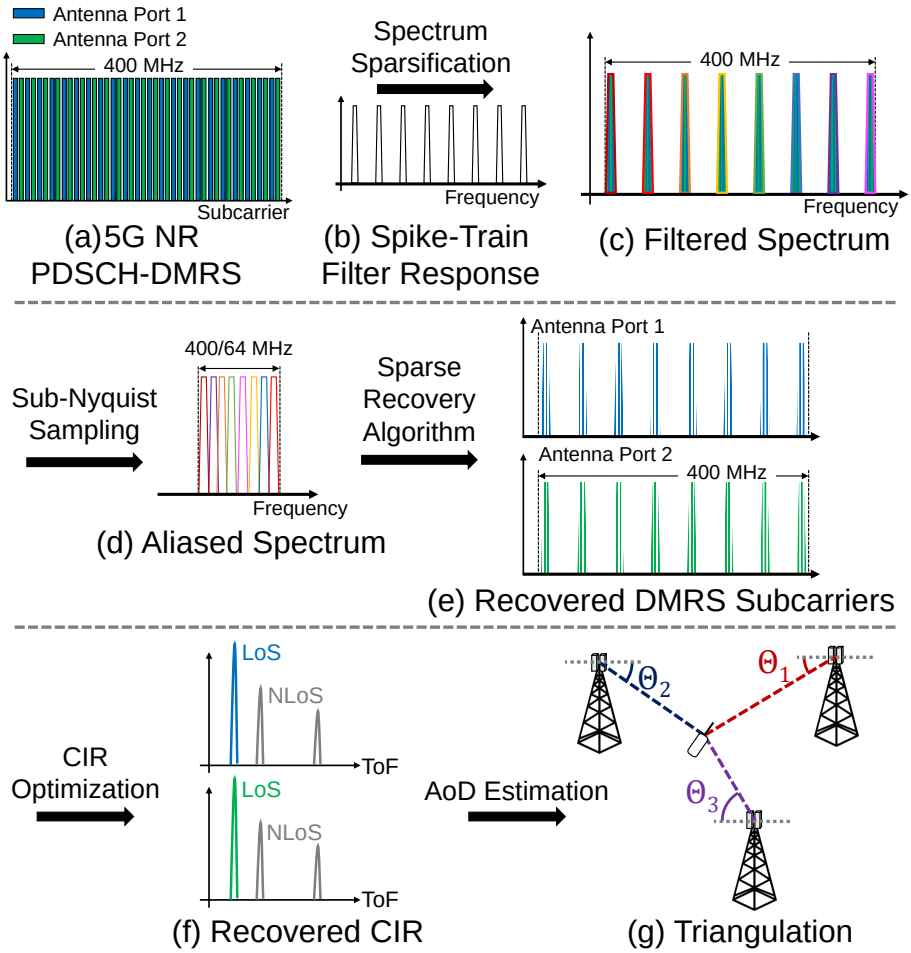


Figure 6.21: *mm-ISLA* **pipeline.** (a) Wideband 5G PDSCH-DMRS Spectrum Allocated to 2 Antenna Ports. (b) MEMS Spike-Train Filter Frequency Response. (c) Filtered Sparse Spectrum. (d) Sub-Nyquist Sampled Spectrum Aliased to the Narrow ADC Bandwidth. (e) Recovered DMRS Subcarriers. (f) Recover Channel Impulse Response. (g) AoD-Based Triangulation Localization.

6.11.2 LoS AoD Estimation with PDSCH-DMRS Waveform

Figure 6.21 illustrates *mm-ISLA*'s system pipeline. *mm-ISLA* adopts the same super-resolution CIR reconstruction method as *mm-ISLA* by formulating an inverse optimization problem. Towards solving the coordination-free challenge, *mm-ISLA* however, takes a completely different approach than *ISLA*. The TDoA-based localization algorithm of *ISLA* is abandoned, because it requires two tightly synchronized RF front-ends, RF chains, and ADCs. The additional RF circuitry and ADC doubles the cost and power-consumption of the IoT nodes, which is even more infeasible in the mmWave frequencies than in the sub-6GHz bands. Restricted to a single antenna front-end, *mm-ISLA* enabled IoT nodes still manage to localize themselves without any coordination with the gNBs. To do so, *mm-ISLA* leverages another unique opportunity in 5G networks – the spatial diversity of the MIMO antenna arrays at the 5G gNBs. *mm-ISLA* tries to measure the ToF differences across antennas in the gNB MIMO antenna array, from which *mm-ISLA* can infer the AoD of the LoS path from the gNB to the IoT node. With AoD estimates of three or more gNBs along with the gNB locations and antenna array orientations, *mm-ISLA* enabled IoT nodes will be able to apply the standard triangulation algorithm to localize themselves. However, to do so, *mm-ISLA* has to first be able to estimate the CIR from each gNB MIMO antenna separately.

The question becomes how can *mm-ISLA* isolate concurrent transmissions from TX MIMO antennas at the 5G gNB and estimate them corresponding CIR separately? Note that signals from different TX antennas have to be transmitted at the same time; otherwise, the transmitting time offset will corrupt the AoD estimation. To overcome this challenge, *mm-ISLA* leverages another unique opportunities in the 5G-NR standards, that is the resource allocation pattern in the 5G-NR PDSCH-DMRS waveforms. PDSCH-DMRS is a specific type of 5G-NR waveform used for decoding the PDSCH data, so it's a preamble-like waveform one can leverage to estimate the channel. When MIMO is enabled at the gNB, to decode the channels from the MIMO antennas, different antenna ports are allocated with a different set of interleaved subcarriers in the resource block [233], as shown in Fig. 6.21(a). Therefore, we can identify the DMRS subcarriers corresponding to each TX antenna and estimate their channels separately. Since the interleaved subcarrier al-

location pattern ensures that the DMRS waveform from all TX antennas covers the entire bandwidth of the resource block, we can still achieve wideband CFR estimations for all TX antennas. Therefore, we can estimate the super-resolution CIRs corresponding to each TX antenna with a small modification to the inversion optimization problem to incorporate the subcarrier allocation in the PDSCH-DMRS waveform. Finally, we compare the ToF differences across the TX antennas to estimate the AoD of the LoS path.

6.12 Limitations and Discussion

- *Power Footprint:* To enable ambient localization, *ISLA* leverages a second antenna and RF chain, which increases the power footprint of the IoT device. However, we would like to note that the power overhead of an additional RF chain is going to be lower than that of a GPS module, which is the likely alternative for localization. This is because the additional RF chain on the IoT device is going to operate in the narrowband with very low sampling rates, whereas GPS incurs high operational power since it needs to receive and correlate long sequences to get the signal power above the noise floor for GPS lock acquisition. Hence, while *ISLA*'s design does lead to an increased power footprint, it is still a better alternative compared to GPS.

- *Loss of SNR:* Since the MEMS spike-train filter is a passive device, the signal suffers from insertion loss when passed through the filter, thus resulting in loss of SNR. This is further exacerbated by the fact that, in practice, the out-of-band rejection of the spike train filter is finite, which results in further loss of SNR. It is possible to reduce the impact of this SNR loss at the circuit level by improving impedance matching and the isolation between input and output ports. We can also compensate for the SNR loss by averaging the channel measurements across multiple OFDM symbols.

