

Radar Sensing Systems for Smarter Indoor Human Environments

A Dissertation

Presented to
the faculty of the School of Engineering and Applied Science
University of Virginia

in partial fulfillment
of the requirements for the degree

Doctor of Philosophy

by

Avinash Kalyanaraman

May 2020

APPROVAL SHEET

The dissertation
in submitted in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy

Avinash Kalyanaraman

AUTHOR

The dissertation has been read and approved by the examining committee:

Kamin Whitehouse (Advisor)

John A. Stankovic (Chair)

Gabriel Robins

Brad Campbell

Robert M. Weikle, II

Accepted for the School of Engineering and Applied Science:

A handwritten signature in black ink, appearing to read 'CHB'. The letters are cursive and connected, with a long horizontal stroke extending from the end of the 'B'.

Craig H. Benson, Dean, School of Engineering and Applied Science

Abstract

Today, on average, humans spend over 90% of their time indoors. Given the considerable amount of time spent indoors, sensing context in these human-centric indoor environments can enable several applications in healthcare, energy, elderly monitoring, personal welfare, etc. Despite the potential slew of applications, people are resistant to having cameras and/or microphones in many indoor human environments such as homes, office cubicles or automobiles, due to privacy concerns. Furthermore, this context sensing must be performed without an onus on the human to wear/carry any device, owing to the well documented *forget to wear, forget to charge* problem.

Given these sensing constraints, this dissertation explores the usage of wireless radar signals to sense context in indoor human environments. Quite unlike their primary usage in outdoor spaces, radars in indoor environments are subject to (a) very strong indoor multipath reflections, (b) power consumption constraints, (c) space constraints, (d) transmit power regulations, and (e) an incomplete observation of the entire sensing region (i.e. partial field-of-view).

We address these challenges by building a suite of hardware and software solutions that builds on past radar literature to sense context in two indoor human environments — homes and cars, by leveraging the structure of the environment. In particular, the components of *FormaTrack* and *Doorpler* perform room-level localization of home occupants using radar sensors mounted atop room transition spots such as doorways via the Doppler Effect. To prevent sensing errors produced by these devices from becoming tracking errors, we build *TransTrack* – a tracking algorithm that uses sensor data from (subsequent) doorway events. Finally, we build *CaraoKey*, a system that repurposes the radar setup that pre-exists in automobiles for keyless entry, as a sensing modality to infer the state of a car. It does so in a manner that is robust to location changes and does not warrant any form of transceiver synchroniza-

tion. The dissertation concludes by pointing out other context sensing applications that can be enabled by this infrastructure. Such features will become increasingly important as driverless cars and shuttles become the norm, resulting in an increased importance to passengers' sense of in-vehicle security and well-being.

Acknowledgements

I owe my deepest gratitude to Kamin Whitehouse, my advisor for his endless support, mentorship and patience during the course of this research. I have learnt a lot from my advisor that has not just made me a better researcher and communicator over time, but also a better person. He stood by me all along through grad school, granting me immense freedom to fail and learn. His emphasis on the importance of communicating research with phrases such as “draw it out”, “your talk and slides should be understandable by a 5th grader” will stay with me forever. I am also thankful for his emphasis on having a balanced and healthy personal life with sayings such as “you need to have at least three (non-research) memorable experiences every year.” Kamin will be my academic role model. I walked out of every meeting with Kamin more inspired and enlightened than I walked into one. Similarly, I hope to carry this legacy moving forward.

I would like to thank my committee members Jack Stankovic, Gabriel Robins, Brad Campbell and Bobby Weikle for being very accessible and supportive to this work. They have always given me time despite their busy schedules during various stages of this work. Bobby Weikle, Steve Bowers and Scott Barker have also been very helpful by lending their RF equipments periodically. I am also thankful to my Master’s advisor, Andrew Grimshaw who taught me something very valuable early in my graduate life – “learning to say no!”.

Thanks to the Wireless Connectivity Group at Bosch Research for a productive internship – Vivek Jain, Yunze Zeng, Sushanta Rakshit and Martin Coors ensured that I had a comfortable learning experience at Bosch. While at Bosch, Ashwin and Sharadha ensured that my Bay Area move and stay was comfortable, even finding a cricket team for me before I reached the area. Thanks also to the Mobile Communication and Networking Group at NEC Labs – Ayon Chakraborty and Karthik

Sundaresan helped provide a wonderful learning opportunity.

Being part of the Smart Home Lab has been a super enjoyable experience. I have spent countless hours biting the heads of my labmates – Juhi Ranjan, Erin Griffiths, Dezhi Hong, Nipun Batra, Elahe Soltanaghaei and Mahmoud Elnaggar. They have all been super patient all along to brainstorm and give feedback at short notice, even after graduation. I am grateful to Vijay Srinivasan, Vidyabhushan Mohan and Sirajum Munir for their constant support as friend and mentor. I am also thankful to my other Rice Hall conversationalists – In Kee Kim, Mohsin Ahmed, Abu Mondol and Jack Lanchantin for their conversations pondering life, universe and everything.

I owe immense gratitude to Divya and Ritambhara, my best friends through grad school. The last 6 years flew thanks to you both. Climbing, backpacking, (freezing weather) running and the 6am gym visits would have been half as fun without you.

I am super thankful to members of the Charlottesville Cricket Club (CVCC) who have been far more than just team-mates aiming to win cricket matches. They have been my family away from home. Playing competitive Cricket with CVCC all over Virginia will be my favorite memory from my Charlottesville stay.

Finally, this work would not have been possible without the unconditional love and sacrifices of my parents and sisters. They have been with me through the ups and downs. They provided me from a very young age, the opportunity and freedom to fail, which I am forever indebted to. Thanks, Amma, Appa, Shalini and Aarthi for all that you are and for everything that you have done for me. This dissertation is dedicated to you.

To my Parents and Sisters

Contents

1	Introduction	7
1.1	Overview and Motivation	7
1.2	Outline	11
1.3	Technical Challenges	13
1.4	Contributions	15
2	Background	18
2.1	Radar Background	18
2.2	Related Work	21
2.2.1	Identity Sensing Systems	21
2.2.2	Device-based localization	23
2.2.3	Radar-based Sensing	23
2.2.4	Occupancy Sensing	25
2.2.5	Multi Target Tracking	26
2.2.6	Non-intrusive Vehicular Sensing	28
3	Room-level Tracking of People based on Body Shape	29
3.1	Introduction	29
3.2	Approach	31
3.2.1	Crossing Detection:	33
3.2.2	Direction Estimation	35

3.2.3	Body Shape Sensing	41
3.3	Experimental Setup	42
3.4	Evaluation	45
3.4.1	System Accuracy	45
3.4.2	Effect of False Positives	46
3.4.3	Effect of Number of People	47
3.4.4	Energy Analysis	47
3.4.5	Down sampling in Slow Time:	48
3.4.6	Down sampling in Fast Time:	48
3.4.7	Effect of Training Size	49
3.4.8	Effect of Objects	49
3.5	Limitations	51
3.5.1	In-situ study	51
3.5.2	Effect of Doors	51
3.5.3	Effect of walking up to the doorway and turning back	51
3.5.4	Effect of running through the doorway	52
3.6	Summary	52
4	Improving Room-level Tracking by Data Fusion	53
4.1	Introduction	53
4.2	Approach	56
4.3	Implementation	58
4.4	Experimental setup	67
4.5	Results	69
4.5.1	TransTrack on FormaTrack data	74
4.6	Summary	75

5	Towards Low Power Room-level Tracking	76
5.1	Introduction	76
5.2	Approach	79
5.2.1	Crossing Detection	80
5.2.2	Direction Estimation	83
5.3	Experimental Setup	91
5.4	Evaluation	93
5.4.1	Doorpler Accuracy	93
5.4.2	Power Consumption	94
5.4.3	Real-timeness of <i>Doorpler</i>	98
5.4.4	Effect of Number of Receive Chains	99
5.4.5	Effect of Transmit Power	99
5.4.6	Effect of Wake-up Time	100
5.4.7	Effect of Degree of Spectrogram Overlap	101
5.5	Limitations	101
5.5.1	The Integrated System	101
5.5.2	Multi-Person Crossings	102
5.5.3	Effect of Pets and Doors	102
5.6	Summary	103
6	Car State Sensing via the UWB Keyless Infrastructure	104
6.1	Introduction	104
6.2	Background	108
6.3	Approach	111
6.3.1	Design Goals	111
6.3.2	Overview	112
6.3.3	Connectivity Test – Transmitter Selection	113

6.3.4	CIR Upsampling	115
6.3.5	CIR Alignment	115
6.3.6	Receiver Selection	116
6.3.7	State Identification	118
6.4	Implementation and Experimental Setup	123
6.5	Evaluation	125
6.5.1	System Accuracy	125
6.5.2	Baseline Comparison	126
6.5.3	Effect of Number of Training Locations	126
6.5.4	Effect of Number of Internal Nodes	127
6.5.5	Power Analysis	128
6.5.6	Effect of Tap Resolution	130
6.5.7	Effect of CIR Length	130
6.6	Discussion	131
6.7	Summary	133
7	Conclusion	134
7.1	Looking Beyond	136

List of Figures

3.1	An example of a radar frame	33
3.2	Energy Reflections During a Crossing	33
3.3	Distance to the head from the radar during a crossing	34
3.4	Range of the distance to head estimates	34
3.5	Range-Doppler Matrix formation	37
3.6	Range-Doppler Matrix in <i>FormaTrack</i>	38
3.7	Doppler during a doorway crossing	38
3.8	Approach-Away Power during a crossing	40
3.9	An example of Reflection Profile	40
3.10	<i>FormaTrack</i> setup	43
3.11	<i>FormaTrack</i> evaluation scenarios	43
3.12	Comparison with a height biometric baseline	46
3.13	Downsampling <i>FormaTrack</i> in slow time	48
3.14	Downsampling <i>FormaTrack</i> in fast time	48
3.15	Effect of training size on <i>FormaTrack's</i> identity accuracy	50
3.16	Effect of objects on <i>FormaTrack's</i> identity accuracy	50
4.1	The overall operation of a Multiple Hypothesis Tracking Algorithm	59
4.2	<i>TransTrack</i> performance with respect to baseline systems	69
4.3	<i>TransTrack</i> Scoring function variants	70
4.4	Sensitivity analysis of <i>TransTrack</i> to the maximum number of targets	70

4.5	<i>TransTrack</i> error distribution in controlled and in-situ studies	70
4.6	Degree of future analysis of <i>TransTrack</i>	72
4.7	Effect of sensing error removal on <i>TransTrack</i>	73
4.8	<i>TransTrack</i> 's target count accuracy	73
4.9	<i>TransTrack</i> 's pruning strategy	73
4.10	Whole home tracking simulation of <i>FormaTrack</i>	74
5.1	The envelope of the Doppler signal reflected back to <i>Doorpler</i> during a doorway crossing (a peak in the figure) is much larger than that in the absence of a crossing. There are 12 crossings in this figure.	79
5.2	Positive and Negative Doppler Shift Examples	80
5.3	Amplitude Spectrogram	82
5.4	Amplitude Spectrogram after Spectrogram Enhancement	82
5.5	Doppler Power during a doorway crossing	82
5.6	Phase Spectrogram	84
5.7	Angle of Arrival (AoA)	84
5.8	Angle of Arrival Spectrogram	84
5.9	Quantized AoA Spectrogram	86
5.10	Consensus AoA Spectrogram	86
5.11	Weighted Consensus Spectrogram	86
5.12	<i>Doorpler</i> Experimental Setup	90
5.13	<i>Doorpler</i> scripted study evaluation scenarios	91
5.14	High-level <i>Doorpler</i> block diagram	97
5.15	Variation of <i>Doorpler</i> transmit power	97
5.16	<i>FormaTrack</i> with a PIR motion sensor	98
5.17	Effect of number of receive chains in <i>Doorpler</i>	99
5.18	Effect of <i>Doorpler</i> wake-up time	100

5.19	Effect of <i>Doorpler's</i> degree of spectrogram overlap	102
6.1	The UWB keyless infrastructure can enable several applications that can benefit from a pre-existing deployment. <i>CaraoKey</i> detects car states using such an infrastructure.	105
6.2	The CIR is representative of the multipath reflections inside the car. Changes in this CIR are leveraged to infer the car state.	106
6.3	An example of two features contained in the <i>multipath profile</i> that aid in state identification. The other features in the profile help improve the state classification.	106
6.4	An UWB packet consists of header and payload symbols. Each UWB symbol contains multiple narrow pulses. The preamble pulses help compute the CIR.	106
6.5	<i>CaraoKey</i> overview	111
6.6	<i>CaraoKey</i> deployment	112
6.7	<i>CaraoKey</i> Connectivity Test	112
6.8	<i>CaraoKey</i> RSS in cars	112
6.9	Example of CIR upsampling	113
6.10	Example of CIRs pre-alignment	113
6.11	Example of CIRs post-alignment	113
6.12	Example of CIRs in indoor and outdoor environments	117
6.13	Correlation of interior and exterior nodes' CIRs	117
6.14	<i>CaraoKey</i> Step 1 Illustration	118
6.15	Multipath Profile	118
6.16	<i>CaraoKey's</i> states of interest	119
6.17	<i>CaraoKey's</i> experimental locations and scenarios	121
6.18	<i>CaraoKey</i> implementation	122

6.19	<i>CaraoKey</i> under leave-one-out cross validation	122
6.20	<i>CaraoKey</i> under pure location independence	122
6.21	<i>CaraoKey's</i> accuracy in comparison with baseline systems	124
6.22	Effect of number of training locations on <i>CaraoKey's</i> accuracy	124
6.23	Effect of number of internal receivers on <i>CaraoKey's</i> accuracy	124
6.24	<i>CaraoKey</i> power draw during a transmit and receive event	129
6.25	Power consumption of a <i>CaraoKey</i> transmitter and receiver, and life- time of <i>CaraoKey</i> under different blinking rates	130
6.26	Effect of fast time downsampling on <i>CaraoKey's</i> accuracy	130
6.27	Effect of CIR window length on <i>CaraoKey's</i> accuracy	130

List of Tables

3.1	<i>FormaTrack</i> participant details	43
3.2	<i>FormaTrack</i> results for metrics of interest	46
4.1	Example of <i>Hidden Target Problem</i>	62
4.2	<i>TransTrack</i> study details	67
5.1	Participant details of <i>Doorpler</i> scripted study	92
5.2	<i>Doorpler</i> accuracy across metrics of interest	94
5.3	<i>Doorpler</i> power consumption comparison	95

Chapter 1

Introduction

1.1 Overview and Motivation

The term *environment* refers to the world around us – the aerial and aquatic life, the mountains, our homes and offices, our automobiles, etc. This environment can be classified into two main heads based on the nature of origin – the *natural environment* and the *built environment* [1]. The natural environment refers to our surroundings that came into existence without human intervention such as oceans, forests etc. On the other hand, *the built environment* refers to the human constructed space around us such as homes, automobiles, bridges, dams, the drainage system, etc.

In this dissertation, we refer to the built environment that is occupied by humans as a *human environment* – e.g. buildings, cars, etc. According to the Environmental Protection Agency (EPA), the average American spends over 90% of one’s life in an indoor human environment – 87% in spaces such as a home or an office, and 6% of one’s life in automobiles [2]. Given the considerable amount of time spent inside an indoor human environment, sensing context of these environments can start to enable several applications that can improve the quality of our lives. Such applications include healthcare, personal welfare, energy savings and safety/security systems to

name a few.

However, many of these indoor human environments such as our homes, our automobiles, our office cubicles, etc. are privacy sensitive. We refer to such built environments as *private indoor human environments*. In such spaces, people do not prefer to have their audio or video recorded as a sensing modality. Put another way, while people accept such privacy invasive sensors in public spaces such as commercial buildings like shopping malls and office hallways, they are resistant to having these sensors in the aforementioned *private indoor human environment*. This motivates the need for a non-invasive and privacy preserving sensing modality to infer context.

To satisfy this sensing need, this dissertation explores the possibility of using wireless signals to sense context in private indoor human environments. Given the multitude of wireless enabled devices that already exist in our private living spaces such as WiFi, Bluetooth, GSM or FM transceivers, people are already accustomed to have wireless devices in their private spaces. Hence, this dissertation explores the usage of such wireless signals as a sensing modality. Wireless signals when emitted by a transmitter (contained in the environment), travel through the indoor human environment, getting reflected by the objects (and humans) present in the environment, before reaching the receiver. As a result, information about the environment is contained within these wireless signals. This dissertation analyzes this complex received wireless signal, which is a combination of a plethora of reflections, to make sense of the environment.

Radars, first built nearly a century ago, aim to achieve this in outdoor environments. These devices drew immense popularity during World War II where they were primarily used to detect enemy aircrafts [3]. These radar devices emit radio frequency (RF) waves which bounce off the outdoor environment. These reflections are analyzed by a receiver to make inferences about the outdoor environment. Quite unlike their primary usage in outdoor spaces, radars in *indoor human environments* such

as homes and automobiles are subject to strong indoor *multipath*. Said differently, in these environments, the transmitted wireless signal reflects not just of the human but also other objects in the environment. To make things worse, these (primary) reflections can get further reflected by (other) objects creating secondary and tertiary reflections. When all these reflections arrive at the receiver, the key challenge for the receiver lies in separating out the reflection of interest. In this dissertation, we develop novel algorithms that help perform context sensing in indoor human environments despite the rich multipath by leveraging the structure of the environment.

Next, despite the potential benefits of context sensing applications to humans, people cannot be expected to wear/carry a device at all times – referred to as the *forget to wear, forget to charge problem* [4]. This motivates the need for a device-free sensing mechanism. Consequently, this dissertation explores the usage of wireless sensing in a device-free manner – i.e. there is no onus on any individual to carry or wear a device.

Also, unlike exterior radar sensing systems where power consumption is not a constraint, *indoor human environments* such as homes and cars treat power as a scarce resource. Power is a precious resource in homes [4] because power outlets might not readily be available, where necessary (particularly near vantage sensing points such as doorways). Furthermore, people cannot be expected to draw long wires to power these sensors, as they become aesthetically very unfriendly to have in a home. While environments such as cars benefit from a battery, the supply is still limited and shared by other components in the car. Consequently, in this research we aim to build low-power sensing solutions that adhere to the power constraints of the environment. Additionally, radars built for outdoor environments typically do not have transmit power restrictions. However, radars designed for indoor human environments must adhere to FCC’s requirement on power levels for safety reasons. The challenge though is weaker the transmitted signal, the weaker becomes the human-induced reflection

that needs to be understood.

Furthermore, radar systems built for outdoor environments have spatial luxury – i.e. there is loose constraint on the space of the antenna array (*diameter*) of the radar. This is because the angular resolution of the radar is inversely proportional to the diameter (length of a linear array). However, for indoor human environments, space is a precious resource because of both aesthetics and the lack of available deployment space (e.g. in cars or in vantage sensing points like doorways of a building). Consequently, in this research we aim to build space-aware radar sensing systems that adhere to the space constraints of the indoor human environment.

Finally, from a convenience and practicality standpoint we want our radar system to be easy to deploy/install. Said differently, we do not want a system that is expensive and cumbersome to deploy – e.g. like smart-floors [5] in homes. Similarly in environments like automobiles, additional sensors come with secondary wiring and harnessing costs for manufacturers. This gets exacerbated as power might not readily be available where the sensors are deployed. Consequently, in this research we aim to build radar sensing systems that is easy to deploy in the indoor human environment of interest.

This leads us to the hypothesis of our research which states that “*we can enable context sensing in indoor human environments in a non-invasive, device-free, low power, easy to deploy/install, space-constrained manner with a higher accuracy than state of the art using FCC-compliant COTS hardware by applying radar principles that leverage the structure of the multipath rich environment*”. To test this hypothesis, we build *Panoptes*, a suite of hardware and software solutions that builds on past radar literature. *Panoptes* deals with the two private human environments we spend the most time in – homes and cars [2]. We perform context sensing in a home by inferring the room location of the occupants in a home (with identity). We perform context sensing in cars by sensing the state of a car – unoccupied, occupied, door open, window

open, trunk open etc. *Panoptes* performs context sensing using consumer electronics available to ordinary uses (i.e. non military hardware). Each component of *Panoptes* is FCC compliant and is implemented in practice - i.e. the components were physically implemented, deployed and empirically evaluated to demonstrate their feasibility.

1.2 Outline

The rest of this dissertation which explains *Panoptes* in greater detail, is organized as follows.

Chapter 2 provides a background of radars, and how the different pieces of *Panoptes* is related to prior works in identity sensing, device-based localization, radar-sensing, occupancy sensing, multi-target tracking and non-intrusive sensing in vehicles.

Chapter 3 introduces *FormaTrack*, an impulse radar based system for inferring the room location of the occupants of a home by sensing at room transition points such as doorways. It does so by introducing body shape as a new weak biometric (i.e. biometrics that are not globally unique but distinct for a small population such as a home). The basic idea is to scan the body with a radar sensor when the person is exactly at the doorway, and to compute the *reflection profile*: the amount of energy that reflects back from each part of the body. By sensing the occurrence of each crossing, the corresponding direction of transition and the person involved, *FormaTrack* identifies the room location of its inhabitants.

Like any piece of hardware, the *FormaTrack* radar can produce sensing errors. These errors can manifest in four main ways -(i) false positives (when *FormaTrack* detects someone walking through a doorway when they have not), (ii) false negatives (when *FormaTrack* misses someone walking through a doorway), (iii) direction errors (*FormaTrack* indicates a person likely moved from room1 to room2, but in reality the person moved from room2 to room1), and (iv) identity errors (the computed

reflection profile looked similar to person P2 as opposed to person P1). Chapter 4 introduces *TransTrack*, inspired by the *multiple hypothesis tracking algorithm* from radar literature [6]. *TransTrack* uses the intuition that *future can disambiguate the past* – i.e. it leverages data from subsequent doorway crossings to fix the aforementioned sensing errors. Said differently, *TransTrack* prevents sensing errors from becoming tracking errors.

Chapter 5 addresses another important metric of practicality, namely power. It makes the observation that the high-power *FormaTrack* identity sensor is powered on at all times. Ideally, it would need to be powered off at all times, and powered on only when someone is at the doorway. To realize this, Chapter 5 introduces *Doorpler*, a Doppler-radar based sensing system that adheres to the time, space and power constraints of the application. It detects a crossing via the *Doppler Principle*, and infers the direction of crossing by measuring the angle-of-arrival of the human reflection. By simply performing crossing detection and direction estimation from atop a doorway, other applications such as automatic heating and lighting control, elderly monitoring in the 13 million elderly single-person households, or “eyes-off” security (e.g. when a person exits a home through the back door, an unlocked front door locks itself) can be enabled by *Doorpler*.

Chapter 6 introduces *CaraoKey* which senses the state of a car (empty, door, window or trunk open, person inside the car, etc.) using the pre-existing Ultra WideBand (UWB) radar infrastructure that is present in cars for keyless entry and start. It does so by building a *multipath profile* based on the Channel Impulse Response (CIR) that is computed by an UWB receiver. This *multipath profile* is indicative of reflections in the environment, and changes as the state of the car changes. *CaraoKey* captures these changes to sense the state of the car.

Chapter 7 concludes the dissertation by summarizing the contributions and dis-

cussing future improvements as well as new directions.

1.3 Technical Challenges

This dissertation addresses a number of key technical challenges in inferring the state of homes and cars using wireless signals, while facilitating this line of research in the future.

- Firstly, as mentioned earlier, unlike outdoor environments, wireless systems in indoor environments are subject to strong multipath reflections. The key challenge in *FormaTrack* and *Doorpler*, which aim to determine the direction and/or identity of every crossing, is to isolate the reflection of the crossing individual. For this, it relies on the Doppler Effect, receiver fusion, transceiver gain mismatching, and a moving target indication (MTI) filter [7] to capture the reflection due to human motion.
- Secondly, *FormaTrack*, which identifies people based on their body shape needs a means to ensure a repeatable (body shape) signature for comparison. For this, it leverages the *Doppler Effect* to identify the exact moment the person is at the doorway. There exists positive Doppler Shift when the person is approaching the doorway, and negative Doppler Shift when the person exits the doorway. Thus, the zero crossing point of Doppler Shift tells the exact moment when the person is at the doorway.
- Thirdly, *Doorpler* aims to be a radar setup that adheres to time, space and power constraints of a doorway based occupancy sensing system. *Doorpler* identifies the direction of human transition by computing the angle of arrival of the human induced reflection. Conventional radar based angle of arrival algorithms take tens of seconds to run on an ultra low power microcontroller unit (MCU)

which make them infeasible for smart lighting applications. As there is a coarser requirement on the angular accuracy (i.e. we need to differentiate whether the angle of arrival of the human reflection is positive or negative depending on the side of the doorway), *Doorpler* employs an FFT-based technique that trades angular accuracy for computational complexity. Furthermore, *Doorpler* suffers from a *space-power* tradeoff. Accordingly, a lower frequency of *Doorpler* operation will result in a lower power consumption [8]. However, a lower operating frequency also results in a large antenna array that can out-span the door (as the array size depends on the wavelength [9]). In other words, *Doorpler* wants to transmit at a low frequency for power sake but also at a high frequency for spatial benefits. To handle this trade-off, *Doorpler* uses a technique called *dual-band wake up radio* [10, 11] by operating radios at two different ISM bands (2.4GHz and 5.8GHz). The lower power 2.4GHz radio performs crossing detection and triggers the higher power 5.8GHz array for direction estimation, only when a crossing is detected.

- Fourthly, *TransTrack* is a generic variant of the multi-target tracking problem in which the tracking region is divided into zones and targets can only be monitored as they transition between these zones. We call this the *transition tracking problem*. This is different from conventional tracking where the sensor can observe the entire field of view at all times. The key challenge in *Transition Tracking* is to estimate the number of targets in the tracking region without being able to sense all targets simultaneously. Unlike most other tracking algorithms that maximize the likelihood of the sensor data, *TransTrack* applies penalty functions to find the minimum number of targets that can explain the sensor data. This is because existing tracking algorithms that solve for a maximum likelihood solution will always overestimate the number of tar-

gets in the tracking region for a transition sensing application. On the other hand, *TransTrack* allows tracks with larger numbers of targets only if they have sufficiently fewer errors than other, alternative tracks.

- Finally, *CaraoKey* which builds a *multipath profile* based on a computed *Channel Impulse Response* (CIR) to identify the car state of interest must deal with two key challenges while building this profile. Unlike atop a “small region” such as a doorway which permits transceiver synchronization, the UWB transceivers in the car are not synchronized as they would require long wiring to be placed along the length and breadth of the car. Because of this absence of synchronization, each CIR computed by a receiver will be randomly shifted with respect to previously computed CIRs from the same transmitter. *CaraoKey* addresses this challenge by identifying the first (direct) path in the CIR, and aligning the CIRs about this path, thus yielding a repeatable signature. Next, unlike a home which does not move, *CaraoKey* must be robust to changes in the location of the automobile (i.e. the same solution must work in a multipath rich indoor garage, a parking lot with cars on the sides, in free space, etc.). To address this challenge, *CaraoKey* leverages the internal UWB nodes to build the multipath profile which were determined to be more robust to location changes. Furthermore, since no automobile in the market has the UWB keyless infrastructure as yet, we start from first principles – i.e. over-instrument a car and narrow down the UWB node locations of interest.

1.4 Contributions

Recognizing these challenges, we make the following contributions in this dissertation:

- To sense the state of the home, we built a proof-of-concept system for *Forma-*

Track which measures body shape using a monostatic pulse radar. The radar measures total reflected energy at a given distance d , producing a single dimensional measurement vector, that does not at all resemble the 3D imaging output produced by sub-millimeter wave radars such as an airport scanner. Yet, this simpler, lower-power, and more compact radar sensor could adequately differentiate people in a home while still achieving higher accuracy than existing weak biometric sensors.

- To prevent (doorway) sensing errors from becoming tracking errors, we build a tracking algorithm called *TransTrack*. This tracking algorithm is applicable to a broader set of real-world problems wherein complete coverage of the sensing region is not practical. For example, vehicle sensors are typically installed only at major intersections and do not cover the entire road network. Similarly, people sensors such as security cameras are typically installed at entryways and corridors but do not cover the entire building. As such, people and vehicles can be tracked as they transition between zones of the building or road network, but their position is not monitored while inside a zone.
- Existing systems that attempt to sense occupancy are power consuming, non real-time, pet unfriendly and/or sensitive to ambient heat, light and air flow. To overcome these limitations, we built *Doorpler*, a Doppler-radar based system that detects occupancy at zone transition points such as doorways by sensing crossings and their direction. We point out that the broader concept of *Doorpler* is applicable to other transition sensing applications – gesture recognition in wearables, smart televisions, smart photo frames etc.
- To sense the state of the car, we built *CaraoKey* using off-the-shelf impulse radars. *CaraoKey* is the first sensing system that uses the RF keyless infrastructure in cars as a sensing modality. By leveraging a pre-existing infrastructure,

CaraoKey mitigates the need for extra hardware resulting in cost and space savings for automobile manufacturers. This notion of using a pre-existing infrastructure (built for a different primary use-case) to sense is referred to as *sensorless sensing* [12]. While several *sensorless sensing* applications that leverage (RF) transceivers (such as WiFi, GSM or FM) [13–15], the microphone-speaker pair [16, 17] or the visible light infrastructure [18, 19] have been built in the past, this work introduces the research community to a new sensorless sensing modality in automobiles. This modality can enable several more applications at minimal added costs (since the infrastructure pre-exists) such as monitoring vital signs of the occupants, counting the number of occupants, detecting human movement near the car, enabling personalization, activity/gesture recognition, and so on while ensuring complete privacy. These features will become increasingly important in driverless cars and shuttles where passengers’ sense of in-vehicle security and well-being will be of paramount importance [20].

Chapter 2

Background

As mentioned earlier, *Panoptes* builds on past literature of radar systems and algorithms, albeit in a completely different context – *private indoor human environment*. Consequently, in this chapter we first provide a quick overview of radars that will better understand the subsequent chapters. After that, we present state-of-the-art in topics related to *Panoptes*.

2.1 Radar Background

A radar is a system that consists of one or more transmitters and receivers that uses radio waves to determine the range (distance), angle, speed or other features like size or type of the target of interest. The transmitter(s) beam(s) out electromagnetic energy into its field of view. This transmitted signal travels through the medium, impinging and reflecting off the object of interest. This reflection is analyzed by receiving antenna(s) to determine properties of the object such as range, direction, velocity, etc. This signal incident on the receiver has a power level which is given by the radar range equation [3]:

$$P_r = \frac{P_t G_t G_r \lambda^2 \sigma}{(4\pi)^3 R_t^2 R_r^2 L} \quad (2.1)$$

where, P_r is the received power, P_t is the transmitted power, G_t and G_r refer to the transmitter and receiver gain, λ is the radar's frequency of operation, σ is the radar cross section, R_t is the range from the transmitter to the target, R_r is the range from the receiver to the target, and L refers to losses such as system and propagation loss.

The urge to use such wireless signals to sense the environment goes as far back as the Second World War, where radar systems were originally designed for detecting and tracking large metallic objects such as airplanes in the sky or tanks on the ground, and that too in open spaces [3]. Since then radars have been built for other outdoor applications such as weather, police speed guns, vehicle proximity sensing, etc. This dissertation explores the usage of radars in multipath rich indoor environments where there exists not just primary reflections off the object of interest, but also a plethora of secondary and tertiary reflections.

A radar system can be classified into different types based on the location of the transmitter and receiver, the type of transmitter and the nature of the transmitted signal. A radar is said to be *monostatic* when the transmitter(s) and receiver(s) are colocated. On the other hand, a radar is said to be *bistatic* (or *multistatic*) when the transmitter(s) and receiver(s) are physically not located in the same place. Two components of *Panoptes*, namely *FormaTrack* and *Doorpler* are examples of *monostatic* radars where the transmitter and receiver are situated atop a doorway. On the other hand, *CaraoKey* which senses the state of the car using the car key infrastructure which has transceivers distributed around the car is an example of a *multistatic* radar. Typically, *monostatic* radars benefit from transceiver synchronization owing to their physical proximity. This helps mitigate sources of noise such as carrier frequency

offset and sampling frequency offset.

A radar system can be classified as an *active* or a *passive* radar depending on the nature of the transmitting source. If a radar system has no active transmission source – i.e. it relies on some form of ambient (pre-existing) radio signal such as FM or TV signals, then such a radar system is said to be *passive*. On the other hand, a radar system is said to be *active*, if it has a dedicated transmitter. All components of *Panoptes* have a dedicated transmitter, and are thus *active radars*.

Finally, a radar system can also be classified into different types based on the nature of the transmitted signal. A radar is said to be a continuous wave (CW) radar when the transmitter continually transmits a signal. On the other hand, a *pulse radar* is one in which the transmitter sends electromagnetic waves for a very short duration of time. The number of pulses transmitted per second is called the *pulse repetition frequency*. The duration of a pulse is referred to as *pulse width*, and the fraction of time the radar is transmitting during one transmit cycle is referred to as the *duty factor* (or *duty cycle*). A CW radar has a duty cycling rate of 100% (i.e. it is on all the time). Amongst the components of *Panoptes*, *Doorpler* is an example of a continuous wave radar. It leverages on the frequency shift of the transmitted continuous wave to perform zone occupancy sensing. Such CW radars that rely on the Doppler shift of the reflected signal are referred to as *Doppler radars*. On the other hand, *FormaTrack* and *CaraoKey* are examples of pulse radars that rely on the computed Channel Impulse Response (CIR) based on the received reflected pulse, for state sensing.

A radar system senses data about targets in the environment. Aside from the detection aspect, many radar systems also have an algorithmic component called the *tracking algorithm* [21]. The goal of the *tracking algorithm* is to associate the observed data with the correct target in the presence of sensing and environmental noise, creating *tracks* that are observed by the same sources. This is referred to as a

data association problem. If multiple targets can be in the field of view of a radar, then it is referred as *multi-target* tracking (MTT). Numerous techniques to perform data association exist, such as *Nearest Neighbor*, *Probabilistic Data Association*, *Multiple Hypothesis Tracking*, etc [22]. These algorithms typically incorporate a motion model factoring in the physics of the objects' motion. *TransTrack* is a Multiple Hypothesis Tracking inspired algorithm that associates data collected by the doorway sensors to the inhabitants of the home.

2.2 Related Work

Panoptes is related to prior works in identity sensing, device-based localization, radar sensing, occupancy sensing, multi-target tracking and vehicular sensing.

2.2.1 Identity Sensing Systems

Non-intrusive identification and tracking of people in the home environment has been an open problem for several years. Early work used smart-floor based sensing systems [5, 23, 24] to track people based on their weights. However, this approach would mis-identify somebody if they wear something heavy such as a backback. Subsequent work tracked people based on height [25, 26], but this approach would mis-identify people if they wore a hat or shoes. This approach was extended by complementing height with smartphone connectivity data [27], infrared sensors [28], or person width measurements [29], but these approaches are subject to compliance issues or error based on body position, respectively. Another system attempts to identify people by sensing their shadows [30, 31] but requires a dense deployment of photodiodes on the floor, which would be difficult to install in homes. Several wireless-based device-free systems exist that attempt to identify people based on their gait [32–35]. These systems typically analyze the variation of the received wireless signal to infer gait

information (such as gait cycle time, walking speed, stride length, torso speed etc). However, these systems require a person to walk in a straight line for about 5m [32, 34], in order to collect enough information about the gait of the person, and gait can be affected by shoes, carried objects, the use of walking aids, and other factors. Vision based systems [36–39] can also identify and track a person. For example, vision systems capture a facial image [38], the gait [36] or iris data [37] as biometrics. Other vision based systems [40, 41] try to extract the silhouette of a person from the video frames and identify them based on their body shape. One limitation of these techniques, however, is the need to deal with lighting issues such as darkness. Additionally, people often do not accept video based systems into their homes, especially given the possibility that the device could be hacked. *FormaTrack* avoids these problems by using wireless signals to sense the identity of people in a non-invasive way. Other RF based sensors can perform much higher resolution imaging of humans to obtain their shape [42, 43]. These systems use sub-millimeter (for e.g. airport scanners) or millimeter waves, but are expensive and large in size [42], making it currently unsuitable for people tracking in homes. *RF-Capture* [44] is a centimeter-wave system similar to ours that senses the body shape of people in a room. However, it requires people to walk towards the device for shape sensing, and also has a more complicated hardware than *FormaTrack* involving 4 transmitters and 16 receivers. *FormaTrack* attempts to infer people’s identity at the doorway with a single transceiver. Finally, depth imaging sensors such as the Microsoft Kinect can detect human body shape with high accuracy [45], but it is not yet a low power device and therefore would be difficult to use for body scanning. However, future versions of this sensor will be able to serve well for body shape sensing.

