

ABSTRACT

Title of dissertation: RADIO ANALYTICS FOR
HUMAN COMPUTER INTERACTION

Sai Deepika Regani
Doctor of Philosophy, 2021

Dissertation directed by: Professor K. J. Ray Liu
Department of Electrical and Computer Engineering

WiFi, as we know it, is no more a mere means of communication. Recent advances in research and industry have unleashed the sensing potential of wireless signals. With the constantly expanding availability of the radio frequency spectrum for WiFi, we now envision a future where wireless communication and sensing systems co-exist and continue to facilitate human lives. Radio signals are currently being used to “sense” or monitor various human activities and vital signs. As Human-Computer Interaction (HCI) continues to form a considerable part of the daily activities, it is interesting to investigate the potential of wireless sensing in designing practical HCI applications. This dissertation aims to study and design three different HCI applications, namely, (i) In-car driver authentication, (ii) Device-free gesture recognition through-the-wall, and (iii) Handwriting tracking by leveraging the radio signals.

In the first part of this dissertation, we introduce the idea of in-car driver authentication using wireless sensing and develop a system that can recognize

drivers automatically. The proposed system can recognize humans by identifying the unique radio biometric information embedded in the wireless channel state information (CSI) through multipath propagation. However, since the environmental information is also captured in the CSI, radio biometric recognition performance may be degraded by the changing physical environment. To this end, we address the problem of “in-car changing environments” where the existing wireless sensing-based human identification system fails. We build a long-term driver radio biometric database consisting of radio biometrics of multiple people collected over two months. Machine learning (ML) models built using this database make the proposed system adaptive to new in-car environments. The performance of the in-car driver authentication system is shown to improve with extending multi-antenna and frequency diversities. Long-term experiments demonstrate the feasibility and accuracy of the proposed system. The accuracy achieved in the two-driver scenario is up to 99.13% for the best case compared to 87.7% achieved with the previous work.

In the second part, we propose *GWrite*, a device-free gesture recognition system that can work in a through-the-wall scenario. The sequence of physical perturbations induced by the hand movement influences the multipath propagation and reflects in the CSI time series corresponding to the gesture. Leveraging the statistical properties of the EM wave propagation, we derive a relationship between the similarity of CSIs within the time-series and the relative distance moved by the hand. Feature extraction modules are built on this relation to extract features characteristic of the gesture shapes. We built a prototype of *GWrite*

on commercial WiFi devices and achieved a classification accuracy of 92% on a set of 15 gesture shapes consisting of the uppercase English characters. We demonstrate that a broader set of gestures could be defined and classified using *GWrite* as opposed to the existing systems that operate over a limited gesture set.

In the final part of this dissertation, we present *mmWrite*, the first high-precision passive handwriting tracking system using a single commodity millimeter wave (mmWave) radio. Leveraging the short wavelength and large bandwidth of 60 GHz signals and the radar-like capabilities enabled by the large phased array, *mmWrite* transforms any flat region into an interactive writing surface that supports handwriting tracking at millimeter accuracy. *mmWrite* employs an end-to-end pipeline of signal processing to enhance the range and spatial resolution limited by the hardware, boost the coverage, and suppress interference from backgrounds and irrelevant objects. Experiments using a commodity 60 GHz device show that *mmWrite* can track a finger/pen with a median error of 2.8 mm close to the device and thus can reproduce handwritten characters as small as $1\text{ cm} \times 1\text{ cm}$, with a coverage of up to 8 m^2 supported. With minimal infrastructure needed, *mmWrite* promises ubiquitous handwriting tracking for new applications in HCI.

RADIO ANALYTICS FOR HUMAN COMPUTER INTERACTION

by

Sai Deepika Regani

Dissertation submitted to the Faculty of the Graduate School of the
University of Maryland, College Park in partial fulfillment
of the requirements for the degree of
Doctor of Philosophy
2021

Advisory Committee:

Professor K. J. Ray Liu, Chair/Advisor

Professor Min Wu

Professor Furong Huang

Dr. Beibei Wang

Dr. Chenshu Wu

Professor Lawrence C. Washington (Dean's representative)

© Copyright by
Sai Deepika Regani
2021

Dedication

To my family —

Hemanth Kumar Regani

Venkateshwaramma Jonnalagadda

Harika Regani

Acknowledgments

I sincerely thank my advisor, Prof. K.J. Ray Liu, for giving me an excellent opportunity to closely observe his research which eventually excited and drew me into his research group. I thank him for being flexible, supportive, and motivating at all times. I just cannot imagine a better Ph.D. advisor and his knowledge and career course will continue to inspire me.

My thanks go to my fellow lab mates in the Signals and Information Group at the University of Maryland for the enlightening discussions and knowledge-sharing sessions. I thank Dr. Qinyi Xu for supporting me during my initial years of Ph.D. and Dr. Beibei Wang for mentoring me. I thank Dr. Feng Zhang, Dr. Chen Chen, and Dr. Chenshu Wu for always being there to answer my questions. I thank my fellow students: Fengyu Wang, Yuqian Hu, Ruomin Ba, Yusen Fan, Zahid Ozturk, and Guozhen Zhu, for helping me with the experiments and for creating a motivating research environment around me with their inspiring works.

I thank my colleagues at Origin Wireless Inc., Dr. Xiaolu Zeng, Dr. Hangfang Zhang and Dr. Yi Han, for providing the hardware for my experiments. The research collaboration during the internships added immense value to this dissertation.

I thank the International Student and Scholar Services (ISSS), the graduate school, and the staff at the ECE department, who has always been friendly, liberal, and favored students in approving petitions. I will miss the student life and

the solicitude from the university, and I feel anxious to face the real world.

My heartfelt thanks to my friends here at College Park: Pallavi Chirumamilla, Harika Vakkantula, Koutilya PNVR, Dwith CYN, Shankar Reddy, Sreeharshavardhan Annu, Raghu Yaramasu, Sriram Vasudevan, Soumya Indela, Spandana Gorantla, Likhith Anvesh, Avinash Bheemineni, and Sai Sreedhar Varada for making my stay in the US the most memorable and enjoyable. Their presence has made the duration of my Ph.D. a fun-filled journey with many memories. They were a constant source of fun and recreation. I am blessed to win the friendships of such sweethearts, and I know that these will last forever.

I owe everything I have accomplished so far to my parents and sister, who always held and supported me in all circumstances. They enjoyed and cherished even the tiniest of my achievements. My gratitude to them is ineffable, and I will be indebted forever.

Table of Contents

List of Tables	viii
List of Figures	ix
1 Introduction	1
1.1 Motivation	1
1.2 Related Works	3
1.2.1 Driver Authentication	3
1.2.2 Gesture Recognition	6
1.2.3 Handwriting Tracking	9
1.3 Dissertation Outline and Contributions	12
1.3.1 In-Car Driver Authentication (Chapter 3)	12
1.3.2 Through-the-Wall Gesture Recognition (Chapter 4)	13
1.3.3 Passive Handwriting Tracking (Chapter 5)	14
2 Primer for RF and Wireless Sensing	15
2.1 2.4/5GHz WiFi	15
2.2 60 GHz WiFi	17
2.2.1 The 60GHz Radar	17
2.2.2 Target Detection	20
2.2.3 Subsample Interpolation	21
3 In-Car Driver Authentication	22
3.1 Challenges	24
3.1.1 Change of In-Car Environment	24
3.1.2 Low Resolution of Multipath	26
3.2 Dataset Preparation	28
3.2.1 Frequency Hopping on Commercial WiFi Devices	29
3.2.2 Data Collection	31
3.2.3 Data Preprocessing	33
3.3 Learning Methodologies	36
3.3.1 K-Nearest Neighbors (K-NN)	36

3.3.2	Support Vector Machine (SVM)	37
3.3.3	Neural Network	38
3.3.4	Grouping	39
3.4	Experimental Results	40
3.4.1	Two-Driver Authentication	41
3.4.2	Multiple Driver Authentication	43
3.5	Discussions	44
3.5.1	Size of the Training Set	45
3.5.2	Similarity of Radio shots	45
3.5.3	Performance with Increasing Gap between Training and Testing Data	47
3.5.4	Effect of Grouping	48
3.5.5	Effect of the Number of Links	49
3.5.6	Effect of the Number of Channels	50
3.6	Summary	51
4	Through-the-Wall Gesture Recognition	55
4.1	System Model	57
4.2	GWrite Algorithm	62
4.2.1	Preprocessing	64
4.2.2	Feature Extraction	66
4.2.2.1	Gesture Segmentation	66
4.2.2.2	Turn Angle Classification	67
4.2.2.3	Intersection Point Detection	72
4.2.3	Gesture Classification	75
4.3	Performance Evaluation	76
4.3.1	Performance of Feature Extraction	77
4.3.1.1	Gesture Segmentation Accuracy	77
4.3.1.2	Turn Angle Classification Accuracy	77
4.3.1.3	Intersection Point Detection Accuracy	78
4.3.2	Gesture Classification Accuracy	78
4.4	Discussions	79
4.4.1	Placement and Coverage	80
4.4.2	Weightage of Features (α)	81
4.4.3	Generalized Gestures	82
4.5	Summary	83
5	Handwriting Tracking	84
5.1	Handwriting Tracking Algorithm	85
5.1.1	Background Subtraction	86
5.1.2	Digital Beamforming	87
5.1.3	Doppler Velocity Extraction	89
5.1.4	Target Detection	92
5.1.5	Target Localization	97
5.1.6	Trajectory Construction	100

5.2	Experimental Evaluation	104
5.2.1	Overall Performance	105
5.2.2	Micro-Benchmarks	108
5.3	Summary	117
6	Conclusion and Future Work	118
6.1	Concluding Remarks	118
6.2	Future Work	120
6.2.1	Driver Authentication	120
6.2.2	Gesture Recognition	121
6.2.3	Handwriting Tracking	122
6.3	General Future Work	123
	Bibliography	125

List of Tables

1.1	Summary of passive handwriting tracking systems	10
1.2	Comparison between <i>mTrack</i> and <i>mmWrite</i>	10
3.1	Information about the testers	42
3.2	Classification accuracy: Comparison between learning-based and TRRS-based approaches for two-driver authentication.	43
3.3	Performance of two driver authentication	44
3.4	Similarity of radio shots and classification accuracy	46
3.5	Performance with and without grouping	50
3.6	Performance with increasing effective bandwidth using Neural network approach.	52
4.1	Theoretical values of f and r for different θ	70
4.2	Probability scores for characters with 3 segments.	76
5.1	Complexity for different modules in <i>mmWrite</i>	116

List of Figures

1.1	Evolution of the WIFI-based gesture recognition approaches. The improvements in each stage motivating the current through-the-wall robust gesture recognition are highlighted in bold text.	8
2.1	Basic radar concepts used in <i>mmWrite</i>	18
2.2	Channel Impulse Response.	18
2.3	Illustration of CA-CFAR for one dimensional signal.	20
2.4	Subsample peak interpolation.	21
3.1	Change of in-car environment with time measured in terms of TRRS. The blue curve shows the TRRS of each day with reference to day 1 and the red curve shows the moving average of the blue curve.	26
3.2	TRRS heatmap between different radio biometrics captured on different days.	27
3.3	Impact of external environment on the in-car CSI.	28
3.4	Location of transceivers in the car. Scenario (a) has transmitter near the speedometer at the back of the steering wheel and in Scenario (b), the transmitter is at the audio system.	30
3.5	Demonstration of channel hopping at the receiver with time. . . .	31
3.6	Demonstration of linear and initial phase compensation. Figure shows the phase (a) of raw CSI, (b) after linear phase compensation, (c) after initial phase compensation.	35
3.7	Average K-NN accuracy for varying value of K.	37
3.8	Average classification accuracy with number of epochs in a neural network.	37
3.9	Neural network architecture.	39
3.10	Grouping technique	40
3.11	Accuracy averaged over several pairs of people with the amount of training data.	46

3.12	Accuracy and maximum TRRS with the difference between train and test days. Red line shows accuracy with the increasing gap between the training and testing days. The blue line shows the TRRS which is the best match of the empty in-car environment from the training database.	49
3.13	Confusion matrices for different number of people.	53
3.14	Classification accuracy with increasing number of links for different sets of people.	54
4.1	Demonstration of the movement of the hand pivoted at the shoulder(<i>O</i>).	57
4.2	Demonstration of the multipath length difference when a reflector moves by a small distance x	60
4.3	TRRS decay in case of hand movement (dynamic scatterers with linearly increasing speed) and a cart movement (dynamic scatterers with one speed).	62
4.4	Overview of <i>GWrite</i> algorithm.	63
4.5	Gesture trajectories of upper case English alphabets with different number of segments. The circular dot indicates the initial point for each shape.	64
4.6	Illustration of MRC of four TRRS links, $l = 1, 2, 3, 4$. (a) Raw TRRS matrices, (b) TRRS matrices after normalizing the static offset, and (c) TRRS matrix after MRC.	65
4.7	Illustration of Gesture segmentation. (a) ‘D’shaped gesture trajectory, and (b) the corresponding motion statistics along with the different gesture segments.	68
4.8	(a) Geometry of a turn in a gesture (b) TRRS decay feature	69
4.9	Distribution of the parameter r for $\theta \approx 0^\circ$ and $\theta \approx 45^\circ$	71
4.10	(a) Illustration of the intersection point detection, (b) TRRS between CSIs of segment 1 and segment 3.	73
4.11	Histogram of T_N values with and without an intersection and the corresponding probability score assignment.	75
4.12	A typical through-the-wall experimental setup.	77
4.13	Confusion matrices for (a) angle classification and (b) intersection point detection.	78
4.14	Confusion matrices for classification of gestures with different number of segments.	79
4.15	Examples of device placement.	81
4.16	Histograms of T_N for two different device placements.	81
4.17	Classification accuracy with α	82
4.18	Examples of <i>GWrite</i> gesture shapes.	83
5.1	Illustration of handwriting tracking using the proposed <i>mmWrite</i> system. (a) Picture of the handwritten trajectory and (b) Reconstructed trajectory of the word “mmWrite”.	85
5.2	Overview of <i>mmWrite</i>	86

5.3	(a) Magnitude of the beamformed CIR, and (b) The maximum Doppler power matrix, $\mathbf{P}$, corresponding to the 8 th CIR tap at one time instance, s . The matrix $\mathbf{P}$ retains the power contributed only by the dynamic targets. This resulted in suppression of one of the two potential targets observed in (a).	89
5.4	(a) Doppler power at range tap 7 for different azimuth and elevation angles, (b) Doppler power after CFAR thresholding. In this example, the target is detected at an elevation of -8° and an azimuth of -4° with a Doppler power of 63.5 dB.	93
5.5	Estimated target locations with time using the two highest Doppler power bins. The traces are marked as T_1 and T_2 , corresponding to the first and second highest Doppler power respectively.	95
5.6	Demonstration of the Doppler power shift from non-zero frequency bin to the zero frequency bin when the target has low or zero radial velocity. Observe that when the target reaches a low radial velocity at time index of about 75, the Doppler power shifts from the non-zero frequency bin to the zero frequency bin and then switches back when the target gains some radial velocity at a time index of about 120.	96
5.7	(a) Doppler power for different range taps with time for a straight line drawn along the range away from the device, (b) Range estimates before and after SPI.	100
5.8	Illustration of the target localization and trajectory construction for the word “beam”. (a) Discrete points of the trajectory from target detection, (b) Finer location estimates obtained from SPI and (c) Trajectory after DCT-based smoothing.	101
5.9	Experimental setup for <i>mmWrite</i> . (Top) Writing on a whiteboard using a marker, (Bottom left to right) Front view of the setup showing the writing surface at nearly zero elevation, writing using a pen and hand.	104
5.10	Visual comparison of ground truth characters and recovered characters. The characters in bold are picked from EMINST dataset and serve as the ground truth (row 1, 3, and 5). The corresponding recovered characters are shown in row 2, 4, and 6.	106
5.11	Character recognition accuracy at different distances from the device with characters written in different scales.	107
5.12	Ground truth and the reconstructed trajectories of a (a) circle and (b) triangle.(c) CDF of tracking error.	109
5.13	Tracking error at different range and azimuth angles from the radar.	109
5.14	Handwriting tracking on different materials	110
5.15	Character recognition accuracy for different users.	110
5.16	Trajectory points obtained for different writing objects.	111
5.17	Demonstration of handwriting tracking for characters within 1 cm.	112
5.18	Demonstration of <i>mmWrite</i> precision	113

5.19 Recovered handwriting trajectories with extreme speeds of hand-writing.	114
5.20 Recovered handwriting trajectories with different w_s	115
5.21 Smoothed trajectories for different smoothing parameter values. .	115

List of Abbreviations

ACF	Autocorrelation Function
AoA	Angle of Arrival
AP	Access Point
CDF	Cumulative Distribution Function
CFO	Channel Frequency Offset
CIR	Channel Impulse Response
CSI	Channel State Information
EM	Electromagnetic
FMCW	Frequency Modulated Continuous Wave
FFT	Fast Fourier Transform
GHz	Gigahertz
IoT	Internet of Things
LOS	Line of Sight
MIMO	Multiple Input Multiple Output
MHz	Megahertz
NLOS	Non Line of Sight
PCA	Principal Component Analysis
RF	Radio Frequency
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
Rx	Receiver
SFO	Sampling Frequency Offset
SNR	Signal-toNoise Ratio
STO	Symbol Timing Offset
ToA	Time of Arrival
ToF	Time of Flight
TRRS	Time Reversal Resonance Strength
TR	Time-Reversal

Tx Transmitter

UWB Ultra-Wide Band

Chapter 1: Introduction

1.1 Motivation

The rapid increase in automation has stimulated the quest for more efficient and convenient Human-Computer Interaction (HCI) approaches. Many of our interactions with computers have been shifting from conventional computer keyboard and mouse to hand gestures and writing in the air. The increasing trend of ubiquitous interactions with machines via in-the-air gestures, handwriting, and voice-controllable systems is also evidenced by recent industrial efforts, including Google Soli [39], Apple UWB radar, Apple Siri, Amazon Alexa, Google Home, and many more.

Wireless signals, particularly WiFi signals, have been exploited to achieve pervasive device-free sensing for various applications [40, 42, 43, 45, 71], such as motion detection, vital sign monitoring, human identification, gait recognition, indoor tracking, etc. [41, 58, 77, 79, 91]. Such WiFi-based systems have been incredibly successful and widely accepted because of four main advantages:- (i) ability to sense information beyond a wall, (ii) preservation of user privacy, (iii) enhanced user experience while sensing without any hand-held sensors, and (iv) reuse of existing infrastructure. While the ever-increasing availability of the RF

spectrum for WiFi continues to push wireless sensing capabilities, the accessibility of compact radars and WiFi APs makes these technologies widely usable.

Motivated by the inherent capabilities of WiFi signals/wireless sensing and the significance of HCI, we design three HCI applications that are often encountered - *driver authentication*, *handwriting tracking*, and *gesture writing recognition*.