- *Line-of-sight*: Similar to many localization systems, *ISLA* assumes the availability of line-of-sight (LoS) paths to the base stations which might not hold under occlusion. This, however, can be addressed by potentially selecting a subset of base stations with LoS paths using similar techniques

demonstrated in [208]. With the dense deployment of 5G base stations, we expect a significant subset of base stations to have LoS path to the node.

- *Fast Mobility*: The current design of *ISLA* is not suitable for highly dynamic applications with fast mobility such as tracking cars. This is because the localization algorithm must receive wideband 5G packets from 4 or more base stations before it can self-localize.

- *Multiple Providers*: *ISLA* can benefit from capturing signals from multiple different providers since the IoT node does not need to associate with the base stations. However, different providers operate in different frequency bands which would require different spike-train filters. This could potentially be addressed by having multiple filters and switching between them similar to our design in section 6.9.

Chapter 7

Conclusion

In this thesis, we introduce new software and hardware primitives that advance wireless technologies towards fulfilling the vision of ubiquitous perception in next-generation wireless networks.

Towards this goal, we first demonstrate AI-enhanced wireless imaging that is able to extract the underlying perceptual and conceptual information in wireless signals to achieve high-resolution imaging and accurate object detection. On this front, one possible future research direction is to combine *HawkEye* and *Radatron* into a hierarchy of neural network models to reconstruct holistic scenes with various types of objects. The blueprint of this complete system is as follows: It first parses the mmWave radar heatmap using a semantic scene understanding network, which can propose regions in the scene where the reflections from each object are and what type of object it is. Hence, we can zoom into each object and image it using a specialized GAN that is trained for that type of object. Finally we merge images of individual objects into a holistic scene.

Besides, this thesis also tries to facilitate the integration of enhanced wireless perception systems into next-generation wireless networks. Towards this goal, we present two approaches.

The first approach is to facilitate the coexistence of perception and communication by allowing perception applications to opportunistically access idle spectra that are not occupied by communication services. To enable such dynamic spectrum sharing spectrum access paradigm, we leverage a novel MEMS spike-train filters to design a efficient wideband spectrum sensing mechanism. However, the coexistence of many wireless services can lead to interference between different users and applications. Therefore, another future direction for me is to design interference avoidance and mitigation schemes for wireless perception systems and especially radars.

The other approach we demonstrate is to reuse the communication network

foundations for perception. We develop a passive IoT self-localization technique by reusing ambient 5G cellular signals. On this front, I plan to leverage opportunities in the antenna beam space provided by mmWave phased array radio front-ends for joint communication and perception design.

Last but not least, there could also be spoofing and even adversarial attacks towards the more ubiquitous wireless perception systems that leads to security issues, so I am also interested in speculating potential spoofing and attack models towards wireless perception systems, as well as prevention and control design.

References

[1] R. Elbakly and M. Youssef, "Crescendo: An infrastructure-free ubiquitous cellular network-based localization system," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–6.

[2] H. Rizk, A. Shokry, and M. Youssef, "Effectiveness of data augmentation in cellular-based localization using deep learning," in *2019 IEEE Wireless Communications and Networking Conference (WCNC)*. IEEE, 2019, pp. 1–6.

[3] A. Bansal, A. Gadre, V. Singh, A. Rowe, B. Iannucci, and S. Kumar, "Owll: Accurate lora localization using the tv whitespaces," in *Proceedings of the 20th International Conference on Information Processing in Sensor Networks (co-located with CPS-IoT Week 2021)*, 2021, pp. 148–162.

[4] J. Xiong and K. Jamieson, "Arraytrack: A fine-grained indoor location system," in *10th USENIX Symposium on Networked Systems Design and Implementation (NSDI 13)*, 2013, pp. 71–84.

[5] D. Vasisht, S. Kumar, and D. Katabi, "Decimeter-level localization with a single wifi access point," in *13th USENIX Symposium on Networked Systems Design and Implementation (NSDI 16)*, 2016, pp. 165–178.

[6] F. Adib, H. Mao, Z. Kabelac, D. Katabi, and R. C. Miller, "Smart homes that monitor breathing and heart rate," in *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, ser. CHI '15, 2015, p. 837–846.

[7] J. Liu, Y. Wang, Y. Chen, J. Yang, X. Chen, and J. Cheng, "Tracking vital signs during sleep leveraging off-the-shelf wifi," in *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, ser. MobiHoc '15, 2015, p. 267–276.

[8] J. Liu, Y. Chen, Y. Wang, X. Chen, J. Cheng, and J. Yang, "Monitoring vital signs and postures during sleep using wifi signals," *IEEE Internet of Things Journal*, vol. 5, no. 3, pp. 2071–2084, 2018.

[9] F. Adib, C. Hsu, H. Mao, D. Katabi, and F. Durand, "Capturing the human figure through a wall," *ACM Trans. Graph.*, vol. 34, no. 6, Oct. 2015.

[10] M. Zhao, T. Li, M. A. Alsheikh, Y. Tian, H. Zhao, A. Torralba, and D. Katabi, "Through-wall human pose estimation using radio signals," in *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, June 2018, pp. 7356–7365.

[11] M. Zhao, Y. Tian, H. Zhao, M. A. Alsheikh, T. Li, R. Hristov, Z. Kabelac, D. Katabi, and A. Torralba, "Rf-based 3d skeletons," in *Proceedings of the 2018 Conference of the ACM Special Interest Group on Data Communication*, 2018, p. 267–281.

[12] M. Zhao, Y. Liu, A. Raghu, H. Zhao, T. Li, A. Torralba, and D. Katabi, "Through-wall human mesh recovery using radio signals," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 10 112–10 121.

[13] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, 2017.

[14] Bloomberg, "Autonomous-car tech investment skyrockets on softbank deals," *Press Release, 2018*.

[15] Bloomberg, "Uber has spent more than $1 billion on driverless cars," *Press Release, 2019*.

[16] Forbes, "The future with level 5 autonomous cars," *Press Release, 2019*.

[17] The Wall Street Journal, "Toyota investing $500 million in uber in driverless-car pact," *Press Release, 2018*.

[18] NVIDIA, "The journey to zero accidents - nvidia drive," May 2019. [Online]. Available: https://resources.nvidia.com/en-us-drive-overview

[19] The New York Times, "Honda to invest 2.75 billion usd in gm's cruise autonomous vehicle unit," *Press Release, 2018*.

[20] Forbes, "Weather creates challenges for next generation of vehicles," *Press Release, 2019*.

[21] M. Laurenzis, F. Christnacher, E. Bacher, N. Metzger, S. Schertzer, and T. Scholz, "New approaches of three-dimensional range-gated imaging in scattering environments," in *Electro-Optical Remote Sensing, Photonic Technologies, and Applications V*, vol. 8186, 2011, pp. 27 – 36.

[22] G. Satat, B. Heshmat, D. Raviv, and R. Raskar, "All photons imaging through volumetric scattering," *Scientific Reports*, vol. 6, no. 1, p. 33946, Sept. 2016.

[23] G. Satat, M. Tancik, and R. Raskar, "Towards photography through realistic fog," in *2018 IEEE International Conference on Computational Photography (ICCP)*, May 2018, pp. 1–10.

[24] Bloomberg Businessweek, "Self-driving cars can handle neither rain nor sleet nor snow," *Press Release*, 2018.

[25] New York Times, "5 things that give self-driving cars headaches," *Press Release, 2016*.