2.2.2 Device-based localization

Aside from the systems described above, many indoor tracking systems sense and track people upto a decimeter level but infer their identities by requiring wearable tags or electronic devices. So called device-based localization systems [46] include Cricket [47], Active Badges [48], RADAR [49], and Blue Sentinel [50] that use ultrasound, infrared, wireless fingerprinting, and Bluetooth Low Energy, respectively. Some other works have attempted to localize people based on FM-signal [13, 51], the powerline infrastructure [52] or GSM signal fingerprinting [53]. More recently, several device-based sensing techniques using WiFi attempt to localize the device (person) by measuring the angle-of-arrival or time of flight with respect to one or more access points [14, 54–57]. In addition, there are motion-capture systems, such as Vicon, Xsens, Zebra etc [58–60], that can capture the human figure and infer their identity, but requires the person to have several markers (sensors) on their body. However, all these systems require a person to carry/wear a device (or at times even rotate the device [56]) to be localized. While such systems may be practical in commercial settings, they are not accepted in homes due to the so-called *forget to wear, forget to charge* problem [4]. We refer the reader to Xiao et al. [61] for a more detailed survey of device-based and device-free localization systems.

2.2.3 Radar-based Sensing

Our system, *Panoptes* is a radar-based system that leverages radar principles to sense the state of a home with and without identity. Far from its initial domains of aviation, navigation or weather forecasting, radars have now entered into homes with applications such as non-intrusive vital signs measurement [62], gait analysis [35], and fall detection [63]. Furthermore, with the advent of Google Soli [64], millimeter-wave based radar systems have been designed for hand-gesture recognition [65], object and

material identification [66], or even to be a musical instrument interface [67]. However, *FormaTrack* is different from these systems in that it uses a radar for identification via body shape and room-level tracking of people.

Similar to *Doorpler*, CW radars have long been used to detect targets and their direction by analyzing the reflections of a transmitted tone signal [21]. However, many of these solutions [68] cannot be directly applied to our use-case, as these systems tend to be space-heavy [9, 69, 70] (i.e. we want *Doorpler* to have a form-factor that fits onto a doorjamb whose width can be as small as 10cm [71]) or time-heavy [72, 73] (i.e. we want *Doorpler* to be real-time). Examples of space-heavy techniques include Time Difference of Arrival [69], Amplitude-based AoA [70] which require a large antenna separation such that RF path loss difference can be used to determine the direction - a spatial luxury unavailable atop a doorway. On the other hand, time-heavy techniques include those that have a large scan-time [74] or employ subspace techniques [72, 73] that are computationally heavy to run in real-time on an ultra low power microcontroller.

However, *Doorpler* adheres to the space, time and power-constraints of the use case. It only requires a coarser angular accuracy (i.e. whether the angle of the reflection from the human is positive or negative depending on the doorway side) in estimating the direction of transition. Consequently, it employs an FFT-based technique that isolates the reflection from the crossing human and trades-off angular accuracy for computational complexity. The coarse nature of the angular accuracy also eliminates the need for large phased arrays, a common space-heavy direction finding solution [9]. The coarseness also permits *Doorpler* to transmit at very low-power and work with low sampling rates.

Secondly, radars employ different techniques to mitigate direct path interference such as delayed sampling (pulse radars), shadowing or beam-steering [75]. In contrast,

Doorpler employs a technique by orienting the nulls of the omni-directional transmit and receive antennas towards each other (i.e. antennas point at each other so that the antenna gains mismatch). Consequently, most of the energy is radiated downwards into the doorway, and the direct path is weakened.

Finally, *Doorpler* identifies the moment a person crosses the doorway via a technique inspired by Doppler and pseudo-Doppler direction finders [76]. In such systems, an RF transmitter is localized by a rotating receiver – there will be positive Doppler when the receiver is rotating towards the active RF source, and negative while moving away, with the zero-crossing informing the direction of the RF source. *Doorpler* uses a similar technique but instead of a rotating radio, it leverages the moving human. A person crossing the doorway causes positive and negative Doppler during approach and exit respectively. The zero-crossing thus tells *Doorpler* when the person was underneath the sensor, creating a temporal reference for AoA comparison.

Furthermore, *FormaTrack* and *Doorpler* use the Doppler principle to identify when the person walks from one side of the doorway to the other. The principle of Doppler has been used for several applications such as inferring hand gestures [64, 77, 78], sleep-sensing [79], connecting mobile devices [80], and tracking phone position [81]. They all rely on the phenomenon that the frequency of the signal changes when the transmitter or receiver or a nearby reflector (*virtual transmitter*) moves. *FormaTrack* and *Doorpler* build on the same principle but uses it in a different context.

2.2.4 Occupancy Sensing

The current state-of-the-shelf system that is commonly deployed in rooms of homes and offices for occupancy sensing is a motion sensor. However, a single motion sensor cannot provide a direction estimate, and movements near the sensor can trigger false positives. Furthermore, these motion sensors are also not real-time, particularly dur-

ing exit-events (i.e. lack of motion for a period of time is perceived as an exit). Since lack of motion is perceived as exit, occupied periods get mis-classified as unoccupied (“the hand-waving at motion sensor problem”). Finally, motion sensors can also get triggered by external factors such as sunlight, car lights (if used in exterior doorways) or HVAC air flows [82] (a limitation even if they are placed on either side of the doorway). Several doorway tracking systems have been built that can perform crossing detection and direction estimation [25, 28, 29, 45, 83]. However, these systems either consume high power [28, 29, 45, 83] (i.e. outside a harvestable limit), are not highly accurate in direction estimation [25, 28], are pet unfriendly (ultrasonic sensors) [25, 29], cannot distinguish between movements near the doorway and doorway crossing events [25] or make assumptions about the heat profile of a human [83, 84]. *Doorpler* mitigates the above limitations of both state-of-the-art and state-of-the-shelf systems by using the phase and amplitude of low-power RF signals that cause no harmful effects, distinguishes near-door events from crossing events, and is independent of the lighting, air flow or the heat profile of the person.

2.2.5 Multi Target Tracking

The problem of multi-target tracking (MTT) has been well explored by many prior works. We refer the reader to Blackman [21] and Pulford [85], for a survey of MTT methods. In this dissertation, we are interested in a variant of the MTT wherein only the transition of targets from one zone to another are sensed. One common approach is to use a sequential Bayesian estimation algorithm such as the Kalman Filter, Hidden Markov Model (HMM), or Particle Filter [86–88]. However, these approaches choose a track by maximizing the likelihood of the data, which is not a viable approach for Transition Tracking because the number of targets is unconstrained and not all targets are sensed. Thus, the maximum likelihood solution will typically contain phantom

targets in order to explain away any sensing errors.

Several other papers have performed multi-target tracking by creating sensing zones. For example, Oh [86], Wilson [89], Kruger [90], Muller [91] treat the home as a graph of zones, each with its own sensors. However, these papers assume that the sensors are located within the zones, whereas our work assumes that the sensors are located on the transitions between the zones. In other words, they assume complete sensing coverage of the tracking region. Additionally, those papers assume the use of motion sensors, which do not have identifying information such as size, color, or shape. Thus, the papers do not need to address the data association problem in the same way. *TransTrack* deals with the additional challenges of observing target transitions rather than target states and, as a result, uncertainty grows quickly about both the number of targets and the state of each target.

The most similar solution to ours was developed for doorway tracking [25, 27, 92] - wherein the identity and direction of a target are sensed as it crosses the doorway. However, the solutions presented here assume a fixed number of targets with known identities, even though the occupants of a typical home come and go at different times, and occasionally bring guests into the home. Therefore, the algorithm that was analyzed did not need to address the complexity of estimating the number of targets in this environment, which is a key part to making the doorway tracking solution practical. Even if an application does not want to track guests, they can cause errors for resident tracking if the system cannot differentiate the guests from the residents. Consequently, there is a need for a tracking algorithm that tracks a variable number of targets by sensing only their transitions.

One common approach to the MTT problem is the Multi-Hypothesis Tracking algorithm (MHT), which is a deferred logic technique that delays uncertain data associations until more data become available by maintaining and scoring several alternative hypotheses. Originally developed for radar tracking systems [6] where the

measured features are a set of discrete blips, it has since seen use in a diverse set of applications like pedestrian tracking [93], eddy current tracking [94], opponent player tracking in autonomous soccer robots [95] etc.

However, inherent to these applications is the concept of periodically observing all targets in the tracking region, which is not a valid assumption in the transition tracking problem. To the best of our knowledge, this is the first work applying MHT in a multi-target transition sensing context.

2.2.6 Non-intrusive Vehicular Sensing

There exist systems that leverage the acoustic or WiFi infrastructure in cars to sense context [96–102]. This context could be in the form of recognizing gestures for hands-free control [96], detecting phone usages while driving [97, 98], tracking the driver's head [99], detecting emotions [100], sensing driver distractions [101], inferring driver actions [102], etc. Our work is different from these systems in that we use a different sensing modality (UWB) to infer car states. To the best of our knowledge, ours is the first system that leverages UWB radios to infer car states. In current automobiles, car states are sensed via explicit sensors [103] – contact sensors for trunk and doors, glass break sensor for windows, pressure sensor for occupancy (only in the front seats for cost reasons). More recently, solutions are being built for occupancy sensing using RF in cars [104–106]. The difference between the above techniques and *CaraoKey* is that we aim to infer car states using a pre-existing infrastructure as opposed to specialized sensors.

Chapter 3

Room-level Tracking of People based on Body Shape

3.1 Introduction

Knowledge of a person's whereabouts in the home is key to context-aware applications, such as personalized heating and cooling, entertainment, task assistance, and behavioral or health analysis. Most tracking systems today require people to wear tags, carry mobile devices, or have cameras or microphones but many people do not accept these technologies in the home [4], and particularly not the aging population that needs in-home care the most. As an alternative, many tracking systems are using so-called *weak biometrics*: physical characteristics of the body that can differentiate people, but that do not necessarily uniquely identify them. For example, several systems over the past several years have tracked people based on height [25, 26], weight [5, 23], and width [29]. Weak biometrics can be effective in a home environment because there are typically only a handful of residents and they often have different biometric features due to age, gender, and/or family role. However, weak biometrics break down in environments with a large number of people (such as office

buildings) or with people who have similar biometric features by chance.

In this dissertation, we propose a new weak biometric feature called *body shape* that differentiates people based on features such as head size, shoulder size, or torso size. The basic idea is to scan the body with a radar sensor and to compute the *reflection profile*: the amount of energy that reflects back from each part of the body. These energy levels are indicative of the relative size of each part of the body. Many people have different body shapes even if they have the same height, weight, or width, which makes body shape a stronger biometric. Even twins often have different body shapes. The most accurate body shape measurements would be achieved with sub-millimeter wave radars, such as those used as airport scanners [42], but we hypothesize that simpler, lower-power, and more compact radar sensors could adequately differentiate people in a home while still achieving higher accuracy than existing weak biometric sensors.

To test this hypothesis, we built a proof-of-concept system called *FormaTrack* using a monostatic pulse radar that measures total reflected energy at a given distance d , without collecting any information about angle of arrival. Thus, it produces a single dimensional measurement vector, as illustrated in Figure 3.1, that does not at all resemble the 3D imaging output produced by an airport scanner. We mounted the sensor at the top of a door frame pointed downward so that it measures the distance to the ground. When a person walks through the door frame, the sensor receives different levels of reflected energy from each part of their body. We use Doppler shift to detect the exact moment when the person is in the door frame, and collect the reflection profile at that moment to ensure the signature is repeatable. Then, we compare the signature to previous measurements in order to recognize the occupant. Additionally, we tilt the sensor in order to create an asymmetric sensing region, similar to prior work [25], in order to detect the walking direction. The identity and direction

combined is sufficient for room-level tracking of occupants in the home.

The sensor that we use has a maximum sensing range of 1m and therefore does not pick up most of the torso size or any gait features observable in the arms and legs. It senses the person from above and therefore features of the face and neck are also occluded. Nevertheless, this sensor demonstrates that body shape can be an effective biometric feature. We evaluate *FormaTrack* using eight participants of varying height and weight who were asked to walk through a doorway in different ways every day for seven days. This process generated over 2800 crossing events while capturing a wide range of factors such as clothing, hats, shoes, and backpacks. Results show that *FormaTrack* can achieve a precision, recall, direction and identity accuracy (over all possible groups of 2 people) of 100%, 99.86%, 99.7% and 95.3% respectively. Additionally, on testing *FormaTrack* for 36 hours (over 12 days) when no one crossed the doorway, *FormaTrack* produced no false detections. To evaluate, how *FormaTrack* translates to a whole-home tracking accuracy, we use the empirical data to emulate 15 different floor plans varying from 3 to 9 rooms and test with data from 2 to 4 participants. We observe that *FormaTrack* can achieve over 99% tracking accuracy with 2 people in a home with 5 or more rooms. Even with 4 people in a home with only 3 rooms, room-level tracking accuracy is still above 92%. Finally, we demonstrate that simple techniques can be used to reduce average power consumption by over 70% while missing fewer than 1% of all doorway crossings.

3.2 Approach

Any doorway tracking system should perform three main tasks : (i) detect when a person crosses through the doorway, (ii) estimate their direction of movement, and (iii) infer the identity of the person who made the doorway transition. Our system *FormaTrack* is a radar-based system mounted atop the doorway that analyzes the

signals reflected back from the environment to perform the aforementioned tasks. Since *FormaTrack* is radar-based, we first provide a quick overview of key radar terminology and concepts.

A pulse radar is a device that emits a pulse of electromagnetic energy towards an expected target area, where a portion of this radiated energy is reflected back by a target. The radar device analyzes this reflected signal to obtain information about the target. For example, the amount of reflected energy indicates the target's radar cross section (size, curvature, reflectivity, etc) and the frequency of the reflected signal indicates the target's velocity, as per the Doppler principle. A radar device is said to be *monostatic* when its transmitter and receiver are collocated. In *FormaTrack*, a *monostatic* radar is mounted atop the doorway. The transmitter of our monostatic radar radiates pulses towards the target area, while the receiver measures the reflected energy from the pulse interfering with targets.

The output of a pulse radar is a *radar frame* that contains information about the total reflected power at varying distances from the radar. These distances are quantized into *bins* called *range bins*, and the size of each bin corresponds to the *range resolution* of the radar (i.e. the minimum distance between two distinguishable targets). This *range resolution* of a radar depends on the sampling rate of the receiver. For example, a receiver that samples at 40GS/s can provide a range resolution of about 4mm. The total number of bins in a *radar frame* depends on the *maximum detectable range* of the radar (i.e. how far can a target be positioned to be detected by the radar). Figure 3.1 shows an example of a *radar frame* with two dominant reflectors at about 0.4m and 0.6m respectively. The size of each *range bin* in this frame is 4mm. There are a total of 256 *range bins* in this frame resulting in a *maximum detectable range* of 1.024m.

Radars typically deal with two time dimensions - *fast time* and *slow time* [7]. *Fast time* refers to the time between two pulses - i.e. it represents the range bins for a given

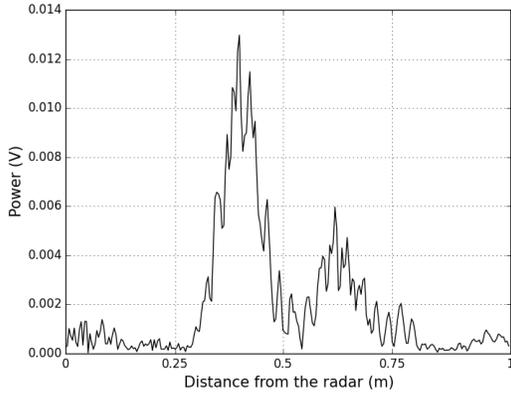


Figure 3.1: An example of a radar frame: the radar measures the reflected power at various distances.

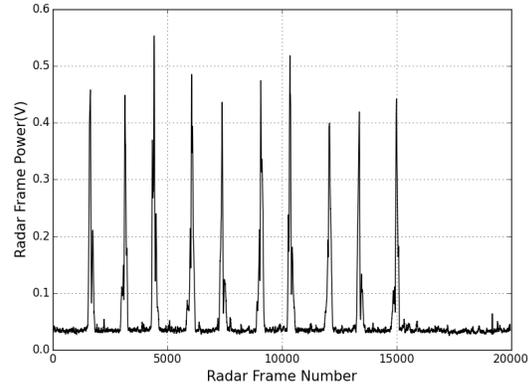


Figure 3.2: The total energy reflected back on to the radar during a crossing (a peak in the figure) is much larger than that in the absence of a crossing.

pulse (frame), and is a function of the sampling rate of the radar. On the other hand, *slow time* refers to the time dimension in the granularity of a pulse. It is a function of the pulse repetition frequency (frame rate) of the radar. For a more detailed review of radar and its principles, we point the readers to [3, 7]. The following sub-sections explain how we use the radar to detect crossings, direction and body shape.

3.2.1 Crossing Detection:

Having seen the basic working of a radar system, we next explain how we use such a system to perform doorway tracking. The first step in any doorway tracking system is to detect when a person actually walks through the doorway. For this, we leverage the intuition that the total amount of energy reflected back on to the radar when a person is crossing the doorway is much higher than when no one walks through.

To capture this intuition, we first pass the captured *radar frame* through a 3-frame moving target indication (MTI) filter [7], to eliminate any unwanted static clutter. Next, we calculate the total power in each radar frame as the sum of the absolute reflected power at each distance in the frame. Formally, the power of the i^{th} radar

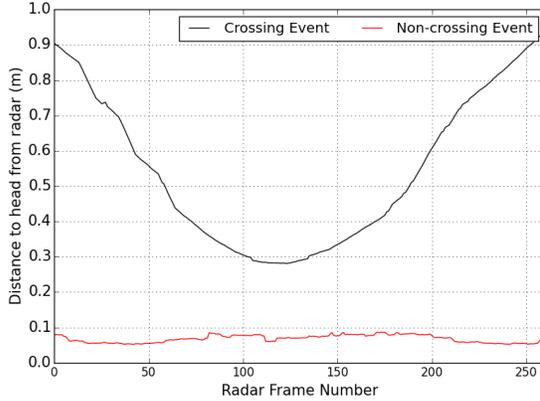


Figure 3.3: The distance to the head from the radar takes a V-shape during a crossing, but occupies the noise-floor during a non-crossing event.

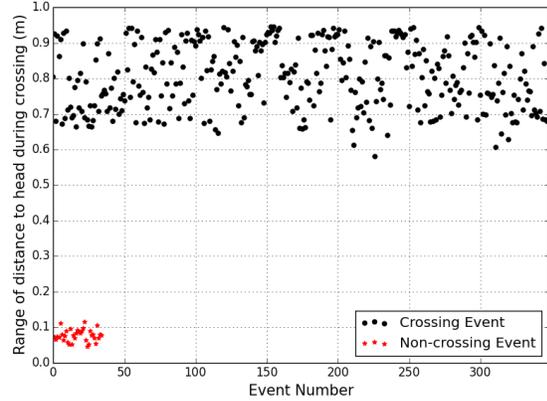


Figure 3.4: The range of the distance to head estimates during a crossing and non-crossing event are vastly different.

frame denoted as $RFP(i)$ (read as radar frame power of the i^{th} frame) is given by:

$$RFP(i) = \sum_{d=0}^{N-1} |P_d(i)|, \quad (3.1)$$

where N is the total number of range bins, and $P_d(i)$ is the reflected power at the d^{th} bin, in the i^{th} radar frame.

We next filter the radar frame power values via a two-stage discrete FIR filter [107] to eliminate any impulse noise. Figure 3.2 shows the result of this filtering, for 10 doorway crossings. We can clearly see that the radar frame power during a crossing is much larger than during its absence. Consequently, we detect a crossing when the total radar frame power exceeds a threshold Th_{cross} (set as 4 times the noise floor). This technique of using the radar frame power to detect a crossing is only the first step towards crossing detection. A second filtering step is needed to eliminate false positives - i.e. to filter out those non-crossing events (events where no one is crossing the doorway) that have a high radar frame power because of transient device noise.

We eliminate these false positive (FP) crossing events by leveraging on the intuition that since there is no one crossing the doorway, the height estimates (or the lack

of it) during a potential crossing event, can be used to filter out FP events. We next explain how these height estimates are obtained. With *FormaTrack* mounted atop the doorway, the head is the dominant reflector during a doorway crossing. Consequently, the distance to the head at radar frame t corresponds to the range-bin with the maximum reflected power. Formally,

$$\text{Distance to Head}(t) = \arg \max_d |P_d(t)|. \quad (3.2)$$

Figure 3.3 shows the distance to the head from the radar during a crossing and non-crossing event (FP). We observe that the distance to head follows the expected V-shaped pattern during a crossing, but occupies the noise floor during a non-crossing. Hence by looking at the range of the distance to head estimates in a small window (we use 100 frames) around the actual doorway crossing frame (obtained in Section 5.2.2), we can filter out false positives¹. Figure 3.4 shows this range estimate calculated for 350 doorway crossing events of a person and for all FP events detected in one of the days of our study (35 non crossing events passed the radar frame power test). We can see a clear separation between true positive and false positive events. We consider an event to be a false positive if it has a range estimate less than a threshold $Th_{head.range}$ (0.24m; determined to be 50% of the 1st percentile of 900 crossing events collected during a testing period).

3.2.2 Direction Estimation

Having detected a crossing event, the next step of a doorway tracking system is to determine the direction of target motion (i.e. from which room to which room

¹On looking at a sufficiently large window around the doorway crossing radar frame for a crossing event, the distance to head measure actually takes an M-shape - i.e. rising from the noise floor when the person comes in the vicinity of the device, and dropping to the noise floor when the person goes outside its vicinity. The noise floor is due to a low-power static path that remains after MTI filtering.

did the person move). For this, we leverage two key factors: (i) the principle of Doppler to identify when the person is exactly at the doorway, and (ii) tilting the radar towards one of the rooms so that the total reflected power from one side of the doorway is greater than the other, during a crossing. From classical physics, a target moving towards the receiver of a transmission causes a positive Doppler shift at the receiver, while a target moving away from the receiver induces a negative Doppler shift. Consequently, we determine the direction of motion by identifying the point of transition from positive to negative Doppler shift, and then comparing the total radar frame power in a small window around this point of Doppler transition. Since the radar is tilted towards one of the rooms, by comparing the reflected power on both sides of the Doppler transition point, we can determine the direction of motion. Given this overview, we next describe the details of our direction estimation - i.e. how we go from the radar frames described in Section 3.2.1 to a direction estimate.

Our first step in Doppler-based direction estimation is to convert a sequence of radar frames into a *range-Doppler* matrix that shows the Doppler shift at varying distances from the radar. As shown in Figure 3.5, this is done by horizontally stacking $K(= 32)$ consecutive radar frames and performing a K -point FFT over each of the $N(= 256)$ range bins in a frame [7]. The outcome of this transformation is the range-Doppler matrix having N rows and K columns. Each row of the range-Doppler matrix corresponds to a range bin, while each column of the range-Doppler matrix corresponds to a Doppler-shift bin. The size of each Doppler bin equals $Radar\ Frame\ Rate/K$ Hz [108], where *Radar Frame Rate* is the total number of emitted pulses (radar frames) per second. For example, with a *Radar Frame Rate* of 170 Hz and a horizontal stacking of $K = 32$ frames, each Doppler bin has a resolution of 5.31 Hz.

The left half of the range-Doppler matrix corresponds to negative Doppler frequencies, and the right half corresponds to positive Doppler shifts. Each cell (i, j)

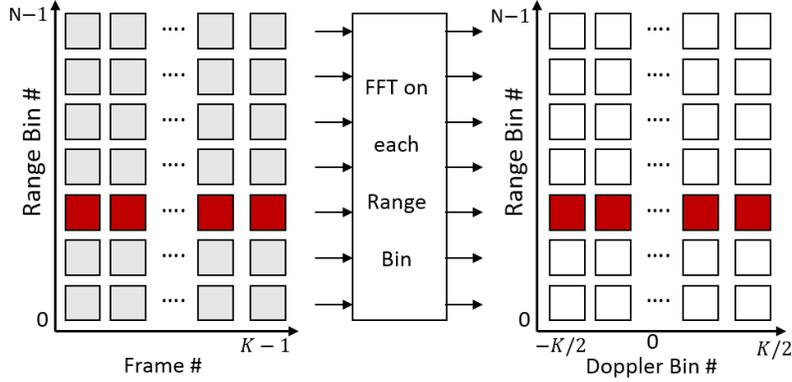


Figure 3.5: The range-Doppler matrix is formed by aggregating K radar frames, and then performing an FFT over each range-bin.

in the matrix represents the Doppler power for the i^{th} range bin and j^{th} frequency bin. Essentially, the range-Doppler matrix captures the amount of frequency shift at each distance. Figure 3.6 shows an example of a range-Doppler matrix captured by *FormaTrack* when a person is approaching the doorway. As the person approaches the doorway, she causes a positive Doppler shift, resulting in higher power in the right half of the matrix.

The Doppler shift caused by a target moving at velocity v at an angle θ relative to the receiver, is given by [109]

$$\Delta f = \frac{2 * f_c * v * \cos\theta}{c} \quad (3.3)$$

where f_c is the transmitter's center frequency and c is the speed of light in the transmission medium.

With a radar frame rate of F frames per second, the X-dimension of the range-Doppler matrix spans from $-F/2$ Hz to $+F/2$ Hz. In other words, we can measure Doppler shifts induced by targets up to $F/2$ Hz. At higher speeds, the frequency "wraps around" the edge of the matrix, leading to *Doppler aliasing* [110]. Given our radar center frequency of 7.2 GHz and a radar frame rate of 170 Hz, from Equation 5.5, we can measure targets moving at speeds up to 1.77 ms^{-1} . This is beyond the average

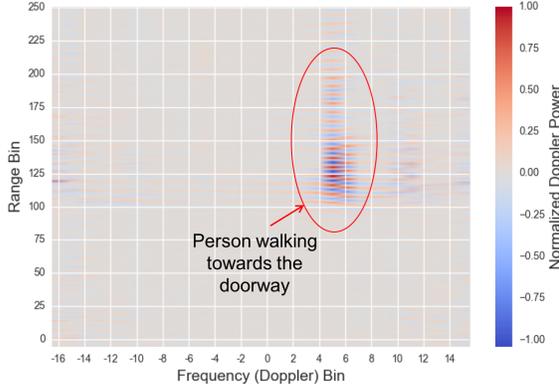


Figure 3.6: Range Doppler Matrix: As the target moves towards the radar, she induces a positive Doppler shift. This is seen by higher power in the right half of the matrix.

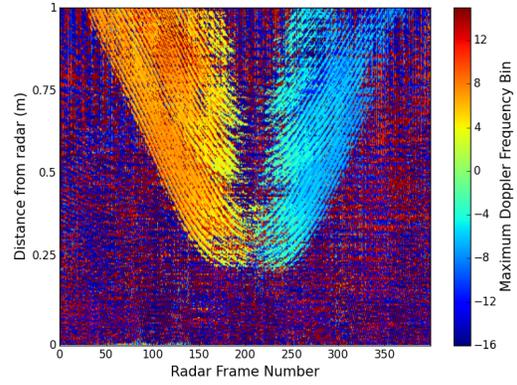


Figure 3.7: Doppler during a doorway crossing: A person causes a positive Doppler shift while walking towards the doorway, and a negative Doppler shift while walking away.

human walking speed of 1.2 to 1.3 ms^{-1} [111].

For a crossing detection frame t_{cross} , identified in Section 3.2.1, we compute RD_{cross} , the set of range-Doppler matrices for w_{rd} frames around t_{cross} . Formally,

$$RD_{cross} = \{RD(t, d, f) \mid t_{cross} - w_{rd} \leq t \leq t_{cross} + w_{rd}\}. \quad (3.4)$$

In our system, we use $w_{rd} = 200$ frames (translating to around 1.2 s). To capture the intuition that a person causes positive Doppler shift while approaching the doorway, and negative while exiting it, we transform RD_{cross} into the *dominant Doppler matrix* $DD(t, d)$ - a matrix that contains the Doppler frequency with the highest reflected power at every distance over time. We perform this transformation by calculating the dominant Doppler frequency for every range-bin, for each range-Doppler matrix in RD_{cross} . Said more formally,

$$DD(t, d) = \arg \max_f RD(t, d, f), \text{ where } RD(t, d, f) \in RD_{cross}. \quad (3.5)$$

In other words, the relationship between $RD(t, d, f)$ and $DD(t, d)$ is as follows: the former captures at each radar frame t , the amount of frequency shift at each

distance, while the latter takes it one step further and gives the frequency with the largest reflected power at each distance, over multiple radar frames. Figure 3.7 shows an example of the dominant Doppler matrix $DD(t, d)$ during a crossing. Here, we can clearly see positive Doppler frequencies dominating when a person approaches the doorway, and negative frequencies dominating during exit. Our next step is to identify the Doppler transition point, which corresponds to when the person is actually at the doorway.

For this we define another term called the *Approach Away Power* (AA). This term weights the *sign* of the dominant Doppler (+1 for positive Doppler and -1 for negative Doppler) by the corresponding Doppler Power, for each range bin in the dominant Doppler Matrix. Formally,

$$AA(t) = \sum_{d=0}^{N-1} \text{sign}(f) * RD(t, d, f), \quad (3.6)$$

$$\text{where } f = DD(t, d), \text{ sign}(f) = \begin{cases} +1, f > 0 \text{ (Positive Doppler)} \\ -1, f < 0 \text{ (Negative Doppler)} \end{cases}. \quad (3.7)$$

Intuitively, AA captures the cumulative dominant Doppler power over the covered range of radar, factoring in the direction of movement. Via this measure, a person approaching the doorway causes positive *Approach Away Power*, while a person exiting the doorway causes negative *Approach Away Power*. Hence, if we calculate $AA(t)$ over the entire crossing window, then capturing the zero-crossing would give us the Doppler transition point. Figure 3.8 shows the normalized *Approach Away Power* during a doorway crossing. We can clearly see our zero crossing of interest around frame 240. However, we also notice other zero-crossings due to noise cropping up (i.e. the *Approach Away Power* oscillates about 0 during a non-crossing).

To identify the zero-crossing of interest, we use the observation that the zero-

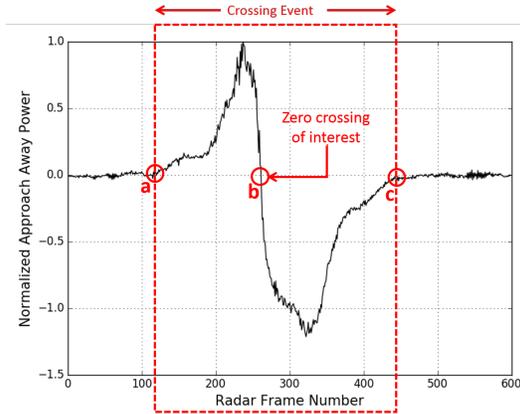


Figure 3.8: The *Approach Away* (AA) Power is used to identify the point of transition from positive Doppler to negative Doppler. There is positive AA power when the person approaches the doorway, and negative AA power when the person leaves. Identifying the zero-crossing tells us when the person crosses the sensor.

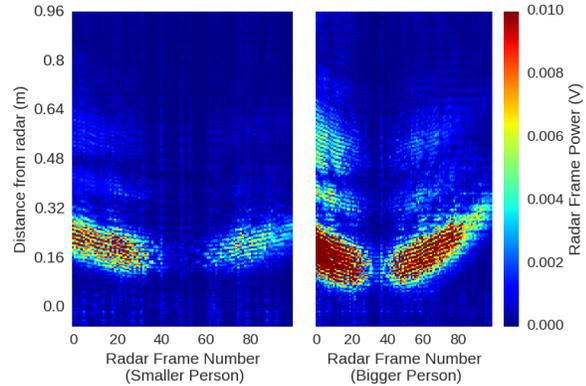


Figure 3.9: Two people of similar height reflect very differently as they walk through the doorway. The bigger person reflects back a stronger signal to *FormaTrack*.

crossing during a doorway crossing is much more widely spaced than during a non-crossing (i.e. the frequency of oscillations about 0 during a non-crossing is much higher than during a crossing). Hence to identify our zero-crossing of interest (corresponding to the person being at the doorway), we first compute the widest zero-crossing interval ($zc1, zc2$). In Figure 3.8, this could correspond to the (a, b) interval or (b, c) interval (i.e. the approach or the exit part of a doorway crossing). For the chosen widest zero-crossing interval ($zc1, zc2$), we next check the *Approach Away Power* values in the interval ($zc1, zc2$). If they are all positive then $(zc1, zc2)$ corresponds to a doorway entry, and we choose the right-end ($zc2$) as our zero-crossing of interest. On the other hand, if they are all negative, then $(zc1, zc2)$ corresponds to a doorway exit, and we choose the left-end ($zc1$) as our zero-crossing of interest. As mentioned earlier, this computed zero-crossing of interest (referred to as $ZC_{interest}$) corresponds to the person actually being at the doorway.

Now that we know the exact point of doorway crossing, we can determine direction. As mentioned earlier, *FormaTrack* is tilted towards one of the rooms adjoining the

doorway. This tilt creates asymmetric reflected power during a crossing; reflected power from one room is always greater than the other. As a result, by comparing the total radar frame power in a small window of $w_{dir}(= 200)$ frames on either side of the $ZC_{interest}$, we can determine direction. If we define motion from tilted side to the non-tilted side as being *IN*, and vice versa as being *OUT*, then we can formally define our direction estimate as:

$$Direction = \begin{cases} \text{IN, if } \sum_{t1=ZC_{interest}-w_{dir}}^{ZC_{interest}} RFP(t1) > \sum_{t2=ZC_{interest}}^{ZC_{interest}+w_{dir}} RFP(t2) \\ \text{OUT, otherwise} \end{cases} . \quad (3.8)$$

3.2.3 Body Shape Sensing

The third piece of any doorway tracking system is to infer the identity of the crossing individual. For this, *FormaTrack* uses knowledge of when the person is exactly at the doorway (Section 5.2.2), and the distance to head estimate (Section 3.2.1), to sense the shape of the person.

According to the radar range equation [7], the total power reflected back on to the radar depends not just on the distance to the target, but also the radar cross section. This radar cross section is a property of the target [64] and is a measure of the target’s ability to reflect signals back on to the radar. In the case of a person walking through a doorway, this becomes a function of the body shape of the person. Moreover, as a bigger person occupies more area, he would also reflect power at more distances than the smaller person. Figure 3.9 shows an example of two people of similar height but very different shapes as they cross the doorway. We see that the bigger person (on the right) reflects back more energy towards the sensor than the smaller person (on the left). *FormaTrack* leverages this intuition and computes a *reflection profile (RP)*

to sense the body shape when the person is at the doorway.

FormaTrack computes the *reflection profile* by capturing the intuition that two differently shaped people will reflect different power at different distances while at the doorway. To compute the *reflection profile*, *FormaTrack* first aggregates a small number of $w_{id}(= 50)$ frames around the doorway crossing point ($ZC_{interest}$). Next for each frame t , *FormaTrack* first computes the distance to the head d_{head} (*Distance to Head* (t) in Section 3.2.1). This distance estimate, which corresponds to the start of the person’s body in a frame, is used as an *anchor point*. Next, we collect the power values at the remaining distances relative to this anchor point, and integrate over the $2*w_{id}$ crossing frames to obtain the *reflection profile* as the following:

$$RP = \left\{ \sum_{t=ZC_{interest}-w_{id}}^{ZC_{interest}+w_{id}} P_d(t) \right\}, \text{ for } d \in [d_{head}, N - 1]. \quad (3.9)$$

We then pass this computed reflection profile of the person (a measure of how the person’s body reflects while at the doorway), which is a vector of aggregated reflected power at various distances, and the distance to head estimate at $ZC_{interest}$, to a support vector machine classifier with the RBF kernel ($C=0.5$, $\gamma=100$) to infer the identity of the person.