1. *Driver authentication* identifies authenticated users for a car. Identifying drivers can automatically enable driver-specific adjustments such as the seat/mirror positions, potentially replace the necessity of a key and prevent thefts. As automobiles continue to play an important role, manufacturers have been seeking and deploying emerging technologies to support drivers, such as driver vital sign monitoring, child presence detection, and seat occupancy detection. Driver authentication forms an essential part of such future of “smart car” technologies [86]. Furthermore, the ability to perform human identification with radio signals can replace all the authentication systems in the future that currently rely on keys or passwords.
2. *Gesture recognition* through-the-wall enables in-the-air interaction with computers/machines and smoothly blends with the future smart environments. It can serve as an alternate technology to voice-operated systems and particularly useful for the elderly and disabled. Interaction via gestures is an intuitive way to communicate with computers and automation systems. With in-the-air gestures, one can operate public machines/systems without touching them, avoiding the necessity to sanitize such frequently touched

surfaces. In a home or an office environment, gestures can be used to operate air-conditioning systems, switches, turn on/off machines remotely, and even communicate sensitive information through-the-wall.

3. *Handwriting tracking* can trace the movement of a hand during writing and reconstruct the handwriting trajectory. Handwriting is a convenient mode of interaction for people and can be considered a more general form of a gesture, which is a promising approach for HCI. Aided by advanced handwriting recognition systems, robust and accurate handwriting tracking can realize countless HCI applications. Handwriting tracking enables smart environments in which any flat region can be a writing surface, and the space for HCI can be extended beyond the touch screens.

1.2 Related Works

1.2.1 Driver Authentication

Current smart vehicle technologies can predict traffic patterns, automate driving and optimize fuel consumption [34]. Further, there are many other works on driver monitoring and activity recognition. The authors in [32,51] used vision-based techniques to detect head, eye, and hand movements to predict driver behavior for accident prevention, although cameras, in general, introduce privacy concerns.

On the other hand, wireless sensing is an innovative approach to achieve

security and privacy at the same time and has been widely used in many IoT applications due to the ubiquitous deployment of WiFi devices. High accuracy indoor localization has been achieved using WiFi fingerprinting [13, 46, 83]. By extracting statistics and identifying other features in the channel state information (CSI), multiple research teams investigated the detection of indoor motion or dynamics [35, 48, 85]. Recently, researchers have been working on using wireless sensing to enable indoor vital sign estimation by extracting the periodic components in the CSI [15, 41]. Other applications such as keystroke recognition [7], movement speed estimation [90], gesture [52] and gait recognition [75], have also been explored through wireless sensing. In the study on smart vehicles using wireless sensing, researchers have investigated driver activity recognition where driving actions have been estimated using received signal strength indicator (RSSI) and CSI amplitudes in a simulated environment [20]. However, not much research has been done on driver authentication based on wireless sensing.

In this work, we use a new type of physical biometrics, radio biometrics, to achieve reliable in-car driver authentication [84]. Radio biometrics is the pattern of a human body introduced into the wireless propagation environment. Researchers have studied the EM wave propagation through a human body [19, 22] and the dielectric properties [23–25]. Alongside these studies, the work in [47] showed that the EM propagation in and around a human body is influenced by height, weight, body water volume, surface area, tissue density, and more. A combination of these features could be potentially unique to a person and serve as a biometric [84]. In an indoor environment, the human radio biometrics can be

recorded within the CSI and determined by each individual’s unique biological characteristics. The captured CSI can be considered a radio signature, and the process of recording a radio signature is termed as a *radio shot*.

The first wireless sensing-based human recognition system [84] used CSI as the feature for individual physical characteristics and utilized the time-reversal technique to compare the similarity of radio biometrics. However, this prior art assumed that the indoor environment remains static throughout the experiment. In reality, this is not the case, and even a small change in the indoor environment will introduce a significant change into the multipath CSI. Different from the existing work, we do not make any such assumptions. Instead, we build machine learning (ML) models that can adapt to the changing in-car environment.

ML techniques require data to learn patterns, draw inferences and generalize to new unseen references [12]. They perform an automated feature selection from the training data. ML and deep learning have achieved great success in computer vision because they have large amounts of data to train on, while the availability of training data essentially limits the performance of deep learning in other fields. To this end, we built the first radio biometric database consisting of radio biometrics of seven people collected over two months and generate ML models.

Previous works have shown that radio propagation pattern through a human body is frequency-dependent [23]. On this basis, we obtain radio biometrics from several channels using frequency hopping. Frequency hopping rapidly switches channels to transmit wireless signals. For example, [70] used frequency

hopping to achieve sub-nano level Time of Flight (ToF) and [13] achieved high accuracy indoor localization by using augmented CSI fingerprint from several channels. We implement frequency hopping on portable commercial WiFi devices that can be fixed in a car. Furthermore, we exploit the multi-antenna diversity in the MIMO systems to obtain more differentiating features.

1.2.2 Gesture Recognition

Initial efforts to achieve gesture recognition used the Received Signal Strength Indicator (RSSI) to realize different applications including gesture recognition, target tracking and breathing estimation [2, 3, 50, 87]. However, RSSI being a single scalar could only provide coarse information about the activities leading to the channel perturbations and was also sensitive to the environmental changes. With the release of the Channel State Information (CSI) tool for commercial WiFi devices [29], fine-grained information on the channel measurement became available, opening up many opportunities to achieve new applications and improve the systems which, till then, depended on RSSI.

When the physical environment is relatively unchanged, a given movement/activity results in a repeatable CSI time series. Thus, many works maintained databases of CSI patterns in the training phase and compared them with the CSIs in the testing phase using a similarity metric such as Dynamic Time Warping (DTW). Although such an approach should work in theory, it is practically impossible to build a training database of CSI prototypes spanning the

wide range of possible locations, users, and environments. This approach also restricted the application to a limited set of pre-defined gestures whose CSI patterns are available. Few works tried to overcome this problem by post-processing the CSI using different filtering and feature selection techniques such as PCA, subcarrier selection, wavelet transforms, low-pass filtering, etc. [6, 7, 31, 37, 64]. However, these attempts were not successful in obtaining location-independent features that are characteristic of the movement pattern.

To completely suppress the location information, the later works started utilizing derived features such as the direction of the gesture [1] and number of repetitions [56]. Such approaches also worked for the Non Line-of-Sight (NLOS) and through-the-wall setups, thereby relaxing constraints on the placement of devices. However, with such features, the number of gestures that can be recognized is limited. Given the number of smart devices that surround us today, a broader set of gestures is necessary to interact with them and add more flexibility to the user. The intuitive way forward to achieve a more comprehensive set of recognizable gestures is to design features that describe the original trajectory of the hand motion or, in other words, that can reconstruct it.

The relative location of the hand with time has to be known to reconstruct the gesture trajectory. For this, information such as Time-of-Flight to determine the distance/range of the hand from the receiver, Angle-of-Arrival (AOA) [62, 67] to determine the angular location, and Doppler shift to determine the radial velocity of the hand [52] were used. However, all these approaches either require modified hardware or a strict LOS setup with a small operational range. *WiDraw*

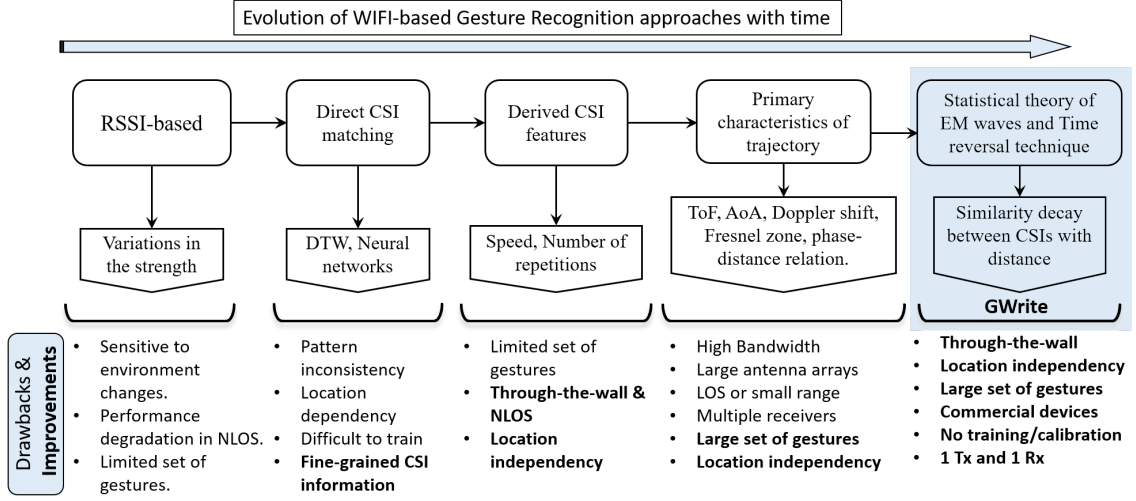


Figure 1.1: Evolution of the WIFI-based gesture recognition approaches. The improvements in each stage motivating the current through-the-wall robust gesture recognition are highlighted in bold text.

relied on the signal strength from multiple APs to determine the location of the hand at different time instances and construct the trajectory [62]. A centimeter accuracy finger tracking was achieved using two receivers placed orthogonally in FingerDraw [82]. These systems are effective only up to a small range (<1 m). As the distance between gesture and the transceiver increases, it becomes increasingly difficult to isolate the signal reflected off the hand.

With *GWrite*, we propose and design a through-the-wall gesture recognition system that can extract features describing the trajectory of the gestures, facilitating a large set of recognizable gestures using COTS WiFi. The statistical theory of EM waves, combined with the unique gesture model, enabled us to extract features that can reconstruct the gesture shape. A summary of the gesture recognition approaches using WiFi is provided in Fig 1.1.

1.2.3 Handwriting Tracking

Handwriting tracking can be achieved in a device-based [4, 74] or device-free manner [30, 60, 78]. Considering user-convenience, device-free or passive approaches have gained more attention over the years. Passive handwriting tracking has been attempted by researches in the past using different approaches such as acoustic, vision, WiFi, RFID, and millimeter waves. A summary of the past works using different approaches is presented in Table. 1.1. Acoustic signal-based approaches use the inbuilt speaker and microphone of the smartphones to derive the sound signal reflected by the moving object. Although these approaches could track handwriting at millimeter-scale accuracy, the performance deteriorates severely with distance, and the operational range is usually within 50 cm from the device [49, 76, 88]. Vision-based methods use a camera to identify and track the fingertip/arm to reconstruct the handwriting trajectory [33, 60]. Intuitively, the tracking accuracy of the vision-based approaches depends on the pixel resolution and the distance from the camera. These approaches are limited by privacy interests and the requirement for ambient light. On the other hand, WiFi-based approaches estimated the displacement using Fresnel zone model [82] or by tracking the phase of the CSI [30]. The transceiver geometry is then exploited to obtain the 2D location of the target. WiDraw [62] uses the Angle of arrival (AoA) of the line-of-sight signals from the densely deployed transceivers to track the motion of the hand in the air. Although WiFi based approaches appear attractive in terms of coverage, the tracking accuracy is of

Table 1.1: Summary of passive handwriting tracking systems

Approach	Reference	Method	Hardware	Coverage	Tracking Accuracy
Acoustic	Strata [88]	Phase tracking	Smart phone	40 cm	3 mm
	LLAP [76]	Phase tracking of sound signals	Smart phone	30 cm	4.6 mm
	FingerIO [49]	Echo profile and OFDM	Smart phone	0.25 m ²	8 mm
Vision	[60]	HMM and multi camera 3D tracking	Camera/Kinect	N/A	N/A
	[33]	Fingertip detection and tracking	USB PC Camera	50 cm [★]	N/A
WiFi	FingerDraw [82]	Fresnel zone, triangulation	3 transceivers	100 cm	1.27 cm
	AirDraw [30]	Phase tracking and triangulation	3 transceivers	1.5 m	2.2 cm
	WiDraw [62]	Analysis of RSSI with direction	30 transceivers	2 feet	5 cm
RFID	RF-Finger [72]	Reflection features from tag array	RFID tags and readers	1 m [★]	~ 2 cm
mmWave	mTrack [78]	Phase tracking and triangulation	3 transceivers	1 m ²	7 mm
	<i>mmWrite</i>	Target detection and localization	One 60 GHz radar	8 m²	2.8 mm

★ Approximate values based on the experimental setup.

Table 1.2: Comparison between *mTrack* and *mmWrite*

Reference	Hardware	Method	Absorbers	Calibration	Coverage	Error at 20 cm
<i>mTrack</i> [78]	1 Tx, 2 Directional horn antenna	Phase tracking & triangulation	Required	Required	1 m ²	7 mm
mmWrite	Phased antenna array	Target localization	Not required	Not required	~ 8 m ²	2.8 mm

the order of centimeters or decimeters, which is not adequate for fine-grained handwriting tracking [30, 62]. Compared to the above modalities, millimeter wave-based methods seem to be an attractive option both in terms of coverage and tracking accuracy [78] [57].

Among the millimeter wave based approaches, *mmWrite* is the first work to reuse a commodity millimeter wave radio as a radar for handwriting tracking. The most relevant past work is *mTrack*, which uses a custom-built 60 GHz software-radio platform (consisting three transceivers) and tracks the writing

object by using the phase of the received signal. While *mTrack* relies on phase changes and triangulation, we leverage more radar capabilities in target localization and Doppler speed estimation, as compared in Table. 1.2. It can be observed that *mmWrite* can achieve lesser tracking error and larger coverage than *mTrack*, while requiring minimal infrastructure support.

Handwriting recognition The field of handwriting tracking is closely linked to that of handwriting recognition. While the end goal of handwriting tracking is to reconstruct the trajectory, handwriting recognition further intends to predict/identify the character. Handwriting recognition can be achieved in two different approaches. The first set of approaches recover the handwriting trajectory as an intermediate step and identify the characters using handwriting recognition modules/classifiers [44, 62]. The second set of approaches extract features representative of the characters and use learning-based techniques to classify them. These approaches suffer from the pattern inconsistency problem, i.e., the extracted features are location and environment-dependent, limiting the practical usage and the allowed set of characters [16, 28]. The current research, therefore, is focused on recovering the handwriting trajectory itself allowing unrestricted device placement and the set of trajectories.

Gesture recognition and passive motion tracking Wireless signals like WiFi and mmWave have also been widely explored for gesture recognition and passive motion tracking. These works either set up multiple transceivers with a specific geometric arrangement for precise gesture tracking [30, 82], or do not need special setup but only allow gesture recognition from a predefined set of

gestures [3, 52, 67]. The latest research resorts to deep learning to improve cross-domain recognition accuracy, but does not fundamentally improve motion tracking resolution [92, 93]. These works, however, could not achieve the required millimeter accuracy for handwriting tracking due to the intense multipath that corrupts the signal reflected from the hand.

Indoor passive motion tracking is also achieved with 5 GHz WiFi. For this, parameters such as Angle-of-Arrival, Doppler velocity and Time-of-flight [38, 53, 54] are estimated from the CSI measurements. Millimeter wave has been used to track one or more people/robots by exploiting the target tracking capabilities of the radar and the phased antenna array structure [80], [27]. However, these works aim at tracking the human/robot body as a whole and the tracking accuracy is not sufficient for the handwriting application.

1.3 Dissertation Outline and Contributions

1.3.1 In-Car Driver Authentication (Chapter 3)

In this Chapter, we propose the first in-car driver authentication system using the human radio biometrics recorded in the wireless CSI. To address the problem of in-car environmental changes, we build a multiple-driver radio biometric database consisting of radio biometrics of seven people collected over a period of two months. To our knowledge, this is the first long-term study conducted for human radio biometric recognition. With the help of this database, we integrate ML techniques to make the proposed driver authentication sys-

tem adaptive to different in-car environments. Later, we study the impact of the multi-antenna diversity and the frequency diversity on the accuracy of the proposed driver authentication system. For experimental evaluation, we have implemented frequency hopping on portable commercial WiFi devices that can be fixed in a car for the long-term. We perform extensive analysis on the dependence of the classification accuracy on different factors including the size of the training set, the similarity of radio shots, the time gap between training and testing days, number of MIMO links and the number of channels.

1.3.2 Through-the-Wall Gesture Recognition (Chapter 4)

In this Chapter, we propose *GWrite*, a gesture recognition system that can work in a through-the-wall setting and recognize gesture shapes composed of straight-line segments. We have developed a complete pipeline of such a system, including gesture segmentation, turn angle classification, and intersection detection. We proposed a unique model for the hand movement during a gesture. Leveraging the model, we derived a relation between the perturbations in the physical environment and the behavior of the similarity of the CSIs in a through-the-wall setting. We believe such an understanding will open up diverse opportunities to develop practical gesture recognition systems and expand the existing WiFi-based systems restricted only to LOS setups. We built a prototype of the proposed system using COTS WiFi devices and evaluated the performance using a set of 15 upper case English alphabets. The classification accuracy achieved

was 92% which can be further improved as more bandwidth becomes available.

1.3.3 Passive Handwriting Tracking (Chapter 5)

In this Chapter, we present *mmWrite* [57], the first handwriting tracking system by reusing a single commodity 60GHz WiFi radio as a mmWave radar. With minimal infrastructure support, *mmWave* can transform any flat surface, be it a conventional whiteboard, a table, or a wall, into an interactive writing area. We present a complete pipeline to track a writing object at millimeter accuracy and thus retrieve handwritten characters as small as 1 cm x 1 cm close to the device and cover up to about 8 m^2 and a maximum range of 3 m. We perform numerous experiments to validate the performance of *mmWrite* in different conditions and environments, including varying distances, surfaces, writing speeds, and device parameters.

Chapter 2: Primer for RF and Wireless Sensing

Electromagnetic waves formed by varying electric and magnetic fields are an omnipresent means of energy transfer. Depending on the source of generation, EM waves differ in their frequency and collectively form the EM spectrum. Radio waves are a subset of the EM spectrum with a frequency range of 3 KHz-300 GHz. The IEEE 802.11 standard specifies several bands in the radio frequency (RF) spectrum that can be used for WiFi, such as 2.4 GHz, 5 GHz, and 60 GHz. Due to the fundamental difference in the frequency, the radio signals from different bands exhibit distinct kinds of propagation. Different propagation characteristics enable them to capture different information and will require specific approaches to harness it.

2.1 2.4/5GHz WiFi

A typical WiFi signal (2.4 GHz or 5 GHz) propagating in the indoor environment undergoes multiple reflections, refractions and scattering due to the physical obstacles. Such a multipath propagation might cause frequency selective fading/attenuation. When the digital transmission uses OFDM scheme to send data over different subcarriers, the optimal allocation to different subcarri-

ers is done by observing the real-time physical channel. The channel characteristics are measured in the form of Channel State Information (CSI) which can be seen as an estimate of the frequency response of the channel. The CSI stores information about the physical state of the environment. We can either analyze one CSI to infer an instance of the environment or study a CSI time-series to analyze indoor motions such as human activities or micro motions such as vital-signs.