[26] Forbes, "How autonomous vehicles will navigate bad weather remains foggy," *Press Release*, 2016.

[27] Tesla, Inc., "Autopilot," 2020. [Online]. Available: https://www.tesla.com/autopilot

[28] K. Beier and H. Gemperlein, "Simulation of infrared detection range at fog conditions for enhanced vision systems in civil aviation," *Aerospace Science and Technology*, vol. 8, no. 1, pp. 63 – 71, 2004.

[29] K. Garcia, M. Yan, and A. Purkovic, "Robust traffic and intersection monitoring using millimeter wave sensors," Texas Instruments, Tech. Rep., 2018.

[30] C. C. Chen, "Attenuation of electromagnetic radiation by haze, fog, clouds, and rain," RAND Corp., Tech. Rep., 1975.

[31] Ford, "Adaptive cruise control," 2020. [Online]. Available: https://www.ford.com/technology/driver-assist-technology/adaptive-cruise-control/

[32] Bosch, "Adaptive cruise control," 2020. [Online]. Available: https://www.bosch-mobility-solutions.com/en/

[33] J. S. Lu, P. Cabrol, D. Steinbach, and R. V. Pragada, "Measurement and characterization of various outdoor 60 ghz diffracted and scattered paths," in *2013 IEEE Military Communications Conference*, Nov. 2013, pp. 1238–1243.

[34] B. Mamandipoor, G. Malysa, A. Arbabian, U. Madhow, and K. Noujeim, "60 ghz synthetic aperture radar for short-range imaging: Theory and experiments," in *2014 48th Asilomar Conference on Signals, Systems and Computers*, Nov. 2014, pp. 553–558.

[35] National Academies of Sciences, Engineering, and Medicine, *Airport Passenger Screening Using Millimeter Wave Machines: Compliance with Guidelines*. The National Academies Press, 2017.

[36] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," in *Advances in Neural Information Processing Systems 27*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds., 2014, pp. 2672–2680.

[37] S. Shahramian, M. J. Holyoak, A. Singh, and Y. Baeyens, "A fully integrated 384-element, 16-tile, w -band phased array with self-alignment and self-test," *IEEE Journal of Solid-State Circuits*, vol. 54, no. 9, pp. 2419–2434, 2019.

[38] S. Zihir, O. D. Gurbuz, A. Kar-Roy, S. Raman, and G. M. Rebeiz, "60-ghz 64- and 256-elements wafer-scale phased-array transmitters using full-reticle and subreticle stitching techniques," *IEEE Transactions on Microwave Theory and Techniques*, vol. 64, no. 12, pp. 4701–4719, Dec. 2016.

[39] Velodyne Lidar, 2020. [Online]. Available: https://velodynelidar.com/

[40] M. M. and S. O., "Conditional generative adversarial nets," *arXiv preprint arXiv:1411.1784*, 2014.

[41] P. Isola, J. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 5967–5976.

[42] O. Ronneberger, P. Fischer, and T. Brox, "U-net: Convolutional networks for biomedical image segmentation," *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2015*, p. 234–241, 2015.

[43] E. Smith and D. Meger, "Improved adversarial systems for 3d object generation and reconstruction," *arXiv preprint arXiv:1707.09557*, 2017.

[44] C. Li, M. Zaheer, Y. Zhang, B. Poczos, and R. Salakhutdinov, "Point cloud gan," *arXiv preprint arXiv:1810.05795*, 2018.

[45] V. Badrinarayanan, A. Kendall, and R. Cipolla, "Segnet: A deep convolutional encoder-decoder architecture for image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 12, pp. 2481–2495, Dec. 2017.

[46] J. Justin, A. Alexandre, and F. Li, "Perceptual losses for real-time style transfer and super-resolution," in *European Conference on Computer Vision*, 2016.

[47] L. A. Gatys, A. S. Ecker, and M. Bethge, "Image style transfer using convolutional neural networks," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 2414–2423.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv preprint arXiv:1409.1556*, 2014.

[49] FUYU Technology, "Linear motion guide," 2020. [Online]. Available: https://www.fuyumotion.com/products/linear-motion-guide

[50] Pasternack, "60 ghz development system," 2014. [Online]. Available: https://www.pasternack.com

[51] Analog Devices, "Adf4159 pll," 2014. [Online]. Available: https://www.analog.com/en/products/adf4159

[52] Ettus Research, "Usrp n210," 2012. [Online]. Available: https://www.ettus.com/all-products/un210-kit

[53] H. Hirschmuller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, Feb. 2008.

[54] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *2017 IEEE International Conference on Computer Vision (ICCV)*, Oct. 2017, pp. 2980–2988.

[55] S. Fidler, S. Dickinson, and R. Urtasun, "3d object detection and viewpoint estimation with a deformable 3d cuboid model," in *Proceedings of the 25th International Conference on Neural Information Processing Systems - Volume 1*, ser. NIPS'12, 2012, p. 611–619.

[56] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The cityscapes dataset for semantic urban scene understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016, pp. 3213–3223.

[57] V. Degli-Esposti, F. Fuschini, E. M. Vitucci, M. Barbiroli, M. Zoli, L. Tian, X. Yin, D. A. Dupleich, R. Müller, C. Schneider, and R. S. Thomä, "Ray-tracing-based mm-wave beamforming assessment," *IEEE Access*, vol. 2, pp. 1314–1325, 2014.

[58] Y. Golovachev, A. Etinger, G. A. Pinhasi, and Y. Pinhasi, "Millimeter wave high resolution radar accuracy in fog conditions—theory and experimental verification," *Sensors*, vol. 18, no. 7, p. 2148, 2018.

[59] Y. Golovachev, A. Etinger, G. Pinhasi, and Y. Pinhasi, "Propagation properties of sub-millimeter waves in foggy conditions," *Journal of Applied Physics*, vol. 125, no. 15, p. 151612, 2019.

[60] D. Glasner, S. Bagon, and M. Irani, "Super-resolution from a single image," in *2009 IEEE International Conference on Computer Vision (ICCV)*, Sept. 2009, pp. 349–356.

[61] A. Guei and M. Akhloufi, "Deep learning enhancement of infrared face images using generative adversarial networks," *Applied Optics*, vol. 57, no. 18, pp. D98–D107, 2018.

[62] C. Ledig, L. Theis, F. Huszár, J. Caballero, A. Cunningham, A. Acosta, A. Aitken, A. Tejani, J. Totz, Z. Wang, and W. Shi, "Photo-realistic single image super-resolution using a generative adversarial network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017, pp. 105–114.

[63] L. Ding and G. Sharma, "Fusing structure from motion and lidar for dense accurate depth map estimation," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Mar. 2017, pp. 1283–1287.

[64] L. Chen, Y. He, J. Chen, Q. Li, and Q. Zou, "Transforming a 3-d lidar point cloud into a 2-d dense depth map through a parameter self-adaptive framework," *IEEE Transactions on Intelligent Transportation Systems*, vol. 18, no. 1, pp. 165–176, Jan. 2017.

[65] Y. He, L. Chen, and M. Li, "Sparse depth map upsampling with rgb image and anisotropic diffusion tensor," in *2015 IEEE Intelligent Vehicles Symposium (IV)*, June 2015, pp. 205–210.