3.3 Experimental Setup

To test our hypothesis, we mounted the Salsa Ancho kit [112] atop the doorway of a home, as shown in Figure 3.10. The Salsa Ancho kit uses the Novelda Xethru X2 ultra-wideband impulse radar transceiver chip [113], which operates at a center frequency of about 7.2 GHz, and provides a range resolution of 4mm (i.e. a sampling rate of 39 GS/s). The kit outputs raw radar baseband frames as shown in Figure 3.1, which are

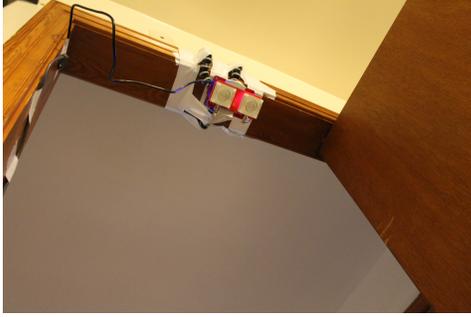


Figure 3.10: The Xethru radar is mounted atop the doorway tilted towards one side.

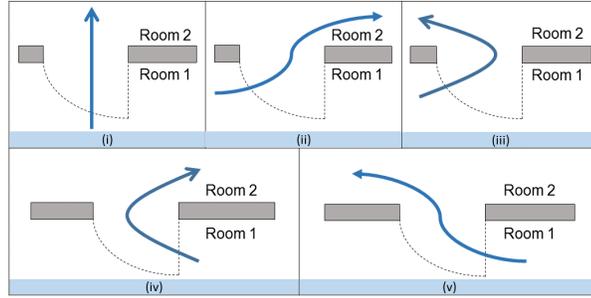


Figure 3.11: Participants were asked to walk through the instrumented doorway in 5 different ways.

Participant #	P1	P2	P3	P4	P5	P6	P7	P8
Height (cm)	161	165	167	169	170	172	175	183
Weight (kg)	60.8	58.1	73.5	77.1	78.0	56.7	97.9	68.0

Table 3.1: The height and weight of the 8 participants who walked through an instrumented doorway for 7 days generating 2800 doorway crossing events

then transferred via a BeagleBone Black to a host computer for analysis. We used a 6dBi sinuous directional antenna on the kit which provided a horizontal and vertical beamwidth of 85° and 65° respectively. We also modified the X2 radar registers to output an average of 170 frames per second to prevent any Doppler aliasing. There are no health concerns associated with *FormaTrack* as the average *transmit power* from the *Salsa Ancho kit* is -13dBm ($50\mu\text{W}$) at 7GHz^2 . This is over 450 times less than the regulations set by the International Commission on Non-Ionizing Radiation Protection (ICNIRP)³[114]. For comparison, the maximum FCC permitted transmit power for an indoor 5GHz WiFi access point is 1W^4 .

We asked 8 participants to walk for 7 days through the instrumented doorway (i.e. one session of 25 back and forth crossings per day for 7 days to capture variations due to speed and clothing). The height and weight of these participants are shown in Table 5.1. On each day, as shown in Figure 5.13, they were asked to walk a total of

²<https://www.xethru.com/chips-salsa-uw-radar-development-kit.html/>

³<http://www.icnirp.org/>

⁴<https://bit.ly/2MBPgF8>

25 times - 5 times back and forth through the doorway in each of the following ways : (i) straight from one room to the other (henceforth referred to as room1 and room2 respectively), (ii) anywhere from the left-side of room1 to the right-side of room2, (iii) anywhere from the left-side of room1 to the left-side of room2, (iv) anywhere from the right-side of room1 to the right-side of room2, and (v) anywhere from the right-side of room1 to the left-side of room2. No restrictions were imposed on the type of clothing the participants wore, or the time of experiment. Our study had participants walking shortly after waking up, after a gym workout, on the way to/from work, during an illness etc. The crossings of the participants were recorded by a video camera, which were then manually analyzed. In all, we collected a total of 2800 doorway crossings.

We evaluate *FormaTrack* via four metrics:

- *Recall*: The fraction of actual doorway crossings that were correctly detected by *FormaTrack*.

$$Recall = \frac{\# \text{ Correctly detected crossings by } FormaTrack}{\# \text{ Ground truth crossings}} \quad (3.10)$$

- *Precision*: Amongst the doorway crossings detected by *FormaTrack*, the fraction that actually occurred.

$$Precision = \frac{\# \text{ Correctly detected crossings by } FormaTrack}{\text{Total } \# \text{ crossings detected by } FormaTrack} \quad (3.11)$$

- *Direction Accuracy*: The fraction of correctly detected doorway crossings having the correct direction.

$$Direction \ Accuracy = \frac{\# \text{ Correct direction crossings by } FormaTrack}{\# \text{ Correct crossings detected by } FormaTrack} \quad (3.12)$$

- *Identity Accuracy*: The fraction of correctly detected doorway crossings classified to be the right person.

$$Identity\ Accuracy = \frac{\# \text{ Correctly classified crossing person's identity by } \mathit{FormaTrack}}{\# \text{ Correct crossings detected by } \mathit{FormaTrack}} \quad (3.13)$$

In this study, we did not ask multiple people to walk through the doorway one behind the other in a platoon fashion. This is because such platooning is not a common scenario in homes (for e.g., in a study, the median time difference between two different people walking through the same doorway in an 8-room home was 10 minutes [28]).

3.4 Evaluation

3.4.1 System Accuracy

Of the 2800 crossings collected, only 4 of them were missed due to the conservative threshold (Th_{cross}), and there were no spurious crossings detected. Among the 2796 crossings correctly detected, 8 of them had an incorrect direction. We evaluated the identity accuracy by testing all possible combinations of 2 people from the 8 participants (8 choose 2), for a total of 28 combinations. For each combination, each person is trained on all crossings collected from a day (one day at a time), and tested on the remaining 6 days. This gives us an overall precision, recall, direction and identity accuracy of 100%, 99.86%, 99.7%, 90.3% respectively. Table 3.2 shows the average accuracy of *FormaTrack*. Using more training data through 7-fold cross validation (i.e. training on 6 days and testing on 1) would give 95.3% identification accuracy, as illustrated in Figure 3.15. The identity accuracy of *FormaTrack* increases with the number of days used for training, and we evaluate this in Section 3.4.7.

Metric	Accuracy (%)
Precision	100
Recall	99.86
Direction Accuracy	99.17
Identity Accuracy	90.3 (95.3)

Table 3.2: *FormaTrack* achieves over 90% accuracy in all metrics of interest for 2800 doorway crossings. The identity accuracy increases from 90.3% with 1 day of training to 95.3% with 6 days of training.

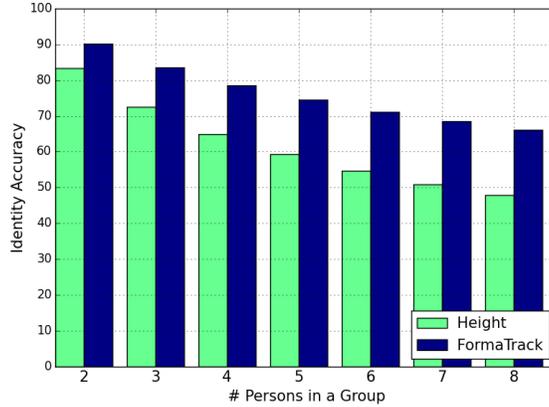


Figure 3.12: *FormaTrack*'s reflection-profile based technique consistently out-performs the height biometric baseline. *FormaTrack* is also more robust - i.e. accuracy decreases more slowly with people. *FormaTrack* achieves nearly 80% identity accuracy even while sensing a group size of 4.

3.4.2 Effect of False Positives

We next test *FormaTrack* for false positives. We do this by collecting 36 hours of data (3 hours for 12 days) from the instrumented doorway on an empty room, where no person crossed the doorway. We compare our *distance filter* algorithm against the no filtering technique (i.e. simply relying on radar frame power alone to detect a crossing). In the absence of any filtering, 152 energy peaks were observed over the 36-hour period. This is due to transient noises on the radar device, as mentioned earlier. However, our *distance filter* technique filtered them all out, resulting in 0 false positives over the study period.

3.4.3 Effect of Number of People

We next evaluate how *FormaTrack's reflection profile technique* performs as we start increasing the number of people being tracked. For an N -person group, we calculate the average *identity accuracy* by generating all possible N -person combinations from the 8 participants. Each combination is trained on a day (one day at a time), and tested on the remaining 6 days. We compare our approach of using *reflection profile* against a baseline algorithm that used *heights* alone. The height estimate is obtained as the distance to the dominant reflector (i.e. the head), as explained in Section 3.2.1, when the person is at the doorway (i.e. the radar frame corresponding to the Doppler sign change).

From Figure 3.12, we notice that *FormaTrack* not just consistently out-performs the *height* alone approach, but is also more robust, i.e. the accuracy drops more slowly as more people are added to the system. We see that for 2, 4 and 8 people, the reflection profile technique is 7%, 14% and 18% better than the height-alone approach. As expected, as the number of people being sensed increases, the *identity accuracy* starts to drop. However, *FormaTrack* still manages to achieve nearly 80% accuracy while sensing a 4-person group.

3.4.4 Energy Analysis

We next explore the potential of a low-power variant of *FormaTrack*. In particular, we explore the following options to save energy:

- Lowering the frame rate of the radar (down-sampling in Slow Time)
- Lowering the sampling rate of the radar (down-sampling in Fast Time)

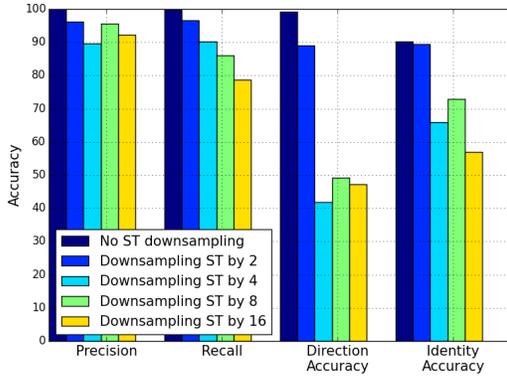


Figure 3.13: Down sampling in slow time (ST) for energy savings: *FormaTrack*'s performance remains comparable even after lowering the frame rate by a factor of 2. At even lower frame rates, *Doppler aliasing* impacts accuracy.

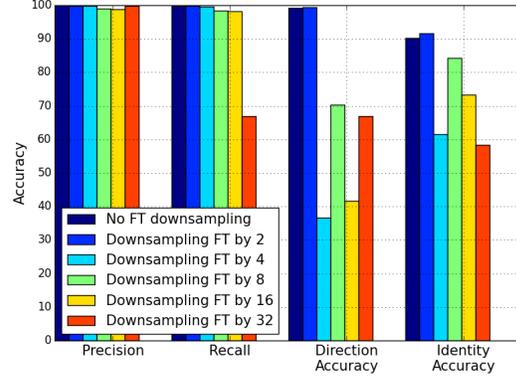


Figure 3.14: Down sampling in fast time (FT) for energy savings: *FormaTrack*'s performance remains unaffected even after lowering the sampling rate of the radar by a factor of 2.

3.4.5 Down sampling in Slow Time:

We next explore the possibility of lowering the frame rate of the radar to save energy (i.e. the radar can start to sleep between every pulse transmission). We do this by considering every N^{th} radar frame, and measuring our metrics of interest. As before, for identity accuracy, we consider all possible 2 person groups. Every group is trained one day at a time, and tested on the remaining 6 days. Figure 3.13 shows that we can down-sample by a factor of 2, and still achieve comparable performance to the non down-sampled case. At greater down-sampling factors, *Doppler aliasing* dominates, affecting the calculation of our zero-crossing of interest, resulting in direction and identity performance loss.

3.4.6 Down sampling in Fast Time:

A third approach to save energy in *FormaTrack* is by lowering the sampling rate of the radar (resulting in lesser stress on the samplers of the radar). This effectively translates to a decrease in the range resolution of the radar. We simulate a fast-time down sampling rate of N , by averaging every N range bins of a radar frame. Figure 3.14

shows that similar to the slow-time case, we can down-sample by a factor of 2 with no performance loss. Even at greater down sampling rates (up to a factor of 16), the precision and recall remains over 98%. However, the direction and identity accuracies start to suffer. This is because the range-Doppler matrix, as seen in Figure 3.6 is not smooth (as the target is moving). An aggregation of range bins means we no longer see a clear transition from approach to away, thus affecting the determination of our desired zero-crossing. As a result, both direction and identity accuracies suffer.

We conclude that the greatest amount of energy savings can be observed by duty-cycling *FormaTrack* with an augmented motion sensor. Furthermore, even while ON, we see that lowering the frame rate and the sampling rate of the sensor by a factor of 2 does not cause any significant performance degradation. In the future, we plan to explore the option of building an energy-harvesting *FormaTrack* that stores energy for most of the day, and uses them during the active times of the day.

3.4.7 Effect of Training Size

Figure 3.15 shows the effect of the number of training samples (back and forth doorway crossings) on *identity accuracy* for all combinations of two people. We see that the identity accuracy increases as we increase the number of training samples with nearly 85% accuracy from just 10 training samples. This shows that *FormaTrack's* training would not be very cumbersome to the users. These training samples can potentially be collected without controlled walk-throughs, by using any events when the person is known to be home alone.

3.4.8 Effect of Objects

We next analyze the effect of carried objects on *FormaTrack's* reflection profile technique. For this, we asked five of the participants (P1, P2, P5, P6 and P7), to walk

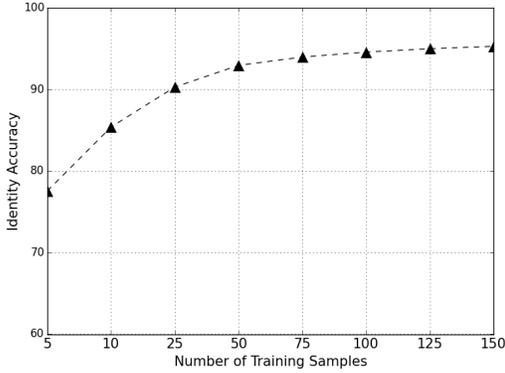


Figure 3.15: Effect of training size on identity accuracy: the identity accuracy increases with more training samples. With just 10 back and forth crossings for training, *FormaTrack* can achieve an identity accuracy of over 85%.

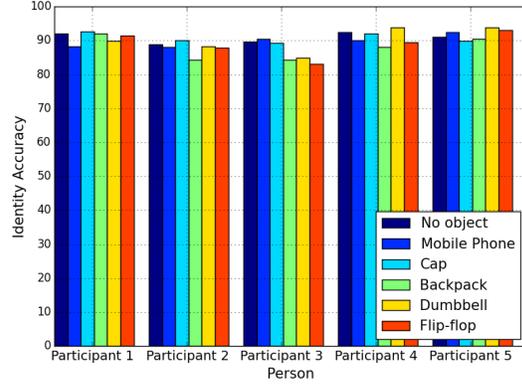


Figure 3.16: The identity accuracy of *FormaTrack* remains comparable even in the presence of some commonly used objects.

through the instrumented doorway, 25 times as mentioned in Section 3.3, but under different scenarios - (i) with a mobile phone near the ear (simulating talking on the phone), (ii) with a baseball cap, (iii) wearing a backpack with a laptop inside, (iv) carrying a 5lb dumbbell in each hand, (v) wearing flip-flops.

We calculate the identity accuracy for each scenario, by considering all possible 2-person groups involving the participant. For every group, we train the two people on each day (one day at a time). We test the object mounted person, with the object mounted walk, and test the non-object mounted participant with each of the remaining 6 days (one at a time).

From Figure 3.16, we see that the identity accuracy is not affected greatly by objects such as mobile phone, cap, backpack and dumbbells, because these objects only occupy a small additional area on the body, and hence do not affect the reflection profile greatly. We also noticed that in the presence of heels, for participants P1 and P7 (two participants comfortable with heels), the identity accuracy was 66% and 89% respectively. We believe this accuracy drop is because heels not just translate the height significantly but also changes the posture of the person while underneath

the sensor. This issue can potentially be fixed via more training data.

3.5 Limitations

3.5.1 In-situ study

FormaTrack is tested on over 2800 crossings in different crossing angles, and a handful of scenarios in Section 3.4.8 to provide a proof-of-concept of sensing the shape of the person when the person is at the doorway. As a future work, we plan to do a more long-term in-situ study after establishing a feasible ground-truth collection method [25, 28].

3.5.2 Effect of Doors

While we do not explicitly consider door interactions in this chapter, we hypothesize that *FormaTrack* can be made to handle these interactions by filtering out the effect of doors. For direction and identity estimation, since the door is taller than an average human, we can filter the door by only considering the dominant Doppler shifts and radar frame power at distances corresponding to the average human height range. An alternate way to address doors would be a data-driven approach that is trained on the door actions. We leave it as a future work to test *FormaTrack* in the presence of door movements.

3.5.3 Effect of walking up to the doorway and turning back

One potential limitation of *FormaTrack* is the case of a person walking all the way to the threshold of the door, turning around and walking back. This motion shows all signs of a crossing event - *RFP* would cross Th_{cross} , there would be positive and negative Doppler, and the *distance to head filter* would see the expected V-shaped

pattern. We believe this can be filtered by measuring the total *RFP* before and after the *Doppler crossing*, and observing them to be similar. An alternate solution is to place a small antenna array that is triggered during a crossing. This array which measures the angle of arrival (-90° to $+90^\circ$) of incident paths [72], can filter out such events by observing the angle of the reflected path to be on one-side of 0° .

3.5.4 Effect of running through the doorway

With drastic speed changes such as running through the doorway, *FormaTrack's* direction estimation algorithm can obtain the correct direction so long as the person's velocity is less than $1.77\text{ms}^{-1} \cos \theta$ (Section 5.2.2). However, the identity algorithm could be impacted as a clear shape signature might not be obtainable. This can potentially be handled by deriving the velocity from the Doppler shift, and using it as a feature or falling back to using height when the speed of the target is significantly different from what was trained. Since this is not a common occurrence, we leave the velocity incorporation as a future work.

3.6 Summary

In this chapter, we present *FormaTrack*, a privacy-preserving radar-based doorway tracking system that estimates the room-location of people in a home. *FormaTrack* determines the direction of room transition of people via the Doppler effect, and infers the identity of people by sensing their shape. We evaluate the system on an instrumented doorway in a home using 2800 doorway crossings. Our results indicate that *FormaTrack* can achieve over 90% tracking accuracy even while tracking 4 people.

Chapter 4

Improving Room-level Tracking by Data Fusion

Thus far, we have been describing the work from the perspective of a single doorway. However, sensing errors (false positives, false negatives, direction errors, identity errors) can happen in that single doorway. In this chapter, we describe how these sensing errors can be retroactively corrected by subsequent doorway crossing events in order to improve the overall tracking accuracy of the system.

4.1 Introduction

The multi-target tracking problem (MTT) [115] is essential to the functioning of many applications including air traffic control, robotics, and biomedical research. The most general form of this problem typically involves an unknown number of targets that move continuously throughout a region and that can appear or disappear [116]. Sensors estimate the positions of the targets at periodic intervals and also estimate identifying properties of the targets such as size, color, or shape. These measurements are subject to noise and the sensors may also generate false positives (a.k.a. false

detections due to clutter) and false negatives (missed detections). In practice, the targets are typically observed with periodically scanning sensors such as a RADAR, an imager, or a LIDAR that can monitor the entire tracking region.

In this dissertation, we consider a variant of MTT in which the tracking region is divided into *zones* and targets can only be monitored as they transition between these zones. We call this the *transition tracking* problem. This problem formulation is representative of an important set of real-world problems where complete coverage of the sensing region is not practical. For example, vehicle sensors are typically installed only at major intersections and do not cover the entire road network. Similarly, people sensors such as security cameras are typically installed at entryways and corridors but do not cover the entire building. As such, people and vehicles can be tracked as they transition between zones of the building or road network, but their position is not monitored while inside a zone.

Unlike traditional MTT, the sensors in transition tracking do not estimate the position of the target. Instead they estimate *the destination zone* of the target as it passes through the transition area. Just like traditional MTT, sensors gather identifying properties of the target (e.g. size, color, or shape) for the purposes of data association, and are subject to three types of errors: sensor noise, false positives, and false negatives.

The key challenge in Transition Tracking is to estimate the number of targets in the tracking region without being able to sense all targets simultaneously. Transition sensors only detect a target when it transitions from one zone to another and so stationary targets are not detectable. No matter how many transitions are detected at any moment in time, that data is consistent with a larger number of targets being in the tracking region, assuming that most of them are stationary. This creates a bias toward estimating a larger number of targets because any sensor value that is not

consistent with true target movement can instead be explained by *phantom targets*: targets that do not actually exist, but that are postulated to have moved at the right place and time to be consistent with the sensor data. Because phantom targets are never inconsistent with the sensor data, they can only increase the likelihood. Therefore, any maximum likelihood solution will overestimate the total number of targets in the tracking region.

In this dissertation, we propose an approach to the Transition Tracking problem called *TransTrack* that jointly estimates the number of targets and their zone locations. Unlike most other tracking algorithms that maximize the likelihood of the sensor data, TransTrack applies penalty functions to find the minimum number of targets that can explain the sensor data. First, it creates a *target penalty* for having a larger number of targets in the tracking region, and applies this penalty only when a sensing error is observed. The intuition behind this approach is to allow tracks with larger numbers of targets only if they have sufficiently fewer errors than other tracks. Second, it creates a *mover penalty* for the number of targets that have moved since the last error. Again, this penalty is only applied when a sensing error is observed. The intuition behind this approach is to eliminate tracks in which different phantom targets are used to explain each sensor reading.

To evaluate, we modify a traditional multi-hypothesis tracking (MHT) algorithm to incorporate the TransTrack principles described above. We used the MHT to reduce computation time and TransTrack could also have been implemented as a HMM or Particle Filter. We then applied this implementation to a dataset created by the Doorjamb sensor [25], which is designed to sense the height and direction of people as they transition between rooms in a home. We use data from 3 controlled studies and 6 days of real-world in-situ deployment involving 2 to 3 participants and totalling 3275 doorway crossings. We observe an average room accuracy of 94.5% and 88.2% in the controlled and in-situ studies respectively.

4.2 Approach

There are two parts typically to a transition tracking system: (i) signal processing phase which handles the raw sensor data and produces a discrete set of transition events (observations), and (ii) tracking phase which operates on the output of signal processing and produces a discrete set of zone locations for each target. This dissertation focuses on the latter. The tracking algorithm must deal with any mistake made by the signal processing algorithm - viz false positives (FP), false negatives (FN), identity errors (IE) and direction errors (DE). Moreover, since we model a variable number of targets, the tracking algorithm must seek to prevent M targets from explaining away N -target data (where $M \neq N$).

Phantom Target Problem : Because of modeling a variable number of targets, maximizing the likelihood would always favor tracks with extra targets ($M > N$). This is because of the *Phantom Target problem*. In *Transition Tracking*, targets can remain stationary for long periods of time in a zone without being detected. Consequently, observations that are not easily explainable by existing targets will trigger the creation of phantom targets : spurious targets created by the tracking algorithm who do not actually exist. For e.g., consider a tracking area with two targets in it. As the two targets move, they cause sensing errors. We wish to prevent choosing those tracks which have extra (phantom) targets that sit idle and then move to explain these error events (caused by the two real targets). Such phantom targets can be brought into the tracking area by a track in several ways - for e.g. (i) via a FP entry event, (ii) by treating the exit observation of a real target to be a FP and retaining the target, (iii) by treating the exit of a real target to be a DE, and bringing in another target, resulting in two phantom targets etc. As these phantom targets are unobservable until they actually move, there is little-to-no penalty in having them sit idle indefinitely, until an event occurs which no other target can explain. Indeed, the likelihood of any

given data set will increase with a larger number of phantom targets

As a result, our goal is to accept the possibility of a new target entering the tracking area, only if it explains the data better. In the case that it does, we want to choose the shortest possible stay of the extra target, i.e. the latest arrival and earliest departure of the extra target that achieves the benefits of the extra target. To address this issue, we define an objective function that penalizes tracks based on the number of targets present, on non-compliance with an observation. The goal is to choose tracks with the minimum number of targets required to explain the observed data. We refer to this balance between phantom targets and unexplained observations as the *target-error tradeoff*.

Hidden Target Problem : However, such an objective function now suffers from the *Hidden Target* problem; since idle targets are unobserved, targets that are idle for a long period of time are evicted out of the tracking area with the goal to minimize the number of targets explaining the observations. To mitigate this, we incorporate a second penalty factor that penalizes a track based on the number of targets who actually move. In other words, a track does not get penalized for having idle targets. The notion of *mover penalty* can be incorporated in many ways: (i) by calculating the number of movers since the last non-compliance, (ii) by ranking the targets based on the total number of moves they have made, and then selecting the maximum movers' rank since last non-compliance etc. We use the former. As a result, the state of a track which normally contains the current location of the targets is augmented with the list of movers.

Finally, among the four error types, IEs alone are not target-agnostic. Therefore, performing data association in the presence of unknown targets requires a notion of identity to be incorporated into a track's state. Addressing this by the inclusion of the identity vector for each target (containing its history of identity assignments) achieves two goals: (i) data association for unknown targets can be performed by

comparing the observed value with past values, (ii) prevent an unknown target from impersonating a known-identity target by better complying with the observations via techniques like T-Test. Summarizing, for tracks to progress and be scored, the current location of targets, the list of movers since last non-compliance and identity vectors for each target become part of a track’s state.

We capture these concepts via the approximate tracking technique of Multiple Hypothesis Tracking (MHT). *TransTrack* differs from classical MHT in that it performs multi-target tracking in the presence of infrequent observation of targets (transitions) as opposed to an entire field-of-view (FoV) scan.

4.3 Implementation

In this section, we explain how we incorporate the TransTrack concepts into the classical Multiple Hypothesis Tracking (MHT) [6]. The TransTrack concepts could equally be incorporated into other tracking algorithms, such as the HMM, but doing greatly increases the state space. We chose to implement with the MHT because several heuristic algorithms enable computational tractability, albeit at the expense of optimality. To understand how the classical MHT must be modified to incorporate TransTrack concepts, we explain each of the key MHT steps below, including our modifications.

Initialization: Let $Z = \{z_0, z_1, \dots, z_{(N-1)}\}$ be the set of N possible zones a target can be in, with z_0 denoting the *outside*. Next, let T be the maximum number of targets trackable by the system. We define a *target state tuple* after the o^{th} observation to be a T-element tuple containing the zone location of the T-targets tracked - viz $\rho_o = (s_1, s_2, \dots, s_T)$ where $s_i \in Z$. Let δ_t be the identity vector of a target t - i.e. the list of identity values for the transitions assigned to target t . Let M be the list of movers since the last non-compliance with the sensor-observation. A

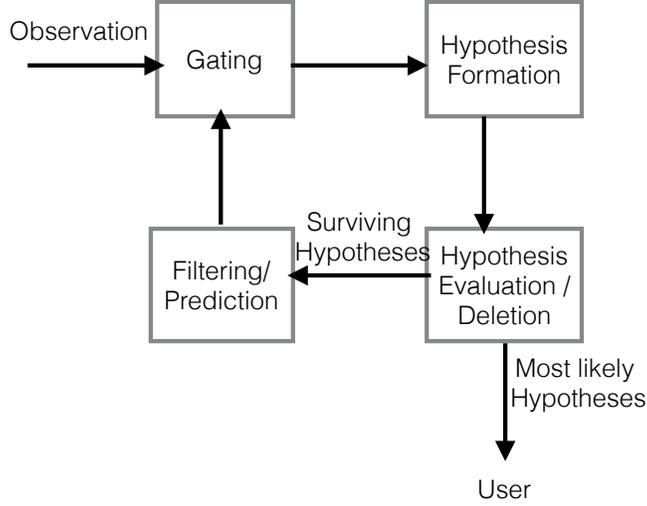


Figure 4.1: The overall operation of a MHT (adopted from [21]).

hypothesis H_i refers to one possible explanation of the o observations, and thus exists as $[(\rho_1, \rho_2, \dots, \rho_o), (\delta_1, \delta_2, \dots, \delta_t), M]$. This can be understood as a sequence of target state tuples according to H_i , identity assignments made to each target, and the list of movers since H_i 's last non-compliance with the sensor-observation. We abbreviate the tracking area (all zones besides the outside) as TA. The last target-state tuple (most recent zone location of targets), the identity vector for each target and the list of movers since last non-compliance constitute the *state* of a hypothesis.

On start-up, *TransTrack* starts with a blank slate, and considers each of the T targets to be equally likely in each of the N zones. As a result, *TransTrack* creates a hypothesis for every (zone, target) combination, resulting in N^T initial hypotheses. The identity vectors of targets with known identity are initialized to the known value, while those of unknown targets are set to ϕ . For e.g. consider a two-target case in a 3 zone state-space $\{z_0, z_1, z_2\}$ with one target of known identity id_1 and another of unknown identity. Then, two of the initial hypotheses are : $H_1 = [[(z_0, z_1)], ([id_1], \phi), \phi]$ and $H_2 = [[(z_1, z_2)], ([id_1], \phi), \phi]$. H_1 thinks only the unknown-identity target is in the TA (in z_1). H_2 is another hypothesis which thinks both targets are inside - one in z_1 and another in z_2 .

Figure 4.10 shows an overview of the classical MHT algorithm, adopted from [21]. We next describe how each block in this diagram behaves in *TransTrack*.

Gating: Gating determines if an observation can be physically caused by a target. In classical MHT, where the entire FoV is scanned, gating helps eliminate certain impossible data associations based on the kinematics of the moving object. However, in transition tracking gating is of little help because the sampling period (how often a target can be observed) is large relative to the potential speed of the target. For e.g., a target can remain idle in a zone for 1 minute but it can also move to the other end of the TA via a small number of FNs within the same 1 minute. This makes most observations in the TA ambiguous. The presence of identity errors exacerbates this problem. As a result, transition tracking does not benefit from gating as each observation become explainable by many targets.

Hypothesis formation: The hypothesis formation step is similar to conventional MHT. Here, the current set of hypotheses are extended by considering all possibilities. In transition tracking this means every observation causes each hypothesis to duplicate itself upto $(2T+1)$ times and progress as:

1. Someone who is inside and has the transition area within his gate, has moved through it in either direction
2. Someone who is outside and has the transition area within his gate, has come in, and moved through it in either direction
3. Observation was a false detection

For example, consider a 3-target scenario in a 3-zone state space ($z0 \leftrightarrow z1 \leftrightarrow z2$), where $z1 \leftrightarrow z2$ is the exterior transition sensor. Upon detecting a $z1 \rightarrow z2$ transition event with observed identity id_{obs} , a hypothesis ending in zone-locations $(z1, z2, z0)$, say [... $(z1, z2, z0)$, $(\delta_1, \delta_2, \phi)$, $\{t1\}$] would duplicate itself $(2T+1)$ times and progress them in the following way:

$$\begin{aligned}
H1 &: [\dots(z1, z2, z0), (\delta_1, \delta_2, \phi), \{t1\}] \xrightarrow{FP} [\dots(z1, z2, z0), (\delta_1, \delta_2, \phi), \phi] \\
H2 &: [\dots(z1, z2, z0), (\delta_1, \delta_2, \phi), \{t1\}] \longrightarrow [\dots(z2, z2, z0), (\delta'_1, \delta_2, \phi), t1] \\
H3 &: [\dots(z1, z2, z0), (\delta_1, \delta_2, \phi), \{t1\}] \xrightarrow{1FN, DE} [\dots(z1, z2, z0), (\delta'_1, \delta_2, \phi), \phi] \\
H4 &: [\dots(z1, z2, z0), (\delta_1, \delta_2, \phi), \{t1\}] \xrightarrow{1FN} [\dots(z1, z2, z0), (\delta_1, \delta'_1, \phi), \phi] \\
H5 &: [\dots(z1, z2, z0), (\delta_1, \delta_2, \phi), \{t1\}] \xrightarrow{DE} [\dots(z1, z1, z0), (\delta_1, \delta'_1, \phi), \phi] \\
H6 &: [\dots(z1, z2, z0), (\delta_1, \delta_2, \phi), \{t1\}] \xrightarrow{1FN} [\dots(z1, z2, z2), (\delta_1, \delta_2, [id_{obs}]), \phi] \\
H7 &: [\dots(z1, z2, z0), (\delta_1, \delta_2, \phi), \{t1\}] \xrightarrow{2FN} [\dots(z1, z2, z1), (\delta_1, \delta_2, [id_{obs}]), \phi] \\
&\text{where, } \delta'_1 = \delta_1 \oplus id_{jobs} \text{ , } \delta'_2 = \delta_2 \oplus id_{jobs} \text{, and } \oplus \text{ denotes append}
\end{aligned}$$

H1 is the hypothesis that thinks the observation is a false detection. H2, H3, H4 and H5 move the two targets inside the TA through the (z1,z2) sensor in either direction. H6 and H7 hypothesize that some target from the outside has come in and transitioned in either direction. Hypotheses H3, H4, H6 and H7 think that some observations have been missed (FN). Note that each hypothesis also appends the identity vector for the hypothesized mover with the observed value.

Next, each hypothesis explores the possibility that someone inside has exited the TA after the current observation via a missed detection. For example, H7 duplicates itself three times, and advances them the following way:

$$\begin{aligned}
H8 &: [\dots(z1, z2, z1), (\delta_1, \delta_2, [id_{obs}]), \phi] \xrightarrow{FN} [\dots(z0, z2, z1), (\delta_1, \delta_2, [id_{obs}]), \phi] \\
H9 &: [\dots(z1, z2, z1), (\delta_1, \delta_2, [id_{obs}]), \phi] \xrightarrow{2FN} [\dots(z1, z0, z1), (\delta_1, \delta_2, [id_{obs}]), \phi] \\
H10 &: [\dots(z1, z2, z1), (\delta_1, \delta_2, [id_{obs}]), \phi] \xrightarrow{FN} [\dots(z1, z2, z0), (\delta_1, \delta_2, \phi), \phi]
\end{aligned}$$

Note that the unknown target's identity vector is reset on exit (H10). This is to capture the intuition that no two unknown targets (*visitors*) are necessarily the same.

Time	τ_0	τ_1	τ_2	τ_3	τ_4	τ_5	τ_6	τ_7	τ_8	τ_9		
GT track (H1)	(z1, z2)	(z1, z3)	(z1, z4)	(z1, z3)	(z1, z3)	(z1, z4)	(z1, z5)	(z1, z4)	(z1, z3)	(z2, z3)		
Inferred error of H1	-	DE	IE	-	FP	-	DE & IE	-	IE	-		
Score (H1)	0	3	6	6	9	9	15	15	18	18		
Chosen Track (H2)	(z1, z2)	(z0, z2)	(z0, z3)	(z0, z4)	(z0, z3)	(z0, z3)	(z0, z4)	(z0, z5)	(z0, z4)	(z0, z3)	(z1, z3)	(z2, z3)
Inferred error of H2	-	FN	DE	IE	-	FP	-	DE & IE	-	IE	FN	-
Score (H2)	0	3	5	7	7	9	9	13	13	15	17	17

Table 4.1: Example of *Hidden Target Problem*: GT track (H1) gets evicted by an alternate hypothesis (H2) at time τ_9 . H2 has a lower score because the idle target was evicted after τ_0 , and brought back in before τ_9 .

Given that there are N^T initial hypotheses, $(N^T) * (T * (2T + 1))^D$ hypotheses are formed after D events. This exponential explosion of hypotheses necessitates track pruning.

Hypothesis evaluation/deletion: Temporal pruning techniques such as *n-scanback* [22, 94, 95] which are commonly employed in conventional MHT, cannot be applied in our case as a target can remain idle in a zone for an indefinite amount of time. The intuition behind *n-scanback* is that ambiguities get resolved in atmost n scans. Secondly, given the large gating challenge, n cannot be high as it will result in storing a large number of hypotheses. For a TA of 8 zones and at most 4 targets, *3-scanback* itself results in the maintenance of over 50 million hypotheses.

Therefore, we develop an alternate two-step pruning strategy that leverages the discretization of states. First, if two hypotheses have the same current zone location of each target, same movers since last inferred error and same identity statistic (e.g. mean of identity vector) for each target, then only the most likely one is retained (i.e. either H1 or H10 in the above example). We refer to this as *equal state* pruning. To further keep the state space tractable and maintain enough diversity and coverage across all possible zone locations, we maintain the top-M (M=4) hypotheses ending in each combination of target state tuple. This results in the constant maintenance of $M * N^T$ hypotheses. However, choosing one hypothesis over another necessitates a scoring function.

Score function: Scoring in MHT is done in an application-specific manner depending on the constraints of the problem. The aim of our scoring algorithm is to

address the *phantom target* and *hidden target* problems. As a result, each hypothesis on non-compliance with an observation suffers a penalty depending on the number of targets present and the number of movers since the last inferred error. We refer to such a non-compliance as an ‘*inferred error*’. For e.g. if the observation says someone moved from zone $z1 \rightarrow z2$, but the hypothesis moves the target from $z2 \rightarrow z1$, then it has inferred a direction error (DE), and suffers a penalty. The other different inferred error types are identity errors (IEs), false positives (FPs) and false negatives (FNs).