When a Dirac delta pulse, $\delta(t)$ is transmitted through the radio channel, the received signal h_t gives the channel impulse response. In a multipath propagation channel at time t with M_t paths, the CIR is given by,

$$h_t(\tau) = \sum_{m \in M_t} \zeta_m e^{-j2\pi f_c \tau_m(t)} \delta(\tau - \tau_m(t)), \quad (2.1)$$

where ζ_m and $\tau_m(t)$ are the path gain and propagation delay of the m^{th} path respectively. At a given time t , the l^{th} tap of the CIR is given by,

$$h[l] = \sum_{m \in M_l} \zeta_m e^{-j2\pi f_c \tau_m(t)}, \quad (2.2)$$

where $M_l = \{m | \tau_m(t) \in [lT - \frac{T}{2}, lT + \frac{T}{2}]\}$ and T is the sampling interval equal to $1/\text{bandwidth}$. Although, we use (2.2) for deriving the relationships between quantities, the commercial WiFi devices can only provide CSI in the frequency domain. The frequency domain equivalent, also known as the Channel Frequency Response (CFR) is given by,

$$H_t(f) = \sum_{m \in M_t} \zeta_m e^{-j2\pi f_c \tau_m(t)} e^{-j2\pi f \tau_m(t)}. \quad (2.3)$$

The CSI from the commercial WiFi devices records a discrete version of (2.3) on different subcarriers. For instance, 802.11n/802.11ac 40 MHz channel has

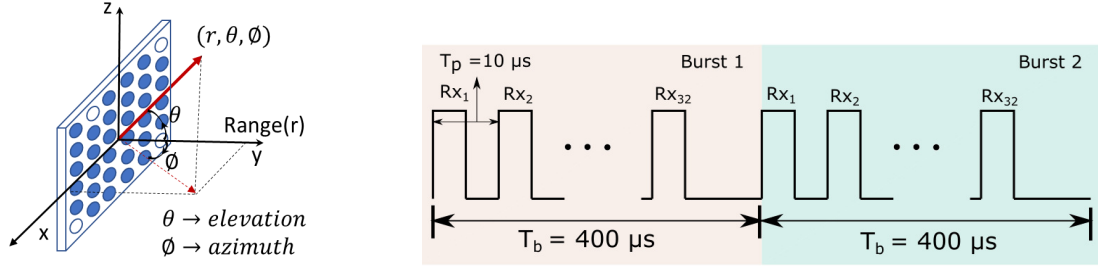
114 accessible subcarriers and the CSI is therefore a 114 dimensional complex vector. One snapshot of CSI will be processed to determine the driver identity in Chapter 3 [58] and a time-series of CSI will be used to recognize gestures in Chapter 4 [55].

2.2 60 GHz WiFi

Recently, two trends have arisen that may underpin much finer-resolution wireless sensing, such as high-precision handwriting tracking. First, the millimeter-wave (mmWave) 60GHz WiFi, introduced for high-rate networking with 802.11ad/ay standards, brings distinct advantages for wireless sensing [73, 81]. Compared to 2.4 GHz/5 GHz WiFi signals, mmWave signals offer shorter wavelengths that create stronger reflections off small objects and enable motion capturing at millimeter scales. Also, mmWave radios usually have a bigger phased antenna array, producing higher directivity and fewer multipaths that allow better spatial resolution. Below, we will discuss the characteristics of the Qualcomm’s 60 GHz radar and the target detection and localization approaches used in this work.

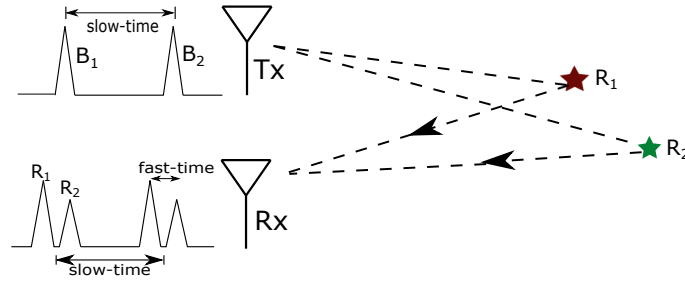
2.2.1 The 60GHz Radar

Our experimental testbed is provided by Qualcomm, which reuses a commodity 60GHz 802.11ad chipset as a pulsed mmWave radar by attaching an additional antenna array to the radio. The co-located Tx and Rx arrays are both equipped with 32 elements, arranged in a 6×6 grid with a separation of 3 mm



(a) Radar coordinate system.

(b) Concept of burst and pulse in radar



(c) Concept of slow-time and fast-time

Figure 2.1: Basic radar concepts used in *mmWrite*.

between the adjacent pair of antennas. The arrangement of the 32 antennas and the coordinate system of the radar is shown in Fig. 2.1(a).

The pulsed radar transmits electromagnetic waves as discrete pulses reflected by various objects and received by the receiver antenna. A burst is a group of 32 pulses and is transmitted by one Tx. Each of the 32 received pulses

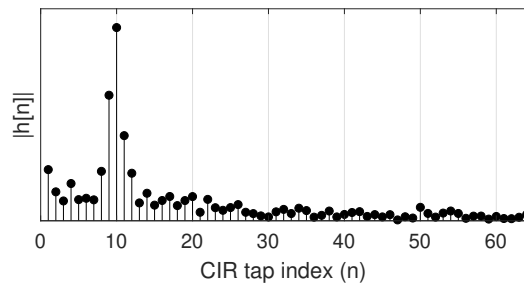
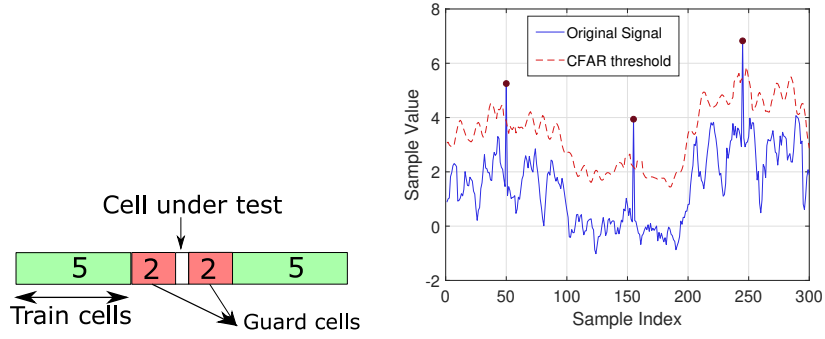


Figure 2.2: Channel Impulse Response.

is detected by the receiving antenna sequentially, and the corresponding CIR is recorded. Fig. 2.1(b) shows two bursts each consisting of 32 pulses. The CIR is computed in the hardware sequentially by switching the antenna elements and uploaded to MATLAB. This CIR is used as the input to the *mmWrite* pipeline.

Two different notions of time will be used to extract information about the writing object. The burst dimension is known as the *slow-time* dimension while the time dimension corresponding to the CIR taps is known as the *fast-time*. The concept of *slow-time* and *fast-time* is demonstrated in Fig. 2.1(c). Consider one Tx, one Rx and two reflectors R_1 and R_2 . Assume the Tx transmits two bursts, B_1 and B_2 , which are reflected and received by the Rx. Here a burst consists of only one pulse since we assume only one Rx. For each burst, the two reflections from R_1 and R_2 result in different time-of-arrival (ToA) due to different path lengths which are captured on different CIR taps. For instance, Fig. 2.2 shows one realization of CIR, $\mathbf{h}$ consisting of 64 taps. A bandwidth of $BW = 3.52 \text{ GHz}$ ¹ on our mmWave platform allows a time resolution of 0.28 ns (Δt), i.e., signals whose propagation delay differ by greater than Δt are recorded on different taps. The CIR tap index n , can therefore give an estimate of the range of the reflector. On the other hand, the *slow-time* captures the temporal difference in the position of a moving target which will be used to differentiate the moving target from the static objects. This will be elaborated more in §5.1. A time resolution of 0.28 ns translates to a range accuracy of $\frac{c\Delta t}{2} = 4.261 \text{ cm}$, where c is the speed of light.

¹Each channel in the 60GHz band of 802.11 ad spans a bandwidth of 2.16 GHz, of which 1.76 GHz is useful. A bandwidth of 3.52 GHz is achieved by channel bonding of two channels.



(a) CFAR window

(b) CFAR threshold for a noisy signal

Figure 2.3: Illustration of CA-CFAR for one dimensional signal.

The antenna response of the antenna array is measured and available on a grid on 2 degrees in the azimuth and elevation dimensions and hence accuracy of 2 degrees can be achieved in those dimensions.

2.2.2 Target Detection

Here, we briefly discuss the CA-CFAR (Cell Averaging-Constant False Alarm Rate) technique to detect the reflections from a target using a sample 1D signal. In §5.1.4, we will build on this technique to detect the moving hand. Fig. 2.3 demonstrates the CFAR technique for a one-dimensional signal. Consider a noisy signal with three targets of interest indicated by dots in Fig. 2.3(b). The CFAR window is designed as shown in Fig. 2.3(a) with the guard cells as zeros and train cells as ones. This CFAR window is convolved with the signal to estimate the average noise level. An additional fixed threshold (equal to 2 in this example) is added to the estimated noise level to obtain the adaptive CFAR threshold. The locations at which the signal is greater than the CFAR threshold are identified as

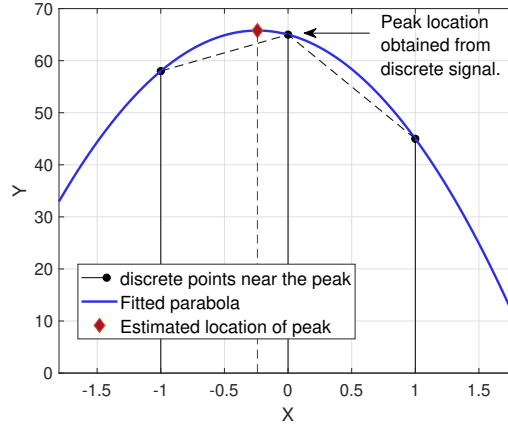


Figure 2.4: Subsample peak interpolation.

target locations.

2.2.3 Subsample Interpolation

The spatial resolution of the radar may not be sufficient to accurately localize the target after it is detected. To enable finer localization, we will use Subsample Peak Interpolation (SPI) technique. SPI is used to determine the location of peak of a signal at sub-sample accuracy in different applications [36, 63]. To include the information from the adjacent taps, we use the parabolic interpolation technique, which is demonstrated in Fig. 2.4. The circular points correspond to the discrete signal around a potential local maximum. By inspection, the center point ($X=0$) is considered as the location of the peak. However, using a parabolic fit through the three points helps us to estimate the location of the peak at sub-sample accuracy. As we will see in §5.1.5, SPI provides a significant improvement in the resolution of the target location.

Chapter 3: In-Car Driver Authentication

The field of the Internet of Things (IoT) continues to extend its capabilities and engulfs many interesting applications within. With the deployment of tremendous smart devices that can sense, exchange and analyze information, the IoT has enabled revolutionary changes in everyday lives. Smart environments and smart vehicles are among the many interesting applications in the IoT [9].

Driver authentication plays an interesting role in the future of smart vehicles. The driver authentication system improves security and can automatically make in-car driver-specific adjustments such as temperature and seat and mirror positions. Nowadays, human authentication is either done by using password-based methods such as encryption keys, PINs, key cards, or by using biometrics. The term biometrics refers to a measurement of biological data. Any biological measurement that is potentially unique to a person is considered as a biometric. Biometric-based methods are gaining popularity due to their inherent uniqueness and convenience, compared to passwords and keys which may be easily forged or forgotten. Biometrics can be broadly classified into two categories: 1) physical characteristics, comprising of fingerprints, face, iris and handprint, and 2) behavioral characteristics, such as gait, keystroke dynamics, specific gestures,

etc. Physical characteristics are more reliable and some may not change significantly with time, while behavioral traits can change over time or be changed intentionally [11].

In the proposed in-car driver authentication system, we focus on cases in which there is only a single driver present in the car with no passengers. The more practical scenario where one or more passengers are present in the car will be studied in future work. Also, the radio biometrics of the driver should have been present in the radio biometric database of the car. In the case of a temporary driver, keys or passwords should be used. Here are some practical applications:

- The proposed system can be used by parents preventing a car driven by kids or an unauthorized driver who might cause accidents. A physical key is easy to access whereas the radio biometric system can be used to differentiate kids from adults.
- In a typical home, there are usually 2-3 daily drivers for a car (e.g., the wife and the husband). In such cases, our proposed system is very useful in both security enhancement and personalization. Also, in the case of recognizing the husband and the wife, the difference in attributes is usually higher and the proposed system performs better.

In this chapter, we first describe the challenges in the proposed in-car driver authentication system in Section 3.1. Section 3.2 explains the procedure for dataset preparation. We present different methodologies to achieve driver authentication in Section 3.3 and the experimental results in Section 3.4. Finally,

we evaluate the impact of various factor on the classification performance in Section 3.5.

3.1 Challenges

The similarity of two CSIs can be defined by the time reversal resonating strength (TRRS). For two channel frequency responses (CFRs) H_1 and H_2 , the TRRS in the frequency domain is given by [84]:

$$TRRS(\mathbf{H}_1, \mathbf{H}_2) = \frac{\max_{\phi} |\sum_{k=0}^{K-1} H_1[k] H_2[k]^* e^{jk\phi}|^2}{(\sum_{k=0}^{K-1} |H_1[k]|^2)(\sum_{l=0}^{K-1} |H_2[l]|^2)}, \quad (3.1)$$

where K is the number of subcarriers. The higher the TRRS is, the more similar the two CFRs are, and thus the more similar the two radio biometric samples are. There are two main challenges in the proposed in-car driver authentication system.

3.1.1 Change of In-Car Environment

The human radio biometrics are highly correlated with the environmental information in the CSI. Hence, when the in-car environment is altered, the CSI containing the driver radio biometrics is also changed. To measure the degree of changes in an in-car environment, we record the CSI of the empty car every day. The similarity between the CSI of an empty car captured on different days and the CSI of an empty car captured on day 1 is calculated by the TRRS. Fig. 3.1 shows the change of in-car environment with time measured in terms of TRRS.

Overall, we can observe that the TRRS decreases as more changes accumulate in the in-car environment over time. The in-car environment is similar to the indoor wireless propagation environment with multipath created by numerous scatterers. The received CSI is a composition of such multipath. When one scatterer S_a is displaced, the multipath involving S_a in their paths are altered and this causes a change in the received CSI. Let the original CSI be CSI_o and the CSI after displacing S_a be CSI_a . Let another scatterer S_b be displaced and the corresponding CSI be CSI_b . The difference between CSI_a and CSI_o is due to the multipath that involved S_a only whereas the difference between CSI_o and CSI_b is due to the multipath which involved S_a only, S_b only and both S_a and S_b . Therefore, as an increasing number of scatterers are displaced with time, more multipath are altered and the CSI becomes more and more distinct from the original CSI. Since the displacement of scatterers in the car is random and there is much multipath involving each scatterer, it is highly unlikely to recreate the same multipath profile by reversing the displacement of scatterers or by a new combination of scatterer locations. Therefore, on average, we see that the TRRS decreases with time as more changes accumulate inside the car.

An existing WiFi-based human identification system used TRRS matching to identify humans [84]. During the training phase, the human radio biometric which is embedded in the indoor CSI is recorded and stored as a database. In the testing phase, the CSI is compared with those in the training database using the TRRS similarity metric. The identity of the human is determined by the class of the highest matching CSI from the training database provided that the high-

est TRRS is greater than a predefined threshold (0.7). With the given degree of changes in the in-car environment, the TRRS matching technique can no longer be applied. For example, consider the CSIs of two drivers H_1 and H_2 , collected on two different days A and B. As shown in Fig. 3.2, the TRRS between CSI of H_2 on day A and CSI of H_1 on day B is 0.81 while TRRS between CSIs of H_2 on different days is 0.63 resulting in a misclassification. In this work, we overcome this challenge by adopting ML techniques to make the system adaptive to new environments.

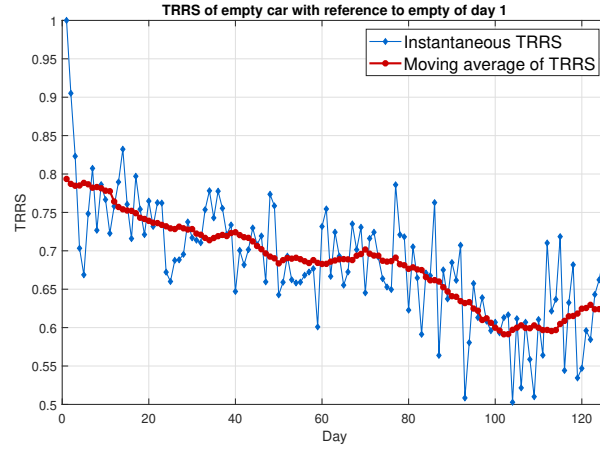


Figure 3.1: Change of in-car environment with time measured in terms of TRRS. The blue curve shows the TRRS of each day with reference to day 1 and the red curve shows the moving average of the blue curve.

3.1.2 Low Resolution of Multipath

Secondly, the CSI recorded at each time instant is a collection of channel information on multipath which has different path lengths. To resolve the multi-

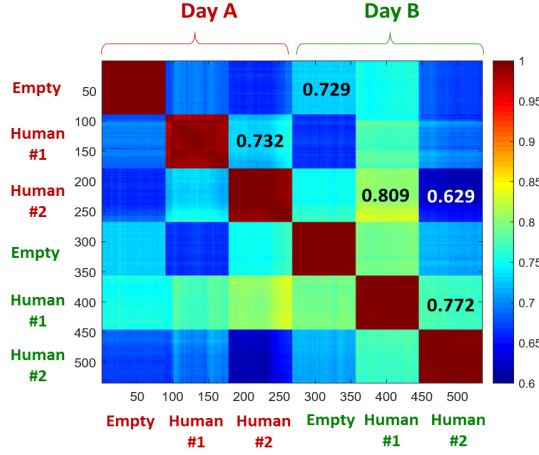


Figure 3.2: TRRS heatmap between different radio biometrics captured on different days.

path with a higher resolution, larger bandwidth is required. Since the bandwidth is fixed for each channel in commercial WiFi devices, we achieve higher effective bandwidths by exploiting the diversity in multiple MIMO links as well as different frequency channels as discussed in §3.2.

Impacts of external environment: The influence of the external environment on the in-car CSI has been studied. Fig. 3.3 shows the experimental set up where the test car is fixed and another car is parked in different locations (1-5) around the test car, in a public parking lot. In-car CSIs are recorded for each of the five scenarios. To quantitatively measure the degree of change of an in-car wireless propagation environment, we have calculated the TRRS matrix for all the recorded CSIs which is shown in Fig. 3.3(b). We can observe that the CSIs are highly correlated and the TRRS values are all nearly equal to 1. The car acts as a metal cage and only a few of the multipath escape to the external environment

through the glass windows and only a smaller fraction of them are reflected by external objects. Such multipath which are reflected into the car through the windows are severely attenuated and almost negligible. We can safely assume that the effect of the external environment is insignificant and focus only on the in-car environment changes.