[66] W. Van Gansbeke, D. Neven, B. De Brabandere, and L. Van Gool, "Sparse and noisy lidar completion with rgb guidance and uncertainty," in *2019 16th International Conference on Machine Vision Applications (MVA)*, May 2019, pp. 1–6.

[67] M. Dimitrievski, P. Veelaert, and W. Philips, "Semantically aware multilateral filter for depth upsampling in automotive lidar point clouds," in *2017 IEEE Intelligent Vehicles Symposium (IV)*, June 2017, pp. 1058–1063.

[68] C. Premebida, L. Garrote, A. Asvadi, A. P. Ribeiro, and U. Nunes, "High-resolution lidar-based depth mapping using bilateral filter," in *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*, Nov. 2016, pp. 2469–2474.

[69] L. Csurgai-Horváth and J. Bitó, "Fog attenuation on v band terrestrial radio and a low-cost measurement setup," in *2010 Future Network Mobile Summit*, June 2010, pp. 1–9.

[70] D. M. Sheen, D. L. McMakin, and T. E. Hall, "Near field imaging at microwave and millimeter wave frequencies," in *2007 IEEE/MTT-S International Microwave Symposium*, June 2007, pp. 1693–1696.

[71] T. Savelyev, X. Zhuge, B. Yang, A. Yarovoy, L. Ligthart, M. Drozdov, and B. Levitas, "Development of uwb microwave array radar for concealed weapon detection," in *11-th INTERNATIONAL RADAR SYMPOSIUM*, June 2010, pp. 1–4.

[72] M. T. Ghasr, M. J. Horst, M. R. Dvorsky, and R. Zoughi, "Wideband microwave camera for real-time 3-d imaging," *IEEE Transactions on Antennas and Propagation*, vol. 65, no. 1, pp. 258–268, Jan. 2017.

[73] Y. Meng, A. Qing, C. Lin, J. Zang, Y. Zhao, and C. Zhang, "Passive millimeter wave imaging system based on helical scanning," *Scientific Reports*, vol. 8, no. 1, p. 7852, May 2018.

[74] R. Appleby and R. N. Anderton, "Millimeter-wave and submillimeter-wave imaging for security and surveillance," *Proceedings of the IEEE*, vol. 95, no. 8, pp. 1683–1690, Aug. 2007.

[75] A. Danzer, T. Griebel, M. Bach, and K. Dietmayer, "2d car detection in radar data with pointnets," in *2019 IEEE Intelligent Transportation Systems Conference (ITSC)*, 2019, pp. 61–66.

[76] C. R. Qi, H. Su, K. Mo, and L. J. Guibas, "Pointnet: Deep learning on point sets for 3d classification and segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 652–660.

[77] S. Fang and S. Nirjon, "Ai-enhanced 3d rf representation using low-cost mmwave radar," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018, p. 414–415.

[78] K. Armanious, S. Abdulatif, F. Aziz, U. Schneider, and B. Yang, "An adversarial super-resolution remedy for radar design trade-offs," *2019 27th European Signal Processing Conference (EUSIPCO)*, Sept. 2019.

[79] J. Guan, S. Madani, S. Jog, S. Gupta, and H. Hassanieh, "Through fog high-resolution imaging using millimeter wave radar," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[80] Y. Wang, G. Wang, H.-M. Hsu, H. Liu, and J.-N. Hwang, "Rethinking of radar's role: A camera-radar dataset and systematic annotator via coordinate alignment," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 2815–2824.

[81] D. Barnes, M. Gadd, P. Murcutt, P. Newman, and I. Posner, "The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 6433–6438.

[82] M. Sheeny, E. De Pellegrin, S. Mukherjee, A. Ahrabian, S. Wang, and A. Wallace, "Radiate: A radar dataset for automotive perception," *arXiv preprint arXiv:2010.09076*, vol. 3, no. 4, p. 7, 2020.

[83] B. Major, D. Fontijne, A. Ansari, R. T. Sukhavasi, R. Gowaikar, M. Hamilton, S. Lee, S. Grzechnik, and S. Subramanian, "Vehicle detection with automotive radar using deep learning on range-azimuth-doppler tensors," in *2019 IEEE/CVF International Conference on Computer Vision Workshop (ICCVW)*, 2019, pp. 924–932.

[84] M. Mostajabi, C. M. Wang, D. Ranjan, and G. Hsyu, "High-resolution radar dataset for semi-supervised learning of dynamic objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 100–101.

[85] A. Ouaknine, A. Newson, J. Rebut, F. Tupin, and P. Pérez, "Carrada dataset: camera and automotive radar with range-angle-doppler annotations," in *2020 25th International Conference on Pattern Recognition (ICPR)*. IEEE, 2021, pp. 5068–5075.

[86] M. Bijelic, T. Gruber, F. Mannan, F. Kraus, W. Ritter, K. Dietmayer, and F. Heide, "Seeing through fog without seeing fog: Deep multimodal sensor fusion in unseen adverse weather," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 11 682–11 692.

[87] X. Dong, P. Wang, P. Zhang, and L. Liu, "Probabilistic oriented object detection in automotive radar," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 102–103.

[88] Y. Wang, Z. Jiang, X. Gao, J.-N. Hwang, G. Xing, and H. Liu, "Rodnet: Radar object detection using cross-modal supervision," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 504–513.

[89] X. Gao, G. Xing, S. Roy, and H. Liu, "Ramp-cnn: A novel neural network for enhanced automotive radar object recognition," *IEEE Sensors Journal*, vol. 21, no. 4, p. 5119–5132, Feb 2021.

[90] K. Qian, S. Zhu, X. Zhang, and L. E. Li, "Robust multimodal vehicle detection in foggy weather using complementary lidar and radar signals," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2021, pp. 444–453.

[91] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE conference on computer vision and pattern recognition*. IEEE, 2012, pp. 3354–3361.

[92] Texas Instruments Inc., "mmWave cascade imaging radar RF evaluation module," 2022, [Online]. [Online]. Available: https://www.ti.com/tool/MMWCAS-RF-EVM

[93] Uhnder Inc., "Uhnder - Digital Automotive Radar," 2022. [Online]. Available: https://www.uhnder.com/

[94] S. Cho and S. Lee, "Fast motion deblurring," in *ACM SIGGRAPH Asia 2009 papers*, 2009, pp. 1–8.

[95] Q. Shan, J. Jia, and A. Agarwala, "High-quality motion deblurring from a single image," *Acm transactions on graphics (tog)*, vol. 27, no. 3, pp. 1–10, 2008.

[96] K. Bansal, K. Rungta, S. Zhu, and D. Bharadia, "Pointillism: Accurate 3d bounding box estimation with multi-radars," in *Proceedings of the 18th Conference on Embedded Networked Sensor Systems*, ser. SenSys '20, 2020, p. 340–353.

[97] X. Gao, G. Xing, S. Roy, and H. Liu, "Experiments with mmwave automotive radar test-bed," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 1–6.

[98] F. E. Nowruzi, D. Kolhatkar, P. Kapoor, F. Al Hassanat, E. J. Heravi, R. Laganiere, J. Rebut, and W. Malik, "Deep open space segmentation using automotive radar," in *2020 IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*. IEEE, 2020, pp. 1–4.

[99] A. Zhang, F. E. Nowruzi, and R. Laganiere, "Raddet: Range-azimuth-doppler based radar object detection for dynamic road users," *arXiv preprint arXiv:2105.00363*, 2021.