More formally, a hypothesis after the i^{th} observation gets penalized according to the following score function:

$$pen(i) = pen(i - 1) + \alpha * \sum_{j \in E} e_j w_j \quad (4.1)$$

where: E : set of inferred errors {FP, FN, DE, IE}

e_j : error penalty associated with the inferred error

w_j : weight of the inferred error type

α : correction term for target-error tradeoff = $(m + t + k)$

t : number of targets in the TA during the error

m : number of movers since last inferred error ($m \leq t$)

k : constant offset to eliminate bias towards certain tracks

We now explain the terms in the above equation, and the intuition behind them.

e_j *Error Penalty* – Each noncompliance by a hypothesis with an observation suffers a penalty depending on the inferred error type. *TransTrack* makes use of a probability value passed up from signal processing, whenever available, and a unit penalty otherwise. FNs suffer a penalty equal to the minimum distance to the transition sensor. IEs are different because the identity of some of the targets are unknown *a priori* during deployment. We refer to such targets as *visitors*. We infer IE for visitors by comparing the observed identity with the hypothesis’ maintained identity

value (e.g. mean) for that visitor. As each visitor is unique, every hypothesis resets its visitor identity vector on the exit of the visitor. Since, we learn the identity of visitors online, a visitor can impersonate a known-identity target by better complying with the observations resulting in incorrect data association. To eliminate this, each hypothesis upon exit of a visitor performs an equal variance T-test ($p = 0.05$) between its visitor identity vector and identity vector of each known-target who has been outside since entry of that visitor. If $p\text{-value} > 0.05$, then the identity distributions are similar, and the visitor hypothesis gets evicted.

w_j : *The weight of the inferred error type* – This term captures the likelihood of each error type across different error types. This is a sensor property. For e.g. if missed detections are less likely than false detections, then $w_{fp} < w_{fn}$.

t : *The number of targets in the sensing-area during the error* – This term is used to address the *Phantom Target* problem. To eliminate any bias on the hypothesis which has all targets outside (i.e. $t = 0$), we use $(t + 1) * \sum e_j w_j$. As mentioned earlier, using the target-factor alone in scoring results in hypotheses being subject to the ‘*hidden target problem*’. Table 4.1 shows an example of the problem with two known-identity targets $t1$ and $t2$. Let us say that the arrangement of the zones are $(z0 \leftrightarrow z1 \leftrightarrow z2 \leftrightarrow z3 \leftrightarrow z4 \leftrightarrow z5)$, where $z0$ denotes the outside-zone. $\tau0$ is the time that target $t1$ becomes idle at zone $z1$ and $\tau9$ is the time $t1$ moves out of $z1$. At $\tau9$, hypothesis H2 evicts H1 with a lower score. This is because H2 having incorrectly evicted the *hidden* target $t1$ out of the TA, ends up with a lower total penalty. For brevity sake the identity vectors are not shown, as they are always identical for H1 and H2.

We next calculate the bounds for the eviction of a hidden target - i.e. the minimum number of errors an idle target can tolerate before being evicted by a hypothesis with a lower score. For simplicity of derivation, we assume uniform unit weighting. Let

τ_a be the time that $t1$ becomes ‘idle’, and τ_b be the time that $t1$ makes a transition again. Let d denote the minimum number of hops from the ‘hidden’ zone to the *outside*. Between τ_a and τ_b , targets $t2$ to tT move and cause e errors. We wish to compare two hypotheses, H1 : $t1$ rightly remains idle, and H2 : $t1$ gets evicted via FNs just after τ_a , and brought in via FNs just before τ_b .

Score (H1) = e errors caused with T targets in TA

Score (H1) = $(T + 1) * e$

Score (H2) = FN to evict $t1$

+ e errors caused with $(T - 1)$ targets in TA

+ FN to enter $t1$

Score (H2) = $(T + 1) * (d) + ((T - 1) + 1) * e$

+ $(T + 1) * (d)$

To retain $t1$, $\text{score}(\text{H1}) < \text{score}(\text{H2})$. Therefore,

$$\begin{aligned} (Te + e) &< 2d * (T + 1) + Te \\ \implies e &< 2d * (T + 1) \end{aligned} \tag{4.2}$$

Consequently, in a 2-target case, a static target in the leaf-node ($d=1$) gets evicted after 6 errors of the other target.

m: *The number of movers since last inferred error* – To mitigate the *hidden target* problem, we add in a second factor that penalizes a hypothesis based on the number of targets who actually move- *the number of movers since last inferred error*. As before, to eliminate any bias towards an all FP track (i.e. $m = 0$), we use the factor of $(m + 1)$. This results in our score formula of: $(t + m + k) * \sum e_j w_j$ where $k = 2$.

We next analyze this scoring function using the same notation. For the sake of simplicity in the derivation, let us consider that every target except $t1$ moved between

each of the e inferred errors. This consideration is just for ease of understanding as the derived inequality is independent of the actual number.

$$\begin{aligned}
\text{Score (H1)} &= e \text{ errors caused with } T \text{ targets in TA} \\
\text{Score (H1)} &= (T + T + 2) + ((T - 1) + T + 2) * (e - 1) \\
\text{Score (H2)} &= FN \text{ to evict } t1 + \\
&\quad e \text{ errors caused with } (T - 1) \text{ targets in TA} + \\
&\quad + FN \text{ to enter } t1 \\
\text{Score (H2)} &= (1 + T + 2) * (d) + ((T - 1) + (T - 1) + 2) * e \\
&\quad + (1 + T + 2) * (d)
\end{aligned}$$

To retain $t1$, $\text{score(H1)} < \text{score(H2)}$. Therefore,

$$\begin{aligned}
(2 + 2T) + (2T + 1) * (e - 1) &< (3 + T) * (2d) + 2Te \\
\implies e &< ((3 + T) * 2d) - 1.
\end{aligned} \tag{4.3}$$

Comparing inequalities 4.2 and 4.3, it can be shown that $((3+T) * 2d) - 1 > 2d*(T+1)$, since $d \geq 1$, confirming that the bounds have increased. In a 2-target case, a static target in the leaf-node now gets evicted after 11 errors of the other target.

In order to avoid growing memory costs, we define a commit policy. After every observation, if all hypotheses agree on a common prefix (i.e. they agree on the zone-locations of each target, from event E_0 to E_i), then the prefix is committed to disk. Subsequent prefix checks happen from event E_{i+1} .

At any time instant, the lowest scored hypothesis is the most likely one. We point out that the recursive nature of the score function makes it unnecessary to have the complete dataset to generate state estimates, making *TransTrack* conducive for near real-time tracking.

Study	Participant Heights	# crossings
Controlled Study1	5'4" , 5'11"	400
Controlled Study2	5'4" , 5'11"	398
Controlled Study3	5'0" , 5'4" , 5'9"	516
In-situ study (6days)	5'0" , 6'2"	1961

Table 4.2: Experiment details: A total of 9 studies with 3275 doorway crossings involving 2 to 3 participants was performed

Filtering/Prediction: In classical MHT, every hypothesis uses a motion model to predict the location of each target for the next scan. However, in transition tracking because of (i) the possibility that a target can stay in a zone for an indefinite amount of time, and (ii) the inevitability of large gates, no prediction is made by a hypothesis on the next location of a target.

4.4 Experimental setup

We first evaluate our tracking algorithm with a doorway tracking application using a Doorjamb-like sensor setup [25] in a detached home of 9 rooms involving 2 to 3 person (targets). The system is mounted on top of each doorway and measured the height and direction of a target as they transitioned through the doorway. We perform 3 controlled studies and 6 days of real-world in-situ deployment. The *diameter* (distance between the farthest two rooms) of the house was 4. This meant one could move from one end of the state space to the other with just 4 FNs, making each doorway transition event became explainable by any of the targets. Table 4.2 describes details of each study and its participants. The first two controlled studies had the same participants. They were asked to leave all doors open in Study1, but open and close doors as they performed the experiment in Study2. This was to study the effect of errors on tracking, as the movement of doors lead to signal processing errors. Controlled Study3 had no constraints, and the participants were asked to enter, exit and walk around as naturally as possible. Ground truth for the study was collected

using cameras installed on the doorway. To ensure that participants lived as naturally as possible the field of view of the cameras were restricted in hardware to only the doorjamb of the doorway. The recorded video was processed, to extract the identity and direction of participant involved in the crossing.

We evaluate tracking using the *Room Accuracy* metric. This metric evaluates if a person is ever detected in the correct room during the time he was in that room. This is calculated as the *F-score* of the room recall and room precision. Room recall is defined as the fraction of the total number of room occupancy periods (the time a person is in a specific room) in ground truth where tracking also correctly placed the same person in the same room at least once during that occupancy period. Room precision is the complement to room recall, where room occupancy periods found by tracking are evaluated. More formally, room precision is defined as the fraction of the number of room occupancy periods in tracking, where ground truth also had the same person in the same room at least once during that period.

The final equation for room accuracy then becomes:

$$\text{Room Accuracy (F-score)} = \frac{2 * \text{Room precision} * \text{Room recall}}{\text{Room precision} + \text{Room recall}}$$

We compare *TransTrack* against three baselines that can track a variable number of targets and have the same set of requirements as *TransTrack*. Our first baseline *Nearest Identity* is a stateless approach to tracking that moves occupants based on the identity data observed at the doorway, with no regard to his previous location. It chooses an occupant based on the height measurement and puts him into a room based on the observed direction. Our second baseline, *Nearest Neighbor* is a well known stateful greedy target tracking approach [21, 117]. Each observation is assigned to the occupant closest to the doorway with heights used as a tie-breaker. The location of the occupant is updated after an assigned observation. Our third baseline K-

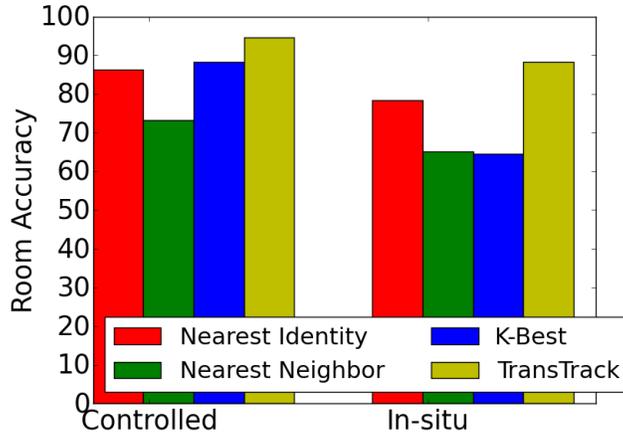


Figure 4.2: *TransTrack* consistently performs better than the baselines. *Nearest Identity* and *Nearest Neighbor* approaches suffer as they do not use future information. The absence of equal state pruning affects accuracy of *K-best*

best is a variant of [118] and maintains the K lowest-scored hypotheses after every observation. The K-value was chosen such that its time and space complexity were identical to that of *TransTrack*. No equal-state pruning is performed here, but the score function is identical to *TransTrack*. To have all algorithms on an equal footing, they are all started with a known initial state of the home.

Finally, we use uniform unit weighting on all errors except FNs. FNs have twice the weight for two key reasons : (i) the signal-processing recall of our system is better than signal-processing precision, and (ii) to increase the bound on the *hidden target problem*. We start off by tracking a maximum of 4 targets in a home (i.e. $T = 4$), and then study the behavior as we vary the maximum number of targets tracked. We refer to the targets with known identities as *residents*, and those with unknown identity as *guests*.

4.5 Results

As seen in Figure 4.2, *TransTrack* observes the highest average accuracy of 94.5% and 88.2% in the controlled and in-situ studies respectively. The *Nearest Identity* approach’s average accuracy of 86.2% and 78.3% in controlled and in-situ respectively

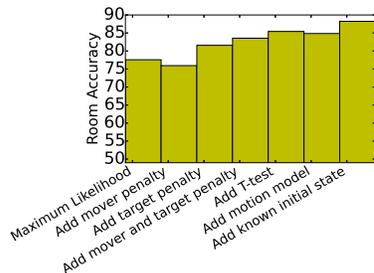


Figure 4.3: Scoring function variants : The addition of target penalty, mover and target penalty, explicit guest reasoning and knowledge of initial state all increase in-situ tracking accuracy over the maximum likelihood approach

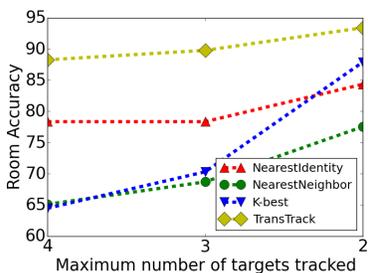


Figure 4.4: Sensitivity analysis on the maximum number of targets (T) tracked: As T decreases, the accuracy of all algorithms increase. However, TransTrack still continues to perform better than other baselines.

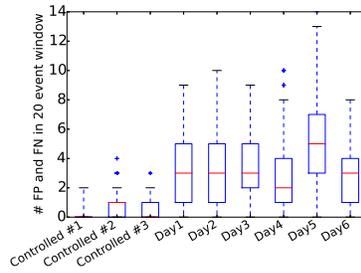


Figure 4.5: Even though the average precision and recall for in-situ was comparable to the controlled studies, the in-situ data has many bursts of 8 or more false positives or negatives within a 20-event window. These bursts help explain the difference in tracking accuracy between the two studies.

is lower than *TransTrack* because it does not use the future to disambiguate the past. The lower 73.2% and 65.1% average accuracy of the *Nearest Neighbor* approach can also be attributed to the lack of use of future. However, the maintenance of state exacerbates the problem here. *K-best* has an average accuracy of 88.2% and 64.5%. It is lower than *TransTrack* because of the absence of equal-state pruning. Consequently, the K-best hypotheses have many redundant hypotheses having the same last state, evicting out the desired hypotheses.

We also performed two additional analysis: (i) On comparing *TransTrack* with Doorjamb’s algorithm, we noted that it performs almost as well as Doorjamb (average less than 5% off) even without assuming the number of targets at all times. (ii) We also evaluated the algorithms by calculating the average resident room accuracy, after making each of the residents as guests. We noticed an identical trend to Figure 4.2 with *TransTrack* suffering an average accuracy drop of 3.8%.

Figure 4.3 shows how different parts of the scoring function affect in-situ tracking accuracy. Simply maximizing the likelihood of the observations gives only 77.5% accuracy. This is because of the presence of phantom persons who sit idle and move

to explain the errors of the real targets. The addition of a mover penalty alone does not increase the accuracy as it still suffers from the phantom effect. However, the addition of a target penalty alone increases the accuracy to 81.6%. This increase is because there is now a penalty of having extra idle persons in the home. However, such an approach suffers from the *hidden target problem*. Adding the mover and target penalty together increases the accuracy to 83.5%. As previously stated, a guest can still impersonate a resident by complying with the observed heights better. We see that the subsequent addition of explicit guest-resident reasoning via the T-Test increases the accuracy to 85.4%. We next notice that the subsequent addition of a motion model does not increase accuracy. The motion model is that it takes a target at least 1 second to pass through every room. In other words, no hypothesis exists wherein a target t can explain the doorway event of another target t' which is H hops away, if the last moved time of t is less than H seconds. Such a motion model does not increase accuracy because we noticed that the percentage of concurrent moves by persons is low, and even during the times of concurrent moves *TransTrack* was already doing a reasonable job in associating the observations to targets. Finally, we notice that starting at a known initial room location for each person increases the accuracy to 88.2%.

We next calculate *room accuracy* as we start varying the maximum number of tracked targets (T) from 4 to 2. As seen in Figure 4.4, decreasing T increases the accuracy of all algorithms. This is because each observation can potentially be explained by a lesser number of persons resulting in lesser ambiguity. *TransTrack* which achieves 93.4% and 89.8% with $T = 2$ and $T = 3$ respectively, still continues to consistently perform better than the other baselines. Since the same trend can be seen in the controlled study too, in the interest of space, we show the graph for in-situ alone.

One of the main reasons for the performance difference between in-situ and con-

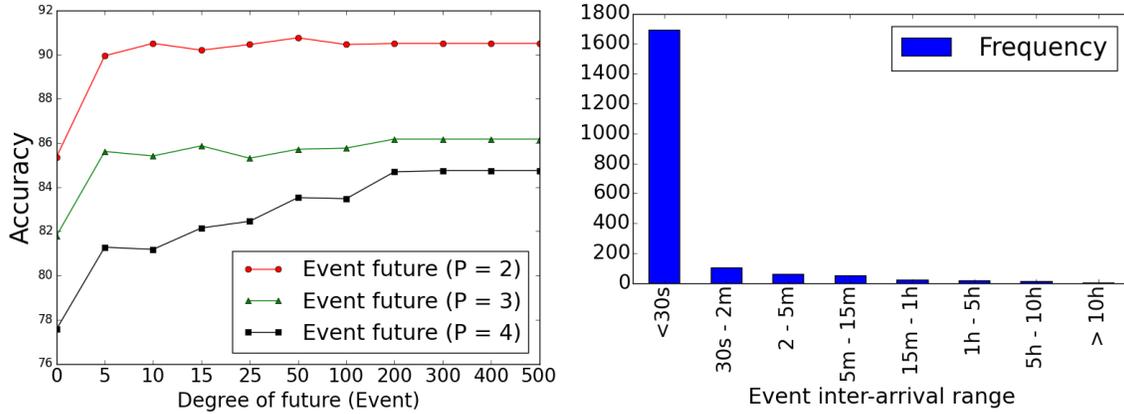


Figure 4.6: Degree of future analysis: Most of the ambiguities get resolved within 5 future events. As the inter-arrival time between nearly 90% of events is less than 30 seconds, this roughly translates to within 150 seconds of future.

trolled studies was due to error-clustering (bursts of errors). The boxplots in Figure 4.5 show the number of FPs or FNs in any 20 event window for each data set. It is seen that in in-situ, there exists several cases in which more than half of the events in a 20-event window are either FPs or FNs. Since *TransTrack*'s scoring assumes a uniform error distribution, when such bursts of errors happen, it tries to explain these erroneous events with extra persons, as the height and direction estimates are inconsistent with the targets at home. Consequently, the accuracy suffers. These bursts of errors were mostly due to a target moving back and forth near a doorway. Other causes were due to crouching, moving hurriedly etc.

Next, we study the effect of removal of each error type on tracking accuracy. Figure 4.7 shows that an increase in accuracy is generally observed with the removal of each error type, approaching 100%. This is because as errors get removed, there is lesser ambiguity in data association for *TransTrack*.

Since, *TransTrack* uses the future to disambiguate the past, we next calculate how much future is required to correctly resolve a doorway crossing event, in the in-situ study. Figure 4.6 shows that most ambiguities can be resolved within 5 future doorway crossing events. To quantify this in terms of time, we looked at the inter-

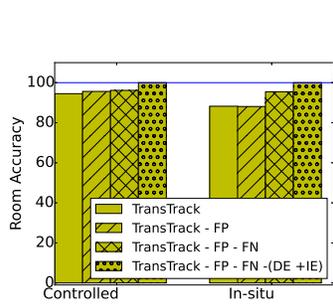


Figure 4.7: Effect of error removal: As errors get removed, there is generally an increase in accuracy owing to lesser ambiguity.

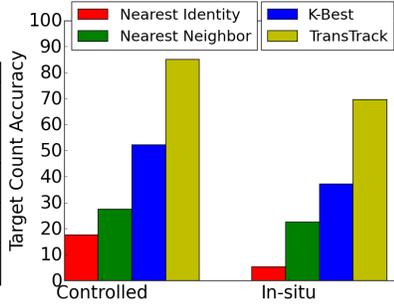


Figure 4.8: TransTrack which addresses the Phantom Target and Hidden Target problems tracked the correct number of targets in a home better than any of the baselines.

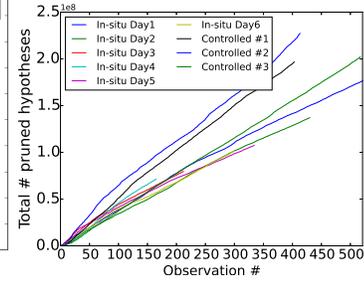


Figure 4.9: Cumulative number of hypotheses pruned by TransTrack’s equal-state pruning strategy : On average 394397 hypotheses are pruned at every observation.

arrival time between consecutive doorway events. We noticed that nearly 90% of events arrive within 30seconds of the previous event. This effectively means 5 future events roughly translates to around 150 seconds of future. These results indicate that *TransTrack* could support applications such as HVAC control.

We next calculated another metric *Target Count Accuracy* defined as the fraction of observations for which tracking and ground truth had the same number of targets. Figure 4.8 shows that *TransTrack* achieves nearly 30% better accuracy than its nearest baseline. The Maximum Likelihood approach gave nearly 0% accuracy. TransTrack performs better than the rest as it addresses the *Phantom Target* and *Hidden Target* problems.

Next, we calculated the benefits of the equal-state pruning strategy. Figure 4.9 shows the cumulative number of hypotheses pruned by TransTrack’s equal-state pruning strategy upto every observation. It was noted that on average 394397 hypotheses were pruned in every observation. This large pruning also explains why *TransTrack* performs better than *K-best*.

We point out that there is an inherent accuracy-complexity trade-off in using *TransTrack*. Even though *TransTrack* achieves higher accuracy than the greedy baselines, it trades-off higher time and space complexity (exponential in number of targets)

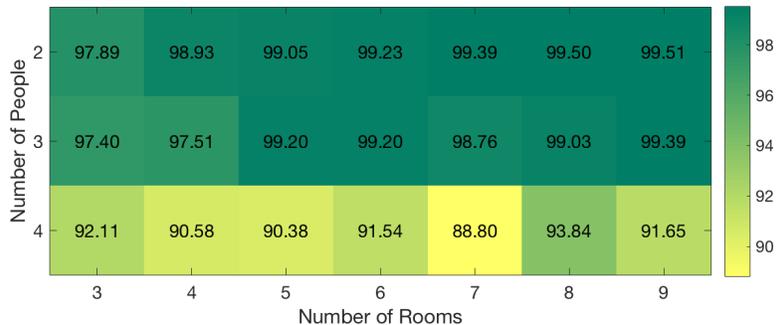


Figure 4.10: Whole home tracking simulation of *FormaTrack* with 15 different floorplans obtained from Google Images and various number of people: *FormaTrack* can achieve an average room-level tracking accuracy of over 99%, 98% and 91% for 2, 3 and 4 people respectively.

as it retains state information to help in disambiguation. However, this is typically not an issue for transition tracking in domains like homes, given the low number of targets at any given time.

4.5.1 TransTrack on FormaTrack data

For cost reasons, we were unable to measure room-level tracking accuracy of *FormaTrack* via a full-home deployment. As an alternative, we perform a tracking simulation by considering 15 different floor plans (obtained from Google Images) of homes varying from 3 to 9 rooms. For each home under test, we consider 2 to 4 inhabitants (limited by the computational complexity of the tracking algorithm [6, 119, 120]). These inhabitants were chosen at random from our pool of 8 participants. For each person we model their room transitions via a uniform distribution, and model their dwell time (in a room) based on 6 days of in-situ data [119]. For each simulated crossing of a person, we pick a random crossing event performed by the corresponding participant on our instrumented doorway. We repeat the entire process 10 times, and obtain a total of 223,982 simulated crossings. We pass these crossings through *TransTrack* [119], a multiple-hypothesis tracking (MHT) [6] algorithm for doorways, and measure the *transition accuracy* [119] (F-score of precision and recall of crossings).

Figure 4.10 shows the average transition accuracy for different number of rooms and people. We see that even though the individual identity accuracies for 2, 3 and 4 person groups (from Figure 3.12) are 90%, 84% and 79% respectively, they translate to an average of over 99%, 98% and 91% of transition accuracy, respectively. This increase is because the tracking algorithm leverages future doorway crossing events to help rectify mistakes in prior crossing events.

4.6 Summary

In this chapter, we present TransTrack, an algorithm to track a variable number of targets by sensing only their transitions. The presence of sensing errors and large sampling period relative to the potential speed of the target leads to uncertainty in the number of targets. We show that there exists a fundamental tradeoff between the number of targets tracked and the sensing errors they cause. Our evaluation of *TransTrack* on 3 controlled studies and 6 days of real-world in-situ data showed that *TransTrack* consistently performed better than the baselines.

We believe that the findings presented here will become more important with time as more diverse and non-invasive sensors get deployed. For instance, we envision the evaluated doorway tracking system to be augmented with motion sensors, which can observe state. This warrants the need for novel fusion tracking algorithms with transition sensors observing identity, and state-observing sensors detecting presence. Such algorithms could augment our findings with the well-studied state sensing literature [21, 22, 86].

Chapter 5

Towards Low Power Room-level Tracking

Thus far, we have been talking about one metric of interest namely *accuracy*. However, there is another important metric of practicality – *power*. This is because most doorways do not have a power outlet near them, and we cannot expect people to draw long wires in order to power these sensors. The next component *Doorpler* addresses this problem by performing crossing detection and direction estimation, and turning on the higher power *FormaTrack* hardware only when someone is at the doorway for identity estimation. At a higher level, since *Doorpler* performs crossing detection and direction estimation, it can alternatively be also thought of as a stand-alone system that performs room occupancy sensing.

5.1 Introduction

Many homes today are logically or physically “zoned” — based on HVACs [121], tasks performed (rooms), physical layout (floors), etc. A sensing technology that can accurately sense this zone occupancy can obtain energy savings (in the order

of 20-30% [122]), perform automatic lighting control (i.e. lights automatically come on when one enters a room, and goes off during an exit in real-time — just like a human would do), aid in energy disaggregation in 35 million single-person households in the US [123], perform elderly monitoring in the 13 million elderly single-person households [124], and help carry out “eyes-off” security (e.g. when a person exits a home through the back door, an unlocked front door locks itself).

The most common off-the-shelf solution for zone occupancy that exists in homes is a motion sensor. However, a single motion sensor has no notion of direction (i.e. it cannot distinguish between a zone crossing and a nearby hover). Furthermore, these motion sensors infer zone exit from lack of motion. Consequently, occupied periods can be mis-classified as unoccupied (“the waving hand at motion sensor problem”), and zone exit events become non real-time. On the other hand, many doorway tracking systems exist in literature [25, 28, 29, 45, 125] that can sense zone occupancy at zone transition spots (e.g. doorway) by detecting a crossing and the direction of movement. However, these systems are high-power [28, 29, 45, 125], cannot distinguish between a near-door event (e.g. hover) and a real crossing [25, 28, 125], are pets unfriendly [25, 28, 29], or depend on the ambient lighting, air flow or temperature [83, 84]. Consequently, in this chapter, we ask the question, how can we build a system that can perform crossing detection and direction estimation at *low-power* (in a harvestable range as most zone transition spots such as doorways do not have a nearby power outlet), in *real-time* and with a small form-factor (since space is at a premium in a doorway), while addressing the above limitations.

To answer this question, we build *Doorpler*, a radar-based sensing system that performs crossing detection and direction estimation using the simplest radio frequency (RF) signal, namely a tone (a continuous wave at a constant frequency), while adhering to the time, space, and power constraints of the application. *Doorpler* is mounted

atop a zone transition spot such as a doorway, and detects a crossing by leveraging the *Doppler Effect* — a person walking towards the radar causes an increase in the frequency of the transmitted signal. It estimates the direction of crossing by computing the angle-of-arrival (AoA) of the signal reflected by the human. It leverages the intuition that a person walking through the doorway creates a few “good” reflections where they reflect directly towards the radar [44], and ensures that these reflections are not inundated by the “bad” multi-path reflections coming off the environment. However, unlike many conventional radar direction finding systems which consume time, space, and power [9, 69, 70, 72, 73], *Doorpler* is *real-time*, *space-efficient*, and *power-aware* (i.e. within harvestable range). As there is a coarser requirement on the angular accuracy (i.e. we need to differentiate whether the angle of arrival of the human reflection is positive or negative depending on the side of the doorway), *Doorpler* employs an FFT-based technique that trades angular accuracy for computational complexity and relies on the phase difference between pairs of receiver elements. *Doorpler* operates these receiver elements in the 5.8GHz ISM band which allows for a compact array size of 7.8cm. Finally, since the interesting crossing events are sparse, *Doorpler* saves power via a dual-band wake-up radio technique [10, 11]. Accordingly, a lower frequency 2.4GHz radar (and hence lower power) is used for crossing detection, while a triggered higher frequency 5.8GHz (and hence higher power) array performs direction estimation.

To evaluate *Doorpler*, we study its accuracy, power consumption and real-timeness. We first conducted a scripted study with 8 participants of varying height and weight who were asked to walk through an instrumented doorway in different ways, every day for 6 consecutive days, producing over 1400 doorway crossing events. Our results show that *Doorpler* can achieve a precision, recall and direction accuracy of over 99% accuracy. Next, we performed two in-situ studies for 200 hours, on an instrumented doorway in a lab and a 2-person home, generating nearly 250 crossings. Despite the

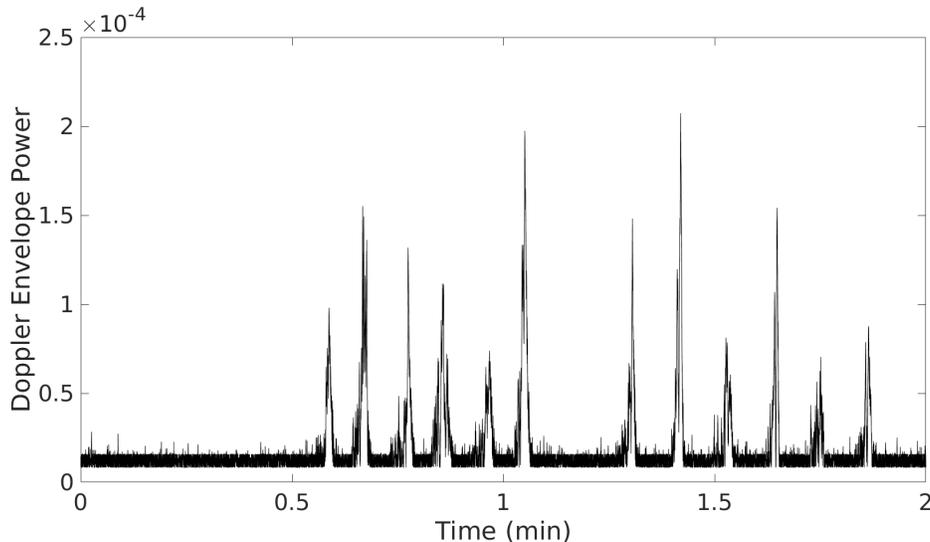


Figure 5.1: The envelope of the Doppler signal reflected back to *Doorpler* during a doorway crossing (a peak in the figure) is much larger than that in the absence of a crossing. There are 12 crossings in this figure.

uncontrolled environment, *Doorpler* achieves an average precision, recall and direction accuracy of 98.7%, 95.4% and 100%, respectively. Next, we estimate that a realization of *Doorpler* when instrumented atop a doorway would consume 6.1mW of power, falling in the harvestable solar range for indoor environments [126]. When *Doorpler* is augmented with a PIR sensor, we estimate that the average power consumption can be further reduced to 2.7mW. To evaluate the real-timeness of the system, we implemented the digital baseband processing on an ultra-low power microcontroller. Our results show an execution time of 13.8ms, thus having the potential to enable several real-time smart home applications like smart-lighting, HVAC control.

5.2 Approach

Doorpler is a radar-based system mounted atop the doorway. It performs crossing detection and direction estimation using only an RF tone. It detects a crossing by leveraging the Doppler effect – i.e. the receiver observes a shift in the transmitted frequency due to human motion. It estimates the direction of human transition by

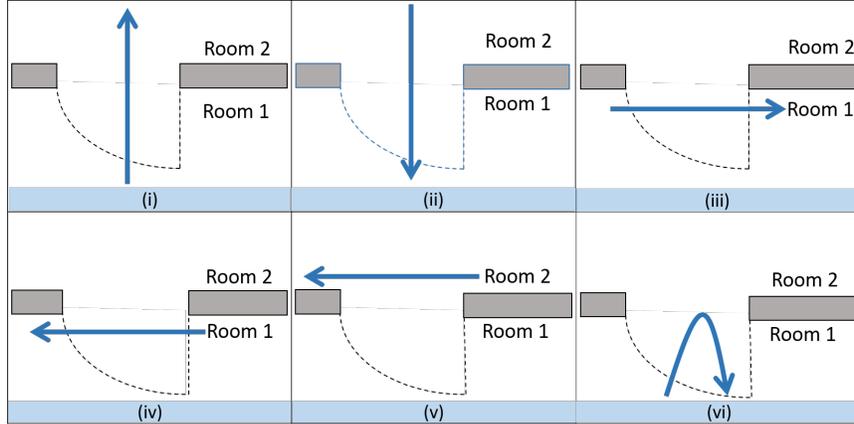


Figure 5.2: A positive Doppler shift followed by a negative Doppler shift can occur not only when a person walks through the doorway (cases (i) and (ii)), but also due to other movements near the doorway such as hovers and U-turns (cases (iii) - (vi)).

computing the angle-of-arrival (AoA) of the reflection coming from the human onto an antenna array. To realize this, *Doorpler* takes a layered approach owing to a *space-power* tradeoff. Accordingly, a lower frequency of operation results in a lower power consumption [8]. However, a lower operating frequency also results in a large antenna array that can out-span the door (as the array size depends on the wavelength [9]). In other words, *Doorpler* wants to transmit at a low frequency for power sake but also at a high frequency for spatial benefits. To handle this trade-off, *Doorpler* uses a technique called *dual-band wake up radio* [10, 11] by operating at two different ISM bands (2.4GHz and 5.8GHz). The lower power 2.4GHz radio performs crossing detection and triggers the higher power 5.8GHz array for direction estimation, only when a crossing is detected. We next explain the design details of *Doorpler*.

5.2.1 Crossing Detection

In order to detect a crossing event, *Doorpler* relies on the *Doppler Effect*. Accordingly, when a target moves towards the receiver during a radio transmission, the target acts as a virtual transmitter by reflecting the transmitted signal with a frequency larger

than the transmitted frequency. This phenomenon is referred to as the *Positive Doppler Effect*. We next describe how this *Positive Doppler* is leveraged by *Doorpler*.

Doorpler uses a 2.4GHz RF transmitter which transmits a carrier signal that is given by [127]

$$x(t) = A \cos(2\pi f_c t) \quad (5.1)$$

where A is the transmit signal magnitude and f_c represents the carrier frequency of 2.4GHz. This transmitted signal propagates through air and is received by an antenna placed at the other end of the doorway. This received signal is given by [128]

$$y(t) = \eta A \cos(2\pi f_c (t - \tau)) \quad (5.2)$$

where η is the attenuation factor, and τ represents the propagation time. However, the transmitted signal does not travel along just one path from the transmitter to the receiver. The transmitted signal gets reflected by the objects in the environment resulting in multiple copies of the same signal arriving at the receiver. This is referred to as *multipath propagation*, and the super-imposed received signal at the receiver due to the N propagation paths is given by

$$y(t) = \sum_{i=1}^N \eta_i A \cos(2\pi f_c (t - \tau_i)) \quad (5.3)$$

where η_i and τ_i represent the attenuation factor and propagation time for the i^{th} path. Now, when a person walks towards the doorway during such a radio transmission, she will reflect a signal which will arrive at the receiver with a frequency (f_c') larger than the transmitted frequency. This is given by

$$y'(t) = \eta A \cos(2\pi f_c' (t - \tau)) \quad (5.4)$$

This frequency difference (Δf) between the transmitted (f_c) and received frequency (f_c') is referred to as the *Doppler shift*. The Doppler shift caused by a target moving at velocity v at an angle θ relative to the receiver, is given by [109]

$$\Delta f = \frac{2 * f_c * v * \cos\theta}{c} \quad (5.5)$$

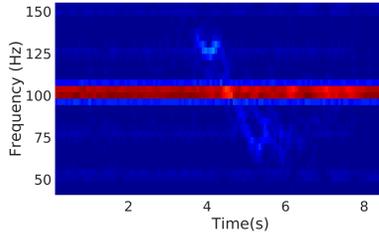


Figure 5.3: Each receiver computes an *amplitude spectrogram* - a measure of the strength of each frequency component over time. The faint signal on either side of the baseband tone shows the Doppler reflections due to a human walking through the doorway¹.