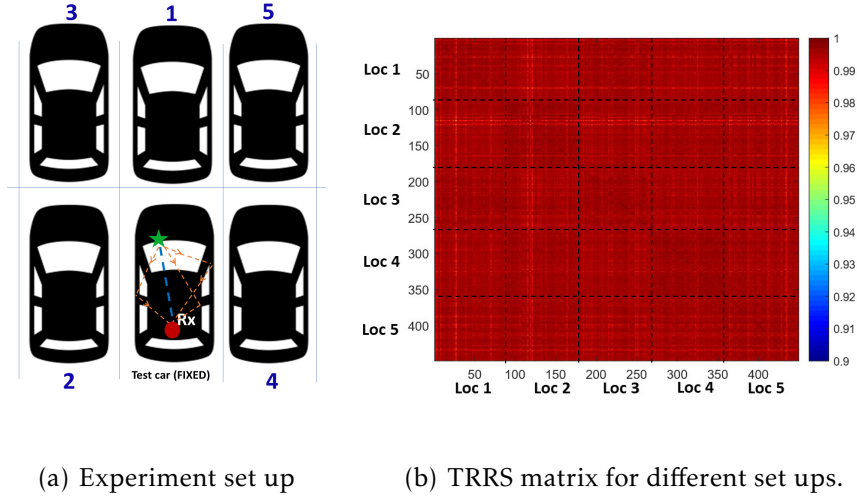


Figure 3.3: Impact of external environment on the in-car CSI.

3.2 Dataset Preparation

The proposed in-car driver authentication system uses commercial WiFi devices as transceivers that are placed in the car. The location of transceivers plays an important role in the performance of the proposed system. The location of the receiver is chosen at the back of the car at the typical location of an in-car RF antenna. The best location for the transmitter would be in front of the driver as we can capture more differentiating features including the face of the driver in a ray-tracing perspective. From our experience, the recorded human radio bio-

metrics were more distinct for different people when the transmitter was placed in front of the driver. This is because the multipath channel is affected the most when the driver intercepts the LOS path between the transmitter and the receiver. Also, a greater number of multipath passing through the driver helps to capture more driver specific radio biometric features. Fig. 3.4 shows two possible transceiver locations and the multipath propagation inside the car. The blue line shows the LOS path which is intercepted when the driver is present in the car. The orange-colored lines show the NLOS paths which are received by the receiver after several reflections inside the car. The transmitter is located at the back of the steering wheel in Fig. 3.4(a), while it is placed near the audio system in Fig. 3.4(b). The slight difference in the locations is because of the space limitation during experiments. The performance for other possible sets of transceiver locations is left for future work. In the following, we will describe the frequency hopping mechanism designed in the proposed system, data collection procedure, and the pre-processing technique.

3.2.1 Frequency Hopping on Commercial WiFi Devices

Frequency hopping refers to changing channels according to a pre-specified schedule/pattern and enables utilization of frequency diversity. Using frequency hopping, we can record CSIs on different channels sequentially. To increase the number of features for in-car driver radio biometrics, we record CSIs on four channels in the 5.2 GHz band, during a radio shot. By doing so, we achieve a

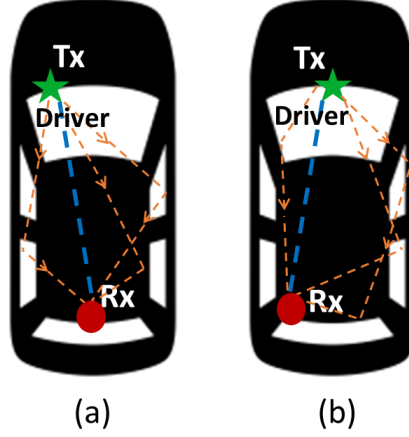


Figure 3.4: Location of transceivers in the car. Scenario (a) has transmitter near the speedometer at the back of the steering wheel and in Scenario (b), the transmitter is at the audio system.

larger effective bandwidth as explained in a later section.

The frequency hopping algorithm that we used is explained in Algorithm 1 and Algorithm 2. The transmitter and receiver function in parallel. The channel index is taken as ch . The transmitter sends the channel information to the receiver in specially designed frames called action frames. These are sent at regular intervals of time and the duration of each channel is specified by the user as dwell time ($\mu = 125$ milliseconds). Fig. 3.5 demonstrates the mechanism of hopping channels at the receiver. It also shows the absolute time of arrival of the CSI frames at the receiver. The stairs-like pattern is caused by the action frames before setting to a new channel during which no CSI samples are recorded. Sometimes, due to channel congestion/packet loss, all of the k action frames might be lost and the receiver continues to stay in the same channel as in regions (B). In

such cases, the next channel is not set until the receiver receives action frames on the existing channel in the next cycle. In §3.5.6, we evaluate the performance of the proposed driver authentication system for a different number of channels.

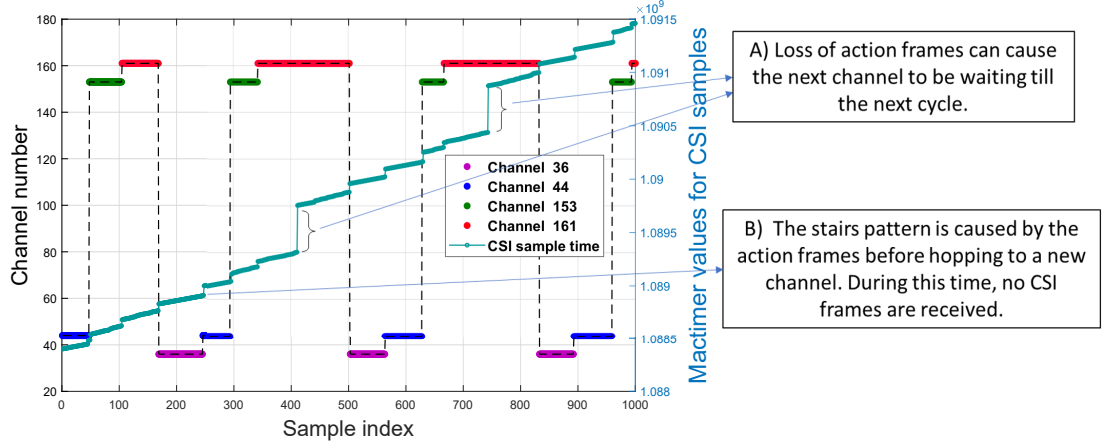


Figure 3.5: Demonstration of channel hopping at the receiver with time.

3.2.2 Data Collection

In our experiments, during the radio shot, the driver sits in the driver's seat of a car and the wireless propagation environment is captured in the CSI. This CSI is used as the radio biometric for that particular driver. For every radio shot of the driver, we also record the CSI of the corresponding in-car environment without the driver.

The in-car driver authentication database was built by collecting radio shots of seven people over two months. On each day, for each test subject, four radio shots were taken in the morning and evening, in a car parked at different locations in a public parking lot. By doing so, a total of 60 different environments

Algorithm 1 Frequency hopping algorithm: Transmitter

```
1:  $channel\_list \leftarrow \{36, 44, 153, 161\}$ 

2:  $ch \leftarrow 0$ 

3: while (1) do

4:   Set channel to  $channel\_list(ch)$ 

5:   Send CSI frames as channel probing signals on channel  $channel\_list(ch)$ 

   for dwell time ( $\mu$ )

6:   Determine the next channel index

    $ch \leftarrow (ch + 1) \bmod 4$ 

7:   Next channel is  $channel\_list(ch)$ 

8:   Construct and send  $k$  action frames with new channel index information

9: end while
```

Algorithm 2 Frequency hopping algorithm: Receiver

```
1:  $channel\_list \leftarrow \{36, 44, 153, 161\}$ 

2: Set channel to  $channel\_list(0)$ 

3: while (1) do

4:   if CSI frame is received then

5:     receive CSI

6:   else if action frame is received then

7:      $ch' \leftarrow$  next channel extracted from action frame

8:     Set channel to  $ch'$ 

9:   end if

10: end while
```

have been considered. Multiple recordings of the radio shots help us to improve the classification accuracy using the grouping technique which is discussed in §3.3.4.

The prototype of the proposed in-car driver authentication system was built using the commercial off-the-shelf WiFi chips with no additional hardware. The CSIs were recorded using a 2×3 MIMO system. The system operated in the 5.2 GHz band over four channels with 114 accessible subcarriers in each channel. Also, multiple CSIs were recorded using a sounding rate of 30 Hz to perform phase cleaning and remove outliers as discussed later in this section. So, for each radio shot, each CSI sample is a $2 \times 3 \times 456$ dimensional complex-valued matrix.

3.2.3 Data Preprocessing

Timing and frequency synchronization errors in the WiFi systems introduce phase offsets in the recorded CSI. The multiple CSIs recorded for each radio shot are highly correlated with each other and thus can be used for phase compensation and outlier removal. In data preprocessing, we compensate for the linear and the initial phase offsets.

Let $\hat{H}_p[k]$ and $\hat{H}_p[k]$ be the p^{th} CFR on the k^{th} subcarrier with and without phase distortions. $\hat{H}_p[k]$ can be written as [13],

$$\hat{H}_p[k] = \text{sinc}(\pi(\Delta\epsilon + \Delta\eta k))H_p[k]e^{j2\pi(\beta_p k + \alpha_p)}, \quad (3.2)$$

where $\Delta\epsilon$ and $\Delta\eta$ are the residual errors of channel frequency offset and sampling frequency offset respectively, and β_p and α_p are the linear and initial phases re-

spectively. Assuming the argument of the sinc function is small, the linear phase can be aligned with a reference CFR [84]. Consider two CFRs $\hat{H}_p[k]$ and $\hat{H}_q[k]$. Let $\hat{H}_p[k]$ be the reference. Then,

$$\hat{H}_p[k] = H_p[k]e^{j2\pi(\beta_p k + \alpha_p)}, \quad (3.3)$$

$$\hat{H}_q[k] = H_q[k]e^{j2\pi(\beta_q k + \alpha_q)}, \quad (3.4)$$

$$\delta\beta = \underset{\phi}{\operatorname{argmax}} |\sum_k \hat{H}_p[k] \hat{H}_q[k]^* e^{j2\pi k \phi}|, \quad (3.5)$$

and the aligned linear phase is obtained by $\hat{H}_q[k]' = \hat{H}_q[k]e^{-j2\pi k \delta\beta}$. The initial phase is equal to the phase of the first subcarrier on each CFR sample. An example is shown in Fig. 3.6 where the linear and initial phases are compensated for CSIs collected using four channels. In this case, the phase compensation should be done independently for each channel as the phase offsets are different for different carrier frequencies. Fig. 3.6(a) shows the phase of raw CSI for four channels. The first 114 subcarriers correspond to channel 1, the next 114 to channel 2 and so on. Fig. 3.6(b) shows the phase after linear phase compensation and Fig. 3.6(c) shows the resultant phase after linear and initial phase compensation. The CSIs are then appended to form the feature vector for the in-car driver authentication system.

After the phase alignment, the combined CSI from the four channels results in a $2 \times 3 \times 456$ (i.e 114 subcarriers per channel) dimensional complex-valued vector which can be flattened to a 5472-dimensional real-valued vector. With such a high dimension of features, the number of parameters that need to be learned in ML models is large and usually, the models require a lot of data to

train. Unlike computer vision techniques, obtaining a large amount of data in our case is expensive. Hence, we perform dimensional reduction using Principal Component Analysis (PCA) to reduce the number of parameters. PCA transforms the original features into a new feature space based on the degree of variance. In this work, we consider the number of features which contribute to 99% of the total variance in the data. For instance, the dimension reduced from 5472 to 270 for the data using all the four channels and 2×3 MIMO links.

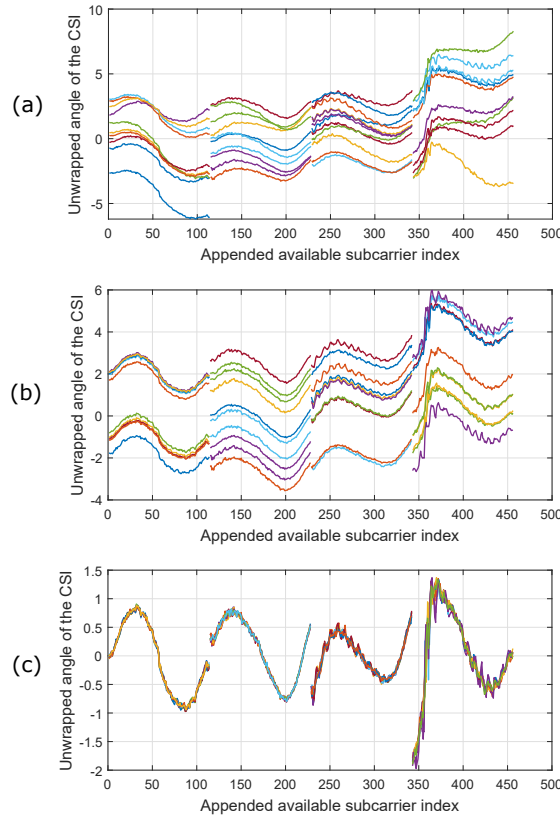


Figure 3.6: Demonstration of linear and initial phase compensation. Figure shows the phase (a) of raw CSI, (b) after linear phase compensation, (c) after initial phase compensation.

3.3 Learning Methodologies

In this section, we introduce the ML techniques and methods that we adopt in the proposed driver authentication system.

3.3.1 K-Nearest Neighbors (K-NN)

We know that the radio biometrics are embedded inside the CSI of the environment and are highly correlated. In the proposed in-car driver authentication system, a new in-car environment is presented on a new day. This can be seen as a new instance of the problem and one baseline approach would be to use instance-based learning methods [5]. K -nearest neighbor approach is the simplest of these methods and often used as a baseline for classification algorithms. In this approach, for a new test sample, we select K nearest neighbors from the existing database and assign the majority label to the test sample. We measure the similarity using the Euclidean distance.

The value of K is a hyperparameter and can be chosen by conducting several experiments and finding the best value of K that gives the maximum average performance. For example, consider radio biometric data of two drivers collected for 40 days. Fig. 3.7 shows the 40-fold cross-validation accuracy with varying number of nearest neighbors (K). The maximum accuracy is achieved for a value of 3. Therefore, for the classification of these drivers, we use 3-nearest neighbors.

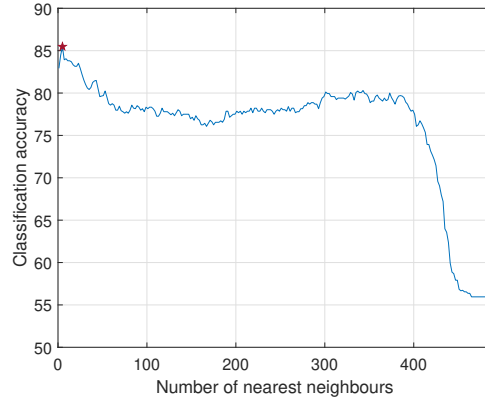


Figure 3.7: Average K-NN accuracy for varying value of K.

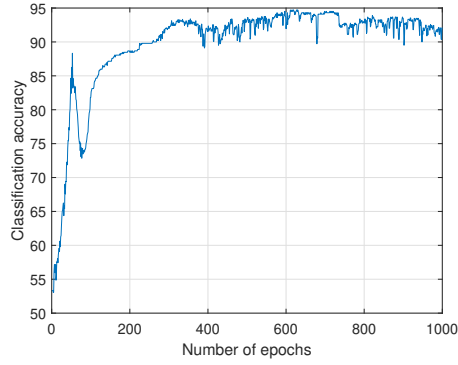


Figure 3.8: Average classification accuracy with number of epochs in a neural network.

3.3.2 Support Vector Machine (SVM)

SVM is the most popular approach for classification algorithms in ML [18]. Linear-SVM aims to find a hyperplane that divides the classes with maximum margin where the margin is defined as the minimum distance of the hyperplane to points from either class. When the data is not linearly separable which is often the case, the “kernel-trick” is used to project the data to higher dimensions where it is linearly separable [69]. In this work, we use linear and RBF kernels

to evaluate the proposed system with the regularization parameter $C = 1.0$. Also, we use a cross-validation technique to report classification accuracy.

3.3.3 Neural Network

As the in-car environment changes with time, we would want the system to be more adaptive and learn human radio biometrics for different environments. Therefore we refer to neural networks (NN) which have been used in ML and deep learning to learn more non-linear and complex decision boundaries for classification problems.

Architecture The hyperparameters in the neural network are tuned using the K-fold validation technique. Consider, for example, the number of training epochs. We find the classification accuracy for all the K experiments for 1000 epochs. It is observed that the model is over-fit much before 1000 epochs. We then calculate the average performance for every epoch. Fig. 3.8 shows the average classification accuracy obtained for the pair A-D with the number of epochs. The maximum value is reported as the final accuracy. Here, the maximum is achieved near epoch 160 with an accuracy of 93.33%. The number of hidden layers and hidden nodes are determined by cross-validation. As we further increased the number of hidden layers or the hidden nodes, the capacity of the network increased and it began to over-fit. The neural network architecture that we use is shown in Fig. 3.9. The network consists of an input layer with a number of input nodes equal to the input features, two hidden layers and an output layer which

gives the class probabilities for a data point. In this work, we adopt the ReLU activation function and cross-entropy loss with the Adam optimizer.

Data We have used about 40 days of data for evaluating the performance using K_v -fold validation. The number of samples per day is 8 and the total data available per class is nearly 320 samples. The data are partitioned by date and in total we have 40 partitions. The 40-fold validation accuracy is used as the evaluation metric in §3.4.

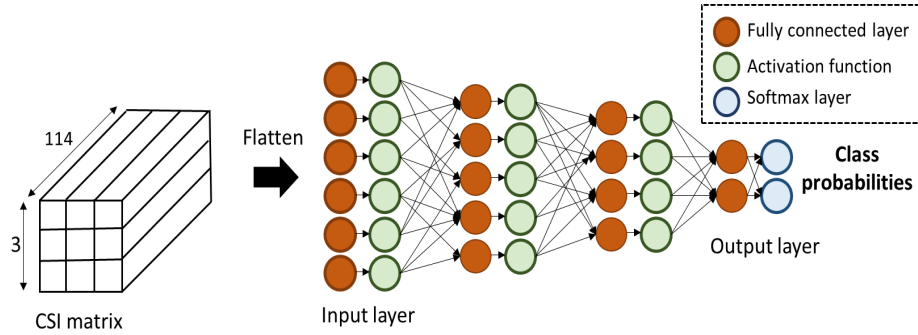


Figure 3.9: Neural network architecture.