[100] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Glaeser, F. Timm, W. Wiesbeck, and K. Dietmayer, "Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges," *IEEE Transactions on Intelligent Transportation Systems*, vol. 22, no. 3, pp. 1341–1360, 2020.

[101] Z. Zhang, Z. Tian, and M. Zhou, "Latern: Dynamic continuous hand gesture recognition using fmcw radar sensor," *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3278–3289, 2018.

[102] T. Li, L. Fan, M. Zhao, Y. Liu, and D. Katabi, "Making the invisible visible: Action recognition through walls and occlusions," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2019, pp. 872–881.

[103] C. X. Lu, S. Rosa, P. Zhao, B. Wang, C. Chen, J. A. Stankovic, N. Trigoni, and A. Markham, "See through smoke: Robust indoor mapping with low-cost mmwave radar," in *Proceedings of the 18th International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '20. New York, NY, USA: Association for Computing Machinery, 2020, p. 14–27.

[104] M. Meyer and G. Kuschk, "Automotive radar dataset for deep learning based 3d object detection," in *2019 16th European Radar Conference (EuRAD)*. IEEE, 2019, pp. 129–132.

[105] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom, "nuscenes: A multimodal dataset for autonomous driving," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, 2020, pp. 11 621–11 631.

[106] O. Schumann, C. Wöhler, M. Hahn, and J. Dickmann, "Comparison of random forest and long short-term memory network performances in classification tasks using radar," in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, 2017, pp. 1–6.

[107] O. Schumann, M. Hahn, J. Dickmann, and C. Wöhler, "Semantic segmentation on radar point clouds," in *2018 21st International Conference on Information Fusion (FUSION)*, 2018, pp. 2179–2186.

[108] M. Meyer, G. Kuschk, and S. Tomforde, "Graph convolutional networks for 3d object detection on radar data," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 3060–3069.

[109] A. Ouaknine, A. Newson, P. Perez, F. Tupin, and J. Rebut, "Multi-view radar semantic segmentation," in *Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV)*, October 2021, pp. 15 671–15 680.

[110] M. Shah, Z. Huang, A. Laddha, M. Langford, B. Barber, S. Zhang, C. Vallespi-Gonzalez, and R. Urtasun, "Liranet: End-to-end trajectory prediction using spatio-temporal radar fusion," 2020.

[111] B. Yang, R. Guo, M. Liang, S. Casas, and R. Urtasun, "Radarnet: Exploiting radar for robust perception of dynamic objects," in *European Conference on Computer Vision*. Springer, 2020, pp. 496–512.

[112] T.-Y. Lim, A. Ansari, B. Major, D. Fontijne, M. Hamilton, R. Gowaikar, and S. Subramanian, "Radar and camera early fusion for vehicle detection in advanced driver assistance systems," *NeurIPS Machine Learning for Autonomous Driving Workshop*, 2019.

[113] J. Kim, Y. Kim, and D. Kum, "Low-level sensor fusion network for 3d vehicle detection using radar range-azimuth heatmap and monocular image," in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.

[114] S. Chadwick, W. Maddern, and P. Newman, "Distant vehicle detection using radar and vision," 2019.

[115] Y. Long, D. Morris, X. Liu, M. Castro, P. Chakravarty, and P. Narayanan, "Radar-camera pixel depth association for depth completion," 2021.

[116] R. Nabati and H. Qi, "Centerfusion: Center-based radar and camera fusion for 3d object detection," in *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision (WACV)*, January 2021, pp. 1527–1536.

[117] Y. Kim, J. W. Choi, and D. Kum, "Grif net: Gated region of interest fusion network for robust 3d object detection from radar point cloud and monocular image," in *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2020, pp. 10 857–10 864.

[118] C. Iovescu and S. Rao, "The fundamentals of millimeter wave sensors," *Texas Instruments*, pp. 1–8, 2017.

[119] T. Zeng, C. Mao, C. Hu, X. Yang, and W. Tian, "Multi-static mimo-sar three dimensional deformation measurement system," in *2015 IEEE 5th Asia-Pacific Conference on Synthetic Aperture Radar (APSAR)*, 2015, pp. 297–301.

[120] F. Robey, S. Coutts, D. Weikle, J. McHarg, and K. Cuomo, "Mimo radar theory and experimental results," in *Conference Record of the Thirty-Eighth Asilomar Conference on Signals, Systems and Computers, 2004.*, vol. 1, 2004, pp. 300–304 Vol.1.

[121] J. Bechter, F. Roos, and C. Waldschmidt, "Compensation of motion-induced phase errors in tdm mimo radars," *IEEE Microwave and Wireless Components Letters*, vol. 27, no. 12, pp. 1164–1166, 2017.

[122] A. Manikas, *Beamforming: Sensor Signal Processing for Defence Applications*. World Scientific, 2015, vol. 5.

[123] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," 2019. [Online]. Available: https://github.com/facebookresearch/detectron2

[124] V. Nair and G. E. Hinton, "Rectified linear units improve restricted boltzmann machines," in *Icml*, 2010.

[125] S. Ioffe and C. Szegedy, "Batch normalization: Accelerating deep network training by reducing internal covariate shift," in *International conference on machine learning*. PMLR, 2015, pp. 448–456.

[126] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770–778.

[127] Stereolabs Inc., "Zed Stereo Camera," 2022. [Online]. Available: https://www.stereolabs.com/zed/

[128] Waymo, "A fog blog," 2021. [Online]. Available: https://blog.waymo.com/2021/11/a-fog-blog.html

[129] O. Sorkine, "Least-squares rigid motion using svd," *Technical notes*, vol. 120, no. 3, p. 52, 2009.

[130] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *European conference on computer vision*. Springer, 2014, pp. 740–755.

[131] T.-Y. Lim, S. A. Markowitz, and M. N. Do, "Radical: A synchronized fmcw radar, depth, imu and rgb camera data dataset with low-level fmcw radar signals," *IEEE Journal of Selected Topics in Signal Processing*, vol. 15, no. 4, pp. 941–953, 2021.

[132] S. Gong, Y. Song, T. Manzaneque, R. Lu, Y. Yang, and A. Kourani, "Lithium niobate mems devices and subsystems for radio frequency signal processing," in *2017 IEEE 60th International Midwest Symposium on Circuits and Systems (MWSCAS)*, 2017, pp. 45–48.

[133] M. Rinaldi, C. Zuniga, C. Zuo, and G. Piazza, "Ultra-thin super high frequency two-port aln contour-mode resonators and filters," in *TRANSDUCERS 2009 - 2009 International Solid-State Sensors, Actuators and Microsystems Conference*, 2009, pp. 577–580.

[134] C. Zuo, N. Sinha, and G. Piazza, "Very high frequency channel-select mems filters based on self-coupled piezoelectric aln contour-mode resonators," *Sensors and Actuators A: Physical*, vol. 160, no. 1, pp. 132 – 140, 2010.

[135] Y. Song and S. Gong, "Wideband spurious-free lithium niobate rf-mems filters," *Journal of Microelectromechanical Systems*, vol. 26, no. 4, pp. 820–828, 2017.

[136] R. Lu, T. Manzaneque, Y. Yang, A. Kourani, and S. Gong, "Lithium niobate lateral overtone resonators for low power frequency-hopping applications," in *2018 IEEE Micro Electro Mechanical Systems (MEMS)*, 2018, pp. 751–754.