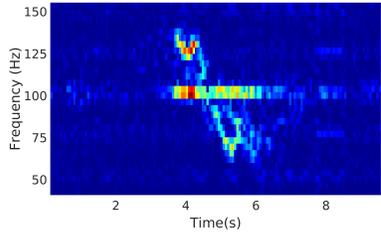


Figure 5.4: Each receiver performs spectrogram enhancement on the amplitude spectrogram resulting in a more visible Doppler signal, and baseband tone mitigation.

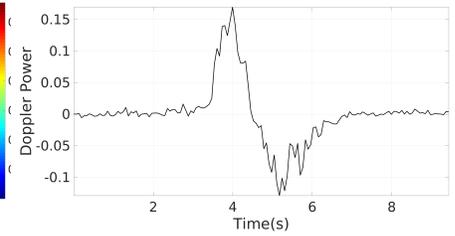


Figure 5.5: The Doppler Power (DP) is a measure to infer the moment, the person crosses the sensor. There is positive DP when the person approaches the doorway, and negative DP when the person exits. Hence, the zero-crossing of DP tells us when the person crosses the sensor.

where f_c is the transmitter's center frequency and c is the speed of light in the transmission medium. Given a center frequency of 2.4GHz, and an average human walking speed of 1.2 to 1.3 ms^{-1} [111], the maximum Doppler shift will be about 21Hz. Similarly, when the person walks away from the doorway during an RF transmission, her reflection will arrive at the receiver with a frequency less than the transmitted frequency, resulting in a *negative Doppler shift*.

Doorpler captures the *positive Doppler shift* in order to detect a crossing (*negative Doppler* happens after the person has crossed the doorway threshold). It does so in two steps. First, it tries to extract $y'(t)$ from the received signal via a Butterworth bandpass filter (cutoff frequency of 3 to 25Hz), such that only the reflection from the human remains. Next, it obtains the envelope of the filtered signal, and detects a crossing only when the envelope power is larger than a threshold (set as 5 times the noise-floor). Fig. 5.1 shows the envelope power of the Doppler filtered signal for 12 doorway crossings. We can clearly see that the envelope power during a crossing is much larger than that during a crossing absence. The advantage of this technique is also that both the filtering and the envelope detection can be performed entirely in analog at just a few microwatts of power [129, 130].

When a potential crossing is detected, the 2.4GHz sensor triggers on a higher power 5.8GHz radio array. The 5GHz radio performs two tasks - (i) direction estimation, and (ii) crossing confirmation. Crossing confirmation is necessary because the crossing detected via the above technique by the 2.4GHz radio can result in false positives. This is because any approaching movement by a person towards the doorway, when she is close to the doorway can cause positive Doppler. For example, all cases shown in Fig. 5.2 will result in positive Doppler (until the person reaches the line of the receiver), followed by a negative Doppler. However, only cases (i) and (ii) are true doorway crossing events. As the above approach will treat all 6 cases to be true crossing events, the 5GHz radio is used to filter out these false positive cases.

5.2.2 Direction Estimation

The triggered 5.8GHz receiver-array is used for direction estimation. We point out that *Doorpler* cannot simply use positive and negative Doppler to obtain direction because irrespective of which side the person crosses the doorway from, she will cause positive Doppler during approach and a negative Doppler during exit. As a result, *Doorpler* estimates the direction of the person crossing the doorway by calculating the *angle-of-arrival* (AoA) of the weak reflected signal coming off the human. The AoA of this signal will be positive when the person is on one-side of the doorway, and negative when the person is on the other side. Finally, *Doorpler* fuses angle estimates from multiple (four) antennas in order to improve the direction estimate. Given this overview, we next explain the details of *Doorpler's* direction estimation.

Upon being triggered by the 2.4GHz radio, each element of the 5.8GHz array transforms the received multipath-rich raw time-domain baseband samples into spectrograms via the Short Time Fourier Transform (STFT). The STFT essentially employs

¹ All spectrogram figures are best viewed in color.

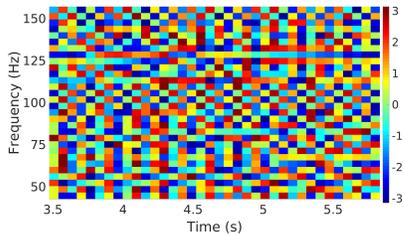


Figure 5.6: Each receiver computes a *phase spectrogram* indicative of the starting phase of each frequency component over time. The phase spectrogram by itself is not very useful, but phase difference computed on a pair of antennas reveals useful direction information.

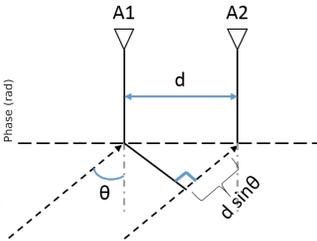


Figure 5.7: If two antennas are separated by a distance d , then a signal incident at angle θ , travels an extra distance of $d \sin\theta$ to the second antenna. This results in an instantaneous phase difference of $2\pi*d*\sin\theta/\lambda$ between the two antennas.

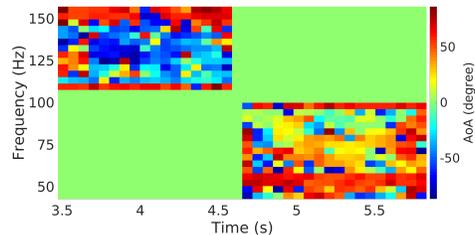


Figure 5.8: Each receiver pair computes an angle-of-arrival (AoA) spectrogram on either side of the doorway crossing point. The AoA of the Doppler signal is predominantly negative when the person is on one side of the doorway, and positive when she is on the other side.

a sliding window over the received time-domain samples and then performs a Fast Fourier transform (FFT) on each window. The resulting spectrogram is essentially a three dimensional plot representing the frequency domain of the received signal over time (i.e. x-axis is time, y-axis is frequency and z-axis is corresponding metric that is analyzed, namely amplitude or starting phase of the signal). We henceforth refer to the spectrogram with an amplitude z-axis as the *amplitude spectrogram*, the one with a phase z-axis as the *phase spectrogram*, and so on. For example, Fig. 5.3 shows an example of an amplitude spectrogram.

Step 1: Compute Amplitude Spectrogram - At first, each array element computes an *amplitude spectrogram* (Fig. 5.3). Each cell (i,j) represents the power of a certain frequency component at a given time. The strong signal at the center of this spectrogram represents the transmitted signal and its multipath reflections (Equation 5.3). The thin contour (4 to 6s) around it shows the Doppler reflections (Equation 5.4) of a human walking through the doorway. In the frequency domain, a spectrogram stretches from 0 to S Hz, where S is the baseband sampling rate. Not all these frequencies are of interest - i.e. we only care about the frequencies around the transmitted tone frequency where the Doppler shifts occur. From Equation 5.5,

for an average human walking speed of 1.3m/s during a 5.8GHz RF transmission, this corresponds to a Doppler shift of about 50Hz. As a result, *Doorpler* only considers a “smaller” amplitude spectrogram that is +/- 50Hz around the transmitted baseband tone frequency (f_{tone}). We refer to this frequency range as f_{min} to f_{max} . This reduces the computational load on the microcontroller that performs the digital baseband processing.

Step 2: Perform Spectrogram Enhancement - The Doppler reflected signals in the frequency bands around the transmitted tone are extremely faint. Consequently, each receiver performs spectrogram enhancement [32, 131] on the aforementioned amplitude spectrogram in order to extract the weak Doppler signal. This is done by first normalizing the amplitude spectrogram with respect to each time bin (i.e. a normalization per column). As a result of this step, the tone-band will have the highest (unit) magnitude. Next, we subtract each column (frequency) of this computed spectrogram from a background column-vector. This background column vector is computed by averaging a similarly normalized background spectrogram of 5 seconds (that is computed initially). Consequently now, the tone band gets mitigated, and the Doppler bands become “visible”, during motion. Fig. 5.4 shows an example of an enhanced amplitude spectrogram ($AS_{enh}(f,t)$), and we see the Doppler bands becoming more visible.

Step 3: Identify Zero Crossing - In order to compare the AoA of the human-reflected signal on each side of the doorway, *Doorpler* first determines the moment the person was at the doorway (i.e. underneath the sensor). It does so by identifying the point of transition from positive to negative Doppler (similar to pseudo-Doppler direction finding radars [76]). It identifies this Doppler transition point by computing a measure called *Doppler Power* (DP). This measure is obtained by weighting the power value from the enhanced amplitude spectrogram with the corresponding

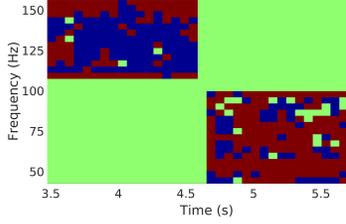


Figure 5.9: Since *Doorpler* only cares about the direction of doorway transition, each receiver pair quantizes the AoA estimates to -1, 0 or +1, depending on the sign of the AoA.

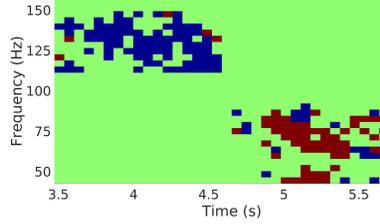


Figure 5.10: *Doorpler* mitigates the effect of secondary Doppler reflections by fusing AoA estimates from multiple antenna pairs. It computes a *consensus spectrogram* where each quantized AoA cell must be agreed upon by all antenna pairs.

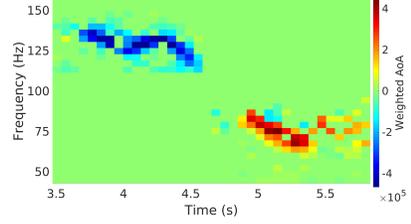


Figure 5.11: *Doorpler* further reduces the effect of secondary reflections by weighting the consensed quantization with the corresponding power value from all the receivers. Consequently, Doppler reflections coming straight from the human get weighted higher than the secondary Doppler reflections.

Doppler sign (+1 for positive Doppler bands and -1 for negative Doppler bands).

More formally,

$$DP(t) = \sum_{f=f_{min}}^{f_{max}} \text{sign}(f) \times AS_{enh}(f, t), \quad (5.6)$$

$$\text{where } \text{sign}(f) = \begin{cases} +1, f > f_{tone} \text{ (Positive Doppler)} \\ -1, f < f_{tone} \text{ (Negative Doppler)} \end{cases}. \quad (5.7)$$

Intuitively, the above is a measure that captures the cumulative Doppler power (in the frequency bands corresponding to human motion), factoring in the manner of movement (approach v/s exit). Via this measure, a person approaching the doorway causes positive Doppler Power, while a person exiting causes negative Doppler Power. Hence, if we calculate Doppler Power over the entire crossing duration, then the zero-crossing would give us the Doppler transition point. Fig. 5.5 shows the Doppler Power during a doorway crossing for one of the antennas. In this figure, we can clearly see the zero-crossing of interest around 4.5 second. *Doorpler* determines this zero-crossing point via a technique similar to FormaTrack [125]. We denote this zero-crossing time as T_{cross} .

Step 4: Compute Phase Spectrogram - Having determined the moment the person is in the doorway, *Doorpler* next determines the direction of transition. As mentioned before, *Doorpler* determines direction by computing the AoA of the human-induced reflection. To measure this AoA, each array element first computes a *phase spectrogram* – x-axis is time, y-axis is frequency and z-axis is the starting phase of the signal. Fig. 5.6 shows an example of a phase spectrogram at a receiver, which appears to reveal little information.

However, as seen in Fig. 5.7, when two antennas A1 and A2 are placed at a distance d apart, a signal arriving at an angle θ to the antennas will travel an extra distance of $d \sin\theta$ to A2. This additional distance results in an instantaneous phase difference [55] between the two antennas of $\Delta\phi = \frac{2*\pi*d*\sin*\theta}{\lambda}$, where λ is the carrier wavelength (=5.17cm for a 5.8GHz signal). When d is half-wavelength ($\lambda/2$), the AoA is given by

$$\theta = \arcsin \frac{\Delta\phi}{\pi} \quad (5.8)$$

Consequently, *Doorpler* takes the *phase spectrogram* for two successive receivers (i.e. $\lambda/2$ apart) , and calculates the phase difference ($\Delta\phi$) between them at every spectrogram cell. This results in a *phase difference spectrogram*, for every antenna pair.

However, the received signal in an antenna is the superposition of multiple paths, and hence the above equation breaks down if applied directly on the received signal. To isolate the reflection from the moving target alone, *Doorpler* computes the phase difference only in the Doppler bands (+/- 6Hz to +/- 50Hz from the tone frequency), leveraging the intuition that the Doppler reflections come from the moving human. We mitigate the effect of secondary reflections - i.e. transmitter -> human -> environment -> receiver via Step 7.

Step 5: Compute Angle-of-Arrival (AoA) Spectrogram - From each com-

puted *phase difference spectrogram*, *Doorpler* next computes an AoA spectrogram based on Equation 5.8. Intuitively, it is the AoA of the Doppler induced reflection (when it exists), for each time step. Next, we leverage the fact that a person causes positive Doppler while approaching the doorway, and negative Doppler while exiting. Consequently, the AoA spectrogram is only computed in the positive Doppler bands before the person reaches the doorway (i.e. T_{cross}), and in the negative Doppler bands after the person exits the doorway. More formally, given a phase difference spectrogram $\Delta\phi_{sgram}$, the AoA spectrogram AoA_{sgram} is given by

$$AoA_{sgram}(f, t) = \begin{cases} \arcsin \frac{\Delta\phi(f,t)}{\pi}, & \text{if (i) } f > f_{tone} \text{ and } t < T_{cross} \\ & \text{(ii) } f < f_{tone} \text{ and } t > T_{cross} \\ 0 & , \text{ else} \end{cases} \quad (5.9)$$

Fig. 5.8 shows an example of an AoA spectrogram. We can see that the AoA in the Doppler bands are mostly on one-side of 0° before T_{cross} (around 4.5 seconds), and on the other side of 0° after T_{cross} .

Step 6: Obtain Quantized AoA Spectrogram - In order to determine the direction of doorway transition, *Doorpler* only needs to know if the human reflection is at a positive or negative angle. Consequently, *Doorpler* quantizes the computed AoA spectrogram such that the cells with positive and negative angles are set to +1 and -1 respectively. This results in the *Quantized AoA spectrogram*, as shown in Fig. 5.9.

Step 7: Secondary Reflections Mitigation - In order to mitigate the effect of secondary reflections coming from the human, *Doorpler* fuses data from multiple antenna pairs. It leverages the intuition that as a person walks through the doorway, there will be a few “good” reflection points where the person reflects directly to the radar [44]. With the antennas located in far-field, all pairs will agree on the angle

quantization at these reflection points. Consequently, *Doorpler* forms a *consensus spectrogram* wherein each cell (i,j) has a quantized angle only when agreed upon by all receiver pairs. The *consensus spectrogram* in Fig. 5.10 shows a reduction in the number of bad angle estimates, compared to Fig. 5.9.

The *consensus spectrogram* could have certain ‘bad’ cells which do not agree with the actual direction of doorway crossing because of noise. However, these noisy cells have low power if they are not a reflection from the human. Consequently, we eliminate these noisy cells by weighting each cell (i,j) of the *consensus spectrogram* with the corresponding power value obtained by summing the (i,j)th cell in the *amplitude spectrogram* of all the antenna pairs. This step further mitigates secondary reflections as they will have lower power compared to those coming directly off the person. Fig. 5.11 shows the spectrogram after amplitude-weighting which exhibits a clear difference on the two sides of the doorway. (amplitude-weighting increases the recall by about 20%). Formally, if CS represents the *consensus spectrogram*, and AS_n represents the amplitude spectrogram of the nth receiver, then the resulting *amplitude weighted consensus spectrogram* (CS_{wt}) is given by

$$CS_{wt}(f, t) = CS(f, t) * \left(\sum_{n=1}^N AS(f, t) \right) \quad (5.10)$$

, where N is the number of receivers.

Step 8: Sign comparison - Finally, *Doorpler* determines the direction of transition by comparing the sum of the sub-matrices, pre and post doorway crossing (i.e. the left and right half of CS_{wt}). This weighted sum will change from positive to negative when the person walks from one side of the doorway to another, and from negative to positive, when she walks the other way. More formally, if we define transition from the positive to the negative side as IN, and vice versa as being OUT, then



(a) *Doorpler* was mounted atop an office doorway. A (b) *Doorpler* was mounted atop the most commonly used scripted study and an 80-hour in-situ study was performed doorway in a 2-person home for 120 hours yielding 113 on this doorway yielding over 1500 doorway crossing events. doorway crossing events.

Figure 5.12: *Doorpler* Experimental Setup

the direction estimate is given by

$$Dir = \begin{cases} IN, & \text{if } \sum_{t < t_{cross}} CS_{wt}(f, t) > 0 \text{ and } \sum_{t > t_{cross}} CS_{wt}(f, t) < 0 \\ OUT, & \text{if } \sum_{t < t_{cross}} CS_{wt}(f, t) < 0 \text{ and } \sum_{t > t_{cross}} CS_{wt}(f, t) > 0 \end{cases}$$

Step 9: Crossing Confirmation - As mentioned earlier, the 2.4GHz radio which performs crossing detection based on positive Doppler shifts can trigger false positives for near-door events such as hovers and U-turns. These false positives are eliminated via the sign comparison in Step 9. If the weighed sums on either side of the doorway crossing point are both positive (or) both negative, then *Doorpler* perceives that the person did not actually cross the doorway, and discards the event.

Finally, we also point out that we have a layered approach (i.e. 2.4GHz triggering on the 5GHz array) because a 2.4GHz phased array with four antenna elements placed half-wavelength apart will occupy a total of 18.75cm. This is longer than most doorjamb widths [71]. A 5.8GHz array on the other hand occupies a much smaller width spanning just 7.8cm, which can fit atop most doorways.

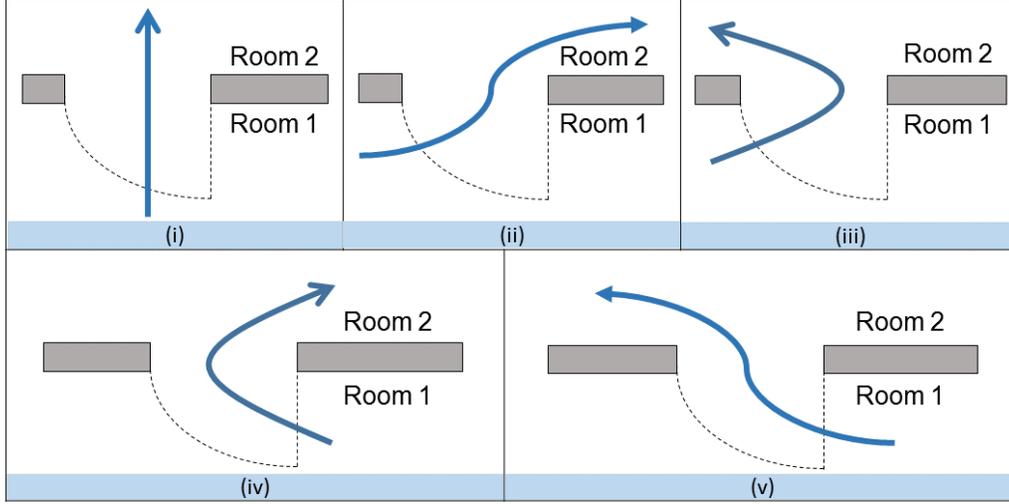


Figure 5.13: In the scripted study, participants walked through the instrumented doorway in 6 different ways yielding 1440 crossings.

5.3 Experimental Setup

To test our hypothesis, we implement the 2.4 GHz and the 5.8 GHz RF transceivers using software defined radios. The 2.4 GHz transmitter and receiver were realized by two USRP N210s [132] with an SBX daughterboard [133] each. The 5.8 GHz transmitter was implemented on a USRP N210 with a CBX daughterboard [134]. The 4-element 5.8 GHz receiver array was implemented on a USRP X310 [135] with two TwinRX daughterboards [136] that provide four phase-coherent RF receive chains. Each transceiver pair was frequency synchronized via a common reference clock, Octoclock-G [137]. Without frequency synchronization, the *Doppler* signal gets submerged in the carrier frequency difference between the transceivers (*Carrier Frequency Offset* [138]). The receivers were time synchronized (sample-aligned) via a pulse-per-second signal provided by the same Octoclock-G. The 5.8GHz transmitter loaded a 100 Hz baseband tone on its carrier whose Doppler was analyzed for both crossing confirmation and direction estimation. Both transmitters transmitted at just -10dBm ($100\mu W$) transmit power, while the receivers sampled at 250Hz. Each RF

Participant	P1	P2	P3	P4	P5	P6	P7	P8
Height (cm)	161	167	168	170	172	175	177	181
Weight (kg)	60.8	61.0	56.1	78.0	56.7	97.9	82.5	80.0

Table 5.1: The height and weight of the 8 participants who walked through an instrumented office doorway for 6 days generating 1440 doorway crossing events

chain was terminated by a 3dBi omni-directional antenna, and these antennas were mounted atop a doorway as shown in Fig. 5.12a and Fig. 5.12b. The transmit and receive antennas were mounted in a *gain-mismatch* fashion (pointing at each other so that the nulls align). This reduced the direct path by 10.2dB. The antennas in the 5.8 GHz phased-array are placed half-wavelength apart (2.58cm).

Finally, there is a constant but repeatable phase-offset between each of the 5.8 GHz receive-chains due to the different local oscillators involved. These phase offsets were eliminated via a one-time calibration [55]. The phase offset if left uncalibrated, would appear as an added phase difference in Equation 5.8, resulting in an incorrect angle estimate. We performed two sets of studies with this setup :

1. A scripted study was performed on an office doorway (Fig. 5.12a), involving 8 participants of varying heights and weights as shown in Table 5.1. The participants were asked to walk for 6 days through the instrumented doorway. On each day, each participant walked 6 times (3 times back and forth) through the doorway in the 6 ways shown in Fig. 5.13. No restrictions were imposed on the type of clothing the participants wore, or the time of the experiment. In all, this study yielded 1440 doorway crossings (+ 144 U-turn events).

2. Two sets of in-situ experiments were performed. A first in-situ study was performed on the same office door for 80 hours which yielded 133 doorway crossings. A second in-situ study was performed on the most commonly used doorway in a 2-person home (Fig. 5.12b). This study was performed for 120 hours, and resulted in 113 doorway crossing events.

The crossings were recorded by a video camera pointed at the doorway, which

were then manually analyzed. We evaluate *Doorpler* accuracy via four metrics:

- *Recall*: The fraction of actual doorway crossings that were correctly detected by *Doorpler*.
- *Precision*: Amongst the doorway crossings detected by *Doorpler*, the fraction that actually occurred.
- *Direction Accuracy* (DirAcc): The fraction of correctly detected doorway crossings having the correct direction.
- *Effective Direction Accuracy* (EffDirAcc): The crossing confirmation (Section 5.2.2) depends on the direction estimate. Consequently, an incorrect direction estimate can manifest itself as a false negative, false positive or a direction error. This metric captures this manifestation as the mean of *recall*, *precision* and *direction accuracy*.

Finally, there are no health concerns with *Doorpler* as its Effective Isotropic Radiated Power (EIRP) after accounting for antenna gain and cable loss is just $125\mu\text{W}$ (-9dBm). In comparison, the maximum FCC permitted transmit power for an indoor 5GHz WiFi access point is 1W [139] (nearly 8000x higher).

5.4 Evaluation

5.4.1 Doorpler Accuracy

Table 5.2 shows that *Doorpler* achieved over 99% and 95% accuracy across all metrics of interest in the scripted and in-situ study respectively. The missed detections in the in-situ study are attributable to the following causes: (a) two people walking one behind the other through the doorway (occlusion), (b) several cases of people walking all the way up to the door, talking to someone in the room for a few seconds, and then continuing motion into the room, (c) direction-errors (i.e. a true crossing event

Study	Metric(%)	Recall	Precision	DirAcc	EffDirAcc
Scripted		99.0	99.9	100.0	99.6
In-situ		95.4	98.7	100.0	98.0

Table 5.2: *Doorpler* achieves over 99% accuracy across all metrics of interest in the scripted study. It also achieves an average accuracy of over 95% across all metrics in the 200 hours of in-situ data. was incorrectly detected as a U-turn/near-door event), (d) not all the 5GHz radio chains were triggered because of the conservative threshold that trades-off precision and recall, and finally (e) a case of a person located very close to the doorway, walks through the doorway. In this case, the positive Doppler received by the 2.4GHz radio was not significant. The causes for the 3 false detections over the 200 hours of data collection were due to direction errors (i.e. a U-turn/near-door event was not detected). In all these cases, the weighted consensus sum on one of the sides was only marginally greater (or lesser) than zero. We leave it as a future work to filter out these low-confidence crossing events.

5.4.2 Power Consumption

We next study the power consumption of *Doorpler*. We cannot take power numbers directly from the USRPs as they are over-engineered for our use-case. For e.g., the components in its radio chain (i) operate over a multi-GHz band (while we operate at a single frequency), (ii) can transmit at over +10dBm (we transmit 100x lower at -10dBm), (iii) can receive signals as low as -130dBm (a human reflection 1m away from the setup comes at -66dBm [140]), (iv) has an ADC with a sampling rate of 200MHz (we sample at 250Hz), etc. Consequently, we come up with an equivalent realization of *Doorpler* based on the USRP radio chain, and obtain power numbers of this realization from literature. We leave it as a future work to engineer the analog integrated system based on the provided design. Fig. 5.14 shows a high-level block diagram of *Doorpler*'s power hungry RF components with the 2.4GHz RF chain

System	Doorjamb [25]	SonicDoor [29]	FORK [45]	PeopleFlow [84]	FormaTrack [125]	Doorjamb 2.0 [28]	Lethe [83]	Doorpler (5GHz)
Technology	Ultrasound	Ultrasound	Depth Camera	IR	UWB Radar	Ultrasound + IR	Thermal Camera	CW Radar
Power (mW)	150	300	26500*	714.9*	2450*	595.5	300	66.3*

Table 5.3: The triggered component of *Doorpler* (5.8GHz array) consumes 2.3x lesser power than the nearest doorway tracking system (*includes signal processing power consumption when real time claim reported by authors).

waking up the 5GHz array upon crossing detection.

2.4GHz radio : Liu et al. [141] built a -10dBm RF front-end similar to *Doorpler* for Bluetooth Low Energy, Zigbee and Medical Body Area Network applications. It consumes 4.6mW for transmission and 3.8mW for reception. However with shareable components like the oscillator, the power consumption becomes 6.6mW. In our studies, we observed that the 2.4GHz radios can be 15% duty-cycled without accuracy loss. This results in a power-consumption of 5.8mW. We point out that this system exceeds *Doorpler's* necessity. For e.g., the receiver sensitivity is -96dBm (*Doorpler* is about -61dBm), the data rate is over 950Kbps (*Doorpler* samples at just 250 samples/sec, i.e. 2Kbps), the radio chains support modulation techniques like GFSK, DQPSK, etc (*Doorpler* runs on an unmodulated tone). The envelope detector to trigger the 5GHz chain can be realized in tens of microwatts of power [129]. Furthermore, we point out that unlike typical transceivers, we do not require components like the oscillators to be stable over a long-term (as they are shared between TX and RXs). Said differently, *Doorpler* is not drift sensitive, and only cares about short-term stability (crossing duration).

5GHz RF chain : Similarly, Homayoun et al. [142] built a 11.6mW receiver for 802.11a applications. With 4 receive chains, the total power consumption becomes 46.4mW. This receiver has a sensitivity of -70dBm at 54Mbps, and a noise figure of 6dB which exceed *Doorpler's* requirements. The transmitter can be realized via a -7dBm, 13.5mW frequency synthesizer [143], eliminating the need for an on-chip power amplifier, as the output power is high enough. This results in a total power consumption of 59.9mW for the 5GHz chain.

Digital Baseband Processing : To measure the power consumption of the micro-controller unit (MCU), we implemented the digital baseband processing of *Doorpler* on an ultra-low power MCU, MSP432 [144], and measured its power consumption. We observed that the MSP432 consumed 6.4mW of power. This results in a total power consumption of the 5GHz chain of 66.3mW (= 59.9 + 6.4)mW.

Average power and comparison with other doorway tracking systems :

The average power consumption of *Doorpler* is given by (*Power Draw of 2.4GHz radio*) + (*On Time of 5GHz array*)*(*Power Draw of 5GHz array*). In our study, we observed that the 2.4GHz radio triggers the array 0.4% of the time. This results in an average power consumption of 6.1 (=5.8 + (0.4% * 66.3)) mW.

Table 5.3 compares the power consumption of the triggered component of *Doorpler* with other doorway-tracking systems. We report the power numbers this way as all of these systems can technically be triggered by the 2.4GHz radar, even though it is not part of their setup. From this table, we observe that the triggered component of *Doorpler* consumes 2.3x lower power than the nearest baseline. An alternate way to interpret its advantage is that should *Doorpler* be part of an environment where near-door events occur frequently (e.g., adjacent to a busy hallway), it will result in 2.3x less power consumption each time it is triggered by a person walking close to the door in the hallway. Furthermore as mentioned earlier, *Doorpler* being an RF-based system does not suffer from many of the limitations of other doorway tracking systems.

Energy harvesting feasibility: Prior work [126] has shown that indoor incident solar irradiation varies from 11-115 μ W/cm². Given a 1m x 10cm solar panel of with 20% efficiency, mounted in the doorway, this translates to a power supply of 2.2 to 23mW. Consequently, *Doorpler's* demand of 6.1mW can be satisfied by many of the doorways. For those doorways with low irradiance, *Doorpler* could still potentially be

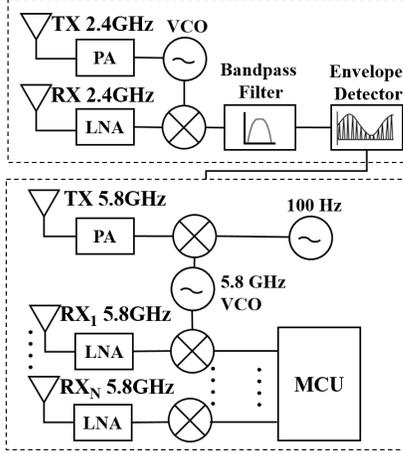


Figure 5.14: High-level *Doorpler* block diagram: The lower power 2.4GHz radio triggers the higher power 5.8GHz chain upon crossing detection.

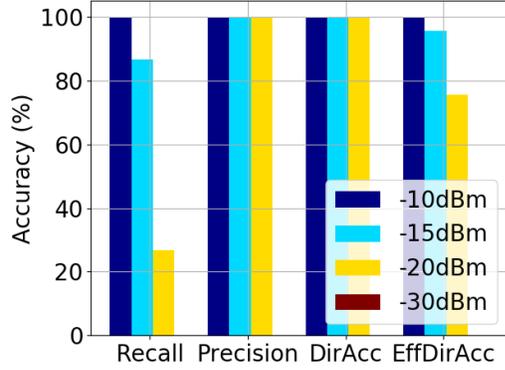


Figure 5.15: As the transmit power of *Doorpler* is reduced, the accuracy starts to decrease. This is because the reflected Doppler signal becomes weaker and starts to submerge itself in the noise-floor.

harvestable by placing larger solar panels and/or placing them above the door frame on either side of the doorway. From Table 5.3, we also point out that *Doorpler* can be within the harvestable range so long as the 5GHz radio chain is triggered for less than 32% ($=23\text{mW}/(5.8+66.3)\text{mW}$) of the time.

PIR Augmentation: We next consider the possibility of a three-tier system wherein the 2.4GHz radio gets triggered by an even lower power system – a PIR motion sensor. For this, we analyze the data from one of our prior in situ studies [119]. The 6-day study contained 1756 doorway crossing events from 8 doors in a home. Each doorway was equipped with two PIR motion sensors facing each of the adjoining rooms. Figure 5.16 shows the *recall* (i.e. the fraction of crossings detectable by a system like *Doorpler* that would be triggered by a motion sensor) compared against the fraction of time a system like *Doorpler* would be ON. We see that we can achieve 99% recall by keeping the 2.4GHz radio of *Doorpler* ON for just 30% of the day. (i.e. the 2.4GHz radio can be turned OFF for 70% off the time). With PIRs consuming $180\mu\text{W}$ and $600\mu\text{W}$ of idle and active power draw [145], this dual PIR augmentation results in an average power consumption of 2.7mW ($= 0.18 * 70\% + (1.2+5.8)*$

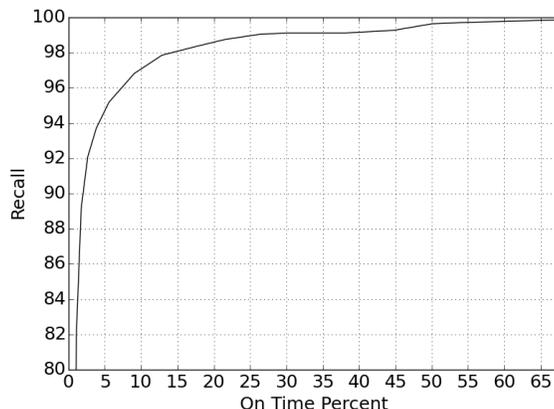


Figure 5.16: By augmenting *Doorpler* with a motion sensor, *Doorpler* can achieve 99% recall with the 2.4GHz radio remaining ON for less than 30% of the time.

$30\% + (0.4\% * 66.3)$. When augmented with the identity sensing system such as *FormaTrack* which consumes 2.2W, the aggregate estimated power draw of the entire system would be 11.5mW ($=2.7 + (0.4\% * 2200)$). This results a lower power draw compared to the 2-tier system and continues to be in the harvestable range of indoor environments.

5.4.3 Real-timeness of *Doorpler*

We next evaluate if *Doorpler* can operate in real-time by measuring the run-time of the baseband processing on the MSP432. We observed that the direction can be estimated by the MSP432 at an average of 13.8ms. With 750ms of crossing data (Section 5.4.6) on each side of the doorway needed to achieve a high accuracy, a direction estimate can be provided just 763.8ms($=750+13.8$ ms) after the person crosses the sensor. Anecdotally, at an average walking speed of 1.2m/s, a human covers just over a step within a 763ms duration. Thus, *Doorpler* can enable several real-time smart home applications like smart-lighting or automatic HVAC control. From an implementation perspective, to save memory on the MCU, steps 1, 2, 4 to 6 of Section 5.2.2 were implemented in a streaming manner (i.e. per arriving FFT

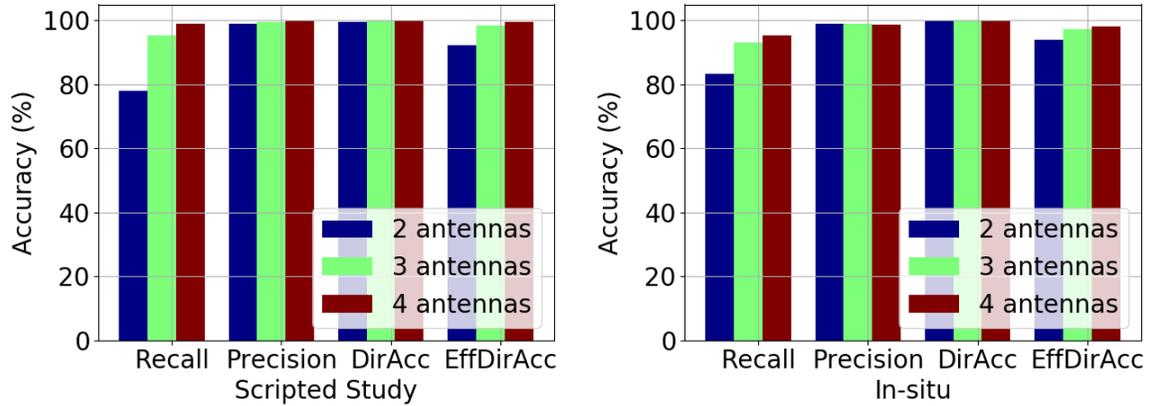


Figure 5.17: Effect of number of receive chains : The recall and effective direction accuracy increase as we add more receivers. This is because AoA errors made by a single receive pair get mitigated while taking a consensus. *Doorpler* uses 4 receive chains.

frame).

5.4.4 Effect of Number of Receive Chains

The number of receivers has an impact on the total power consumption of *Doorpler*. From Fig. 5.17, we see that the recall and effective direction accuracy increases as we add more receivers. This is because AoA errors made by a single receiver pair gets mitigated during consensus. We point out that there is an inherent *power-accuracy trade-off* here - with more receivers, even though the accuracy increases, the power consumption also increases by 11.6mW for every RX-chain. However, *Doorpler* still uses all 4 RX-chains because the array comes on for only a small fraction (0.4%) of the time.