3.3.4 Grouping

During the process of radio shots, slight variations in the seating positions of the driver can cause a change in the CSI and might sometimes lead to misclassification. To capture and compensate these small variations, we collect multiple radio shots for the same in-car environment and take a combined decision. We call it the grouping technique which is explained in Fig. 3.10. During the testing phase, for each test subject, assume the four radio shots are indexed as i ;

$i = 1, 2, 3, 4$. Let P_{Ai} and P_{Bi} represent the predicted class probability of the i^{th} radio shot for class A and class B, respectively. Then the identity of the test subject is determined as class A, if $\sum P_{Ai} < \sum P_{Bi}$ and vice versa. If $\sum P_{Ai} = \sum P_{Bi}$, the test subject cannot be determined and we considered such samples as incorrectly classified in our accuracy calculations. We used four radio shots since, for our test subjects, more than four radio shots led to repetitions of the CSI. This can easily be extended to more number of radio shots based on the consistency of seating postures of the test subjects.

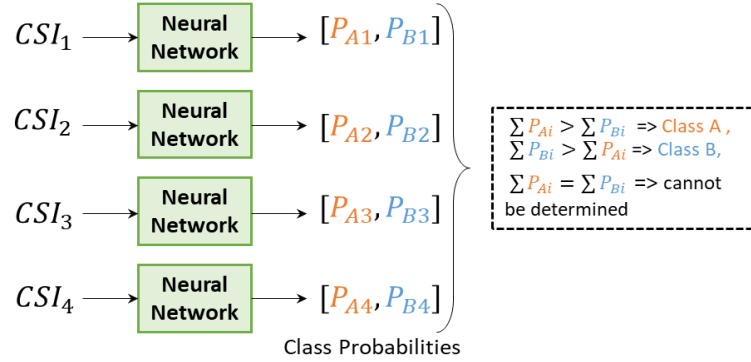


Figure 3.10: Grouping technique

3.4 Experimental Results

We evaluate the performance of the proposed in-car driver authentication system using the ML models discussed in the previous section. First, we discuss the special case of two driver authentication which can serve a similar purpose as the existing memory seating facilities in cars alongside providing authentication. Later, we evaluate the performance of the proposed system in the multi-driver

scenario.

One of the main challenges in evaluating the in-car driver authentication system is the availability of data. The study of the trend of radio biometric data with the time required that the testers be available throughout the experiment duration which is two months. Since the amount of data is limited, the accuracy values obtained are dependent on the split of train and test data. This is because the selected train data may or may not be able to generalize well to the test data. To overcome this, cross-validation techniques are used in ML models [61]. In the simplest cross-validation technique, the entire data set is divided into K_v parts and K_v experiments are performed with each part as testing data and the remaining $K_v - 1$ parts as the training data. In this work, the value of K_v is taken as the number of days of available data i.e we treat the data corresponding to each day as a partition. By doing so, we make sure that the same instance of the in-car environment is not present in both training and testing data. Throughout this work, the reported accuracies are calculated using K_v -fold validation and the value of K_v is taken as the total number of days of data available.

3.4.1 Two-Driver Authentication

In this scenario, we classify a driver into one of the two known drivers i.e., a two-class problem. We consider five test subjects denoted by A, B, C, D and E. More information about the testers is shown in Table-3.1. The accuracy of the proposed system with different ML techniques is evaluated using a 40-fold val-

idation with CSI from one channel and with the proposed grouping technique. Table-3.3 - III shows the classification accuracy for different sets of drivers from the in-car driver radio biometric database. The performance of the K-NN in the first column can be taken as the baseline performance. On average, the NN approach gave about 7% increase in accuracy. The highest accuracy achieved is 99.13% for the pair A-D.

Table 3.1: Information about the testers

Name	Gender	Age	Ht (<i>cm</i>)	Wt (<i>kg</i>)
A	F	25	163	56.2
B	F	28	165	58.5
C	M	30	168	82.5
D	M	23	172	85
E	M	25	180	73

Comparison with state-of-the-art approach To our best knowledge, there is only one prior work that uses radio biometrics embedded in the CSI of wireless signals for human recognition [84]. However, as discussed in §3.1, the performance of the previous work will be compromised by the in-car environment changes because it heavily relied on TRRS to compare the similarity between different radio biometrics embedded in the CSI. Table-3.2 demonstrates the improvement delivered by the proposed learning-based approach. In all the cases, the learning-based approach outperforms the state-of-the-art TRRS-based ap-

Table 3.2: Classification accuracy: Comparison between learning-based and TRRS-based approaches for two-driver authentication.

Classes	TRRS-based(%)	Learning-based(%)
A-B	76.51	96.55
A-C	83.09	98.27
A-D	87.73	99.13
A-E	82.18	93.10
B-C	80.88	96.55
B-D	81.58	93.67
B-E	76.13	87.06
C-D	64.93	72.12
C-E	74.60	91.37
D-E	73.41	90.22

proach by at least 7% and up to 20%.

3.4.2 Multiple Driver Authentication

We also evaluate the performance of the proposed system in identifying more than two drivers. Fig. 3.13 shows the confusion matrices for multiple driver classification using the NN approach. The average detection rate for an individual among the three drivers is 84.33% while among seven drivers is 53.85%. This is much greater than the accuracy achieved by random guessing of identities

Table 3.3: Performance of two driver authentication

Classes	K-NN(%)	Linear SVM(%)	SVM-RBF(%)	NN(%)
A-B	88.93	92.52	90.22	96.55
A-C	91.88	90.87	93.03	98.27
A-D	90.37	94.39	93.10	99.13
A-E	88.50	94.39	90.44	93.10
B-C	89.79	89.15	90.51	96.55
B-D	85.70	88.93	90.08	93.67
B-E	75.71	84.69	85.48	87.06
C-D	65.80	70.83	72.12	60.63
C-E	83.11	86.99	85.77	91.37
D-E	82.68	86.99	88.36	90.22

among seven people i.e., 14.28%. The performance decreases with an increasing number of drivers. The increasing off-diagonal elements from confusion matrices between three to seven people also indicate an increasing false alarm. The average false alarm increased from 15.33% in the three driver case to 46.14% in the seven driver case.

3.5 Discussions

The performance of a human radio biometric-based system is dependent on many factors, such as the physical characteristics of the people, the number

of channels used to obtain CSI, the environment, the amount of training data present, the number of classes present, etc. In this section, we analyze and evaluate the impact of the various factors on the in-car driver authentication system using the NN approach. The hyperparameters are tuned using the K_v -fold validation technique.

3.5.1 Size of the Training Set

We evaluate the performance of the proposed system as the size of the training set increases. Fig. 3.11 shows the performance of the system averaged over several pairs of people. As the amount of training data increases, the classification accuracy improves. When a new user is added, the performance will improve as the proposed adaptive neural network continues learning and generalizing the distinctiveness between his/her and other radio biometrics. From Fig. 13, we can see that the accuracy improves drastically and reaches about 90% in 15 days. As the user continues to use the car, the proposed system can capture more radio biometric information of him/her and then improve the recognition accuracy.

3.5.2 Similarity of Radio shots

We can observe from previous experiments that the classification performance largely depends on the set of people that we aim to differentiate. This is because some people can have more similar radio biometrics compared to others. Classification of such people might be more challenging than others.

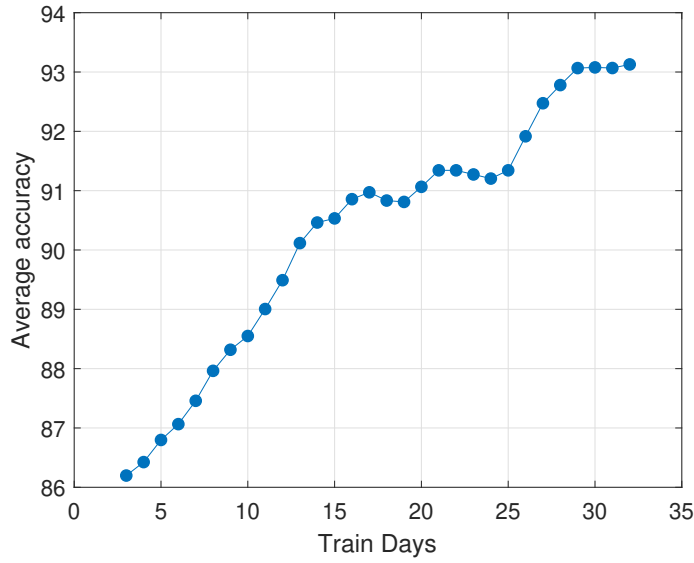


Figure 3.11: Accuracy averaged over several pairs of people with the amount of training data.

Table 3.4: Similarity of radio shots and classification accuracy

Classes	Average TRRS	Classification accuracy
A-D	0.7094	99.36%
C-D	0.7773	84.19%

To support our observation, we have calculated the similarity of radio biometrics in the same environment for different sets of people, where similarity is measured in terms of TRRS as defined in (1). In Table-3.4, we show the TRRS calculated for the pairs with maximum (A-D) and minimum accuracy (C-D) averaged over all days. We can say that the similarity of CSIs is one of the many factors affecting the classification accuracy. The accuracy is lower for the pair with more similar CSIs. However, many other factors can influence the classifi-

cation accuracy in addition to the similarity of radio biometrics. A few of them are listed below.

1. Consistency of seating position: If the seating position of a tester is consistent and similar during the training and testing period, the classification accuracy is improved. On the other hand, if the tester tends to sit in different seating postures or positions every time, it might lead to a decreased classification accuracy but can get mitigated with an adaptive training data set which keeps refreshing with newly added radio biometrics samples.
2. Difference in the in-car environment: As the human radio biometrics are highly correlated with the in-car environment, if the difference in the in-car environments (measured quantitatively by the TRRS) during the training and the testing period is significant, the classification accuracy tends to decrease.
3. Difference in the training and testing data: For example, if a tester wears a thick jacket during the training phase and no jacket during the testing phase, the classification accuracy might decrease.

3.5.3 Performance with Increasing Gap between Training and Testing Data

As discussed in §3.1, with time, the changes in the in-car environment accumulate and the TRRS with reference to day 1 continues to decrease. This causes

a decrease in classification accuracy. Fig. 3.12 shows the classification accuracy with an increasing gap between the training and testing data. The maximum TRRS achieved by the test sample with the samples from the training database is shown in blue. The red line shows the average accuracy achieved in the case of two-driver authentication. We can see that with increasing difference between train and test times (days), the maximum matching TRRS and the classification accuracy have a decreasing trend. The classification accuracy does not monotonically decrease since it also depends on other factors such as variation in the seating positions, type of clothing, etc. From this observation, the best performance of the system can be achieved when it is used regularly and by constantly updating the database. The more regular and longer this system is used, the better is the performance.

3.5.4 Effect of Grouping

Grouping technique (§3.3.4) uses multiple radio shots to determine driver identity. In Table-3.5, we show the classification accuracy with and without grouping for one channel using the NN approach. We observe that in most cases, the grouping technique can significantly improve the classification accuracy and hence using multiple radio shots to predict the identity is more reliable. Few exceptions in the case of K-NN could be due to a large variation in the seating position for each radio shot.

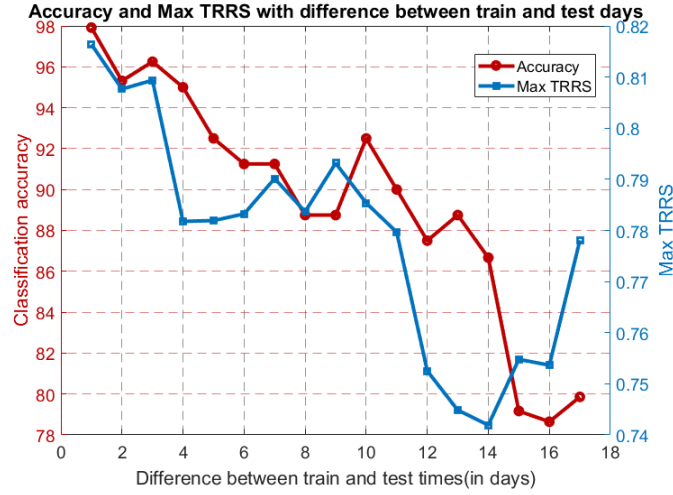


Figure 3.12: Accuracy and maximum TRRS with the difference between train and test days. Red line shows accuracy with the increasing gap between the training and testing days. The blue line shows the TRRS which is the best match of the empty in-car environment from the training database.

3.5.5 Effect of the Number of Links

Through exploiting the antenna diversity provided by multiple links of the MIMO system, we can explore different multipath in the environment and there is a potential increase in the number of independent features. Fig. 3.14 shows the classification accuracy with an increasing number of links for two driver authentication. Overall, we can see that the performance, increases with the number of links.

Table 3.5: Performance with and without grouping

Classes	K-NN	K-NN(G)	NN	NN(G)
A-B	88.93	90.22	88.93	96.55
A-C	91.88	94.25	92.81	98.27
A-D	90.37	90.51	94.18	99.13
A-E	88.50	90.80	87.42	93.10
B-C	89.79	91.09	90.22	96.55
B-D	85.70	85.91	87.93	93.67
B-E	75.71	71.26	80.45	87.06
C-D	65.80	61.78	57.75	60.63
C-E	83.11	86.20	84.77	91.37
D-E	82.68	85.91	82.39	90.22

3.5.6 Effect of the Number of Channels

The effective bandwidth has been defined as $We = D \times W \times N$, where W is the bandwidth per channel per link which is 40 MHz in the proposed system, D is the number of channels and N is the number of links [13, 14, 71]. The radio shots are taken on a single channel at a time, according to the proposed frequency hopping mechanism. In our work, we use a maximum of four channels and achieved the largest effective bandwidth of $2 \times 3 \times 40 \times 4$, i.e 960 MHz. Table-3.6 shows the performance using the NN approach with training data of 12 days for a different

number of channels. While in a few cases, there is a marginal difference in the accuracy, in other cases such as B-E, a significant increase in the classification accuracy is observed from 64.13% to 80.65%. The effective bandwidth that we achieve here is different from the physical bandwidth. Although we increase the number of channels using frequency hopping, the system uses only one channel at a time which cannot improve the resolution of multipath. However, using different channels and thus different carrier frequencies, more features can be extracted. The variation in the performance improvement for different pairs of people with the number of channels is yet to be analyzed by involving a greater number of subjects in the study.

3.6 Summary

In this Chapter, we propose the first in-car driver authentication system using the human radio biometrics recorded in the wireless CSI. To address the problem of in-car environmental changes, we build a multiple-driver radio biometric database consisting of radio biometrics of seven people collected over a period of two months. To our knowledge, this is the first long-term study conducted for human radio biometric recognition. With the help of this database, we integrate ML techniques to make the proposed driver authentication system adaptive to different in-car environments. Later, we study the impact of the multi-antenna diversity and the frequency diversity on the accuracy of the proposed driver authentication system. For experimental evaluation, we have

Table 3.6: Performance with increasing effective bandwidth using Neural network approach.

Classes	1 ch	upto 2 ch	upto 3 ch	upto 4 ch
A-B	77.95	78.69	81.63	82.51
A-C	92.39	92.39	92.39	92.39
A-D	78.14	82.58	88.26	88.26
A-E	68.37	79.97	83.37	83.37
B-C	90.53	92.18	92.18	92.18
B-D	83.64	83.64	83.64	83.64
B-E	64.13	73.80	79.04	80.65
C-D	73.65	76.06	76.06	76.06
C-E	88.97	88.97	88.97	88.97
D-E	77.90	83.50	83.50	83.50

implemented frequency hopping on portable commercial WiFi devices that can be fixed in a car for the long-term. We perform extensive analysis on the dependence of the classification accuracy on different factors including the size of the training set, the similarity of radio shots, the time gap between training and testing days, number of MIMO links and the number of channels.

True Class	#1	82.0%	10.0%	8.0%
	#2	4.0%	87.0%	9.0%
	#3	6.0%	9.0%	85.0%
		#1	#2	#3
		Predicted Class		

(a) 3 people

True Class	#1	83.0%	8.0%	5.0%	4.0%
	#2	4.0%	82.0%	10.0%	4.0%
	#3	9.0%	9.0%	74.0%	8.0%
	#4	5.0%	8.0%	28.0%	59.0%
		#1	#2	#3	#4
		Predicted Class			

(b) 4 people

True Class	#1	78.0%	14.0%	7.0%	1.0%	0.0%
	#2	9.0%	74.0%	1.0%	6.0%	10.0%
	#3	6.0%	3.0%	47.0%	41.0%	3.0%
	#4	2.0%	12.0%	17.0%	66.0%	3.0%
	#5	2.0%	9.0%	19.0%	12.0%	58.0%
		#1	#2	#3	#4	#5
		Predicted Class				

(c) 5 people

True Class	#1	78.0%	11.0%	6.0%	1.0%	1.0%	3.0%
	#2	4.0%	66.0%	1.0%	4.0%	10.0%	15.0%
	#3	3.0%	3.0%	44.0%	39.0%	3.0%	8.0%
	#4	0.0%	6.0%	18.0%	59.0%	0.0%	17.0%
	#5	2.0%	9.0%	22.0%	5.0%	50.0%	12.0%
	#6	10.0%	7.0%	7.0%	1.0%	10.0%	65.0%
		#1	#2	#3	#4	#5	#6
		Predicted Class					

(d) 6 people

True Class	#1	68.0%	11.0%	4.0%	0.0%	6.0%	10.0%	1.0%
	#2	1.0%	63.0%	3.0%	3.0%	6.0%	9.0%	15.0%
	#3	2.0%	2.0%	41.0%	39.0%	3.0%	5.0%	8.0%
	#4	0.0%	9.0%	9.0%	59.0%	6.0%	6.0%	11.0%
	#5	0.0%	8.0%	14.0%	8.0%	53.0%	5.0%	12.0%
	#6	9.0%	4.0%	4.0%	6.0%	6.0%	34.0%	37.0%
	#7	6.0%	9.0%	6.0%	3.0%	10.0%	7.0%	59.0%
		#1	#2	#3	#4	#5	#6	#7
		Predicted Class						

(e) 7 people

Figure 3.13: Confusion matrices for different number of people.

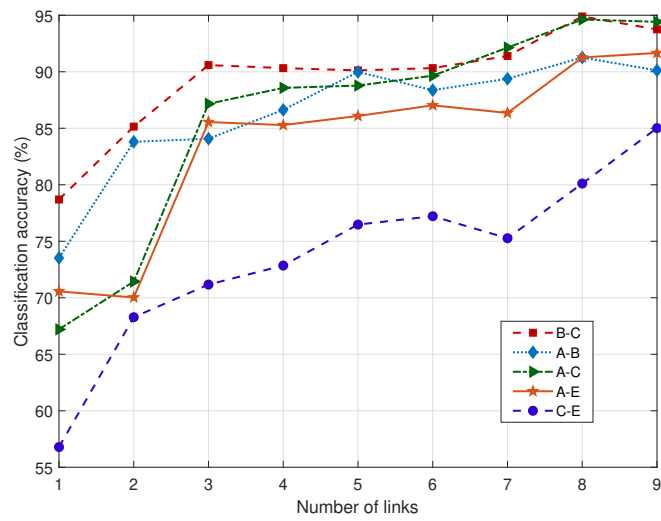


Figure 3.14: Classification accuracy with increasing number of links for different sets of people.