[137] R. Lu, T. Manzaneque, Y. Yang, J. Zhou, H. Hassanieh, and S. Gong, "Rf filters with periodic passbands for sparse fourier transform-based spectrum sensing," *Journal of Microelectromechanical Systems*, vol. 27, no. 5, pp. 931–944, 2018.

[138] M. Bao and H. Yang, "Squeeze film air damping in mems," *Sensors and Actuators A: Physical*, vol. 136, no. 1, pp. 3 – 27, 2007, 25th Anniversary of Sensors and Actuators A: Physical.

[139] M. Kadota, S. Tanaka, Y. Kuratani, and T. Kimura, "Ultrawide band ladder filter using sh0 plate wave in thin linbo3 plate and its application," in *2014 IEEE International Ultrasonics Symposium*, 2014, pp. 2031–2034.

[140] Federal Communications Commission, "Fcc adopts new rules for the 6 ghz band, unleashing 1,200 megahertz of spectrum for unlicensed use," https://www.fcc.gov/document/fcc-opens-6-ghz-band-wi-fi-and-other-unlicensed-uses, Apr. 2020.

[141] Federal Communications Commission, "3.5 ghz band overview," https://www.fcc.gov/wireless/bureau-divisions/mobility-division/35-ghz-band/35-ghz-band-overview, Jan. 2020.

[142] Federal Communications Commission, "White space," https://www.fcc.gov/general/white-space, Jan. 2020.

[143] Y. Ghasempour, C. R. C. M. da Silva, C. Cordeiro, and E. W. Knightly, "Ieee 802.11ay: Next-generation 60 ghz communication for 100 gb/s wi-fi," *IEEE Communications Magazine*, vol. 55, no. 12, pp. 186–192, 2017.

[144] Y. M. Greshishchev, J. Aguirre, M. Besson, R. Gibbins, C. Falt, P. Flemke, N. Ben-Hamida, D. Pollex, P. Schvan, and S. Wang, "A 40gs/s 6b ADC in 65nm CMOS," in *2010 IEEE International Solid-State Circuits Conference - (ISSCC)*, 2010, pp. 390–391.

[145] B. Murmann, "A/d converter trends: Power dissipation, scaling and digitally assisted architectures," in *2008 IEEE Custom Integrated Circuits Conference*, 2008, pp. 105–112.

[146] DigiKey, "Data acquisition - analog to digital converters (ADC)." [Online]. Available: https://www.digikey.com/products/en/integrated-circuits-ics/data-acquisition-analog-to-digital-converters-adc

[147] *12-Bit, 2.6 GSPS/2.5 GSPS/2.0 GSPS, 1.3 V/2.5 V Analog-to-Digital Converter AD9625 Data Sheet*, Analog Devices, Norwood, MA, USA.

[148] R. T. Yazicigil, T. Haque, M. R. Whalen, J. Yuan, J. Wright, and P. R. Kinget, "Wideband rapid interferer detector exploiting compressed sampling with a quadrature analog-to-information converter," *IEEE Journal of Solid-State Circuits*, Dec 2015.

[149] A. A. Cheema and S. Salous, "Digital fmcw for ultrawideband spectrum sensing," *Radio Science*, vol. 51, no. 8, pp. 1413–1420, 2016.

[150] S. Subramaniam, H. Reyes, and N. Kaabouch, "Spectrum occupancy measurement: An autocorrelation based scanning technique using usrp," in *2015 IEEE 16th Annual Wireless and Microwave Technology Conference (WAMICON)*, 2015, pp. 1–5.

[151] U.S. Government, "CFR title 47 section 96.67 environmental sensing capability," Jan. 2020.

[152] S. K. Sharma, E. Lagunas, S. Chatzinotas, and B. Ottersten, "Application of compressive sensing in cognitive radio communications: A survey," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 1838–1860, 2016.

[153] J. N. Laska, W. F. Bradley, T. W. Rondeau, K. E. Nolan, and B. Vigoda, "Compressive sensing for dynamic spectrum access networks: Techniques and tradeoffs," in *2011 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*, 2011.

[154] M. Mishali and Y. C. Eldar, "From theory to practice: Sub-nyquist sampling of sparse wideband analog signals," *IEEE Journal of Selected Topics in Signal Processing*, Apr. 2010.

[155] H. Hassanieh, L. Shi, O. Abari, E. Hamed, and D. Katabi, "Ghz-wide sensing and decoding using the fourier transform," in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, Apr. 2014.

[156] J. Yoo, S. Becker, M. Loh, M. Monge, E. Candes, and A. Emami-Neyestanak, "A 100mhz–2ghz 12.5 x sub-nyquist rate receiver in 90nm cmos," in *2012 IEEE Radio Frequency Integrated Circuits Symposium*. IEEE, 2012, pp. 31–34.

[157] B. Ghazi, H. Hassanieh, P. Indyk, D. Katabi, E. Price, and L. Shi, "Sample-optimal average-case sparse fourier transform in two dimensions," in *2013 51st Annual Allerton Conference on Communication, Control, and Computing (Allerton)*. IEEE, 2013, pp. 1258–1265.

[158] Y. Guddeti, R. Subbaraman, M. Khazraee, A. Schulman, and D. Bharadia, "Sweepsense: Sensing 5 ghz in 5 milliseconds with low-cost radios," in *16th USENIX Symposium on Networked Systems Design and Implementation (NSDI 19)*, Feb. 2019.

[159] M. Rashidi, K. Haghighi, A. Panahi, and M. Viberg, "A nlls based sub-nyquist rate spectrum sensing for wideband cognitive radio," in *2011 IEEE International Symposium on Dynamic Spectrum Access Networks (DySPAN)*. IEEE, 2011, pp. 545–551.

[160] J. A. Tropp, J. N. Laska, M. F. Duarte, J. K. Romberg, and R. G. Baraniuk, "Beyond nyquist: Efficient sampling of sparse bandlimited signals," *IEEE transactions on information theory*, vol. 56, no. 1, pp. 520–544, 2009.

[161] J. Yoo, C. Turnes, E. B. Nakamura, C. K. Le, S. Becker, E. A. Sovero, M. B. Wakin, M. C. Grant, J. Romberg, A. Emami-Neyestanak, and E. Candes, "A compressed sensing parameter extraction platform for radar pulse signal acquisition," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 2, no. 3, pp. 626–638, 2012.

[162] M. Mishali, Y. C. Eldar, O. Dounaevsky, and E. Shoshan, "Xampling: Analog to digital at sub-nyquist rates," *IET Circuits, Devices Systems*, Jan. 2011.

[163] L. Shi, H. Hassanieh, and D. Katabi, "D-bigband: Sensing ghz-wide non-sparse spectrum on commodity radios," in *Proceedings of the 6th Annual Workshop on Wireless of the Students, by the Students, for the Students*, ser. S3 '14, 2014, p. 13–16.

[164] O. Abari, F. Chen, F. Lim, and V. Stojanović, "Performance trade-offs and design limitations of analog-to-information converter front-ends," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2012, pp. 5309–5312.

[165] O. Abari, F. Lim, F. Chen, and V. Stojanović, "Why analog-to-information converters suffer in high-bandwidth sparse signal applications," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 60, no. 9, pp. 2273–2284, 2013.