5.4.5 Effect of Transmit Power

We next study the effect of lowering the transmit power, as a lower transmit power typically results in lower power consumption [8]. One person was asked to walk a total of 120 times in varying directions (Fig. 5.13), through an instrumented doorway

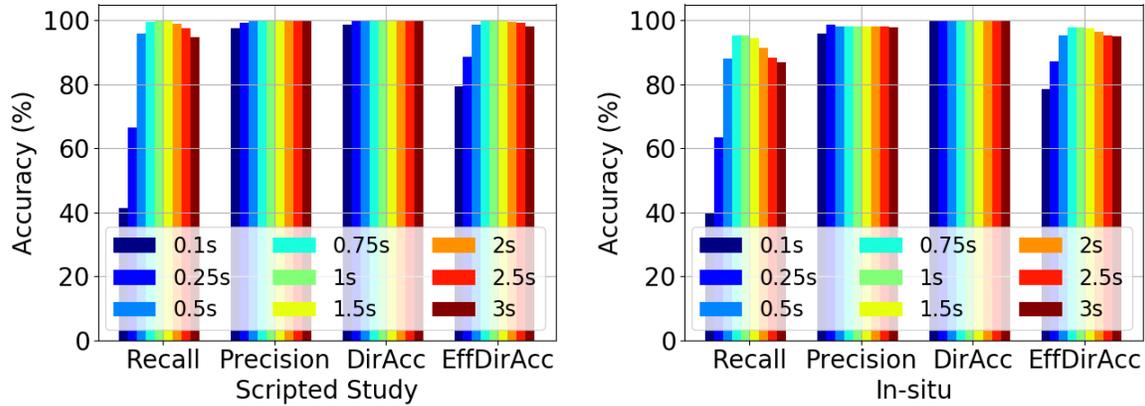


Figure 5.18: The 5GHz array works with high accuracy as long as it is triggered at least 750ms before the person crosses the doorway. With a late trigger, the accuracy suffers because the Doppler is not significant. With an early trigger, the accuracy suffers as noisy spectrogram cells, or data from a prior crossing starts to dominate.

(Fig. 5.12b). Fig. 5.15 shows that as we lower the transmit power from -10dBm to -30dBm (by attaching attenuators), the ability to detect crossings decreases. This is because the reflected Doppler becomes weaker and submerges itself in the noise-floor. The precision and direction accuracy do not suffer because the fraction of crossings detected is low. As before, there is a power-accuracy tradeoff here. However, the accuracy gain by transmitting at -10dBm outweighs the power benefits at -15dBm and lower. Hence, *Doorpler* transmits at -10dBm.

5.4.6 Effect of Wake-up Time

We next study the effect of the wake-up time of the 5GHz chain on accuracy and timing. If the 2.4GHz radio triggers too late, then the Doppler is not significant ($\theta \approx 0^\circ$), while if it triggers too early then noisy spectrogram cells, or data from a prior crossing starts to dominate. From Fig. 5.18, we observe that the 5GHz array can be woken up as late as 750ms before a doorway crossing event. This also says that any future work that replaces the 2.4GHz radio with an alternate sensor must ensure that the 5GHz array is woken up at least 750ms before the person reaches

the doorway threshold. Since we consider an equal time window on either side of the doorway crossing point to determine direction, this result places a lower bound on the real-timeness of the system (i.e. at least 750ms after the person crosses the doorway threshold).

5.4.7 Effect of Degree of Spectrogram Overlap

The degree of spectrogram overlap is a measure of how often an FFT must be performed by the MCU. A high overlap results in a tighter real-time bound as lesser ‘fresh’ samples are required for each subsequent FFT (stream processing of Steps 1, 2, 4 to 6). From Fig. 5.19, we see that as the overlap decreases, the accuracy also starts to decrease. This is because with lower overlap, the useful signal data gets “out-weighted” by the noisy spectrogram cells. Consequently, *Doorpler* uses 90% overlap. Despite the high overlap, the MCU performs all the streaming FFT-related operations on a single batch of 64 samples from all receive chains, on average 13ms before the next set of 6 samples (90% overlap) arrives. A high overlap also results in a higher power draw as the on-time of the MCU increases. This does not impact *Doorpler* as the MCU consumes only 6mW, and the accuracy benefits of a high overlap outweigh the MCU “sleeping” benefits of low overlap.

5.5 Limitations

5.5.1 The Integrated System

We have demonstrated that *Doorpler* can perform doorway crossing detection and direction estimation using just a -10dBm tone via software defined radios. However, this prototype is both expensive and bulky. Our next step is to engineer an integrated system based on Section 5.4.2. We would then aim to incorporate it with identity

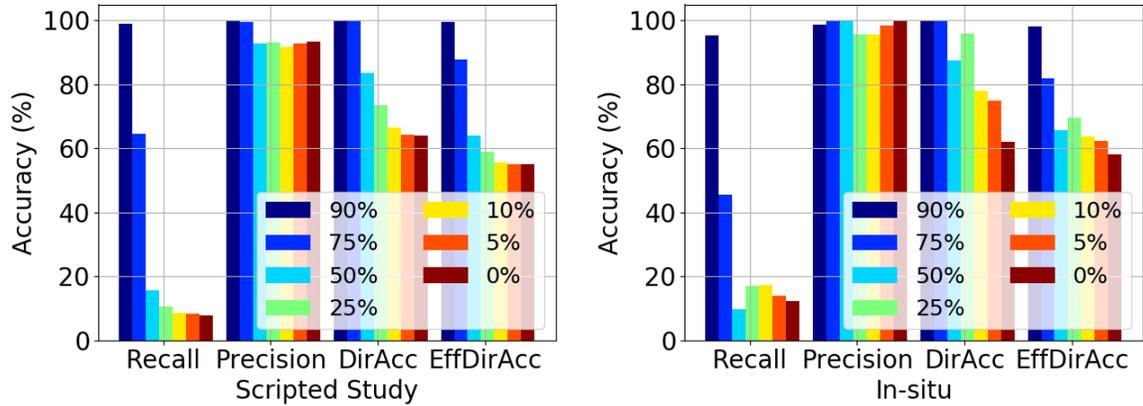


Figure 5.19: As the degree of overlap between FFT samples in the spectrogram decreases, the accuracy of *Doorpler* starts to decrease. This is because with lesser overlap, the useful Doppler data gets out-weighted by noisy spectrogram cells.

sensing from Doorjamb [25] or FormaTrack [125]. Such a system would take us closer to the vision of *plug-and-play* doorway tracking systems [4, 25].

5.5.2 Multi-Person Crossings

Doorpler has trouble when multiple walk through the doorway one behind the other (occlusion), a strong point of FORK [45]. However, this is typically not a common scenario in homes (for instance, in a prior study, the median time gap between two different individuals walking through the same doorway in an 8-room home was 10 minutes [119]).

5.5.3 Effect of Pets and Doors

Doorpler can be triggered by crossing pets as they too create Doppler. However, their gait is different from humans [146]. We leave it as a future work to differentiate pets based on gait.

While we do not explicitly consider door interactions in this work, we hypothesize that *Doorpler* can be made to handle the common case of *a person walking up to the*

door, opening/closing it and continuing motion. The more challenging scenario occurs when the door moves simultaneously in the direction opposite to human motion, resulting in simultaneous Doppler from both sides. However, this can potentially be addressed by leveraging the cyclicity of human gait. We leave it as a future work to test *Doorpler* with door movements.

5.6 Summary

In this paper, we present *Doorpler*, a low-power and real-time Doppler-based zone occupancy sensing solution that performs crossing detection and direction estimation using just an RF tone signal. *Doorpler* infers the direction of human transition by computing the angle-of-arrival of the reflection coming from the human. We evaluate *Doorpler* via a scripted study and two in-situ studies. Our results indicate that *Doorpler* can achieve over 99% and 95% accuracy across all metrics of interest in the scripted and in-situ studies, respectively. Our analyses also estimate that an analog realization of *Doorpler* would consume 6.1mW of power, falling in the indoor harvestable solar range. Our implementation of the baseband processing on an ultra low power MCU took 13.8ms, thus having the potential to enable several real-time smarthome applications like smart-lighting, HVAC control, etc.

Chapter 6

Car State Sensing via the UWB Keyless Infrastructure

Thus far we have described evidence for our claim from the perspective of one indoor human environment – namely homes. This chapter describes how context sensing is performed in another multi-path rich indoor human environment, namely automobiles by sensing the state of a car – occupied, unoccupied, door open, window open, trunk open, etc.

6.1 Introduction

The keying system in automobiles has evolved significantly from the initial usage of a mechanical key. Car manufacturers have the passive keyless entry vision [147, 148] – a car automatically unlocks itself when the person (who carries a compatible device such as a key fob or smartphone) is in its vicinity, and the car can be started only when the device is inside the car. To realize this vision, the current state-of-the-art solution uses LF-UHF (a combination of low-frequency and ultra high frequency) channels. However, these systems can be subject to relay attacks [148, 149] – as recent

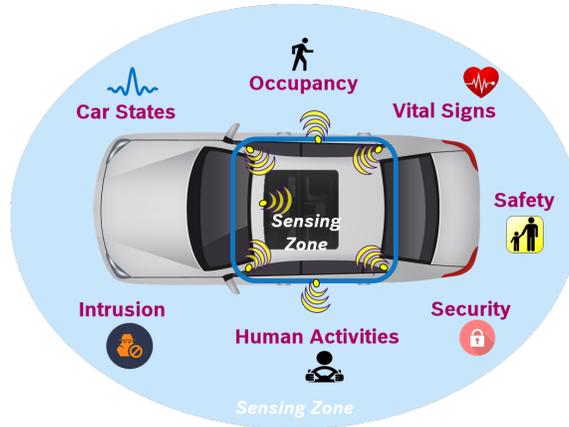


Figure 6.1: The UWB keyless infrastructure can enable several applications that can benefit from a pre-existing deployment. *CaraoKey* detects car states using such an infrastructure.

as October 2018 [150]. Consequently, car manufacturers are now developing Ultra Wideband (UWB)-based solutions for keyless entry [151–153]. These UWB systems are more robust to relay attacks as the IEEE 802.15.4-2015 UWB standard [154] explicitly incorporates timing information. In fact, the IEEE 802.15.4z Enhanced Impulse Radio Task Group is currently tasked with developing more accurate ranging methods with UWB keyless access as one of its main pilot applications [155]. Given this impending installation of UWB radios for keyless entry, we ask the question if we could multi-purpose these UWB radios – i.e. leverage them for secondary use-cases beyond keyless entry. More specifically, we explore the possibility of using the UWB keyless infrastructure as a sensing modality.

Sensing systems that leverage an already existing infrastructure (e.g. WiFi [55], acoustic [16], visible light infrastructure [18], etc.) can be used in three main ways – (i) in a standalone manner that mitigates the need for extra hardware resulting in cost, space and/or power savings, or (ii) be used in combination with other sensing systems to improve data fidelity, or (iii) to trigger a power hungry or more privacy invasive sensing system like cameras. In this chapter, we are the first to explore the UWB keyless infrastructure of automotives as a sensing modality. Such a sensing system that piggybacks on the existing UWB infrastructure can enable several

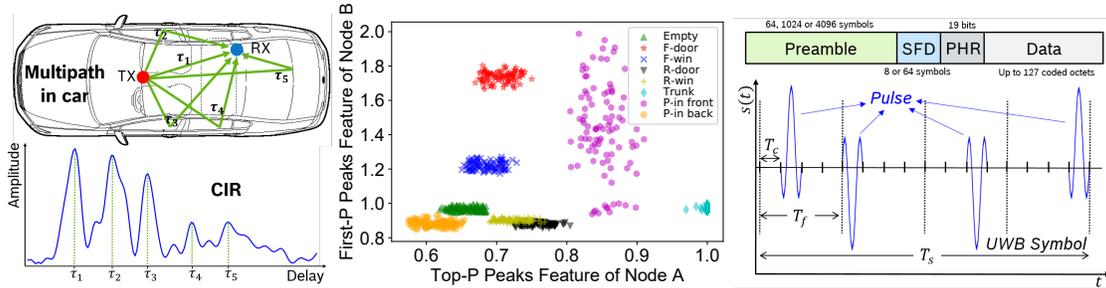


Figure 6.2: The CIR is representative of the multipath reflections inside the car. Changes in this CIR are leveraged to infer the car state.

Figure 6.3: An example of two features contained in the *multipath profile* that aid in state identification. The other features in the profile help improve the state classification.

Figure 6.4: An UWB packet consists of header and payload symbols. Each UWB symbol contains multiple narrow pulses. The preamble pulses help compute the CIR.

applications. As shown in Figure 6.1, it can potentially enable a car to detect its state (unoccupied, occupied, door open, window open, trunk open, etc.), monitor vital signs of the occupants, count the number of occupants, detect human movement near the car, detect intrusion in the car as a function of several states in succession, enable personalization, activity/gesture recognition, and so on while ensuring complete privacy. These features will become increasingly important in driverless cars and shuttles where passengers' sense of in-vehicle security and well-being will be of paramount importance [20]. In this research, we build *CaraoKey*, which explores the possibility of inferring car states using these UWB sensors. Such a car state sensing system could be used in a standalone manner for state-sensing resulting in cost benefits for manufacturers (since sensing comes for 'free' from a pre-existing hardware), or be used as a building block for other applications (for e.g. occupancy counting or child/pet presence detection can benefit from knowledge of state), or be used along with other sensing systems (e.g. a door contact sensor – a door contact sensor cannot detect an intrusion when an intruder opens the door after the door was not closed properly when the user left the car).

CaraoKey performs UWB-based sensing of car states by leveraging the channel impulse response (CIR) computed by UWB receivers. As shown in Figure 6.2, the

CIR is indicative of reflections in the environment, and changes as the state of the car changes. *CaraoKey* captures these CIR changes to identify the car state in two steps. First, it prunes the state space (identifies the most likely states) by correlating the CIRs observed by the receivers with a corpus of reference CIRs. Next, it narrows down the car state by computing a *multipath profile*, a measure of how the car is reflecting in each state as observed by the UWB nodes. Figure 6.3 shows an example of how two features contained in the *multipath profile* help separate some of the states. However, *CaraoKey* must deal with two key challenges while building the *multipath profile*. Firstly, the UWB transceivers in the car are not synchronized. Consequently, each CIR computed by a receiver will be randomly shifted with respect to previously computed CIRs from the same transmitter. *CaraoKey* addresses this challenge by identifying the first (direct) path in the CIR, and aligning the CIRs about this path, thus yielding a repeatable signature. Secondly, *CaraoKey* must be robust to changes in the location of the automobile (i.e. the same solution must work in a multipath rich indoor garage, a parking lot with cars on the sides, in free space, etc.). To address this challenge, *CaraoKey* leverages the internal UWB nodes to build the multipath profile which are more robust to location changes.

To test our hypothesis, we deploy 14 UWB nodes (placed inside and outside the car) in two different configurations of a sedan. These node locations are chosen from a superset of possible node locations [156–158]. This allows us to determine optimal number of nodes and their locations for *CaraoKey* which can further motivate manufacturers to install UWB nodes at those locations. We evaluate *CaraoKey* at 7 different physical locations – 2 indoor locations in a multipath rich garage, and 5 outdoor locations with cars, people, walls on the sides, etc. We show that using 8 UWB nodes, *CaraoKey* can distinguish between the 8 states of interest with an accuracy of 98%. In comparison, a system that is based on received signal strength (RSS) (such as Bluetooth), achieves only 49% accuracy. We then show that using

only 4 nodes, *CaraoKey* can achieve a comparable accuracy of nearly 94%, but at a lower power draw. With a measured power draw of 28.2mW at a 1Hz blink rate, *CaraoKey* consumes 20.8Wh over a 30-day period, or runs for nearly 2.5 years (885 days) on a typical 12V 50A-hour car battery.

6.2 Background

To understand *CaraoKey* better, we first provide some background on the evolution of the automobile key infrastructure and the UWB PHY Layer which *CaraoKey* leverages as a sensing modality.

Automobile Key Infrastructure: Automobile keys have evolved significantly from the traditional access control method involving physical keys. This conventional method was augmented with remote access wherein users can remotely lock or unlock their vehicles via a key press on an explicit key fob. In the last decade, automobile manufacturers have started introducing a passive keyless system [159] (i.e. no user action is required) - a convenience system which permits users to lock, unlock or start the car in a “touch-less” manner, even if their key fobs were in their pockets. These systems typically use a combination of LF and UHF channels to detect if the key fob is *Inside* or in *Close Proximity* (within 2m).

However, these passive keyless entry systems can be subject to relay attacks [148, 149], as recent as a theft in October 2018 [150]. In such an attack, the attacker positions one device in the proximity of the key (or phone), and another in the vicinity of the car. The attacker then creates a relay channel between the key and the car enabling the car to be unlocked and started [148]. In other words, the relay channel makes the car falsely believe that the user is in close proximity to it. To eliminate these attacks, car manufacturers are considering the possibility of using UWB radios around the car for keyless entry [152, 160–162]. UWB based keyless entry systems

are more robust to relay attacks as they explicitly carry timing information as part of the IEEE 802.15.4-2015 UWB standard [154] (i.e. the messages sent and received by the UWB node in the car are timestamped). As a result, a packet coming via the relay channel will be delayed as the valid key/phone signal which is being relayed is physically distant. This delay allows the car to infer that the user is not in its vicinity, thus thwarting the attack. We point out that such timestamping is now possible in WiFi with the recent IEEE 802.11-2016 [163] which standardized a Fine Time Measurement (FTM) protocol that enables a pair of WiFi cards to estimate the distance between them [164]. However, these timestamps are not as accurate as UWB systems due to the narrow bandwidth of these signals (i.e. 20 to 80MHz vs. 500MHz to 1.2GHz). This narrow bandwidth makes it difficult to determine the exact arrival time of the signal, especially in multi-path rich environments, yielding several meters of error [165]. Consequently, automobile manufacturers prefer the usage of UWB for keyless entry. Hence, in this research, we ask the question “*can we leverage these UWB nodes to sense the state of the car - viz empty, door open, trunk open, window open, person inside, etc.?*”

UWB PHY Primer: In keyless entry cars, the UWB nodes help localize the keyfob’s position when it is in the car’s vicinity. During other times, these nodes remain idle. In this idle time, *CaraoKey* uses the UWB nodes for sensing state. We next introduce some UWB terms that will help understand the approach better.

In *CaraoKey*, a transmitter (referred to as a tag) periodically beacons (blinks) an UWB message. This blink message is in the IEEE 802.15.4 UWB format [154]. As shown in Figure 6.4, an UWB Packet contains header (preamble, start of frame delimiter, PHY header) and payload symbols. Each transmitted UWB symbol (as seen in Figure 6.4) can be represented as [166, 167]:

$$s(t) = \sqrt{E_p} \sum_{j=0}^{N_f-1} b_j \omega(t - jT_f - c_j T_c) \quad (6.1)$$

where $\omega(t)$ denotes the UWB pulse of duration T_p , T_f is the duration of a frame (i.e. a symbol is divided into N_f frames), $b_j \in \{-1, +1\}$ denotes the polarity code, c_j denotes the hopping sequence - $c_j \in \{1, 2, \dots, N_h\}$ (N_h is the number of hopping slots) - i.e. the hopping code determines the location of the pulse within the N_h slots of the frame, T_c is the chip duration, and E_p represents the energy of the symbol. Figure 6.4 shows an UWB Symbol with a chipping sequence of $\{1, 0, 2, 3\}$, and four pulses with a polarity of $+1, -1, -1, +1$ respectively.

These UWB symbols travel over the air across multiple paths and reach the receiver UWB nodes (slaves). This aggregate signal received by a node from L different paths can be represented as :

$$r(t) = \sqrt{E_p} \sum_{j=0}^{N_f-1} \sum_{l=1}^L \alpha_l b_j \omega(t - \tau_l - jT_f - c_j T_c) \quad (6.2)$$

where α_l and τ_l refer to the complex attenuation and time of flight of the l^{th} path respectively.

An UWB receiver uses the perfect periodic autocorrelation property of the known preamble sequence [168] to compute the impulse response of the channel. Said differently, it runs a correlator that correlates the received signal with the known preamble sequence to compute a channel impulse response (CIR) which is given by :

$$h(t) = \sum_{l=1}^L \alpha_k \delta(t - \tau_l) \quad (6.3)$$

where $\delta(\cdot)$ refers to the Dirac delta function. *CaraoKey* uses this CIR which is indicative of the L reflected paths, to identify the car state by leveraging the intuition that the different states affect the CIR differently.

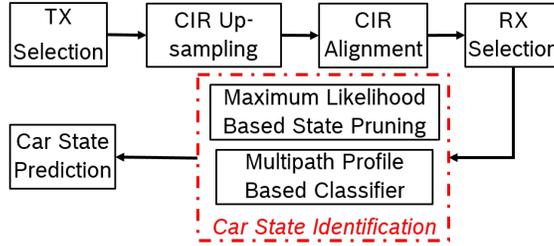


Figure 6.5: System overview of *CaraoKey*: Processing steps to convert the raw CIRs to car state.

6.3 Approach

6.3.1 Design Goals

As mentioned earlier, car manufacturers are considering the possibility of placing UWB nodes in cars for keyless entry. These nodes are leveraged by *CaraoKey* for car state identification. We build *CaraoKey* with the following design goals. **(i) Low Cost:** Cost is a major factor for car manufacturers as a unit additional sensor has a multiplicative effect on cost based on the number of produced cars. It is for this reason that cars typically do not have pressure sensors on the rear seats. Furthermore, an extra sensor comes with secondary costs such as wire harnessing, installation costs etc. Hence, *CaraoKey* is a sensorless sensing system. **(ii) Low Power:** Unlike many RF sensors deployed in buildings which have a continuous power source, *CaraoKey* must run on battery power. Consequently, we want *CaraoKey* to be low-power so as to run for months on a typical 12V 50Ah car battery. **(iii) Synchronization-free:** We leverage UWB nodes placed in and around the car to sense. However, these transceivers are not synchronized as the primary use case of keyless entry does not warrant synchronization. Consequently, we want *CaraoKey* to address the lack of synchronization from the computed CIRs (i.e. without increasing the cost or complexity of the system). **(iv) Robustness to location changes:** As cars move from place to place, we want *CaraoKey* to be robust to changes in location - i.e. the solution should work in a multipath rich indoor garage, in free-space, with

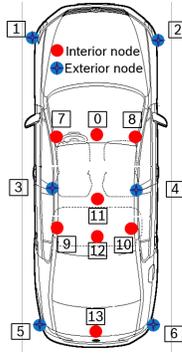


Figure 6.6: 14 nodes deployment: We over instrument a car to identify possible links.

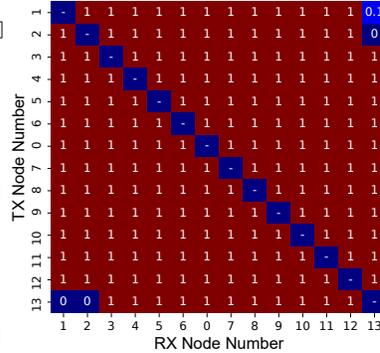


Figure 6.7: Connectivity test : Almost all the 14 nodes can deliver UWB packets to one another irrespective of their position. The links are also symmetrical with respect to packet delivery.

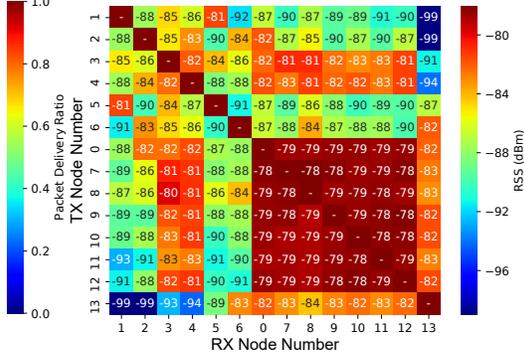


Figure 6.8: The average RSS for an interior UWB transmission is higher because of the lack of any significant attenuating object such as a car frame. The links are also mostly symmetrical.

cars on the sides, etc.

6.3.2 Overview

Figure 6.5 provides an overview of *CaraoKey's* working. As no automobile that performs UWB-based passive keyless entry exists yet in the market, *CaraoKey* over-instruments a car with 14 nodes placed in and around the car (Figure 6.6). With this over-deployment, *CaraoKey* first selects a transmitter based on connectivity (UWB packet deliver ratio), strength of the received packet and location of the nodes. Next, the resolution of each CIR computed by a receiver is increased by interpolating and upsampling in the frequency domain to aid in accurate alignment (Section 6.3.5) and feature extraction (Section 6.3.7). Thirdly, in the absence of transceiver synchronization, the CIRs computed by a receiver are randomly shifted with respect to one another. In the alignment step, each receiver aligns its CIR by pivoting its direct path at a reference tap. Next, *CaraoKey* must be robust to location changes as it is deployed in automobiles. In receiver selection, *CaraoKey* narrows down the set of potential receivers to those nodes that are more robust to location changes. We point out that the transceiver selection is a one-time event and is not repeated at

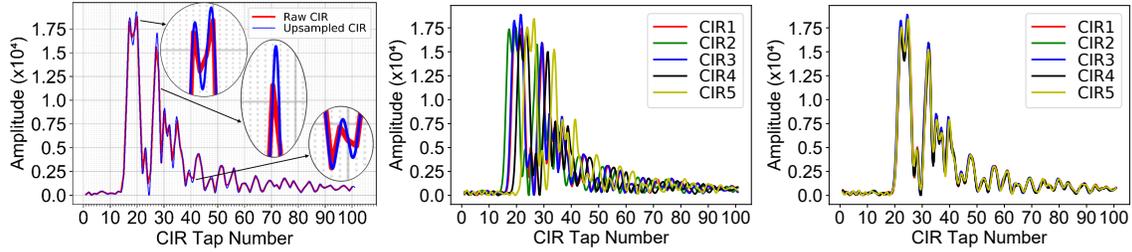


Figure 6.9: CIR upsampling: Each receiver upsamples its CIR via an FFT. This takes the discrete CIR closer to its analog waveform, aiding CIR alignment and feature extraction.

Figure 6.10: CIRs pre-alignment: The CIRs computed by a receiver are misaligned due to the absence of transceiver synchronization and receiver hardware imperfections.

Figure 6.11: CIRs post-alignment: Each receiver performs CIR alignment by identifying the tap corresponding to the first (direct) path, and shifting it to a pivot tap.

runtime. Finally, having chosen the transceivers of interest and converted the CIRs to a desired format, *CaraoKey* is ready to identify the car state. It does so in two phases. In the first phase, it identifies the K-most likely states by correlating the CIRs observed by the nodes with a reference corpus. In the second phase, features are extracted from the observed CIRs of the nodes—referred to as *multipath profile*. This *multipath profile* is used to identify the car state from the K-shortlisted states. With this overview, we provide details of *CaraoKey*'s working.

6.3.3 Connectivity Test – Transmitter Selection

There exists no automobile yet in the market that performs UWB-based passive keyless entry. Hence, in order to identify node locations to perform sensing of car states via UWB nodes, we start from first principles. In other words, we start by over-instrumenting a car with 14 nodes, as shown in Figure 6.6, and then narrow down the node positions of interest for car-state sensing. These 14 nodes are distributed both inside and outside the car as UWB nodes for keyless entry are expected to be placed both inside and outside to precisely localize the access device. In the keyless system, these nodes help the car unlock itself and set itself up by adjusting seats, mirrors, powering on the rear-view camera, HVAC, etc., depending on the “profile”

of the associated person who is in the vicinity of the car, and also start the car only when the keyfob is ascertained to be inside the car. The 14 UWB nodes are deployed as follows: four exterior bumper nodes (nodes 1, 2, 5 and 6) - i.e. two nodes on either side of the front and rear bumpers, two exterior nodes at the center top on each side (nodes 3, 4), four interior nodes on the four interior corners (nodes 7-10), a node on the rear-view mirror (node 0), a node each behind the glove compartment (node 11), on the interior light switch (i.e. center rooftop) (node 12), and inside the trunk (node 13).

As a first step, in nodes selection, we test connectivity - i.e. we ask the question, “*which nodes can communicate with which other nodes in the car?*” For this, we park the car in an indoor garage, and ask each of the 14 nodes to send 6000 blinks sequentially, and compute the blink delivery rate at each node. Figure 6.7 shows the connectivity matrix (i.e. blink delivery ratio between every pair of nodes) from our test. We observe that most nodes can communicate with one another except those in the trunk and the front bumper whose connectivity suffers because of the distance and presence of multiple signal attenuators along the path(s). From Figure 6.8, we also observe that the RSS for an interior transmission is on average much larger than a transmission from an interior (or exterior) node to an exterior (or interior) node, (or) between two exterior nodes. This is primarily because of the lack of any significant attenuating objects for an indoor transmission such as metal (car frame). We also observe *link symmetry* - i.e. given a pair of nodes N_i , and N_j , the RSS and blink delivery rates in link $L(N_i, N_j)$ is similar to the one in link $L(N_j, N_i)$.

With these observations, we set the node on the rear-view mirror (node 0) as the transmitting node (tag), and the remaining nodes as receivers (slaves). We choose node 0 as the transmitter for the following reasons : (i) it can communicate with all the nodes in the car at reasonably high power, (ii) it creates a symmetric sensing region in the car from an experimental standpoint, and (iii) every car has a rear-view

mirror.

6.3.4 CIR Upsampling

As mentioned earlier, each node that receives a blink (transmitted by node 0) will have an associated CIR. Given a Nyquist rate of 1GHz, each CIR tap is 1ns apart. We increase the resolution of this CIR by upsampling [165, 169]. This upsampling process takes us closer to the original analog waveform, and thus helps in more accurate alignment (Section 6.3.5). Figure 6.9 shows an example of upsampling. Specifically, we take a *fast Fourier transformation* (FFT) of the time-domain CIR y . Next, we zero-pad this frequency domain signal by factor of $N*(K-1)$, where N is the number of taps in the CIR and K is the upsampling factor. Finally, we obtain the upsampled CIR \hat{y} by taking the the inverse-FFT of the upsampled frequency domain signal. Said formally,

$$y_{freq} = fft(y) \tag{6.4}$$

$$y_{\hat{freq}} = [y_{freq}^{(1)}, \dots, y_{freq}^{(N/2-1)}, 0^{(1)}, \dots, 0^{(N(K-1))}, y_{freq}^{(N/2)}, \dots, y_{freq}^{(N)}] \tag{6.5}$$

$$\hat{y} = ifft(y_{\hat{freq}}) \tag{6.6}$$

6.3.5 CIR Alignment

As the transmit and receive nodes are not synchronized, the CIR frames computed by a receiver node are randomly shifted with respect to one another. Figure 6.10 shows this misalignment in 5 CIRs computed by a receiver node. While it is possible to mitigate this by synchronizing the nodes via a common reference clock, it will result in added cost and complexity as the keyless setup does not require synchronization for its primary use-case. Consequently, each node performs alignment by identifying an “event” that occurs in all CIRs independent of the environment, and then shifting the

location (tap) of that “event” to a reference pivot tap. Said differently, each CIR is shifted differently with the degree of a CIR shift depending on the tap corresponding to the arrival of the first (direct) path. As a result of this shifting, the first path of every CIR computed by a node now occurs at the pivot. We point out that this first path is not necessarily the strongest path and that the nodes in the bumpers can also observe the first path, albeit heavily attenuated. Figure 6.11 shows the aligned version of the 5 misaligned CIRs shown in Figure 6.10. The tap location corresponding to the arrival of the first path, referred to as *First Path Index* (FP_{idx}) is typically exported by UWB chips such as the Decawave DW1000 [170]. This first-path based alignment benefits from upsampling as the FP_{idx} is at a much finer resolution of 15.625ps [170] (compared to the raw CIR 1ns tap resolution). We also verified the alignment process by computing the “lag” between any two CIRs received by a node via cross correlation. We observed that the “lag” corresponds to the difference between their first path indices. Formally, we obtain the aligned CIR $\hat{y}'(t)$ where t refers to a tap as :

$$\hat{y}'(t) = \hat{y}'(t + \Delta), \quad (6.7)$$

$$\text{where } \Delta = FP_{idx} - Pivot \quad (6.8)$$

6.3.6 Receiver Selection

Having aligned the CIRs received by a node with respect to each other, we next ask the question, “*which of the remaining nodes can actually become receivers?*” To answer this, we first park the car at 4 different locations - two spots in an indoor garage, an outdoor location with cars on either side and a free-space setup (i.e. the car has nothing in its vicinity). We set node 0 to transmit and the rest to receive. In each location, we collect CIRs of different car states (empty, occupied, door, window and

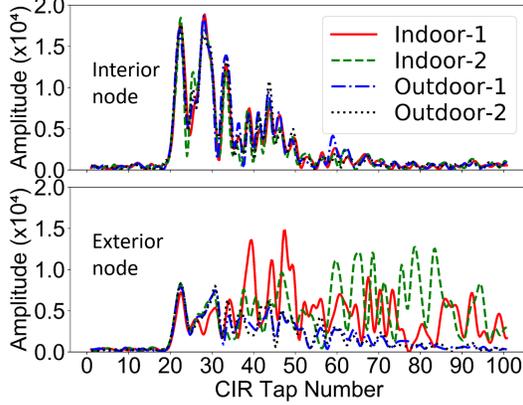


Figure 6.12: Raw CIRs for an empty car as observed by an interior and an exterior node at 4 different locations. The interior nodes are more robust to location changes than the exterior nodes.

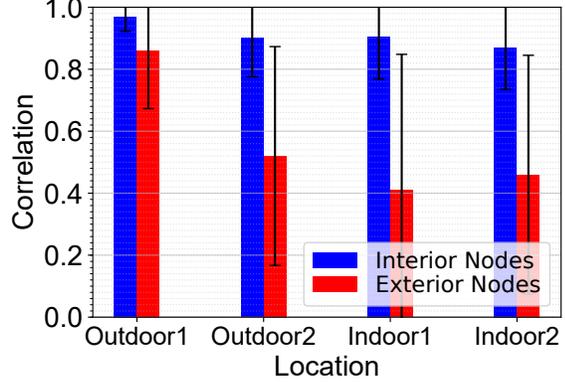


Figure 6.13: For each state, the CIRs collected in a location (outdoor1 in this example), correlate highly with CIRs collected at other locations for the interior nodes but not for the exterior nodes.

trunk open). Next, we compute the average Pearson correlation coefficient between the CIRs (of a given state) computed in free-space (called *Outdoor1*) and all the 4 locations, for each of the internal and external nodes. This correlation coefficient \hat{R} between two CIRs x and y of duration t taps is given by:

$$\hat{R}_{xy} = \frac{\sum_{i=1}^t (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^t (x_i - \bar{x})^2} \sqrt{\sum_{i=1}^t (y_i - \bar{y})^2}} \quad (6.9)$$

where x_i, y_i refer to the CIR amplitude in the i^{th} tap of CIRs x and y respectively, and \bar{x}, \bar{y} refer to the sample mean of the two CIRs.

From Figures 6.12 and 6.13, we observe that irrespective of the location, the CIR of the internal nodes correlate much higher than their external counterparts. This is because the internal nodes are more robust to location changes than the external nodes. The robustness arises primarily because the metallic car frame acts like a shield from outside reflections. When the doors/windows are open, the exterior reflections potentially incident on some nodes (depending on the angle of reflection) only affect a small portion of the CIR. Consequently, the CIRs of a state continue to correlate better with itself than any other state. Furthermore, we also look only at

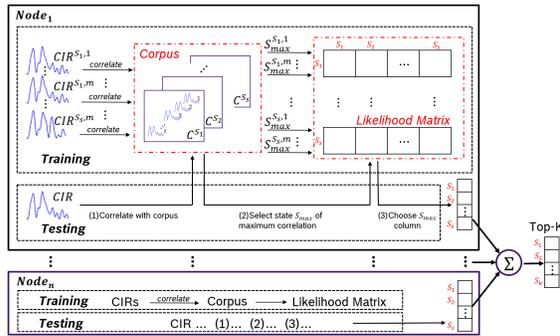


Figure 6.14: *CaraoKey* prunes the state space by choosing the Top-K most likely states via a Maximum Likelihood Approach based on CIR correlation with a reference corpus.