Chapter 4: Through-the-Wall Gesture Recognition

Proliferating number of intelligent connected devices and computers are transforming the space around us. Human-Computer Interaction (HCI) forms a significant portion of our daily routine, and there is a need to simplify it as much as possible. With time, researchers around the world have come up with innovative solutions and approaches to ease HCI. We have come a long way from using manual operations to push buttons, touch sensors to motion sensors, and the fancier fingerprint and face ID systems for authenticated services. We evolved from using keyboard inputs to using in-the-air gestures and voice commands to talk to computers. These efforts underscore our desire to ease interaction with computers and facilitate human lives. In-the-air gesture recognition blends easily with the future smart environments.

Researchers have experimented with different types of signals to build practical gesture recognition systems. Vision-based approaches tracked the movement of hand/finger to reconstruct the hand trajectory. Commercial applications such as gaming have become widely popular with the Microsoft Kinect and LeapMotion. The vision-based approaches specifically target LOS applications, require ambient light, and potentially raise privacy concerns. Sensor-based tech-

niques provide reliable solutions to gesture recognition, and many such systems have been developed [4, 74]. However, passive or device-free approaches are preferred due to better user convenience and experience. Few works used RF-signals to track hand motion and perform handwriting/gesture recognition [74]. Such systems require dedicated hardware and incur additional costs. On the other hand, passive WiFi-based approaches have the most suitable features for many exciting applications, including practical gesture recognition [40, 71]. Some of these include easy penetration of walls relaxing the LOS constraints, ubiquitous infrastructure, privacy preservation, and better user experience. In this work, we propose *GWrite*, a passive gesture recognition system using commodity WiFi devices that can work in through-the-wall scenarios.

At the core of the *GWrite* system lies a statistical model of the WiFi channel time-variance. The model allowed us to simplify the expression for the adopted similarity metric between two CIRs. The proposed method of hand gestures further helped us derive a correspondence between the relative distance moved by the hand and the similarity decay. Leveraging this property, we extracted differentiable features for gesture shapes consisting of straight-line segments. We designed algorithms to automatically identify different straight line segments, determine the angle between two segments and identify intersections between segments. The gesture classification is performed by selecting the character with the highest accumulated probability score for its constituent features.

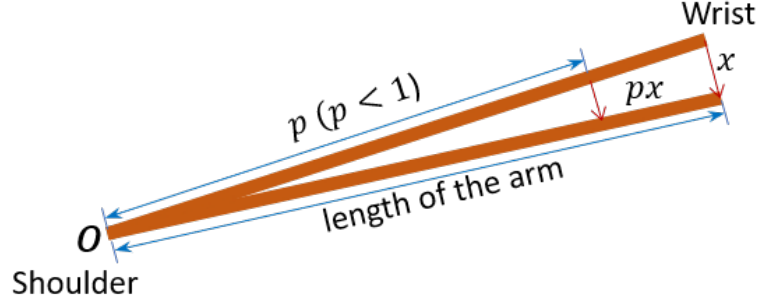


Figure 4.1: Demonstration of the movement of the hand pivoted at the shoulder(O).

4.1 System Model

In this section, we develop a statistical model to explain the changes in the wireless channel induced by a hand gesture. An instance of a wireless channel can be characterized by its Channel State Information (CSI) or equivalently by its Channel Impulse Response (CIR). A gesture is a continuous movement of the hand and can thus be represented by a CIR/CSI time series. It is useful to quantify the relationships/similarities between any two CSIs in the time series to develop a correspondence between the hand movement and the wireless channel variations. For this, we adopt the Time Reversal Resonating Strength (TRRS) metric.

Assume the wireless channel between a transmitter (Tx) and a receiver (Rx) is denoted by the CIR $h(l)$, where $l = 0, 1, \dots, L - 1$ and L is the total number of taps. Also assume that the CIR is normalized i.e., $\sum_{l=0}^{L-1} |h(l)|^2 = 1$. If the Tx sends an impulse $\delta(l)$, the Rx receives $y(l) = h(l)$. Note that the wireless channel from

$T_x \rightarrow R_x$ and $R_x \rightarrow T_x$ can be considered equivalent following the channel reciprocity. Now, if the R_x sends $h^*(-l)$ i.e., a time reversed and a conjugate version of the channel, then the received signal at the T_x becomes $y'(l) = h^*(-l) \star h(l)$, where $\star$ denotes convolution. The zeroth tap of $y'(l)$ becomes:

$$y'(0) = \sum_{l=0}^{L-1} h^*(l)h(l) = \sum_{l=0}^{L-1} |h(l)|^2 = 1, \quad (4.1)$$

i.e., we can know if the channel at a later time instance is different from the initial channel by sending a time inversed and conjugate version of the initial CIR from R_x and recording $y'(0)$ at T_x . It can be concluded that the channel has changed if $y'(0) \neq 1$. Alternatively, we can compute,

$$\eta(\mathbf{h}_1, \mathbf{h}_2) = \left| \frac{\sum_{l=0}^{L-1} h_1(l)h_2(l)^*}{\sqrt{\sum_{l=0}^{L-1} |h_1(l)|^2} \sqrt{\sum_{l=0}^{L-1} |h_2(l)|^2}} \right|, \quad (4.2)$$

which gives the TRRS between two CIRs $\mathbf{h}_1$ and $\mathbf{h}_2$. The value of η lies between 0 and 1 where 0 is achieved by a pair of uncorrelated CIRs. Assessing the extent of the physical perturbation in the channel could reveal more information about the hand movement than mere detection of a change. Let us further analyze (4.2) for the case of a hand gesture to understand the relation between the TRRS drop and the extent of hand movement.

Let $\mathbf{h}_0$ denote the CIR when the hand is at an initial position. Due to the multiple reflections in an indoor environment, each tap of the CIR is composed of several multipath components (MPCs) which can be explicitly written as:

$$h_0[l] = \sum_{m \in M} \zeta_{l,m} e^{-j2\pi f_c \tau_{0,l}(m)}, \quad (4.3)$$

where M is the set of multipath, ζ is the complex path gain, f_c is the carrier frequency and τ is the path delay. Let $\mathbf{h}_x$ denote the CIR after the hand moved by a short distance x . Then,

$$h_0[l]h_x[l]^* = \sum_{m \in M} \zeta_{l,m} e^{-j2\pi f_c \tau_{0,l}(m)} \sum_{m \in M} \zeta_{l,m}^* e^{j2\pi f_c \tau_{x,l}(m)}, \quad (4.4)$$

where the set M is considered unchanged during the short time. Assuming a sufficiently large bandwidth such that the significant MPCs are captured on distinct CIR taps [89], we can approximate the numerator of $\eta(\mathbf{h}_0, \mathbf{h}_x)$ as:

$$\sum_{l=0}^{L-1} h_0[l]h_x[l]^* \approx \sum_{l=0}^{L-1} |\zeta_l|^2 e^{j2\pi f_c (\tau_{x,l} - \tau_{0,l})}. \quad (4.5)$$

Eq. (4.5) does not indicate a direct relationship between the distance moved, x and the drop/decay in the TRRS value. To make this correspondence more evident, we simplify the equation by exploiting the following observations.

Path length difference: The change in the path length when the hand moves by a small distance x can be calculated using the angle of reflection v_l and the angle between the surface normal and the moving direction, ψ_l as indicated in Fig. 4.2. F_1 and F_2 are the Tx and Rx locations, which can be considered an ellipse's foci. By the geometric definition of the ellipse, a small displacement along the surface does not impact the total length of the path. Therefore, the change in the total path length depends only on the displacement along the normal to the reflecting surface/ellipse. The difference in the delays of a path l when the hand moves a distance x can be written as:

$$\tau_{x,l} - \tau_{0,l} = \frac{2x \cos \psi_l \cos v_l}{c}. \quad (4.6)$$

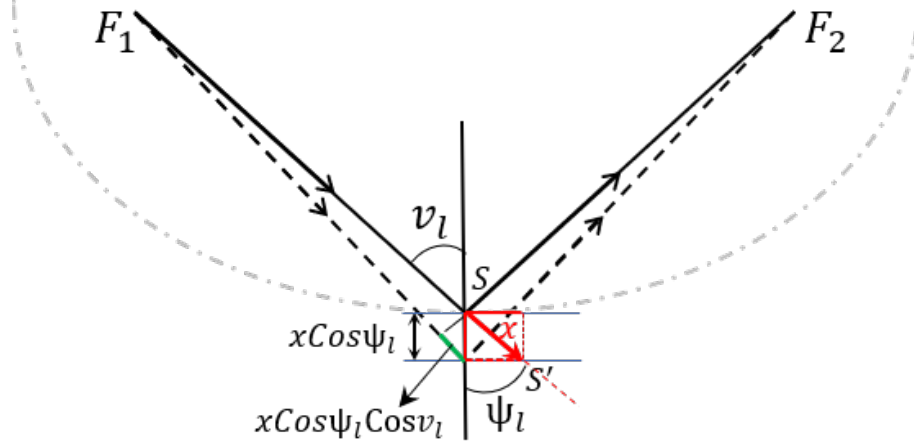


Figure 4.2: Demonstration of the multipath length difference when a reflector moves by a small distance x .

Uniform scattering: In a rich indoor environment with a lot of multipath, we can assume that the MPCs are uniformly distributed in the space. This allows us to model the angle of incidence of the MPCs at any point in space to be uniformly distributed in $(0, 2\pi]$, converting the summation in (4.5) to integration over the angles as,

$$\sum_{l=0}^{L-1} h_0[l] h_x[l]^* \approx K \int_{v_l=0}^{2\pi} \int_{\psi_l=0}^{2\pi} e^{j2\pi f_c \frac{2x \cos \psi_l \cos v_l}{c}} dv_l d\psi_l, \quad (4.7)$$

where K is the normalization constant.

Fraction of multipaths: Amongst the dense multipath, only a fraction of the MPCs are influenced by the hand's movement. If we denote the fraction of the multipath consisting of reflections from the static environment by T_S , the RHS of (4.7) can be written as,

$$K \int_{v_l=0}^{2\pi} \int_{\psi_l=0}^{2\pi} T_S + (1 - T_S) e^{j2\pi f_c \frac{2x \cos \psi_l \cos v_l}{c}} dv_l d\psi_l \quad (4.8)$$

$$= K T_S (2\pi)^2 + K (1 - T_S) (2\pi)^2 J_0^2 \left(\frac{2\pi f_c x}{c} \right), \quad (4.9)$$

where $J_0(\cdot)$ is a zeroth-order Bessel function of the first kind [66].

Stretched hand gesture: During a hand gesture, different parts of the hand move at different speeds and in different directions. Modeling an arbitrary hand movement is, therefore, complicated. In this work, we propose a unique way of performing a gesture with a hand, as shown in Fig. 4.1 with the arm stretched and pivoted about the shoulder, O . The implications of such a constrained movement are three-fold:

- The spatial orientation of all the parts of the arm can be determined from the wrist's location relative to the shoulder.
- The locus of the moving scatterers (different parts of the hand) is unique and does not overlap/intersect with each other. In a gesture, the only way for a part of the arm to repeat a spatial location is when the whole arm repeats its location.
- When the hand is moved per Fig. 4.1, the hand can be modeled as a combination of multiple scatterers whose displacement increases linearly from 0 to x , where x is the displacement of the wrist. Observe that if the wrist is displaced by x , then a point on the arm at a fractional length pl from the shoulder is displaced by px . This results in a summation term simplifying RHS of (4.9) to

$$KT_S(2\pi)^2 + \frac{K(1 - T_S)(2\pi)^2}{N_D} \sum_{i=1}^{N_D} J_0^2\left(\frac{2\pi f_c x i}{c N_D}\right), \quad (4.10)$$

where $J_0(\cdot)$ is a zeroth-order Bessel function of the first kind [66].

The denominator of (4.2) becomes,

$$\sqrt{\sum_{l=0}^{L-1} |h_1[l]|^2} \sqrt{\sum_{l=0}^{L-1} |h_2[l]|^2} = K \int_{v_l=0}^{2\pi} \int_{\psi_l=0}^{2\pi} 1 dv_l d\psi_l. \quad (4.11)$$

Using (4.10) and (4.11), the TRRS between $\mathbf{h}_0$ and $\mathbf{h}_x$ can be written as,

$$\eta(\mathbf{h}_o, \mathbf{h}_x) = T_S + \frac{(1 - T_S)}{N_D} \sum_{i=1}^{N_D} J_o^2\left(\frac{2\pi f_c x_i}{c N_D}\right). \quad (4.12)$$

Eq.(4.12) specifies a relation between the TRRS decay and the distance moved by the wrist with a stretched arm. For $N_D = 50$, the RHS is plotted and is as shown in Fig. 4.3 indicating a monotonous decay. We exploit this monotonous decay to develop a likelihood-based approach for gesture classification in the next section.

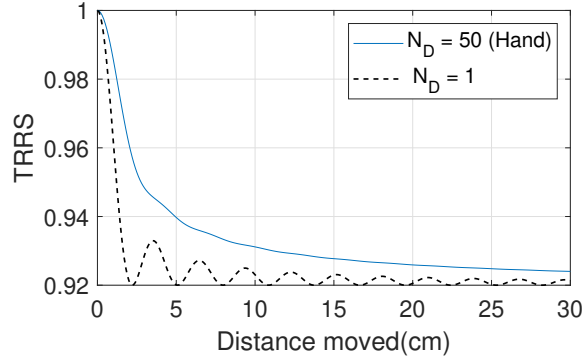


Figure 4.3: TRRS decay in case of hand movement (dynamic scatterers with linearly increasing speed) and a cart movement (dynamic scatterers with one speed).

4.2 GWrite Algorithm

This section designs a pipeline to extract unique features for each gesture shape and use them for classifying gestures. We consider gestures consisting of

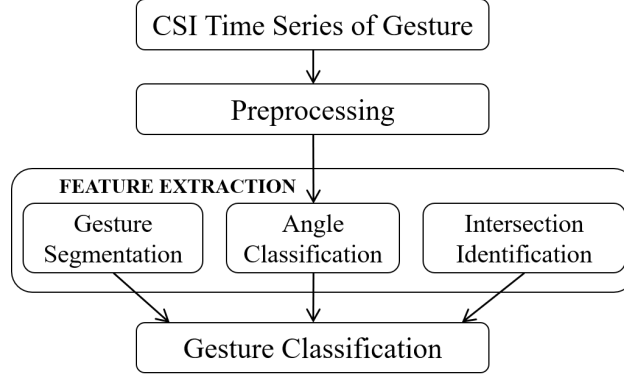


Figure 4.4: Overview of *GWrite* algorithm.

only straight-line segments in this work. It is sufficient to know the number of segments, the angle between two adjacent segments, and the locations of the intersection points between line segments, if any, to describe any shape composed of straight-line segments. An overview of *GWrite* module is shown in Fig. 4.4. Since we do not have the information about the absolute/relative direction of motion of the hand, we focus only on the relative shape of the trajectory. Thus, the features for the trajectories of letters ‘N’ and ‘Z’ will be the same with three line segments, acute angles between adjacent segments, and no intersection points. Similarly, the trajectory features for ‘M’ and ‘W’ will be the same. Sample trajectories from the upper case English alphabets with unique features are shown in Fig. 4.5. The *GWrite* prototype proposed in this work will demonstrate gesture classification on this set of 15 gesture shapes.

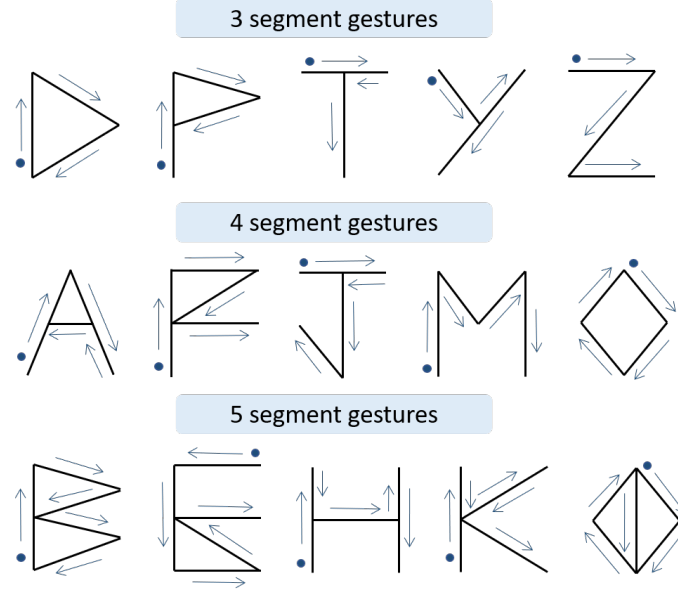


Figure 4.5: Gesture trajectories of upper case English alphabets with different number of segments. The circular dot indicates the initial point for each shape.

4.2.1 Preprocessing

The *GWrite* prototype is built on a WiFi chipset equipped with 2x2 MIMO. In the preprocessing step, we calculate the TRRS similarity between all pairs of CSIs and generate the TRRS matrix for each link. We then combine the matrices from the four links using Maximal-Ratio Combining (MRC), and the resultant matrix will be used as an input for the feature extraction module.

For each link l , the TRRS matrix is denoted by TR^l and the $(i, j)^{\text{th}}$ entry is the TRRS similarity between i^{th} and j^{th} CSI in the time series. For two CSIs H_i and H_j , $TR^l(i, j)$ is defined in the frequency domain as [13],

$$TR^l(i, j) = \frac{|\sum_{k \in \mathcal{V}} H_i[k] H_j[k]^*|}{\sqrt{\sum_k |H_i[k]|^2 \sum_k |H_j[k]|^2}}, \quad (4.13)$$

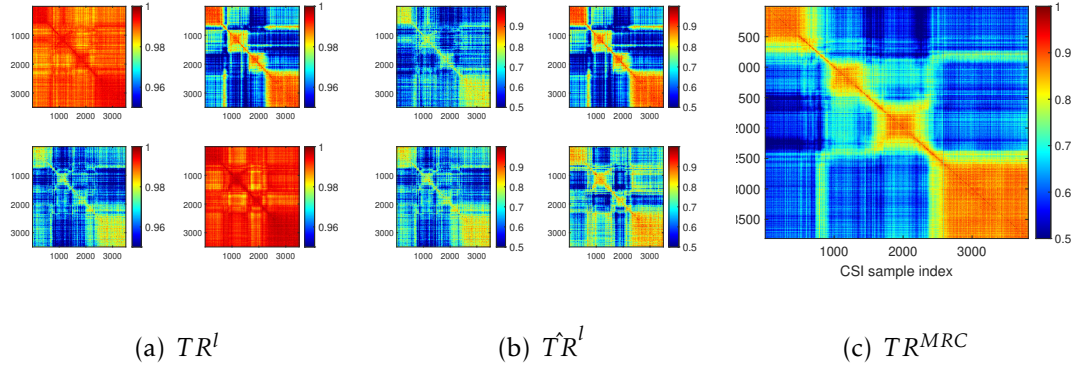


Figure 4.6: Illustration of MRC of four TRRS links, $l = 1, 2, 3, 4$. (a) Raw TRRS matrices, (b) TRRS matrices after normalizing the static offset, and (c) TRRS matrix after MRC.

where ν is the set of subcarriers and $(.)^*$ is the complex conjugate operation. One realization of TR^l with $l = 1, 2, 3, 4$ is shown in Fig. 4.6(a). It can be observed that different links present information of different significance due to the spatial diversity presented by the MIMO system. We combine the information from different links using the MRC technique by weighing them according to their signal and noise levels. From (4.9), T_S denotes the contribution of the static scatterers to the TRRS value. Different links could have different levels of T_S based on their multipath propagation. As T_S decreases, we observe a larger change in the TRRS for the same degree of hand movement. By normalizing the contribution from the static environment, we equalize the signal level on different links. The normalized matrices will be denoted by $\hat{TR}^l$ (Fig. 4.6(b)) and defined as

$$\hat{TR}^l = \frac{TR^l - T_S}{1 - T_S}. \quad (4.14)$$

When the speed of the hand is zero at the location of the turns, the TRRS between CSIs should be equal to 1. Due to the environmental factors (noise due to

ambient motion) and human factors (minor motions in other body parts/hand), the $TRRS < 1$. The offset of the average $TRRS$ value at these zero speed instances from 1 indicates the noise level. The MRC weights are taken as inversely proportional to the square of the noise level. The combined $TRRS$, TR^{MRC} (Fig. 4.6(c)) is then given by,

$$TR^{MRC} = \frac{\sum_{l=1}^4 \frac{\hat{TR}(l)}{(1-TR_l^{\max})^2}}{\sum_{l=1}^4 \frac{1}{(1-TR_l^{\max})^2}}. \quad (4.15)$$

TR^{MRC} shows greater variation in its values for a hand gesture with a reduced noise level and will be further analyzed by the feature extraction module.