[166] Y. Ma, Y. Gao, A. Cavallaro, C. G. Parini, W. Zhang, and Y. Liang, "Sparsity independent sub-nyquist rate wideband spectrum sensing on real-time tv white space," *IEEE Transactions on Vehicular Technology*, vol. 66, no. 10, pp. 8784–8794, 2017.

[167] C. T.-C. Nguyen, "Vibrating RF MEMS overview: applications to wireless communications," in *Micromachining and Microfabrication Process Technology X*, 2005.

[168] A. Ali and W. Hamouda, "Advances on spectrum sensing for cognitive radio networks: Theory and applications," *IEEE Communications Surveys Tutorials*, vol. 19, no. 2, pp. 1277–1304, 2017.

[169] Y. Arjoune and N. Kaabouch, "A comprehensive survey on spectrum sensing in cognitive radio networks: Recent advances, new challenges, and future research directions," *Sensors (Basel, Switzerland)*, vol. 19, no. 1, p. 126, 01 2019.

[170] H. Sun, A. Nallanathan, C. Wang, and Y. Chen, "Wideband spectrum sensing for cognitive radio networks: a survey," *IEEE Wireless Communications*, vol. 20, no. 2, pp. 74–81, April 2013.

[171] D. Bhargavi and C. R. Murthy, "Performance comparison of energy, matched-filter and cyclostationarity-based spectrum sensing," in *2010 IEEE 11th International Workshop on Signal Processing Advances in Wireless Communications (SPAWC)*, June 2010, pp. 1–5.

[172] S. Kapoor, S. Rao, and G. Singh, "Opportunistic spectrum sensing by employing matched filter in cognitive radio network," in *2011 International Conference on Communication Systems and Network Technologies*, June 2011, pp. 580–583.

[173] A. Nasser, A. Mansour, K. C. Yao, H. Charara, and M. Chaitou, "Efficient spectrum sensing approaches based on waveform detection," in *The Third International Conference on e-Technologies and Networks for Development (ICeND2014)*, April 2014, pp. 13–17.

[174] E. Axell and E. G. Larsson, "Optimal and sub-optimal spectrum sensing of ofdm signals in known and unknown noise variance," *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 2, pp. 290–304, Feb. 2011.

[175] S. Hong and S. Katti, "DOF: a local wireless information plane," in *Proceedings of the ACM SIGCOMM 2011 conference*, 2011, pp. 230–241.

[176] T. Yucek and H. Arslan, "A survey of spectrum sensing algorithms for cognitive radio applications," *IEEE communications surveys & tutorials*, vol. 11, no. 1, pp. 116–130, 2009.

[177] P. Bahl, R. Chandra, T. Moscibroda, R. Murty, and M. Welsh, "White space networking with wi-fi like connectivity," *ACM SIGCOMM Computer Communication Review*, vol. 39, no. 4, pp. 27–38, 2009.

[178] H. Rahul, N. Kushman, D. Katabi, C. Sodini, and F. Edalat, "Learning to share: narrowband-friendly wideband networks," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 4, pp. 147–158, 2008.

[179] S. Yoon, L. E. Li, S. C. Liew, R. R. Choudhury, I. Rhee, and K. Tan, "Quicksense: Fast and energy-efficient channel sensing for dynamic spectrum access networks," in *2013 Proceedings IEEE INFOCOM*, April 2013.

[180] J. Su and W. Wu, "Wireless spectrum prediction model based on time series analysis method," in *Proceedings of the 2009 ACM workshop on Cognitive radio networks*, 2009, pp. 61–66.

[181] L. Shi, P. Bahl, and D. Katabi, "Beyond sensing: Multi-ghz realtime spectrum analytics," in *12th USENIX Symposium on Networked Systems Design and Implementation (NSDI 15)*, May 2015.

[182] The Software Defined Radio Forum Inc, "Application of management technologies in dynamic spectrum sharing," Jul. 2019.

[183] M. Khazraee, Y. Guddeti, S. Crow, A. C. Snoeren, K. Levchenko, D. Bharadia, and A. Schulman, "Sparsdr: Sparsity-proportional backhaul and compute for sdrs," in *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, ser. MobiSys '19, 2019, p. 391–403.

[184] C. T. C. Nguyen, "Mems-based rf channel selection for true software-defined cognitive radio and low-power sensor communications," *IEEE Communications Magazine*, vol. 51, no. 4, pp. 110–119, 2013.

[185] T. Mukherjee, G. K. Fedder, H. Akyol, U. Arslan, J. Brotz, F. Chen, A. Jajoo, C. Lo, A. Oz, D. P. Ramachandran, V. K. Saraf, M. Sperling, and J. Stillman, "Reconfigurable mems-enabled rf circuits for spectrum sensing," in *Government Microcircuit Applications and Critical Technology Conference*, 2005.

[186] X. G. Xia, "On estimation of multiple frequencies in undersampled complex valued waveforms," *IEEE transactions on signal processing*, vol. 47, no. 12, pp. 3417–3419, 1999.

[187] X. G. Xia, "An efficient frequency-determination algorithm from multiple undersampled waveforms," *IEEE Signal Processing Letters*, vol. 7, no. 2, pp. 34–37, 2000.

[188] P. P. Vaidyanathan and P. Pal, "Sparse sensing with co-prime samplers and arrays," *IEEE Transactions on Signal Processing*, vol. 59, no. 2, pp. 573–586, 2011.

[189] C. Herley and Ping Wah Wong, "Minimum rate sampling and reconstruction of signals with arbitrary frequency support," *IEEE Transactions on Information Theory*, vol. 45, no. 5, pp. 1555–1564, 1999.

[190] R. Venkataramani and Y. Bresler, "Perfect reconstruction formulas and bounds on aliasing error in sub-nyquist nonuniform sampling of multiband signals," *IEEE Transactions on Information Theory*, vol. 46, no. 6, pp. 2173–2183, 2000.

[191] N. Tzou, D. Bhatta, S. Hsiao, H. W. Choi, and A. Chatterjee, "Low-cost wideband periodic signal reconstruction using incoherent undersampling and back-end cost optimization," in *2012 IEEE International Test Conference*, 2012, pp. 1–10.

[192] A. J. Silva, "Reconstruction of undersampled periodic signals," *MIT Technical Report*, 1986.

[193] D. Pfammatter, D. Giustiniano, and V. Lenders, "A software-defined sensor architecture for large-scale wideband spectrum monitoring," in *Proceedings of the 14th International Conference on Information Processing in Sensor Networks*, 2015, pp. 71–82.

[194] A. Chakraborty, M. S. Rahman, H. Gupta, and S. R. Das, "Specsense: Crowdsensing for efficient querying of spectrum occupancy," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[195] Y. Zeng, V. Chandrasekaran, S. Banerjee, and D. Giustiniano, "A framework for analyzing spectrum characteristics in large spatio-temporal scales," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[196] H. Sudo, K. Kosaka, H. Kanemoto, N. Gejoh, T. Yasunaga, and M. Uesugi, "Study of spectrum sensing scheme using received power within preamble signals," in *2017 20th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, 2017, pp. 592–597.

[197] A. Thierer and A. Castillo, "Projecting the growth and economic impact of the internet of things," *George Mason University, Mercatus Center, June*, vol. 15, 2015.