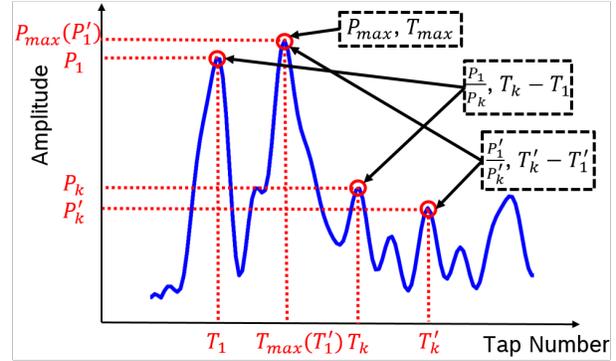


Figure 6.15: *CaraoKey* computes a *multipath profile* by extracting peak-based features from each node’s observed CIR.

a narrow CIR window of interest after the arrival of the direct path (Section 6.5.7). Consequently, given the increased robustness to location changes, we use the internal nodes as receivers.

6.3.7 State Identification

CaraoKey leverages the CIR to identify the car state. At a high level, it uses the fact that the changes in a car state (door open, window open, trunk open, person inside, etc.) alter the multipath reflections inside the car, which is observed in the CIR. For example, an open door will eliminate (or create) reflections that previously existed (or did not exist). Such reflection changes are leveraged to infer the state. *CaraoKey* performs this state inference in two steps. In Step 1, each node correlates its observed CIR with reference CIRs (in a maintained corpus) to identify the top-K likely states (Figure 6.14). Having pruned the state space, *CaraoKey* next extracts features from the CIR – referred to as the *Multipath Profile* (Figure 6.15), in order to identify the car state. We next explain how *CaraoKey* differentiates the states of interest shown in Figure 6.16.



Figure 6.16: *CaraoKey* attempts to distinguish 8 car states – empty, front and rear door open, front and rear window open, trunk open, and a person in the front or the rear of the car.

Step 1: Maximum Likelihood Based State Pruning:

CaraoKey leverages the intuition that some states can be captured better by some nodes, while other nodes observe a CIR similar to *Empty*. We capture this intuition in Step 1 via a Maximum Likelihood approach. Here, given an observed CIR (by each node) due to a transmitted blink, each node votes on a particular state based on correlation. *CaraoKey* then computes the likelihood of being in each of the possible states, given this vote. It then fuses the likelihood estimates from all the nodes, to obtain the top-K (we use $K=4$) most likely states. This is achieved in two phases : a *training phase* and a *testing phase*, as shown in Figure 6.14.

Training Phase: We explain this phase by first defining some notations. Let $\mathbf{R} = \{R_1, R_2, \dots, R_n\}$ be the set of n (receiver) nodes deployed in the car. Let $\mathbf{S} = \{S_1, S_2, \dots, S_s\}$ be the set of s car states of interest. Let $\mathbf{C}_i = \{C_i^{S_1}, C_i^{S_2}, \dots, C_i^{S_s}\}$ ($1 \leq i \leq n$) be a corpus of reference CIRs maintained for each state by a receiver R_i . Each node R_i first builds a likelihood matrix LM_i of dimensions $s \times s$. Each cell (x,y) of the likelihood matrix LM_i essentially denotes the probability of the car being in state S_x , when node R_i votes that the observed CIR is in state S_y (where $S_x, S_y \in \mathbf{S}$). We next explain how a node votes and builds the likelihood matrix.

Given a CIR of state S_x ($S_x \in \mathbf{S}$), a receiver R_i correlates this CIR with (other)

CIRs in its corpus \mathbf{C}_i , and computes the mean correlation with the reference CIRs of each state. This results in a correlation vector CV_i of dimension $s \times 1$. Node R_i then chooses the state of maximum correlation $S_{max}^x(i)$ as its vote. Formally, $S_{max}^x(i) = \text{argmax}(CV_i)$. This process is repeated for m different CIRs ($m = 50$ in *CaraoKey*) of state S_x by the node R_i , resulting in a maximum vote vector $M_i^x = [S_{max}^{x,1}(i), S_{max}^{x,2}(i), \dots, S_{max}^{x,m}(i)]$. From this vector, node R_i computes a row of the likelihood matrix which can be formally represented as $P(S_x|S_y), \forall S_y \in \mathbf{S}$, where $P(S_x|S_y) = \frac{\# \text{ of occurrences of } S_y \text{ in } M_i^x}{m}$. Anecdotally, this vector can be understood as: “the probability of being in state S_x (say Empty) when the node votes Empty, node votes front door open, node votes front window open etc. A node that can detect a particular state S_x well will have a high $LM_i(S_x, S_x)$, while a node which cannot detect a particular state well will have $LM_i(S_x, S_x)$ similar to $LM_i(\text{Empty}, S_x)$ and $LM_i(S_x, \text{Empty})$. This process is repeated for each of the s states, and for each of the n nodes, resulting in $n \times s$ likelihood matrices.

Testing Phase: In the testing phase, each node R_i correlates its observed CIR with the corpus (as in the training phase), and makes a vote based on maximum correlation ($S_{max}(i)$). From this vote, R_i obtains a likelihood vector (a column of the likelihood matrix), LV_i . Formally, $LV_i = LM_i(S_x, S_{max}(i)) \forall S_x \in \mathbf{S}$. This s -element vector essentially says : ‘when node R_i votes $S_{max}(i)$, how likely is the car to be in each of the s states. *CaraoKey* repeats the process for each of the n nodes and then fuses the likelihood vector from each node via a vector sum (i.e. the probability value of being in each state according to every node is summed). *CaraoKey* then passes the top-K most likely states for this observed CIR, on to the next step.

Step 2: Multipath Profile based State Inference:

In this step, *CaraoKey* identifies the state of the car. It does so by extracting features from the CIRs observed by the nodes – referred to as the *multipath profile*. We next



Figure 6.17: Experimental locations and scenarios: We evaluate *CaraoKey* at 7 different locations (two indoors and five outdoors), under various scenarios. Locations 1 and 2 are different indoor environments of varying multipath, in Location 3 the car is parked in free space, in Location 4 there is a person walking on the side, in Locations 5 and 6 car(s) are parked on one and both side(s) respectively, and in Location 7, the car is parked next to a concrete wall.

explain how this *multipath profile* is computed.

As mentioned earlier, the CIR is representative of how the environment impacts the transmitted signal. The peaks in the CIR represent the reflections from the environment. These peaks look different when the state of the car changes. Consequently, *CaraoKey* performs “peak-driven” feature extraction to build the *multipath profile*. These peak-based features are extracted based on position and amplitude. More precisely, as shown in Figure 6.15, each node extracts the following features from its observed CIR – (i) ratio of the power (amplitude) of the first p peaks - $(\frac{P_1}{P_2}, \frac{P_1}{P_3}, \dots, \frac{P_1}{P_p})$, where P_k refers to the k^{th} peak ordered by position, (ii) ratio of the power of the top p peaks - $(\frac{P'_1}{P'_2}, \frac{P'_1}{P'_3}, \dots, \frac{P'_1}{P'_p})$, where P'_k refers to the k^{th} peak ordered by power and $P'_1 = P_{max}$, (iii) relative (tap) distance between the first p peaks $(T_2 - T_1, T_3 - T_1, \dots, T_p - T_1)$, where T_k refers to the tap of the k^{th} peak ordered by location, (iv) relative (tap) distance between the top p peaks $(T'_p - T'_1)$, where T'_k refers to the tap of the k^{th} peak sorted by power such that $T'_1 = T_{max}$, (v) power of the maximum valued peak (P_{max}), (vi) position of the maximum valued peak (T_{max}). (*CaraoKey* uses $p = 3$). Figure 6.3 shows an example of a pair of features – $\frac{P_1}{P_2}$ and $\frac{P'_1}{P'_2}$ from two nodes helping separate many of the states. By adding the remaining features

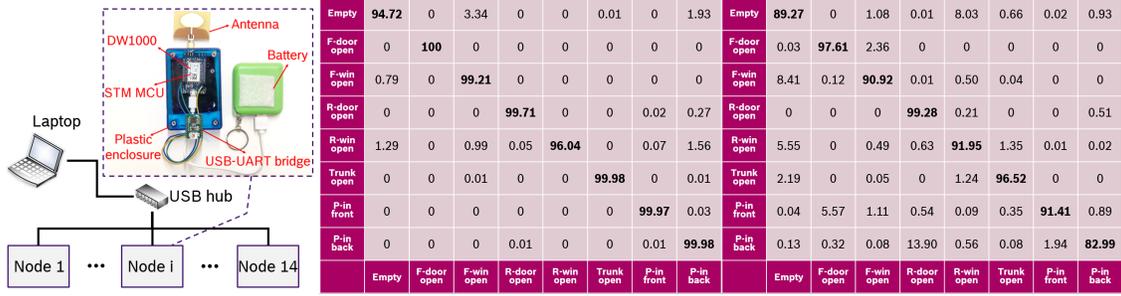


Figure 6.18: Internals of a *CaraoKey* node. Data transferred between 14 such (i.e. by training on 6 locations and a host via a USB and testing on the remaining 1 location).

Figure 6.20: Pure location independence : *CaraoKey* achieves over 92% average state classification accuracy by training on just 1 location (with cross validation).

and nodes, *CaraoKey* starts to better distinguish the states of interest. Furthermore, in *Step 1*, the correlation values obtained when the test CIR is correlated with the corpus, are reduced to a single value (max). In this step, the correlation values are also used as features. More precisely, let $c_i^1, c_i^2, \dots, c_i^s$ be the mean correlation value obtained by node R_i on correlating the test CIR with elements of the corpus $C_i^{S_1}, C_i^{S_2}, \dots, C_i^{S_s}$ respectively. As the correlation values from the different nodes are not in the same scale (i.e. the correlation values are locally ordered per node, but not globally ordered across nodes), we compute a relative correlation vector : $[c_i^1 - c_i^1, c_i^2 - c_i^1, \dots, c_i^s - c_i^1]$. Said differently, the relative correlation vector is a measure of change relative to a reference state, namely the *empty state* (i.e. c_i^1 is mean correlation with respect to the corpus of *empty* CIRs).

These features are computed by each of the n nodes, and are together referred to as *multipath profile*. The resulting 147-element feature vector is next passed through a Random Forest Classifier (100 estimators) which identifies the car state from the K shortlisted states.

6.4 Implementation and Experimental Setup

Implementation : Figure 6.18 shows an overview of our implementation. To implement *CaraoKey*, we program 14 Radino32 Spider boards [171]. Each board consists of a Decawave DW1000 transceiver chip [170] that complies with the IEEE802.15.4 UWB standard [154], and an STM32L151 microcontroller [172] to control the chip. These boards are programmed to output a 250-tap CIR at 33Hz. Each tap is represented as a 4-byte complex number from which the amplitude is computed as its magnitude. We point out that *CaraoKey* only uses 25 of the 250 taps. However, we read out 250 taps to perform sensitivity analysis (Section 6.5.7). These “truncated” CIRs are read from each Radino board via its serial port that is connected to a host laptop via a Silicon Labs CP2102N USB-UART bridge [173] and a USB 3.0 hub. These CIRs are upsampled by a factor of 4 in the host laptop. Each board is terminated by a 3.3dBi omni-directional antenna [174]. Each node is powered via USB through an 1800mAh SM3921 power bank [175], and placed inside a plastic enclosure. The boards are programmed to use Channel 5 of the UWB standard - occupying the 6.24 to 6.74 GHz frequency band. We use a 1024-length preamble with a 64MHz pulse repetition frequency. Finally, the CIRs that are read onto the host laptop are processed (Section 6.3) in Python.

Experimental Setup : To test our hypothesis, we mount the 14 nodes as previously shown in Figure 6.6, on two configurations of a Volkswagen CC. We place the instrumented cars at 7 different locations of varying multipath – two indoors (L1 and L2) (in a multipath rich garage) and five outdoors (L3-L7), as shown in Figure 6.17. The five chosen outdoor locations were all different and tested different factors. In L3 the car was parked in free space (i.e. no person or vehicle besides the instrumented car), in L4 a person was walking close (roughly 50cm) to the car, in L5 a car (SUV) was parked on one side, in L6 (different) cars (a hatchback on the left, a sedan on the

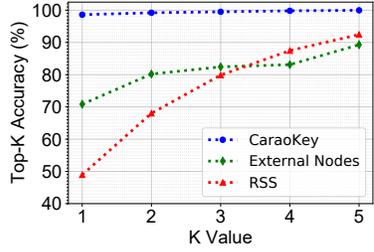


Figure 6.21: *CaraoKey* outperforms baselines that use RSS and the exterior nodes by 49% and 28% respectively. These baselines suffer from lack of multipath information and location dependence respectively.

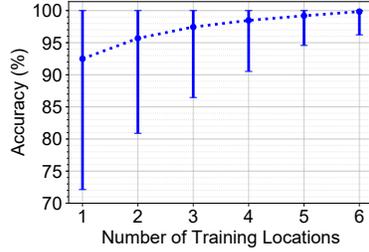


Figure 6.22: As the number of training locations increase, the accuracy increases because of the observation of a similar scenario in training. *CaraoKey* achieves 92% accuracy with just 1 training location.

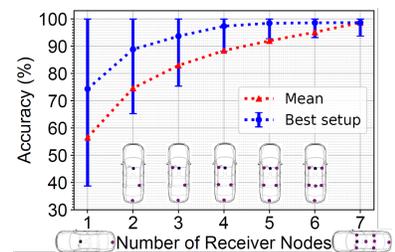


Figure 6.23: As the number of internal receivers increase, the accuracy increases. With just 3 receiver nodes (one each in the front, rear and trunk), *CaraoKey* can achieve nearly 94% accuracy.

right) were parked on both sides, and in L7 the car was parked next to a concrete wall. In each location, we collect 3 minutes of data for each state – empty, front door open, front window open, rear door open, rear window open, person in front, person in back and trunk open. The experiments for all states (except empty and trunk open) were performed on the left-side of the car alone. This is because of the symmetric nature of the setup. For the person inside experiments, two different people sat inside the car. During the study, other vehicles and people moved freely in the adjacent lanes of the parking lot (or) adjoining roads. We evaluate *CaraoKey* in terms of two metrics – *accuracy* and *power consumption*. To measure power consumption, we use the Keysight N6705B DC Power Analyzer [176]. From a power dissipation standpoint, there are no health concerns associated with *CaraoKey* as it is UWB-standards compliant. Its Effective Isotropic Radiated Power after accounting for antenna gain and cable loss is just $79\mu\text{W}$. In comparison, a typical 5GHz WiFi access point transmits at 200mW which is nearly 2500x higher. (FCC limit is 1W [177]).

6.5 Evaluation

6.5.1 System Accuracy

We first evaluate *CaraoKey*'s ability to differentiate the states of interest. In this evaluation, we first perform a leave-one-out cross validation (i.e. train on 6 locations and test on the remaining 1 location), and report the average. From Figure 6.19, we observe that *CaraoKey* has an average state classification accuracy of about 98% (across all states). Each state individually also has an average classification accuracy of nearly 95% and above. At times, the *Empty* state tends to get confused with one of the window open states (or vice-versa), but not the door open states. This is because doors (unlike windows) are typically made of metal, and hence their open/close actions affect the *multipath profile* more drastically, thus making them more distinct from the *empty* state. Figure 6.20 shows the individual state classification accuracy under pure location agnosticism. Here the location under test uses training data from just one location. Each cell in the matrix is the average state classification accuracy of all 42 possible combinations per dataset (${}^7C_1 \times 6$) of training on 1 location, and testing on one other location. Even under such conditions, *CaraoKey* achieves an average state classification accuracy of 92% (across all states). This accuracy is lower than the leave-one-out scenario because of the absence of observation of similar scenarios in training. *CaraoKey* increases the accuracy by adding more diverse training data. However, the accuracy can also be improved by increasing the number of transmitters (i.e. a round-robin set of transmitters). If N is the number of nodes in the car, then this creates $N(N-1)/2$ links that can be sensed, instead of $(N-1)$, albeit at the cost of complexity. We leave it as a future work to increase the number of transmitters.

6.5.2 Baseline Comparison

We compare *CaraoKey*'s performance with two baselines – an RSS-based baseline and a *CaraoKey* variant that uses only the exterior nodes. In this comparison, all systems employ leave-one-out cross validation.

RSS-based baseline: This baseline is representative of keyless systems that only use RSS information which are currently exported by technologies like Bluetooth [178]. Using RSS alone, the average accuracy drops down to nearly 49%. This is because *CaraoKey* uses features derived from the CIR which provides information about the multipath within the car. On the other hand, RSS is a single aggregate metric with no information about individual paths incident on a receiver, and which also does not change significantly between states (roughly 3dBm). As seen in Figure 6.21, even the top-5 classification accuracy (the fraction of blinks where the ground truth state is in the top-5 most likely states of the classifier) of an RSS-based system is lower than *CaraoKey*'s top-1 accuracy.

***CaraoKey* with exterior nodes :** As *CaraoKey* only uses internal nodes (Section 6.3.6), we compare its results with a *CaraoKey* variant that only uses the external nodes. As seen in Figure 6.21, using the exterior nodes causes the average accuracy to drop to 70%. This is because as mentioned earlier the exterior nodes are less robust to location changes than the interior nodes. Hence, at two different locations, a given state S_i looks “more similar” to another state S_j , than itself. However, the external node variant still has a higher (top-1) accuracy than the RSS-baseline as it uses the CIR which contains more details than the aggregate RSS metric.

6.5.3 Effect of Number of Training Locations

We evaluate how *CaraoKey*'s accuracy changes as we vary the number of training locations. If t is the number of training locations, we generate all ${}^7C_t \times (\gamma-t)$ scenarios

of training and testing, and compute the classification accuracy across all states, for each scenario, per dataset. From Figure 6.22, we observe that even if we train on just one location (i.e. pure location independence), we can obtain an average state classification accuracy of 92%. We also observe that as we start to increase the number of diverse training locations, the accuracy starts to increase, with over 95% accuracy by training on any two locations. With 50 training CIRs per state (Section 6.3.7), 14 states (states of interest can happen on either side of the car) and 6 training locations, the total training time for *CaraoKey* is only 127s ($=\frac{50}{33}\times 14\times 6$). We point out that such a training can be performed on only one model of a car (in the factory), and not necessarily on every shipped model of a car.

6.5.4 Effect of Number of Internal Nodes

We next study how the number of internal receiver nodes affects the accuracy of *CaraoKey*. This attempts to answer the question “*given a node budget by a car manufacturer, what is the highest achievable state classification accuracy?*” For this evaluation, we vary the number of nodes and perform a leave-one-out cross validation. Furthermore, for a given node count (N) we take the (a)symmetry properties of our experiments into account. For example, as we experiment with only the left front door open, and in the scenario where we consider only one front node, we emulate the effect of either door being open by taking the average accuracy of both the front nodes. Hence, say for $N = 1$, the highest average state classification accuracy (ASCA) is given by $\max(\text{ASCA}(\text{Node 7, Node 8}), \text{ASCA}(\text{Node 9, Node 10}), \text{ASCA}(\text{Node 11}), \text{ASCA}(\text{Node 12}), \text{ASCA}(\text{Node 13}))$. From Figure 6.23, we notice that in general the ASCA increases as we add more nodes. We also observe that with just 3 internal nodes, *CaraoKey* can achieve nearly 94% accuracy in the best arrangement of the 3 nodes. Figure 6.23 also shows the node combination which gives us the highest ASCA

for a given node count. We observe that the trunk node is the most important receiver node, if only one receiver is permitted. This is because it is central and can receive reflections from both sides of the car, and the trunk itself. Similar to Section 6.5.1, we point out that the accuracy for a given receiver node count can potentially be improved by increasing the number of transmitters, thus creating more sense-able links.

6.5.5 Power Analysis

We next measure the power consumption of *CaraoKey*. The average current consumption of a transmitter and receiver during a transmit and receive event is 80.5mA and 94mA respectively. Figure 6.24 shows the detailed breakdown of the current consumption measured via the Keysight N6705B DC Power Analyzer [176] during a transmit and receive event. We make two main observations: (i) in addition to the active current draw, the idle/listening current draw for the receiver is much larger than that of the transmitter (154mA vs. 22mA) because of the power-hungry correlator that computes the CIR, and (ii) during the actual UWB transmission (or reception) of the packet, the current draw varies. This is because different parts of the UWB packet (Figure 6.4) are modulated differently. By duty-cycling the device (sleep current draw is $2\mu\text{A}$) between the blinks, the average current consumption of the transmitter and receiver reduces to 12.6mA and 57.8mA respectively. This is seen in Figure 6.25 as power draw of 63mW and 289mW respectively (current draw * 5V), at 33Hz. We wake the receiver 10ms before scheduled wake-up time to account for lack of synchronization. When the 7 interior receivers are used along with the single tag, the total current consumption is 417.2mA ($= 12.6 + 57.8*7$). However, from Section 6.5.4, we observed that we can obtain 94% accuracy using just 3 receivers. This results in a current draw of 186mA ($= 12.6 + 57.8*3$). For a typical 12V 50Ah

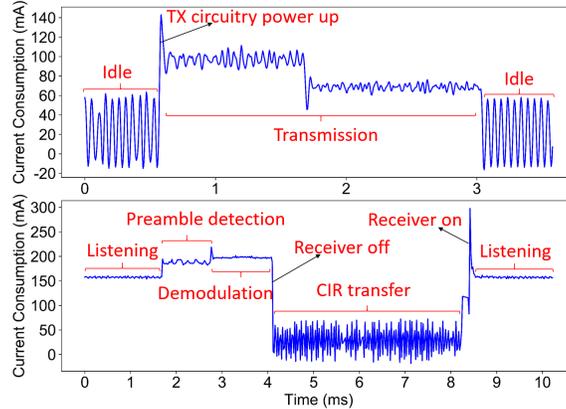


Figure 6.24: From our measurements, a *CaraoKey* transmitter draws an average of 80.5mA during transmission, while a receiver draws an average of 94mA during reception. The receiver draw is higher due to the preamble correlation for computing the CIR.

car battery, the system can be powered up for only 27 days.

However, *CaraoKey* can reduce power consumption by lowering the blink rate from 33Hz. This is called *down-sampling in slow time* [179]. The required operational blink rate of *CaraoKey* depends on the duration of the event being detected. In other words, a single blink must exist within the start and end of an event (else the event will be missed). For example, as it takes a minimum of 3.5s (empirically determined) to perform an event such as (open door, take an object from the seat and close the door), *CaraoKey* is operational so long as a blink occurs within this 3.5s event window. This permits *CaraoKey* to have an operational blink rate of 0.3Hz. Figure 6.25 shows the transceiver power draw (= current draw * 5V) and lifetime of *CaraoKey* on a 12V, 50Ah car battery, as we lower the blink rate from 33Hz to 1Hz. For a conservative blink rate of 1Hz (which is higher than the desired 0.3Hz), a 3-receiver *CaraoKey* now consumes 28.2mW of power, yielding a lifetime of 885 days on the same car battery. This is also interpretable as a power draw of 20.8Wh over a 30-day period (a metric of interest for car manufacturers).

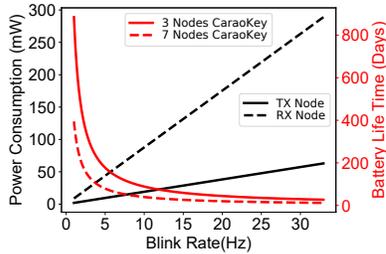


Figure 6.25: A 3-receiver, 33Hz blink rate *CaraoKey* lasts only 27 days on a 12V 50Ah battery. By lowering the blink rate to 1Hz, the lifetime increases to nearly 2.5 years.

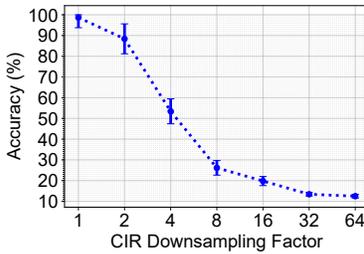


Figure 6.26: As a CIR is down-sampled in fast time, the accuracy starts to decrease. This is because the CIR peaks begin to fuse or get missed.

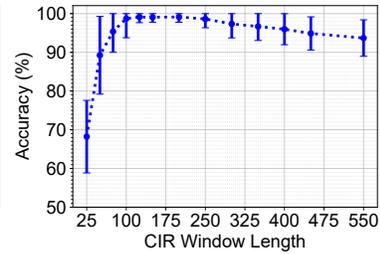


Figure 6.27: At low CIR window lengths, an insufficient number of separable peaks between states affects accuracy. At high window lengths, noise affects the correlation.

6.5.6 Effect of Tap Resolution

CaraoKey uses the DW1000 chipset which has a 1ns tap resolution. We study *CaraoKey*'s working as we lower this tap resolution. This is referred to as *fast time down-sampling* [179]. This can potentially further reduce power consumption as it places a lower stress on the ADC. We simulate a fast-time down sampling rate of N , by taking every N^{th} CIR tap. From Figure 6.26, we observe nearly 90% accuracy after downsampling by a factor of 2 (= increasing tap resolution to 2ns). However after that, we notice that the accuracy decreases rapidly as we downsample in fast time. This is because multiple peaks start to fuse as one (or are missed), and the multipath profile that is leveraged to distinguish states becomes no longer distinguishable.

6.5.7 Effect of CIR Length

CaraoKey looks at a window of 100 taps, (roughly 6m two-way distance after upsampling) to compute the multipath profile. We next study the effect of tap duration, as a smaller window places lesser stress on execution. We observe that as we start to shrink this tap window, the accuracy starts to decrease. This is because with a smaller window there are lesser peaks to build the multipath profile and the profile for the different states start to look similar to one another. Similarly as we start

to increase the window length the accuracy begins to decrease. This is because the “noisy” part in the tail of the CIR begins to dominate, and the CIRs (of different states) start to correlate with one another.

6.6 Discussion

Effect of Configuration Changes: The *multipath profile* can be affected by configuration changes - i.e. changes in the positions of the seats, mirrors, etc. However, this is typically not an issue, as most modern cars have a notion of “memory” [180]. Consequently, whenever the car is locked, the car can move its seats and mirrors back to its preset configuration. Similarly, when the car is unlocked, the car can adjust itself back to the user’s configured location.

Effect of Objects: We evaluate *CaraoKey*’s state identification in the presence of objects inside the car. With small commonly used objects like backpacks, the accuracy is not impacted. With a large moving box placed inside the car, *CaraoKey* mis-classified the *empty* car to have a person inside 40% of the time. This is addressed by fusing data from multiple CIRs and measuring the variance of a given tap (across time). Intuitively, over time, a human inside the car shows some form of macro (limbs movement, head movement) or micro (chest displacement due to breathing or heart rate) movement, while an inanimate object shows no such movement.

Multiple Transmitters: *CaraoKey* uses a single transmitter (tag). With N nodes, $(N-1)$ RF links are treated as sensors. However, if the N nodes also transmit in a round-robin manner, then $N(N-1)/2$ links can be treated as sensors. This can potentially improve the robustness of the system at a lower node count, and lower average power draw (since an UWB transmitter draws lesser current than an UWB receiver). We leave it as a future work to make *CaraoKey* multi-tag based.

Use of Dedicated Sensors: Car states can be determined via the use of ded-

icated sensors too (e.g. contact sensors for trunk and doors, glass break sensor for windows, pressure sensor for occupancy). However, besides primary and secondary costs these systems have other drawbacks – e.g. a door contact sensor cannot detect an intrusion which occurs when the door was not closed properly when the user left the car, glass break sensor cannot detect if the window is open/closed, and the occupancy sensor is only in the front seats (for cost reasons). Furthermore, from a logistical standpoint in automotives, even if door open/close information can be determined by the door contact sensor, *CaraoKey* still needs to "understand" the door state because the keyless infrastructure is independent from other sensor systems (which are typically developed by different manufacturers). This "self-compiled" knowledge of car states can be used as the building block for other sensing applications (e.g. child presence detection or vital signs monitoring) based on keyless infrastructure.

Effect of multi-state and sub-states: In the current version, we can only detect a single state. This suffices for many practical scenarios involving a parked car, where unexpected states will happen in succession and not altogether. For e.g. during an intrusion, an intruder will open/break window followed by opening a door. However, a car can potentially be in more than one state or can have partially open doors, windows too. This can be handled in two ways - (i) training for combination of states or adding sub-states, or (ii) using the probability values of the classification process to identify the multi-state or sub-states possibility. We point out that *CaraoKey* is the first system that shows the feasibility of using the keyless infrastructure of automobiles as a sensing modality via a state sensing application. However, for any practical solution, we need to maintain a balance between usability and complexity of the developed system. As a future work, we plan to work with car manufactures to identify the limited number of additional states that suffices to realize a practical solution.

Other applications Using the Infrastructure: In this paper, we introduce a novel application of car state sensing using the UWB keyless infrastructure that will be present in future cars. However, this is just one application. Numerous other applications can be enabled using this infrastructure such as vital signs monitoring, person counting (for shared car applications), emotion recognition, distracted driving detection, etc.

6.7 Summary

In this chapter, we present *CaraoKey*, a sensorless sensing system that estimates the car state via the UWB keyless infrastructure. It does so by leveraging the multipath information contained in the CIR computed by an UWB receiver. We implement a 14-node setup and evaluate it in 7 different locations and scenarios. Our results indicate that *CaraoKey* can detect the car state with 98% accuracy using 8 nodes, and 94% accuracy using just 4 nodes while drawing 28.2mW of power, at 1Hz blink rate. These numbers can potentially be further improved by using multiple RF links. *CaraoKey* is an example of using the UWB keyless infrastructure as a sensing modality. This infrastructure can also be used to build several other applications like monitoring the vital signs of the occupants, performing occupancy counting, in-car activity recognition, etc. Such features will become increasingly useful in future driverless cars, shuttles and taxis where passengers' in-vehicle security and well-being will become increasingly important.

Chapter 7

Conclusion

This dissertation introduces novel algorithms and software-hardware systems that leverage radar principles to sense the state of two *private indoor human environments* – the home and the car. The technologies presented in this dissertation provide evidence that we can sense the occupancy of rooms in a home, the corresponding identity of the occupant, and the state of a car (viz empty, occupied, door, window or trunk open, etc.) by only leveraging the RF reflections coming from the environment. All this is done in a privacy-preserving (i.e. without cameras or microphones), device-free (i.e. without placing any onus on the user to carry or wear a device), low-power and space-compliant manner in complex multipath-rich enclosed environments. From a ubiquitous sensing perspective, this dissertation introduces radars as context sensors for multipath rich indoor human environments. From a wireless networking perspective, this dissertation builds on the recent trend that wireless media which has thus far been predominantly used as a means to communicate, can also be used as a sensing medium. In this regard, it also introduces a new sensing modality in automobiles. From a biometric standpoint, this dissertation introduces a new (stronger) weak biometric namely "body shape" that can be used to separate small populations of people. From a tracking perspective, this dissertation introduces and

solves a new form of tracking called *transition tracking* which is representative of many real-world applications that warrant sensing in transition spots.

Our hypothesis was that we could sense the state of a private indoor human environment in a non invasive, device-free, low power manner with higher accuracy than state of the art, by applying radar principles and techniques. To test this hypothesis, we built *Panoptes*, a suite of solutions containing *FormaTrack*, *TransTrack*, *Doorpler* and *CaraoKey*. In Chapter 3, we explained *FormaTrack*, a radar based system for room-level tracking of occupants based on their body shape. We showed that *body shape* is a better weak biometric than state-of-the-art. We computed this body shape via a radar sensor that computed a *reflection profile*, a measure of how different parts of a person’s body reflects the transmitted RF signal. We observed that a simple radar that measures 1-dimensional reflected energy which does not at all resemble the 3D imaging output produced by sub-millimeter wave radars such as an airport scanners, is sufficient to distinguish between people in a home with a higher accuracy than existing systems. However, like any piece of hardware, *FormaTrack* can also produce sensing errors. To prevent sensing errors from becoming tracking errors, we designed *TransTrack* in Chapter 4. *TransTrack* is a multi-target tracking algorithm that deals with the challenges of only observing transitions between one state to another. *TransTrack* applies penalty functions to find the minimum number of targets that can explain the sensor data. This is because a conventional maximum likelihood solution will always overestimate the number of targets in the tracking region. In Chapter 5, we presented *Doorpler*, a Doppler-radar based system for detecting if a room is occupied or not (i.e. state sensing without identity). *Doorpler* relied on the *Doppler* reflections from the crossing human, and was designed to adhere to the time, space and power requirements of the application. Finally, in Chapter 6, we presented *CaraoKey*, a system that uses the existing UWB keyless infrastructure in cars to sense its state. It does so in a manner that is robust to location changes and does

not warrant any form of synchronization between the transceivers.

7.1 Looking Beyond

The techniques provided in this dissertation have broader applicability than the use-cases mentioned here. For example, the applicability includes :

- *Internet of "Intelligent" Things* : The Internet of Things era has resulted in connectivity provided to several everyday objects such as appliances, doors, mirrors, etc. The wireless systems that are in place to ensure connectivity can be leveraged to create "smart objects". As an example, *FormaTrack*, explained in Chapter 3 uses the *reflection profile* to identify the crossing individual. It does so by identifying the point of transition from positive to negative Doppler, with the zero crossing indicating the exact moment of the individual under the door sensor. Such a technique of computing the *reflection profile* at the opportune moment, can also be used to identify the user of appliances such as microwaves, refrigerators etc. Besides the personalization benefits, this information can help provide individual utility bills which are known to cut an individual home's bills by 20-50% [181], and thus taking us closer to the vision of a green smart home.
- *Beyond Transition Tracking in Doorways: TransTrack*, explained in Chapter 4 prevents sensing errors of *FormaTrack* from becoming tracking errors. More specifically, it uses multiple (future) *FormaTrack* observations to fix any sensing errors. Such a tracking algorithm that operates by only sensing at transition points has applicability beyond simply doorways. For example, while tracking vehicles for smart city applications, it is practically impossible to deploy sensors that cover the entire road network. Such sensors would typically be installed only at major intersections. Similarly, the operator of a shopping mall/store

for provisioning sake, might only be interested in knowing the sections of the mall (viz bakery, jewellery, toy store) frequented by an individual, but do not care about their specifics within a section. Under such circumstances, only the “section transition spots” would be instrumented. The algorithm and principles of the *hidden target* and *phantom target* find applicability in the above use-case. Akin to the *travelling salesman problem*, we anecdotally call this problem for future researchers as the *frog-lily pad problem* – a *frog* cannot be sensed while on a lily pad, but it can only be sensed while it jumps (transitions) from one lily pad to another. The goal of this problem is to form tracks of all frogs in the pond.

- *Low Power Transition Sensing : Doorpler*, presented in Chapter 5, identifies the direction of a person crossing the doorway to infer the occupancy of the adjoining room. Its principle of trading-off computational complexity for angular accuracy to be real-time, the *space-power* tradeoff handled by a dual-band wake up radio, and the ability to isolate the reflection of interest via the *Doppler Effect* can all see applicability beyond simply a doorway into a general class of applications called *transition sensing*. A *transition sensing* application essentially needs to determine when an object of interest is crossing, and its corresponding direction. Hence, for instance, the principles of *Doorpler* can be used to sense gestures (where a gesture is a sequence of transitions) in wearables, smart picture frames, smart televisions etc.
- *Safety Applications:* Chapters 3-5 describe the usage of radar principles to perform room-level occupancy sensing in a known environment. However, during firefighting or first responder operations the environment is unknown, and the goal of these first responders is to detect whether each room is occupied or not as they walk through the length and breadth of the building. *Panoptes*

can be extended to handle such environments with the first responder mounted with a radar device, and room occupancy being sensed by this device via the application of synthetic aperture radar techniques.

- *Other Applications using the Car Key(less) Infrastructure:* *CaraoKey*, presented in Chapter 6, demonstrates car state sensing using the keyless infrastructure in automobiles. This application has only scratched the surface of possibilities using the keyless infrastructure. The fact that “sensing comes for free”, given that the infrastructure exists for keyless entry results in minimal added costs – a key factor for automobile manufacturers due to the multiplicative effect of cost with the cars sold. The potential applications using the infrastructure include *person counting*, vital signs monitoring to answer questions like ‘*is the person in the car still alive?*’, child monitoring, gesture recognition for hands-free control, etc. We expect these applications to become more important with time, as driverless cars and taxis become more commonplace in the foreseeable future, thus resulting in increasing importance to passengers’ in-vehicle security and well-being.