4.2.2 Feature Extraction

This module extracts the features characteristic of the gesture trajectory which will be later used for gesture classification.

4.2.2.1 Gesture Segmentation

Each straight line in a gesture shape is termed a segment. For example, different gesture shapes are shown in Fig. 4.5 grouped based on the number of segments. Gesture segmentation aims to determine the number of segments in a given gesture shape.

During a gesture, the speed is minimal when the hand changes the direction of motion, i.e., at the turns. Assuming the user performs the gesture with reduced/zero speed during turns, the number of turns can be inferred from the number of instances at which the speed is very low/zero. We leverage the motion

statistics proposed in [91] to achieve this goal.

Motion statistics represent the extent of motion/disturbance in the wireless channel as perceived by the transceivers. As a result, a small dynamic object close to a transceiver and a larger dynamic object farther away from the transceiver could cause a similar amount of disturbance and result in the same motion statistics. Similarly, a small object in rapid motion and a larger object with slower motion could have the same motion statistics. Motion statistics are calculated from the correlation of the CSIs as detailed in [91]. The theoretical values lie between -1 and 1, with 1 indicating the highest motion/disturbance level.

During a gesture, the person is at a fixed distance from the transceivers. Therefore, the motion statistics can capture the relative rapidness of the hand motion. Consider a ‘D’ shaped gesture shown in Fig. 4.7(a) and the corresponding motion statistics shown in Fig. 4.7(b). Observe that the motion statistics are lower at the location of the turns as marked in circles in Fig. 4.7. The number of straight-line segments in a gesture can be determined from the number of local minima/valleys in the motion statistics minus 1.

4.2.2.2 Turn Angle Classification

A sudden change in the direction of movement of the hand is referred to as a “turn” event. For instance, Fig. 4.8(a) shows a gesture trajectory beginning at the Origin (‘O’), following a straight line till a “turn” occurs at the point ‘T.’ The trajectory after the point ‘T’ is again a straight line till it reaches the endpoint,

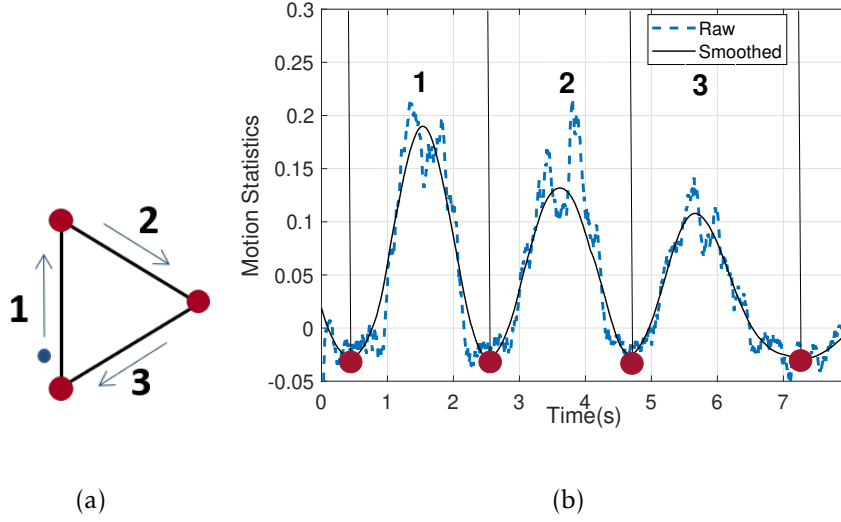


Figure 4.7: Illustration of Gesture segmentation. (a) ‘D’-shaped gesture trajectory, and (b) the corresponding motion statistics along with the different gesture segments.

‘F.’ The angle between the two straight line segments is referred to as the turn angle and is denoted by θ . We classify the turn angle into three groups/classes, namely, (i) $\theta \approx 0^\circ$, (ii) $\theta \approx 45^\circ$, and (iii) $\theta \approx 90^\circ$.

Consider any point ‘S’ on the second segment of the trajectory in Fig. 4.8(a). Point ‘P’ is the closest point to ‘S’ on the first segment. If we compute the TRRS similarity between the CSI at the point ‘S’ and all the CSIs along the trajectory, moving backward, we observe a valley for the CSI at the point ‘T’ and a peak for the CSI at the point ‘P’ as shown in Fig. 4.8(b). Let us call this plot as S_Δ for any point ‘S’ on the second segment. The direct correspondence between the relative distance and the TRRS is a result of the monotonous decay pattern derived in Section. 4.1. Leveraging the relative pairwise TRRS relationships between ‘S’, ‘P’ and ‘T’, we define two features, r and f , and use them to assign probability scores

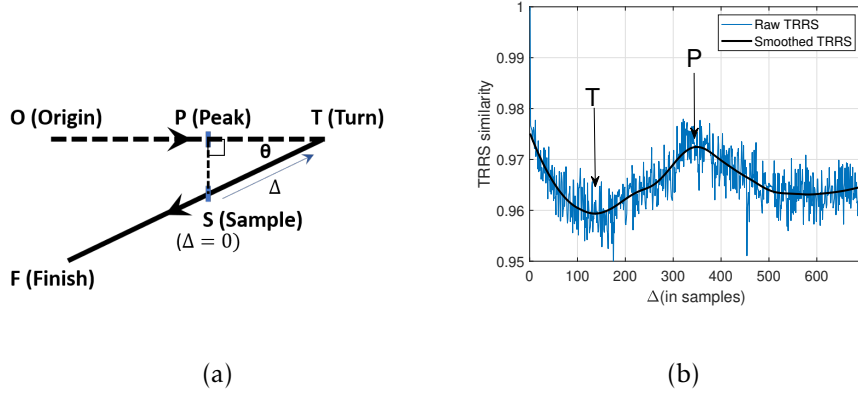


Figure 4.8: (a) Geometry of a turn in a gesture (b) TRRS decay feature

for different θ .

1. f : Fraction of the second segment from the turn location for which a peak can be observed on the S_{Δ} plot. If 'S' is the farthest point on the second segment from 'T' for which a peak is observed on the S_{Δ} plot, then f is defined by:

$$f = \frac{\sum_{i \in (T,S)} \text{Motion Statistics}(i)}{\sum_{i \in (T,F)} \text{Motion Statistics}(i)}. \quad (4.16)$$

As the monotonous decay of the TRRS curve is observed only upto a certain distance, a peak may not be observed if the distance SP is beyond a certain limit. Although this limit is not universal, we can roughly obtain a range of values of f for different θ . In this work, $f \leq 0.2$ is assigned to $\theta \approx 90^\circ$, and f should be equal to 1 for $\theta \approx 0^\circ$.

2. r : For each point 'S' on the second segment, the corresponding point 'T' and 'P' can be obtained from the S_{Δ} curve. The curve $y = SP$ is obtained from the TRRS (S,P) for all the pairs of (S,P) where S lies on the second segment. For a given turn angle θ , the parameter r is defined as the ratio

of areas between the curves $y = T_{max} - SP$ and $y = T_{max} - ST$ where T_{max} is the highest TRRS between points on the two segments. Ideally $T_{max} \approx 1$ as the TRRS between points close to the point 'T' lying on different segments is close to 1. If $\theta = 45^\circ$, $SP = PT$ and assuming a uniform environment, $TRRS(S,P) \approx TRRS(P,T)$ resulting in $r \approx 1$. For $\theta = 0^\circ$, $TRRS(S,P) \approx T_{max}$ resulting in $r \approx 0$.

Table 4.1: Theoretical values of f and r for different θ .

Turn angle	r	f
$\theta \approx 0^\circ$	0	1
$\theta \approx 45^\circ$	1	1
$\theta \approx 90^\circ$	Undefined	0

The ideal values of the parameters f and r are shown in the Table-4.1. However, in reality, due to the following non-idealities, the observed values differ from their ideal values.

- The monotonous decay of the TRRS is not evident over a large distance. Therefore, for higher acute angles, the peaks may not be detected for the entire length of the second segment and $f < 1$.
- $TRRS(SP)$ is not always equal to 1 for $\theta \approx 0^\circ$. This is because the human body and hand are not rigid objects and the CSI may not match exactly even though the person intends to trace the same path. Therefore, $r > 0$ for $\theta \approx 0^\circ$.

- Since there is no reference when drawing gestures in the air, it may be difficult for the user to make a precise 45° turn. Therefore, the value of r varies from 0.5 to greater than 1 for an acute angle turn.

To understand the distribution of the parameter in a practical setup, we gathered several data samples and calculated r . The empirical distribution of r is shown in Fig. 4.9. Although there is no single ideal value for the $\theta \approx 45^\circ$ class, the value

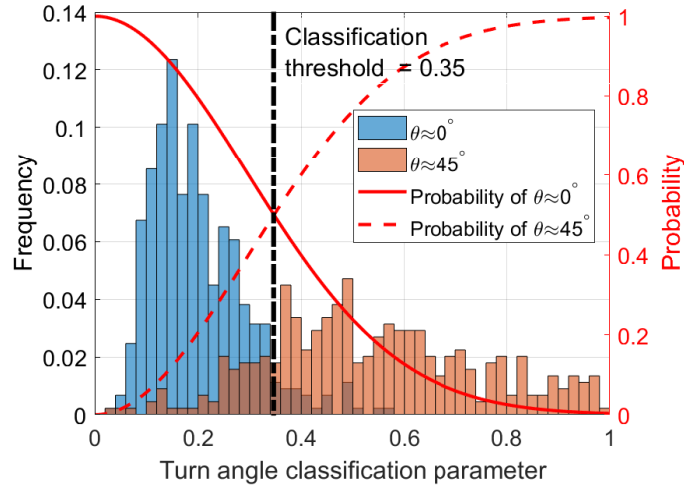


Figure 4.9: Distribution of the parameter r for $\theta \approx 0^\circ$ and $\theta \approx 45^\circ$.

of r for $\theta \approx 0^\circ$ is ideally equal to 0. The histogram contains 400 samples for each class, and the classification threshold is determined as the one which achieves the highest classification accuracy. The probability curve for $\theta \approx 0^\circ$ is defined as the half-normal density centered at $r = 0$ such that the value is 0.5 at the threshold ($= 0.35$), as shown in Fig. 4.9. We use a data-driven approach here to understand the distribution of the parameter r and design the probability score accordingly. Although the probability values are not accurate for each realization and shape, the relative values will enable us to do the gesture classification later. In the

extreme case of r being very high, the probability of $\theta \approx 45^\circ$ is one.

4.2.2.3 Intersection Point Detection

Identifying and locating intersections between two line segments can be a useful feature to differentiate gesture shapes. For instance, the location of intersection in 'D' shape and 'P' shaped gestures can be denoted by $(f_1, f_3) = (0, 1)$ and $(f_1, f_3) = (0.5, 1)$ respectively, where $f_s \in [0, 1]$ denotes the fractional distance of the intersection point from the start point of the segment s . If an intersection/crossing occurs in a gesture trajectory, it implies that the spatial orientation/location of the arm is almost the same at those two time instances, resulting in a relatively high TRRS value between the two CSIs (ideally equal to 1). However, occurrence and the location of the intersection point are not the only deciding factors for gesture classification. Defining a probability score could be more informative for the overall classification rather than a binary decision on the intersection. There exists two main challenges ahead.

- It is evident that if an intersection occurs, it will be the location/instance corresponding to the maximum TRRS among all the pair-wise TRRS similarities between the CSIs from the two segments. However, we do not know the absolute distance moved between any two time instances, which makes the estimation of f_s non-trivial.
- Although the value of TRRS similarity at an intersection point is ideally equal to 1, the environmental and human factors decrease the value to less

than 1. The influence of these factors being unpredictable and difficult to model, makes the probability score assignment more involved.

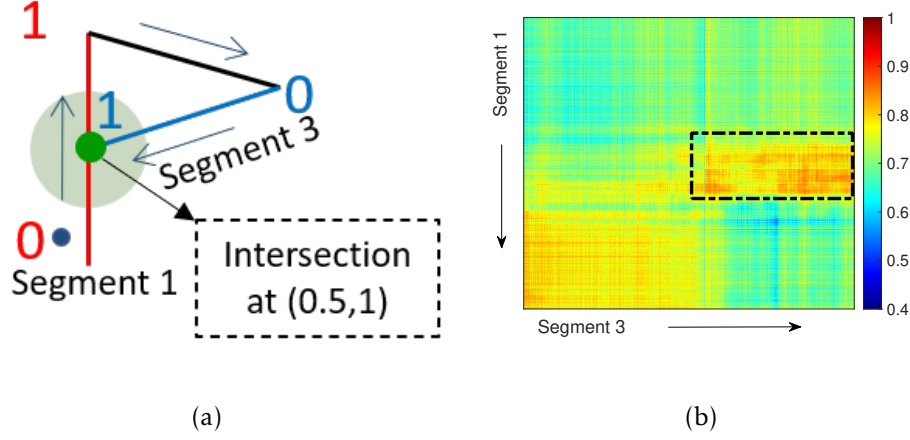


Figure 4.10: (a) Illustration of the intersection point detection, (b) TRRS between CSIs of segment 1 and segment 3.

Observe that we only need a relative location of the intersection point on each of the two segments under consideration. Given that each segment follows a straight-line path, the distance moved between any two instances can be estimated by accumulating the distance traveled between every two successive time instances between them. Recall that motion statistics were used in Section. 4.2.2.1 to determine the relative speed of the gesture at every instance. Assuming that the CSIs are equally spaced in time, we can use the motion statistics as an approximate indicator of the relative distance moved, by accumulating the motion statistics between any two time instances. The approximation holds because a similar degree in the movement will cause similar motion statistics within a single gesture realization. We assume that the total length of each segment is equal to 1 and the distance between any two points X and Y on a segment AB is

given by,

$$\frac{\sum_{i \in (X,Y)} \text{MotionStatistics}(i)}{\sum_{i \in (A,B)} \text{MotionStatistics}(i)} \quad (4.17)$$

Consider an P-shaped gesture shown in Fig. 4.10. The intersection point between segment one and segment three is $(f_1, f_3) = (0.5, 1)$ and is marked by the darker circle. In reality, since the user does not have any reference of the spatial trajectory drawn till then, it is difficult to cross segments at the desired location. We, therefore, allow a margin of 0.25 as indicated by the extended circular region in Fig. 4.10. If the intersection point lies within the margin of 0.25 of the desired location on both the segments, we estimate a probability score for an intersection point.

We use a similar approach to that in 4.2.2.2 for the probability score assignment. We designed a heuristic $T_N = \frac{T_{\max} - T_{\min}}{1 - T_{\min}}$, where $T_{\max}$ and $T_{\min}$ are the maximum and minimum values of TRRS between the two CSIs from the two segments. Fig. 4.11 shows a histogram of T_N values for different pairs of segments with and without an intersection. The value of T_N for an intersection point is ideally equal to 1. The histogram contains 600 samples for each class, and the classification threshold is the one with maximum classification accuracy. The probability curve for the ‘intersection point’ class is defined as the half-normal density centered at 1 such that the probability score value is 0.5 at threshold ($= 0.7$) as shown in Fig. 4.11. As mentioned before, the probability values may not be accurate for each instance but the relative values are useful for the classification. In the extreme case of T_N being very low, i.e., if the two segments are far

apart, then the probability of an intersection point is equal to 0.

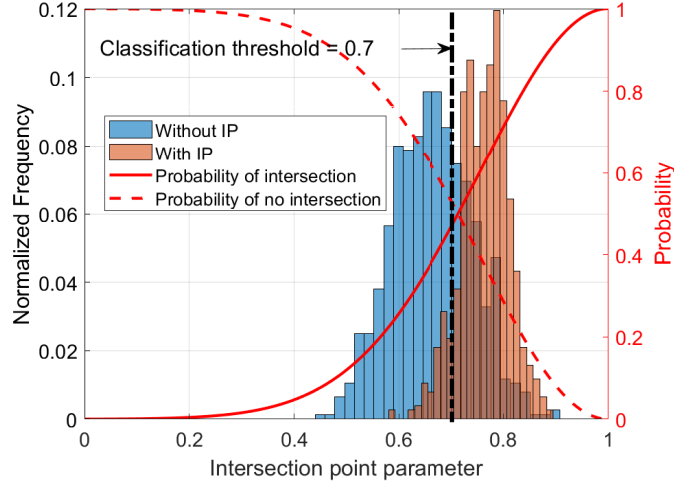


Figure 4.11: Histogram of T_N values with and without an intersection and the corresponding probability score assignment.

4.2.3 Gesture Classification

Gestures can be classified using the probability scores of the features calculated in the feature extraction module. First, the gesture group is decided based on the number of segments. The probability scores for each character within the group are then computed as follows:

$$P(\Omega) = \frac{\alpha}{N_s - 1} \sum_{a=1}^{N_s-1} P(\theta_a^\Omega) + \frac{(1-\alpha)}{N_s-1} C_2 \sum_{i=1}^{N_s-1} \sum_{j=i+2}^{N_s} P(\{f_i, f_j\}^\Omega), \quad (4.18)$$

where the actual values of the features for the trajectory shape Ω are given by θ_i^Ω and $\{f_i, f_j\}^\Omega$. For example, $\{\theta_1^P, \theta_2^P, \{f_1, f_3\}^P\} = \{45^\circ, 45^\circ, \{0.5, 1\}\}$. The parameter α is a hyperparameter that can be tuned to weight the contribution of the angle classification and intersection point detection modules. Table. 4.2 shows

Table 4.2: Probability scores for characters with 3 segments.