[198] D. Vasisht, Z. Kapetanovic, J. Won, X. Jin, R. Chandra, S. Sinha, A. Kapoor, M. Sudarshan, and S. Stratman, "Farmbeats: An iot platform for data-driven agriculture," in *14th USENIX Symposium on Networked Systems Design and Implementation (NSDI 17)*, 2017, pp. 515–529.

[199] S. Kumar and A. Jasuja, "Air quality monitoring system based on IoT using raspberry pi," in *2017 International Conference on Computing, Communication and Automation (ICCCA)*. IEEE, 2017, pp. 1341–1346.

[200] E. Manavalan and K. Jayakrishna, "A review of internet of things (IoT) embedded sustainable supply chain for industry 4.0 requirements," *Computers & Industrial Engineering*, vol. 127, pp. 925–953, 2019.

[201] R. Nandakumar, V. Iyer, and S. Gollakota, "3d localization for sub-centimeter sized devices," in *Proceedings of the 16th ACM Conference on Embedded Networked Sensor Systems*, 2018, pp. 108–119.

[202] M. Ibrahim and M. Youssef, "A hidden markov model for localization using low-end gsm cell phones," in *2011 IEEE International Conference on Communications (ICC)*. IEEE, 2011, pp. 1–5.

[203] J. Xiong, K. Sundaresan, and K. Jamieson, "Tonetrack: Leveraging frequency-agile radios for time-based indoor wireless localization," in *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, 2015, pp. 537–549.

[204] Y. Hao, M. Chen, L. Hu, J. Song, M. Volk, and I. Humar, "Wireless fractal ultra-dense cellular networks," *Sensors*, vol. 17, no. 4, p. 841, 2017.

[205] P. Subrahmanya and A. Farajidana, "5g and beyond: Physical layer guiding principles and realization," *Journal of the Indian Institute of Science*, vol. 100, pp. 263–279, 2020.

[206] 3GPP, "Study on narrow-band internet of things (NB-IoT) / enhanced machine type communication (eMTC) support for non-terrestrial networks (NTN)," 3rd Generation Partnership Project (3GPP), Technical Report (TR) 36.763, 06 2021.

[207] G. A. Akpakwu, B. J. Silva, G. P. Hancke, and A. M. Abu-Mahfouz, "A survey on 5g networks for the internet of things: Communication technologies and challenges," *IEEE access*, vol. 6, pp. 3619–3647, 2017.

[208] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti, "Spotfi: Decimeter level localization using wifi," in *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, 2015, pp. 269–282.

[209] Y. Xie, J. Xiong, M. Li, and K. Jamieson, "md-track: Leveraging multi-dimensionality for passive indoor wi-fi tracking," in *The 25th Annual International Conference on Mobile Computing and Networking*, 2019, pp. 1–16.

[210] M. Ibrahim and M. Youssef, "Cellsense: An accurate energy-efficient gsm positioning system," *IEEE Transactions on Vehicular Technology*, vol. 61, no. 1, pp. 286–296, 2011.

[211] J. Paek, K.-H. Kim, J. P. Singh, and R. Govindan, "Energy-efficient positioning for smartphones using cell-id sequence matching," in *Proceedings of the 9th international conference on Mobile systems, applications, and services*, 2011, pp. 293–306.

[212] H. Aly and M. Youssef, "Dejavu: an accurate energy-efficient outdoor localization system," in *Proceedings of the 21st ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2013, pp. 154–163.

[213] A. Shokry, M. Torki, and M. Youssef, "Deeploc: a ubiquitous accurate and low-overhead outdoor cellular localization system," in *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, 2018, pp. 339–348.

[214] F. Wen, H. Wymeersch, B. Peng, W. P. Tay, H. C. So, and D. Yang, "A survey on 5g massive mimo localization," *Digital Signal Processing*, vol. 94, pp. 21–28, 2019.

[215] J. Palacios, G. Bielsa, P. Casari, and J. Widmer, "Communication-driven localization and mapping for millimeter wave networks," in *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*. IEEE, 2018, pp. 2402–2410.

[216] J. Palacios, P. Casari, and J. Widmer, "Jade: Zero-knowledge device localization and environment mapping for millimeter wave systems," in *IEEE INFOCOM 2017-IEEE Conference on Computer Communications*. IEEE, 2017, pp. 1–9.

[217] H. Sallouha, A. Chiumento, and S. Pollin, "Localization in long-range ultra narrow band IoT networks using rssi," in *2017 IEEE International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–6.

[218] S. Gezici and Z. Sahinoglu, "Uwb geolocation techniques for ieee 802.15.4a personal area networks," *MERL Technical report*, 2004.

[219] F. Gustafsson and F. Gunnarsson, "Mobile positioning using wireless networks: possibilities and fundamental limitations based on available wireless network measurements," *IEEE Signal processing magazine*, vol. 22, no. 4, pp. 41–53, 2005.

[220] J. Wang and D. Katabi, "Dude, where's my card? rfid positioning that works with multipath and non-line of sight," in *Proceedings of the ACM SIGCOMM 2013 conference on SIGCOMM*, 2013, pp. 51–62.

[221] I. Guvenc and C.-C. Chong, "A survey on toa based wireless localization and nlos mitigation techniques," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 3, pp. 107–124, 2009.

[222] M. Bouet and A. L. Dos Santos, "Rfid tags: Positioning principles and localization techniques," in *2008 1st IFIP Wireless Days*. IEEE, 2008, pp. 1–5.

[223] V. Iyer, R. Nandakumar, A. Wang, S. B. Fuller, and S. Gollakota, "Living iot: A flying wireless platform on live insects," in *The 25th Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '19, 2019.

[224] S. Naderiparizi, Y. Zhao, J. Youngquist, A. P. Sample, and J. R. Smith, "Self-localizing battery-free cameras," in *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, 2015, pp. 445–449.

[225] C. Yang and H.-R. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.

[226] S. Boonsriwai and A. Apavatjrut, "Indoor wifi localization on mobile devices," in *2013 10th International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology*. IEEE, 2013, pp. 1–5.

[227] A. Rai, K. K. Chintalapudi, V. N. Padmanabhan, and R. Sen, "Zee: Zero-effort crowdsourcing for indoor localization," in *Proceedings of the 18th annual international conference on Mobile computing and networking*, 2012, pp. 293–304.

[228] C. Zhang, P. Patras, and H. Haddadi, "Deep learning in mobile and wireless networking: A survey," *IEEE Communications surveys & tutorials*, vol. 21, no. 3, pp. 2224–2287, 2019.

[229] A. Marcaletti, M. Rea, D. Giustiniano, V. Lenders, and A. Fakhreddine, "Filtering noisy 802.11 time-of-flight ranging measurements," in *Proceedings of the 10th ACM International on Conference on emerging Networking Experiments and Technologies*, 2014, pp. 13–20.

[230] S. Boyd, S. P. Boyd, and L. Vandenberghe, *Convex optimization*. Cambridge university press, 2004.

[231] A. Omri, M. Shaqfeh, A. Ali, and H. Alnuweiri, "Synchronization procedure in 5g nr systems," *IEEE Access*, vol. 7, pp. 41 286–41 295, 2019.

[232] CellMapper, "Signal tiles and towers." [Online]. Available: https://www.cellmapper.net

[233] 3GPP, "5G; NR; Physical channels and modulation," 3rd Generation Partnership Project (3GPP), Technical Specification (TS) 138.211, 07 2020, version 16.2.0.