Bibliography

- [1] W. Fox. “Introduction: ethics and the built environment”. In: *Ethics and the built environment*. Routledge, 2012, pp. 17–28.
- [2] N. E. Klepeis, W. C. Nelson, W. R. Ott, J. P. Robinson, A. M. Tsang, P. Switzer, J. V. Behar, S. C. Hern, and W. H. Engelmann. “The National Human Activity Pattern Survey (NHAPS): a resource for assessing exposure to environmental pollutants”. In: *Journal of Exposure Science and Environmental Epidemiology* 11.3 (2001), p. 231.
- [3] M. I. Skolnik. “Radar handbook”. In: (1970).
- [4] T. W. Hnat, V. Srinivasan, J. Lu, T. I. Sookoor, R. Dawson, J. Stankovic, and K. Whitehouse. “The hitchhiker’s guide to successful residential sensing deployments”. In: *Proceedings of the 9th ACM Conference on Embedded Networked Sensor Systems*. ACM. 2011, pp. 232–245.
- [5] M. Addlesee, A. Jones, F. Livesey, and F. Samaria. “The ORL active floor [sensor system]”. In: *Personal Communications, IEEE* 4.5 (1997), pp. 35–41.
- [6] D. Reid. “An algorithm for tracking multiple targets”. In: *IEEE transactions on Automatic Control* 24.6 (1979), pp. 843–854.
- [7] M. A. Richards, J. A. Scheer, W. A. Holm, et al. *Principles of modern radar*. Citeseer, 2010.

- [8] R. Rajan and C. Microsemi. “Ultra-low power short range radio transceivers”. In: *Microsemi CMPG, May* (2012).
- [9] R. J. Mailloux. *Phased array antenna handbook*. Vol. 2. Artech House Boston, 2005.
- [10] J. Ansari, D. Pankin, and P. Mähönen. “Radio-triggered wake-ups with addressing capabilities for extremely low power sensor network applications”. In: *International Journal of Wireless Information Networks* 16.3 (2009), p. 118.
- [11] M. Del Prete, D. Masotti, A. Costanzo, M. Magno, and L. Benini. “A dual-band wake-up radio for ultra-low power wireless sensor networks”. In: *Wireless Sensors and Sensor Networks (WiSNet), 2016 IEEE Topical Conference on*. IEEE. 2016, pp. 81–84.
- [12] K. Woyach, D. Puccinelli, and M. Haenggi. “Sensorless sensing in wireless networks: Implementation and measurements”. In: *Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks, 2006 4th International Symposium on*. IEEE. 2006, pp. 1–8.
- [13] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. “FM-based indoor localization”. In: *Proceedings of the 10th international conference on Mobile systems, applications, and services*. ACM. 2012, pp. 169–182.
- [14] M. Kotaru, K. Joshi, D. Bharadia, and S. Katti. “Spotfi: Decimeter level localization using wifi”. In: *ACM SIGCOMM Computer Communication Review*. Vol. 45. 4. ACM. 2015, pp. 269–282.
- [15] A. Varshavsky, E. De Lara, J. Hightower, A. LaMarca, and V. Otsason. “GSM indoor localization”. In: *Pervasive and Mobile Computing* 3.6 (2007), pp. 698–720.

- [16] R. Nandakumar, V. Iyer, D. Tan, and S. Gollakota. “Fingerio: Using active sonar for fine-grained finger tracking”. In: *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*. ACM. 2016, pp. 1515–1525.
- [17] S. Yun, Y.-C. Chen, H. Zheng, L. Qiu, and W. Mao. “Strata: Fine-Grained Acoustic-based Device-Free Tracking”. In: *Proceedings of the 15th Annual International Conference on Mobile Systems, Applications, and Services*. ACM. 2017, pp. 15–28.
- [18] Y.-S. Kuo, P. Pannuto, K.-J. Hsiao, and P. Dutta. “Luxapose: Indoor positioning with mobile phones and visible light”. In: *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM. 2014, pp. 447–458.
- [19] L. Li, P. Hu, C. Peng, G. Shen, and F. Zhao. “Epsilon: A Visible Light Based Positioning System.” In: *NSDI*. Vol. 14. 2014, pp. 331–343.
- [20] A. O. Salonen. “Passenger’s subjective traffic safety, in-vehicle security and emergency management in the driverless shuttle bus in Finland”. In: *Transport policy* 61 (2018), pp. 106–110.
- [21] S. Blackman and R. Popoli. “Design and Analysis of Modern Tracking Systems (Artech House Radar Library)”. In: *Artech house* (1999).
- [22] S. S. Blackman. “Multiple hypothesis tracking for multiple target tracking”. In: *Aerospace and Electronic Systems Magazine, IEEE* 19.1 (2004), pp. 5–18.
- [23] Y.-L. Shen and C.-S. Shin. “Distributed sensing floor for an intelligent environment”. In: *IEEE Sensors Journal* 9.12 (2009), pp. 1673–1678.
- [24] W.-H. Liau, C.-L. Wu, and L.-C. Fu. “Inhabitants tracking system in a cluttered home environment via floor load sensors”. In: *IEEE Transactions on Automation Science and Engineering* 5.1 (2008), pp. 10–20.

- [25] T. W. Hnat, E. Griffiths, R. Dawson, and K. Whitehouse. “Doorjamb: unobtrusive room-level tracking of people in homes using doorway sensors”. In: *Proceedings of the 10th ACM Conference on Embedded Network Sensor Systems*. ACM. 2012, pp. 309–322.
- [26] V. Srinivasan, J. Stankovic, and K. Whitehouse. “Using height sensors for biometric identification in multi-resident homes”. In: *International Conference on Pervasive Computing*. Springer. 2010, pp. 337–354.
- [27] S. Lee, D. Ahn, S. Lee, R. Ha, and H. Cha. “Personalized energy auditor: Estimating personal electricity usage”. In: *Pervasive Computing and Communications (PerCom), 2014 IEEE International Conference on*. IEEE. 2014, pp. 44–49.
- [28] E. Griffiths, A. Kalyanaraman, J. Ranjan, and K. Whitehouse. “An Empirical Design Space Analysis of Doorway Tracking Systems for Real World Environments”. In: *ACM Transactions on Sensor Networks (TOSN)* 13.4 (2017).
- [29] N. Khalil, D. Benhaddou, O. Gnawali, and J. Subhlok. “Nonintrusive Occupant Identification by Sensing Body Shape and Movement”. In: *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*. ACM. 2016, pp. 1–10.
- [30] T. Li, C. An, Z. Tian, A. T. Campbell, and X. Zhou. “Human sensing using visible light communication”. In: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM. 2015, pp. 331–344.
- [31] C. An, T. Li, Z. Tian, A. T. Campbell, and X. Zhou. “Visible light knows who you are”. In: *Proceedings of the 2nd International Workshop on Visible Light Communications Systems*. ACM. 2015, pp. 39–44.

- [32] W. Wang, A. X. Liu, and M. Shahzad. “Gait recognition using wifi signals”. In: *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2016, pp. 363–373.
- [33] J. L. Geisheimer, W. S. Marshall, and E. Greneker. “A continuous-wave (CW) radar for gait analysis”. In: *Signals, Systems and Computers, 2001. Conference Record of the Thirty-Fifth Asilomar Conference on*. Vol. 1. IEEE. 2001, pp. 834–838.
- [34] K. Kalgaonkar and B. Raj. “Acoustic doppler sonar for gait recognition”. In: *Advanced Video and Signal Based Surveillance, 2007. AVSS 2007. IEEE Conference on*. IEEE. 2007, pp. 27–32.
- [35] C.-Y. Hsu, Y. Liu, Z. Kabelac, R. Hristov, D. Katabi, and C. Liu. “Extracting Gait Velocity and Stride Length from Surrounding Radio Signals”. In: *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. ACM. 2017, pp. 2116–2126.
- [36] I. Bouchrika, M. Goffredo, J. Carter, and M. Nixon. “On using gait in forensic biometrics”. In: *Journal of forensic sciences* 56.4 (2011), pp. 882–889.
- [37] K. W. Bowyer, K. Hollingsworth, and P. J. Flynn. “Image understanding for iris biometrics: A survey”. In: *Computer vision and image understanding* 110.2 (2008), pp. 281–307.
- [38] Z. Liu and S. Sarkar. “Outdoor recognition at a distance by fusing gait and face”. In: *Image and Vision Computing* 25.6 (2007), pp. 817–832.
- [39] A. Kale, A. Rajagopalan, N. Cuntoor, and V. Kruger. “Gait-based recognition of humans using continuous HMMs”. In: *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE. 2002, pp. 336–341.

- [40] A. Bălan and M. Black. “The naked truth: Estimating body shape under clothing”. In: *Computer Vision–ECCV 2008* (2008), pp. 15–29.
- [41] R. T. Collins, R. Gross, and J. Shi. “Silhouette-based human identification from body shape and gait”. In: *Automatic Face and Gesture Recognition, 2002. Proceedings. Fifth IEEE International Conference on*. IEEE. 2002, pp. 366–371.
- [42] R. Appleby and R. N. Anderton. “Millimeter-wave and submillimeter-wave imaging for security and surveillance”. In: *Proceedings of the IEEE* 95.8 (2007), pp. 1683–1690.
- [43] K. B. Cooper, R. J. Dengler, N. Llombart, T. Bryllert, G. Chattopadhyay, E. Schlecht, J. Gill, C. Lee, A. Skalare, I. Mehdi, et al. “Penetrating 3-D imaging at 4-and 25-m range using a submillimeter-wave radar”. In: *IEEE Transactions on Microwave Theory and Techniques* 56.12 (2008), pp. 2771–2778.
- [44] F. Adib, C.-Y. Hsu, H. Mao, D. Katabi, and F. Durand. “Capturing the human figure through a wall”. In: *ACM Transactions on Graphics (TOG)* 34.6 (2015), p. 219.
- [45] S. Munir, R. S. Arora, C. Hesling, J. Li, J. Francis, C. Shelton, C. Martin, A. Rowe, and M. Berges. “Real-Time Fine Grained Occupancy Estimation using Depth Sensors on ARM Embedded Platforms”. In: *Real-Time and Embedded Technology and Applications Symposium (RTAS), 2017 IEEE*. IEEE. 2017, pp. 295–306.
- [46] D. Lymberopoulos, J. Liu, X. Yang, R. R. Choudhury, V. Handziski, and S. Sen. “A realistic evaluation and comparison of indoor location technologies: Experiences and lessons learned”. In: *Proceedings of the 14th international*

- conference on information processing in sensor networks*. ACM. 2015, pp. 178–189.
- [47] N. B. Priyantha, A. Chakraborty, and H. Balakrishnan. “The cricket location-support system”. In: *Proceedings of the 6th annual international conference on Mobile computing and networking*. ACM. 2000, pp. 32–43.
- [48] R. Want, A. Hopper, V. Falcao, and J. Gibbons. “The active badge location system”. In: *ACM Transactions on Information Systems (TOIS)* 10.1 (1992), pp. 91–102.
- [49] P. Bahl and V. N. Padmanabhan. “RADAR: An in-building RF-based user location and tracking system”. In: *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*. Vol. 2. Ieee. 2000, pp. 775–784.
- [50] G. Conte, M. De Marchi, A. A. Nacci, V. Rana, and D. Sciuto. “BlueSentinel: a first approach using iBeacon for an energy efficient occupancy detection system.” In: *BuildSys@ SenSys*. 2014, pp. 11–19.
- [51] S. Shi, S. Sigg, and Y. Ji. “Joint localization and activity recognition from ambient FM broadcast signals”. In: *Proceedings of the 2013 ACM conference on Pervasive and ubiquitous computing adjunct publication*. ACM. 2013, pp. 521–530.
- [52] S. N. Patel, K. N. Truong, and G. D. Abowd. “Powerline positioning: A practical sub-room-level indoor location system for domestic use”. In: *UbiComp*. Vol. 6. Springer. 2006, pp. 441–458.
- [53] V. Otsason, A. Varshavsky, A. LaMarca, and E. De Lara. “Accurate GSM indoor localization”. In: *International conference on ubiquitous computing*. Springer. 2005, pp. 141–158.

- [54] 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)*. USENIX Association. 2016, pp. 165–178.
- [55] J. Xiong and K. Jamieson. “ArrayTrack: A Fine-Grained Indoor Location System.” In: *NSDI*. 2013, pp. 71–84.
- [56] S. Kumar, S. Gil, D. Katabi, and D. Rus. “Accurate indoor localization with zero start-up cost”. In: *Proceedings of the 20th annual international conference on Mobile computing and networking*. ACM. 2014, pp. 483–494.
- [57] E. Soltanaghaei, A. Kalyanaraman, and K. Whitehouse. “Multipath triangulation: Decimeter-level wifi localization and orientation with a single unaided receiver”. In: *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*. ACM. 2018, pp. 376–388.
- [58] *VICON Motion Capture System*. <http://www.vicon.com>. May 2017.
- [59] *XSens 3D Motion Capture*. <https://www.xsens.com>. May 2017.
- [60] *Zebra Sports Solutions*. <https://www.zebra.com/us/en.html>. May 2017.
- [61] J. Xiao, Z. Zhou, Y. Yi, and L. M. Ni. “A survey on wireless indoor localization from the device perspective”. In: *ACM Computing Surveys (CSUR)* 49.2 (2016), p. 25.
- [62] 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*. ACM. 2015, pp. 837–846.

- [63] M. G. Amin, Y. D. Zhang, F. Ahmad, and K. D. Ho. “Radar signal processing for elderly fall detection: The future for in-home monitoring”. In: *IEEE Signal Processing Magazine* 33.2 (2016), pp. 71–80.
- [64] J. Lien, N. Gillian, M. E. Karagozler, P. Amihoud, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev. “Soli: Ubiquitous gesture sensing with millimeter wave radar”. In: *ACM Transactions on Graphics (TOG)* 35.4 (2016), p. 142.
- [65] S. Wang, J. Song, J. Lien, I. Poupyrev, and O. Hilliges. “Interacting with soli: Exploring fine-grained dynamic gesture recognition in the radio-frequency spectrum”. In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM. 2016, pp. 851–860.
- [66] H.-S. Yeo, G. Flamich, P. Schrempf, D. Harris-Birtill, and A. Quigley. “Radar-cat: Radar categorization for input & interaction”. In: *Proceedings of the 29th Annual Symposium on User Interface Software and Technology*. ACM. 2016, pp. 833–841.
- [67] F. Bernardo, N. Arner, and P. Batchelor. “O Soli Mio: Exploring Millimeter Wave Radar for Musical Interaction”. In: (2017).
- [68] P. J. Gething. *Radio direction finding and superresolution*. 33. IET, 1991.
- [69] A. Dersan and Y. Tanik. “Passive radar localization by time difference of arrival”. In: *Military Communications Conference (MILCOM 2002)*. IEEE. 2002, pp. 1251–1257.
- [70] S. Brown, J. Irvine, and A. Metcalfe. *Antenna characterisation for amplitude comparison in electronic warfare systems*. 2010.
- [71] *Typical Doorjamb Sizes*. <http://www.doornmore.com/help/jamb-width.html>. Aug. 2018.

- [72] R. Schmidt. “Multiple emitter location and signal parameter estimation”. In: *IEEE transactions on antennas and propagation* 34.3 (1986), pp. 276–280.
- [73] R. Roy and T. Kailath. “ESPRIT-estimation of signal parameters via rotational invariance techniques”. In: *IEEE Transactions on acoustics, speech, and signal processing* 37.7 (1989), pp. 984–995.
- [74] D. Short. “Radar Scan Strategies for the Patrick Air Force Base Weather Surveillance Radar, Model-74C, Replacement”. In: (2008).
- [75] H. Griffiths and C. Baker. “The signal and interference environment in passive bistatic radar”. In: *Information, Decision and Control, 2007. IDC’07. IEEE.* 2007, pp. 1–10.
- [76] W. Read. *Review of conventional tactical radio direction finding systems*. Tech. rep. 1989.
- [77] Q. Pu, S. Gupta, S. Gollakota, and S. Patel. “Whole-home gesture recognition using wireless signals”. In: *Proceedings of the 19th annual international conference on Mobile computing & networking*. ACM. 2013, pp. 27–38.
- [78] S. Gupta, D. Morris, S. Patel, and D. Tan. “Soundwave: using the doppler effect to sense gestures”. In: *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*. ACM. 2012, pp. 1911–1914.
- [79] T. Rahman, A. T. Adams, R. V. Ravichandran, M. Zhang, S. N. Patel, J. A. Kientz, and T. Choudhury. “Dopplesleep: A contactless unobtrusive sleep sensing system using short-range doppler radar”. In: *Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2015, pp. 39–50.

- [80] M. T. I. Aumi, S. Gupta, M. Goel, E. Larson, and S. Patel. “DopLink: using the doppler effect for multi-device interaction”. In: *Proceedings of Ubicomp 2013*. ACM. 2013, pp. 583–586.
- [81] S. Yun, Y.-C. Chen, and L. Qiu. “Turning a mobile device into a mouse in the air”. In: *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*. ACM. 2015, pp. 15–29.
- [82] *Panasonic PIR Motion Sensor Brochure*. <https://www.mouser.com/pdfdocs/PanasonicPIRSensorsBrochure.pdf/>. Aug. 2018.
- [83] E. Griffiths, S. Assana, and K. Whitehouse. “Privacy-preserving Image Processing with Binocular Thermal Cameras”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.4 (2018), p. 133.
- [84] H. Mohammadmoradi, S. Munir, O. Gnawali, and C. Shelton. “Measuring People-Flow Through Doorways using Easy-to-Install IR Array Sensors”. In: *Distributed Computing in Sensor Systems (DCOSS), 2017 International Conference on*. 2017.
- [85] G. Pulford. “Taxonomy of multiple target tracking methods”. In: *Radar, Sonar and Navigation, IEE Proceedings-*. Vol. 152. 5.
- [86] S. Oh and S. Sastry. “Tracking on a graph”. In: *Proceedings of the 4th international symposium on Information processing in sensor networks*. IEEE Press. 2005, p. 26.
- [87] D. De and et al. “Findinghumo: Real-time tracking of motion trajectories from anonymous binary sensing in smart environments”. In: *Distributed Computing Systems (ICDCS), 2012 IEEE 32nd International Conference on*. IEEE. 2012, pp. 163–172.

- [88] G. Pulford and R. Evans. “A survey of hidden Markov model tracking with emphasis on OTHR-first report to High Frequency Radar Division”. In: *University of Melbourne, Tech. Rep. CSSIP 7* (1995), p. 95.
- [89] D. H. Wilson and C. Atkeson. “Simultaneous tracking and activity recognition (STAR) using many anonymous, binary sensors”. In: *International Conference on Pervasive Computing*. Springer. 2005, pp. 62–79.
- [90] F. Kruger, M. Kasparick, T. Mundt, and T. Kirste. “Where are My Colleagues and Why? Tracking Multiple Persons in Indoor Environments”. In: *Intelligent Environments (IE), 2014 International Conference on*. IEEE. 2014, pp. 190–197.
- [91] A. Müller Sebastian Hein. “Multi-Target Data Association in Binary Sensor Networks”. In: *Proceedings of AMBIENT 2015, The Fifth International Conference on Ambient Computing, Applications, Services and Technologies*. 2015.
- [92] N. Nasir et al. “Fusing Sensors for Occupancy Sensing in Smart Buildings”. In: *Distributed Computing and Internet Technology*. Springer, 2015, pp. 73–92.
- [93] J. Sherrah, B. Ristic, and D. Kamenetsky. “A pedestrian multiple hypothesis tracker fusing head and body detections”. In: *Digital Image Computing: Techniques and Applications (DICTA), 2013 International Conference on*. IEEE. 2013, pp. 1–8.
- [94] J. H. Faghmous et al. “Multiple Hypothesis Object Tracking For Unsupervised Self-Learning: An Ocean Eddy Tracking Application.” In: *AAAI*.
- [95] T. Schmitt, M. Beetz, R. Hanek, S. Buck, et al. “Watch their moves: Applying probabilistic multiple object tracking to autonomous robot soccer”. In: *AAAI/IAAI*. 2002, pp. 599–604.

- [96] P. Melgarejo, X. Zhang, P. Ramanathan, and D. Chu. “Leveraging directional antenna capabilities for fine-grained gesture recognition”. In: *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. ACM. 2014, pp. 541–551.
- [97] J. Yang, S. Sidhom, G. Chandrasekaran, T. Vu, H. Liu, N. Cekan, Y. Chen, M. Gruteser, and R. P. Martin. “Detecting driver phone use leveraging car speakers”. In: *Proceedings of the 17th annual international conference on Mobile computing and networking*. ACM. 2011, pp. 97–108.
- [98] J. M. Rodríguez-Ascariz, L. Boquete, J. Cantos, and S. Ortega. “Automatic system for detecting driver use of mobile phones”. In: *Transportation Research Part C: Emerging Technologies* 19.4 (2011), pp. 673–681.
- [99] X. Xie, K. G. Shin, H. Yousefi, and S. He. “Wireless CSI-based head tracking in the driver seat”. In: *Proceedings of the 14th International Conference on emerging Networking EXperiments and Technologies*. ACM. 2018, pp. 112–125.
- [100] M. Raja and S. Sigg. “RFexpress! Exploiting the wireless network edge for RF-based emotion sensing”. In: *Emerging Technologies and Factory Automation (ETFA), 2017 22nd IEEE International Conference on*. IEEE. 2017, pp. 1–8.
- [101] M. Raja, V. Ghaderi, and S. Sigg. “Detecting Driver’s Distracted Behaviour from Wi-Fi”. In: *2018 IEEE 87th Vehicular Technology Conference (VTC Spring)*. IEEE. 2018, pp. 1–5.
- [102] S. Duan, T. Yu, and J. He. “WiDriver: Driver Activity Recognition System Based on WiFi CSI”. In: *International Journal of Wireless Information Networks* 25.2 (2018), pp. 146–156.

- [103] T. Ogino and T. Sato. *Car security apparatus and car security system*. US Patent 6,100,792. 2000.
- [104] *Driver vital sign detection demonstration using mmWave radar sensors*, CES 2018. <https://training.ti.com/driver-vital-sign-detection-demonstration-using-mmwave-radar-sensors>. Mar. 2019.
- [105] A. R. Diewald, J. Landwehr, D. Tatarinov, P. D. M. Cola, C. Watgen, C. Mica, M. Lu-Dac, P. Larsen, O. Gomez, and T. Goniva. “RF-based child occupation detection in the vehicle interior”. In: *2016 17th International Radar Symposium (IRS)*. IEEE. 2016, pp. 1–4.
- [106] *Caaresys*. <https://caaresys.com/>. Mar. 2019.
- [107] D. F. Elliott. *Handbook of digital signal processing: engineering applications*. Academic press, 2013.
- [108] G Trunk and S Brockett. “Range and velocity ambiguity resolution”. In: *Radar Conference, 1993., Record of the 1993 IEEE National*. IEEE. 1993, pp. 146–149.
- [109] W. G. Carrara, R. S. Goodman, and R. M. Majewski. *Spotlight synthetic aperture radar*. 2007.
- [110] A. V. Oppenheim. *Discrete-time signal processing*. Pearson Education India, 1999.
- [111] T. Oberg, A. Karsznia, and K. Oberg. “Basic gait parameters: reference data for normal subjects, 10-79 years of age.” In: *Journal of rehabilitation research and development* 30.2 (1993), p. 210.
- [112] *Salsa Ancho Kit*. <https://www.xethru.com/chips-salsa-uwbradar-development-kit.html/>. May 2017.

- [113] *Novelda Xethru X2 SoC*. <https://www.xethru.com/x2-uw-radar-chip.html/>. May 2017.
- [114] *Novelda Xethru Sensor Emissions*. <https://www.xethru.com/community/resources/xethru-sensor-emissions.53/>. July 2017.
- [115] M. Mallick, B. Vo, T. Kirubarajan, and S. Arulampalam. “Introduction to the issue on multitarget tracking”. In: *Selected Topics in Signal Processing, IEEE Journal of* 7.3 (2013), pp. 373–375.
- [116] S. Oh, S. Russell, and S. Sastry. “Markov chain Monte Carlo data association for general multiple-target tracking problems”. In: *Decision and Control, 2004. CDC. 43rd IEEE Conference on*.
- [117] D. B. Chelton, M. G. Schlax, and R. M. Samelson. “Global observations of non-linear mesoscale eddies”. In: *Progress in Oceanography* 91.2 (2011), pp. 167–216.
- [118] L. J. Cox and S. L. Hingorani. “An efficient implementation of Reid’s multiple hypothesis tracking algorithm and its evaluation for the purpose of visual tracking”. In: *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 18.2 (1996), pp. 138–150.
- [119] A. Kalyanaraman, E. Griffiths, and K. Whitehouse. “TransTrack: Tracking Multiple Targets by Sensing Their Zone Transitions”. In: *Distributed Computing in Sensor Systems (DCOSS), 2016 International Conference on*. IEEE. 2016, pp. 59–66.
- [120] A. Kalyanaraman and K. Whitehouse. “An event-based data fusion algorithm for smart cities”. In: *Adjunct Proceedings of the 2015 ACM International Joint Conference on Pervasive and Ubiquitous Computing and Proceedings of the*

- 2015 ACM International Symposium on Wearable Computers*. ACM. 2015, pp. 1575–1582.
- [121] V. Smith, T. Sookoor, and K. Whitehouse. “Modeling building thermal response to HVAC zoning”. In: *ACM SIGBED Review* 9.3 (2012), pp. 39–45.
- [122] E. Soltanaghaei and K. Whitehouse. “WalkSense: Classifying Home Occupancy States Using Walkway Sensing”. In: *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*. ACM. 2016, pp. 167–176.
- [123] *The Rise of Living Alone*. <https://bit.ly/2Nvgkpi>. Aug. 2018.
- [124] *Number of single-person households in the U.S. in 2014, by sex and age*. <https://www.statista.com/statistics/242025/number-of-single-person-households-in-the-us-by-sex-and-age/>. Aug. 2018.
- [125] A. Kalyanaraman, D. Hong, E. Soltanaghaei, and K. Whitehouse. “Forma track: tracking people based on body shape”. In: *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 1.3 (2017), p. 61.
- [126] L. Yerva, B. Campbell, A. Bansal, T. Schmid, and P. Dutta. “Grafting energy-harvesting leaves onto the sensor net tree”. In: *Proceedings of the 11th international conference on Information Processing in Sensor Networks*. ACM. 2012, pp. 197–208.
- [127] R. W. Heath. *Introduction to Wireless Digital Communication: A Signal Processing Perspective*. Prentice Hall, 2017.
- [128] T. Wei and X. Zhang. “mtrack: High-precision passive tracking using millimeter wave radios”. In: *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*. ACM. 2015, pp. 117–129.

- [129] B. Kellogg, V. Talla, and S. Gollakota. “Bringing Gesture Recognition to All Devices.” In: *NSDI*. Vol. 14. 2014, pp. 303–316.
- [130] A. Hylamia, A. Varshney, A. Soleiman, P. Papadimitratos, C. Rohner, and T. Voigt. “Towards Battery-free Radio Tomographic Imaging”. In: *Proceedings of the 11th ACM Conference on Security & Privacy in Wireless and Mobile Networks*. ACM. 2018, pp. 293–295.
- [131] Y. Ephraim and D. Malah. “Speech enhancement using a minimum-mean square error short-time spectral amplitude estimator”. In: *IEEE Transactions on acoustics, speech, and signal processing* 32.6 (1984), pp. 1109–1121.
- [132] *USRP N210*. <https://www.ettus.com/product/details/UN210-KIT>. Oct. 2018.
- [133] *SBX Daughterboard*. <https://www.ettus.com/product/details/SBX>. Oct. 2018.
- [134] *CBX Daughterboard*. <https://www.ettus.com/product/details/CBX>. Oct. 2018.
- [135] *USRP X310*. <https://www.ettus.com/product/details/X310-KIT>. Oct. 2018.
- [136] *TWINRX Daughterboard*. <https://www.ettus.com/product/details/TwinRX>. Oct. 2018.
- [137] *Octoclock-G*. <https://www.ettus.com/product/details/OctoClock-G>. Oct. 2018.
- [138] H. Meyr, M. Moeneclaey, and S. Fechtel. *Digital communication receivers: synchronization, channel estimation, and signal processing*. John Wiley & Sons, Inc., 1997.

- [139] *Electronic Code of Federal Regulations*. <https://bit.ly/2MBPqF8>. Oct. 2018.
- [140] G. Melia. “Electromagnetic absorption by the human body from 1-15 GHz”. PhD thesis. University of York, 2013.
- [141] Y.-H. Liu, X. Huang, M. Vidojkovic, A. Ba, P. Harpe, G. Dolmans, and H. de Groot. “A 1.9 nJ/b 2.4 GHz multistandard (Bluetooth Low Energy/Zigbee/IEEE802.15.6) transceiver for personal/body-area networks”. In: *Solid-State Circuits Conference Digest of Technical Papers (ISSCC), 2013 IEEE International*. IEEE. 2013, pp. 446–447.
- [142] A. Homayoun and B. Razavi. “A low-power CMOS receiver for 5 GHz WLAN”. In: *IEEE Journal of Solid-State Circuits* 50.3 (2015), pp. 630–643.
- [143] S Pellerano, S Levantino, C Samori, and A. Lacaita. “A 13.5-mW 5-GHz frequency synthesizer with dynamic-logic frequency divider”. In: *IEEE Journal of Solid-State Circuits* 39.2 (2004), pp. 378–383.
- [144] *MSP432P401R LaunchPad Development Kit*. <http://www.ti.com/tool/MSP-EXP432P401R>. Oct. 2018.
- [145] *Motion Mote*. <https://lowpowerlab.com/guide/motionmote/pir-sensor-options/>. Dec. 2019.
- [146] M. Otero. “Application of a continuous wave radar for human gait recognition”. In: *Signal Processing, Sensor Fusion, and Target Recognition XIV*. Vol. 5809. International Society for Optics and Photonics. 2005, pp. 538–549.
- [147] T. J. Waraksa, K. D. Fraley, R. E. Kiefer, D. G. Douglas, and L. H. Gilbert. *Passive keyless entry system*. US Patent 4,942,393. 1990.

- [148] A. Francillon, B. Danev, and S. Capkun. “Relay attacks on passive keyless entry and start systems in modern cars”. In: *Proceedings of the Network and Distributed System Security Symposium (NDSS)*. Eidgenössische Technische Hochschule Zürich, Department of Computer Science. 2011.
- [149] A. Humayed and B. Luo. “Cyber-physical security for smart cars: taxonomy of vulnerabilities, threats, and attacks”. In: *Proceedings of the ACM/IEEE Sixth International Conference on Cyber-Physical Systems*. ACM. 2015, pp. 252–253.
- [150] *Tesla stolen via keyfob hack*. <https://electrek.co/2018/10/21/tesla-stealing-video-keyfob-hack/>. Mar. 2019.
- [151] *Automotive News: NXP and VW share the wide possibilities of UWB fine ranging capabilities*. <https://media.nxp.com/news-releases/news-release-details/nxp-and-vw-share-wide-possibilities-ultra-widebands-uwbfine>. Oct. 2019.
- [152] *Automotive Security: Why UWB Measures Up*. <https://www.decawave.com/automotive-security-why-uwbfine-measures-up-eeacatalogue-us/>. Mar. 2019.
- [153] *FIRA Consortium : UWB Use Cases*. <https://www.firaconsortium.org/discover/use-cases>. Oct. 2019.
- [154] “IEEE Standard for Low-Rate Wireless Networks”. In: *IEEE Std 802.15.4-2015 (Revision of IEEE Std 802.15.4-2011)* (2016), pp. 1–709. DOI: 10.1109/IEEESTD.2016.7460875.
- [155] *IEEE 802.15 WPAN Task Group 4z Enhanced Impulse Radio*. <http://www.ieee802.org/15/pub/TG4z.html>. Mar. 2019.
- [156] K. Golsch and S. Sute. *Passive entry/passive start systems and methods for vehicles*. US Patent 10,328,898. 2019.

- [157] A. E. Assaad. *Angle detection for locating mobile terminals*. DE 102017214435.1. 2018.
- [158] R. Mazraani, M. Saez, L. Govoni, and D. Knobloch. “Experimental results of a combined TDOA/TOF technique for UWB based localization systems”. In: *2017 IEEE International Conference on Communications Workshops (ICC Workshops)*. IEEE. 2017, pp. 1043–1048.
- [159] *Turn-Key Passive Entry/ Passive Start Solution*. http://ww1.microchip.com/downloads/en/DeviceDoc/Article_AC10_Turn-Key-Passive-Entry.pdf. Mar. 2019.
- [160] T. R. ETSI. “103 416 V1. 1.1 (Jul. 2016)”. In: *European Telecommunications Standards Institute, France* (2016).
- [161] D. Knobloch. “Practical challenges of particle filter based UWB localization in vehicular environments”. In: *2017 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE. 2017, pp. 1–5.
- [162] T. P. Oman and K. J. Hawes. *Relay attack prevention for passive entry passive start (PEPS) vehicle security systems*. US Patent 8,930,045. 2015.
- [163] “IEEE Standard for Information technology–Telecommunications and information exchange between systems Local and metropolitan area networks–Specific requirements - Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications”. In: *IEEE Std 802.11-2016 (Revision of IEEE Std 802.11-2012)* (2016), pp. 1–3534. DOI: 10.1109/IEEESTD.2016.7786995.
- [164] M. Ibrahim, H. Liu, M. Jawahar, V. Nguyen, M. Gruteser, R. Howard, B. Yu, and F. Bai. “Verification: Accuracy Evaluation of WiFi Fine Time Measure-

- ments on an Open Platform”. In: *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*. ACM. 2018, pp. 417–427.
- [165] B. Großwindhager, C. A. Boano, M. Rath, and K. Römer. “Concurrent Ranging with Ultra-Wideband Radios: From Experimental Evidence to a Practical Solution”. In: *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*. IEEE. 2018, pp. 1460–1467.
- [166] I. Guvenc, S. Gezici, and Z. Sahinoglu. “Ultra-wideband range estimation: Theoretical limits and practical algorithms”. In: *Proc. IEEE International Conference on Ultra-Wideband (ICUWB 2008)*. Vol. 3. 2008, pp. 93–96.
- [167] Y. Kilic. *Device-free detection and localization of people using UWB networks*. University of Twente, 2016.
- [168] V. Ipatov. “Ternary sequences with ideal periodic autocorrelation properties”. In: *Radio Engineering and Electronic Physics* 24 (1979), pp. 75–79.
- [169] A. Dhekne, M. Gowda, Y. Zhao, H. Hassanieh, and R. R. Choudhury. “LiquID: A Wireless Liquid Identifier”. In: *MobiSys ’18*. ACM, 2018, pp. 442–454.
- [170] *Decawave DW1000 User Manual*. https://www.decawave.com/sites/default/files/resources/dw1000_user_manual_2.11.pdf. Feb. 2019.
- [171] *In-Circuit Radino Spider Datasheet*. https://wiki.in-circuit.de/images/8/85/305000086A_radino_Spider_RP-SMA.pdf. Feb. 2019.
- [172] *STM32L151 Datasheet*. <https://www.st.com/resource/en/datasheet/cd00277537.pdf>. Mar. 2019.
- [173] *Silicon Labs CP2102n USB-to-UART Bridge*. <https://www.silabs.com/documents/public/data-sheets/cp2102n-datasheet.pdf>. Feb. 2019.

- [174] *In-Circuit Ultra-Wideband (UWB) Antenna, 3-8GHz*. https://shop.in-circuit.de/product/_info.php?cPath=22_26&products_id=187. Mar. 2019.
- [175] *SM3921 Emergency Keychain Power Bank*. https://www.pcna.com/bullet/en-us/bullet-technology-power/product/sm-3921_reg. Mar. 2019.
- [176] *Keysight N6705B DC Power Analyzer*. <https://www.keysight.com/en/pd-1842303-pn-N6705B/dc-power-analyzer-modular-600-w-4-slots>. Mar. 2019.
- [177] *Electronic Code of Federal Regulations*. <https://bit.ly/2MBPgF8>. Mar. 2019.
- [178] *Bluetooth 5.0 Standard*. https://www.bluetooth.org/DocMan/handlers/DownloadDoc.ashx?doc_id=421043. Mar. 2019.
- [179] M. A. Richards. *Fundamentals of radar signal processing*. Tata McGraw-Hill Education, 2005.
- [180] *2018 Acura NSX Personalized Settings*. http://m.acura.com/pdf/owners/2018/NSX/2018_NSX_Personalized_Settings.pdf. Mar. 2019.
- [181] J. Ranjan, E. Griffiths, and K. Whitehouse. “Discerning electrical and water usage by individuals in homes”. In: *Proceedings of the 1st ACM Conference on Embedded Systems for Energy-Efficient Buildings*. ACM. 2014, pp. 20–29.