	$\theta_i \approx 0^\circ$	$\theta_i \approx 45^\circ$	Ω	Probability score
			D	0.42
$i = 1$	0	1	P	0.82
			T	0.17
$i = 2$	0.31	0.69	Y	0.32
			Z	0.42

an example of the calculated probability scores for a P-shaped gesture, where $\alpha = 0.5$ and $P(\{(f_1, f_3)\}) = 0.8$. The probability score for $\Omega = P$ is the highest (0.82) amongst others and the gesture shape can therefore be classified as a P-shape.

4.3 Performance Evaluation

To evaluate the performance, we built a prototype of *GWrite* on off-the-shelf commercial WiFi chipsets with a bandwidth of 80 MHz, 2×2 MIMO using a sampling rate of 350 Hz. A typical through-the-wall setup is as shown in Fig. 4.12. The transmitter (Tx) and the receiver (Rx) are placed in different rooms. Gestures are performed in a different room with at least one wall between each transceiver and the gesture location.

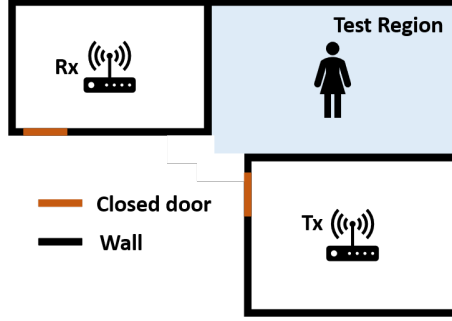


Figure 4.12: A typical through-the-wall experimental setup.

4.3.1 Performance of Feature Extraction

4.3.1.1 Gesture Segmentation Accuracy

The gesture segmentation module accurately estimates the number of segments in a gesture trajectory. We have collected 100 samples of each of the gestures with 3, 4 and 5 segments for evaluation. The number of segments was correctly identified 100% of the time. Since the gesture segmentation module can accurately estimate the number of segments in a gesture trajectory, in the gesture classification accuracy evaluation, we evaluate classifying gestures with the same number of segments.

4.3.1.2 Turn Angle Classification Accuracy

The angle classification module discussed in section 4.2.2.2 is evaluated using about 400 samples each with $\theta \approx 0^\circ$ and $\theta \approx 45^\circ$. The confusion matrix of the classification is shown in Fig. 4.13(a). The average classification accuracy is 90.4%.



Figure 4.13: Confusion matrices for (a) angle classification and (b) intersection point detection.

4.3.1.3 Intersection Point Detection Accuracy

The intersection point detection module discussed in 4.2.2.3 is evaluated with about 600 realizations each with and without an intersection point. The confusion matrix of the classification is as shown in Fig. 4.13(b) with an overall classification accuracy of 81.1%.

4.3.2 Gesture Classification Accuracy

For evaluating the overall performance of the gesture classification algorithm, we group sets of uppercase English characters with the same number of segments and perform gesture classification among them. This is because the gesture segmentation is highly accurate and we found zero misclassification based on the number of segments. We divide the characters with unique gesture shapes into three groups as shown in Fig. 4.5 into,

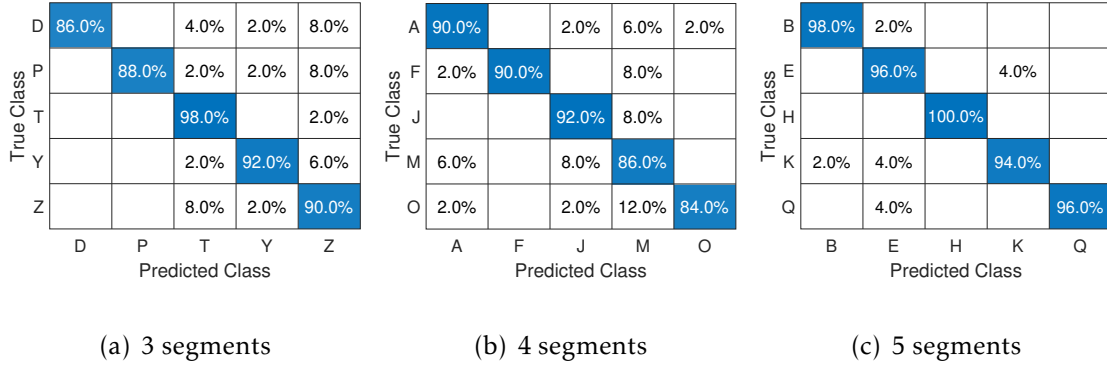


Figure 4.14: Confusion matrices for classification of gestures with different number of segments.

- 3 segments: D, P, T, Y and Z shapes.
- 4 segments: A, F, J, M and O shapes.
- 5 segments: B, E, H, K and Q shapes.

The confusion matrix for each group is as shown in Fig. 4.14. The average classification accuracies for 3, 4 and 5 segment groups are 90.8%, 88.4% , and 96.8% respectively. The higher classification accuracy for 5 segment gestures is because of more differentiating features which decreases the probability of misclassification. The average classification accuracy for all the 15 gesture is 92%.

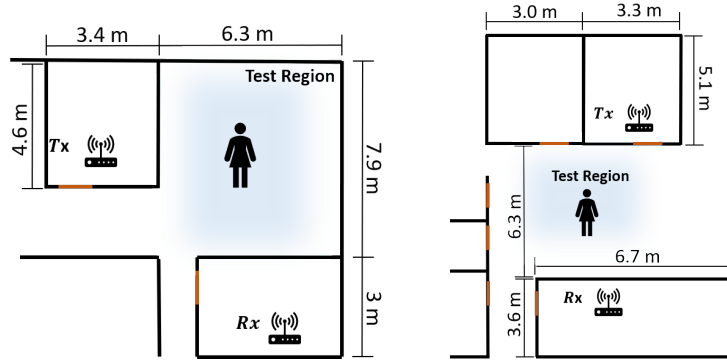
4.4 Discussions

In the previous section, we have studied the performance of *GWrite* in the default scenario. Below, we discuss the factors that could influence the performance of *GWrite*.

4.4.1 Placement and Coverage

The transceiver placement is flexible with *GWrite* as the features for classification are characteristic of the trajectory shape rather than the geometry of the placement. While the exact placement is irrelevant, there is a specific operation region in which *GWrite* can work best. If the transceivers are placed far away from each other, then the channel perturbation caused due to the movement of the hand may not be perceived. On the other hand, if the devices are placed very close to each other, then the uniform scattering assumption may not hold good, and the performance might degrade. We show few experimental setups in Fig. 4.15 suitable for *GWrite*.

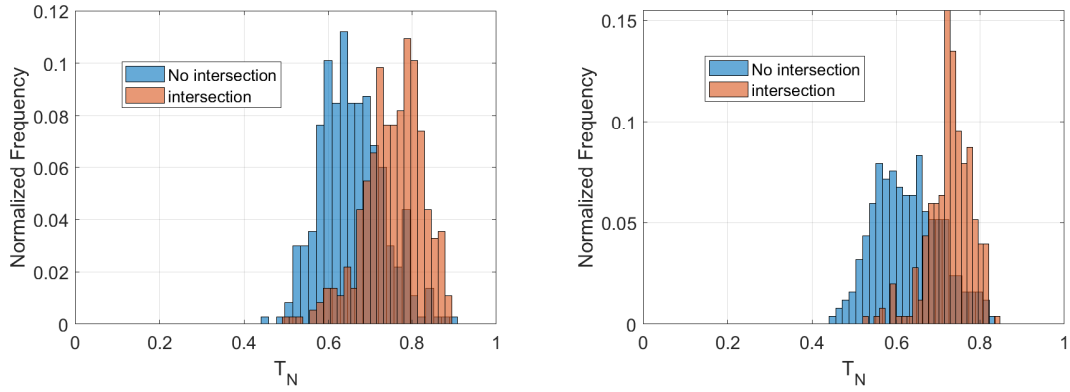
The term T_S in (4.12) denotes the extent of the contribution from the static environment to the TRRS and is dependent on the relative location of the transceivers, gesture and the environment. We compensate for T_S while combining the TRRS from different links in the preprocessing stage. As a result, the features for angle classification and intersection detection are independent of the location/distance between the transceivers within the region of operation. Fig. 4.16 shows the histograms of T_N for two locations. It can be observed that the classification thresholds are similar in both the cases based on which we design the probability score curves. Location independence is one of the crucial advantages provided by *GWrite*.



(a) Location 1

(b) Location 2

Figure 4.15: Examples of device placement.



(a) Location 1

(b) Location 2

Figure 4.16: Histograms of T_N for two different device placements.

4.4.2 Weightage of Features (α)

The parameter α in (4.18) can be adjusted to give different weightage to the angle classification and the intersection point modules. The optimal value depends on the set of gestures to be classified. For example, in the extreme case of using a gesture set consisting only of shapes with acute angles, α can be set to 0, in which case the classification is purely based on the intersection detection

module. The average classification accuracy of 3 segment gestures for different α is shown in Fig. 4.17.

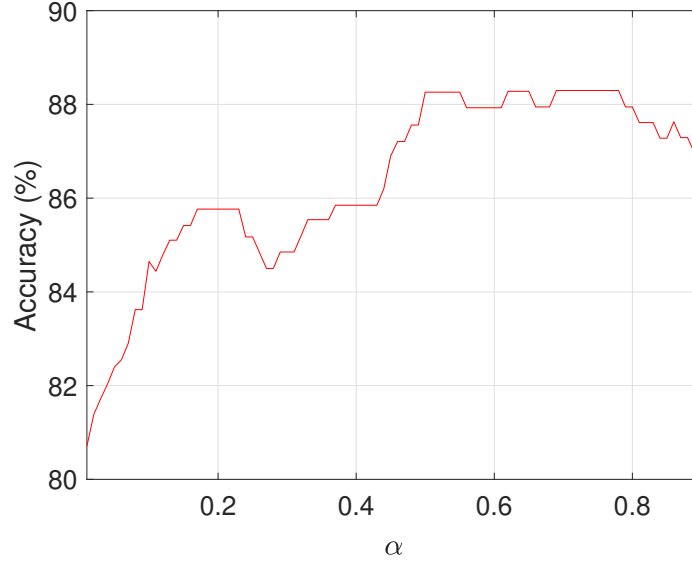


Figure 4.17: Classification accuracy with α .

4.4.3 Generalized Gestures

We evaluated the performance of *GWrite* on a set of 15 upper case English characters. However, this set can be expanded to include generalized shapes with straight lines, such as in Fig. 4.18. *GWrite* can classify gestures composed of straight-line segments that differ in terms of the turn angles and the intersection points. More the number of such differentiating features, the higher is the classification accuracy.

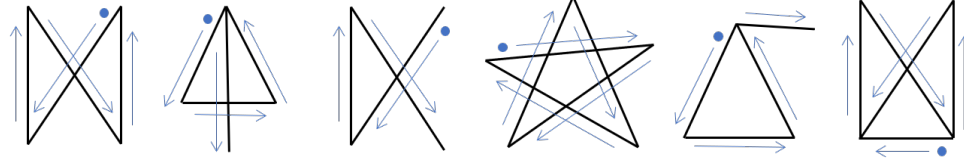


Figure 4.18: Examples of *GWrite* gesture shapes.

4.5 Summary

In this Chapter, we propose *GWrite*, a gesture recognition system that can work in a through-the-wall setting and recognize gesture shapes composed of straight-line segments. We have developed a complete pipeline of such a system, including gesture segmentation, turn angle classification, and intersection detection. We proposed a unique model for the hand movement during a gesture. Leveraging the model, we derived a relation between the perturbations in the physical environment and the behavior of the similarity of the CSIs in a through-the-wall setting. We believe such an understanding will open up diverse opportunities to develop practical gesture recognition systems and expand the existing WiFi-based systems restricted only to LOS setups. We built a prototype of the proposed system using COTS WiFi devices and evaluated the performance using a set of 15 upper case English characters. The classification accuracy achieved was 92% which can be further improved as more bandwidth becomes available.

Chapter 5: Handwriting Tracking

Rapid increase in automation has stimulated the quest for more efficient and convenient approaches for Human-Computer Interaction (HCI). Touch screens and smart surfaces (e.g., electronic whiteboards) have emerged as a more user-friendly alternative to the traditional input devices such as the keyboard and the computer mouse. However, the smart surfaces are usually in small form factors limiting the space for HCI and hindering our vision of ubiquitous smart environments. Instead, we see an increasing trend of ubiquitous interactions with machines via in-the-air gestures, handwriting, and voice-controllable systems, evidenced by recent industrial efforts, including Google Soli [39], Apple UWB radar, Apple Siri, Amazon Alexa, Google Home, and so on. Handwriting is a convenient mode of interaction for people and can be considered as a more general form of a gesture, which is a promising approach for HCI. Aided by the advanced handwriting recognition systems [8, 21, 65], enabling robust and accurate handwriting tracking can realize countless applications in the field of HCI.

In this chapter, we present *mmWrite*, the first passive motion tracking system that can recover the trajectory of handwritten traces with high precision us-



Figure 5.1: Illustration of handwriting tracking using the proposed *mmWrite* system. (a) Picture of the handwritten trajectory and (b) Reconstructed trajectory of the word “mmWrite”.

ing a single commodity 60GHz mmWave radio. *mmWrite* captures the signals reflected off a writing object (e.g., a finger or a pen) and employs a pipeline of signal processing to reproduce the moving trajectory. And unlike gesture recognition systems, *mmWrite* does not require any calibration or supervised learning. Instead, it directly tracks the location of the writing target and recovers the entire trace. With one single radio and without instrumenting the writing object, *mmWrite* achieves handwriting tracking with millimeter accuracy over a large area of up to 8 m^2 , thus converting a conventional surface into an interactive trackpad with minimal supporting infrastructure. We have shown one instance of handwriting tracking in Fig. 5.1.

5.1 Handwriting Tracking Algorithm

This section describes the handwriting tracking algorithm of *mmWrite* in detail. The algorithm aims to recover the handwriting trajectory from the CIR time series recorded by the radar while writing. The CIR time series is processed

in different modules, which are shown in Fig. 5.2. First, background subtraction is performed to reduce the contribution of the static objects in the environment to the received signal. Next, digital beamforming is used to obtain the spatial information followed by the transformation to the Doppler domain, aiming to differentiate the moving/writing object from other static objects in the environment. The target is detected using the 3D-CFAR and clutter mapping techniques, which is further finely localized within the detected spatial bins using the subsample interpolation technique. In the final step, the trajectory points corresponding to the target of interest are gathered and combined to form a raw trajectory. The raw trajectory is further smoothed using the DCT-based smoothing technique to obtain the *mmWrite* handwriting trajectory.

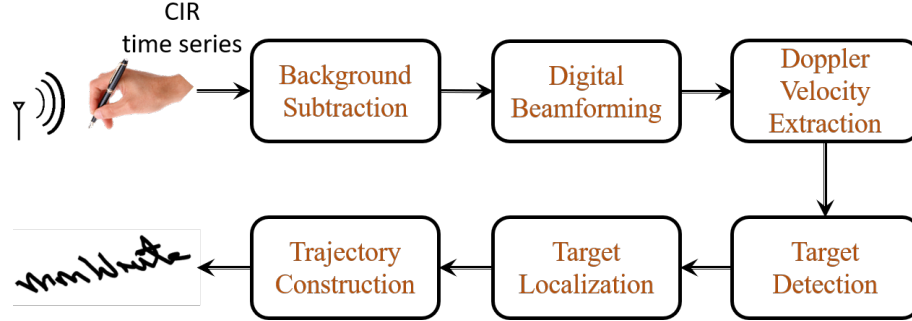


Figure 5.2: Overview of *mmWrite*.

5.1.1 Background Subtraction

Let the CIR recorded by the r^{th} Rx at time instance t be denoted by $h[r, n, t]$, where n is the tap index of the CIR. To highlight the target of interest (i.e., the writing object) and reduce the contribution from other static objects in the envi-

ronment, the background CIR is subtracted from $\mathbf{h}$. In this work, the background CIR is estimated by taking the average of the CIR time series along the *slow-time* dimension. However, when designing a causal system or when the duration of handwriting is short, a calibration step may be included to record the background without the writing object. The background subtraction can be mathematically written as:

$$\hat{h}[r, n, t] = h[r, n, t] - \overline{h[r, n, .]}, \quad (5.1)$$

where $\overline{\mathbf{h}}$ is the average CIR over time and $\hat{\mathbf{h}}$ is the CIR after background subtraction. In the following steps, the notation $\mathbf{h}$ will be used instead of $\hat{\mathbf{h}}$ to denote the CIR after background subtraction. The tap index n corresponds to the *fast-time* and contains the range information of the reflecting object. However, there could be multiple signals from several directions at the same range, making the target identification ambiguous. The received signal is spatially filtered to address this, thereby extracting the direction information by leveraging the phased antenna array structure. In the next stage of *mmWrite*, digital beamforming is performed to extract the spatial information.

5.1.2 Digital Beamforming

As we have the raw CIR at each receiving antenna, we can perform digital beamforming to separate the target-reflected signals from other irrelevant reflection signals. In this work, classical beamforming is used [68]. The steering vector used for beamforming was measured by Qualcomm and is available with the

device. The Fraunhofer distance, which determines the near-field limit, is calculated as 18 cm. Hence, in this work, a range greater than 20 cm can be considered a far-field.

Let K reflected signals be incident on the receiving antenna array and let $\mathbf{g}_k$ denote the k^{th} incoming reflected signal. Let the azimuth and elevation angles corresponding to $\mathbf{g}_k$ be represented by az_k and el_k respectively. If the measured antenna response of the receiver r is denoted by $\mathbf{a}_r$, then the combined received signal $\mathbf{h}$ at the receiver antenna can be modeled as:

$$h[r, n, t] = \sum_{k=1}^K a_r(az_k, el_k) g_k[az_k, el_k, n, t], \quad (5.2)$$

where n is the CIR tap index and t is the time instance. The beamformed CIR, $\mathbf{y}$ is a four-dimensional matrix obtained by compensating for the array response as:

$$y[az, el, n, t] = \mathbf{a}^H(az, el) h(., n, t), \quad (5.3)$$

where $\mathbf{a}(az, el)$ is the 32-dimensional vector of antenna responses of all the receiver antennas for a signal arriving at an angle (az, el) and $(.)^H$ is the Hermitian operator. To simplify the notation, a vector $\mathbf{T}$ is used to denote the triplet (az, el, n) as follows:

$$h[r, n, t] \xrightarrow{\text{beamforming}} y[\underbrace{az, el, n}_{\mathbf{T}}, t] = y[\mathbf{T}, t]. \quad (5.4)$$

One instance of the beamformed CIR is shown in Fig. 5.3(a). The variation of the absolute value of $\mathbf{y}$ at a range tap of 8, corresponding to a range of 34 cm, is shown with different azimuth and elevation angles.

At this stage, the range and direction of the reflected signals can be deter-

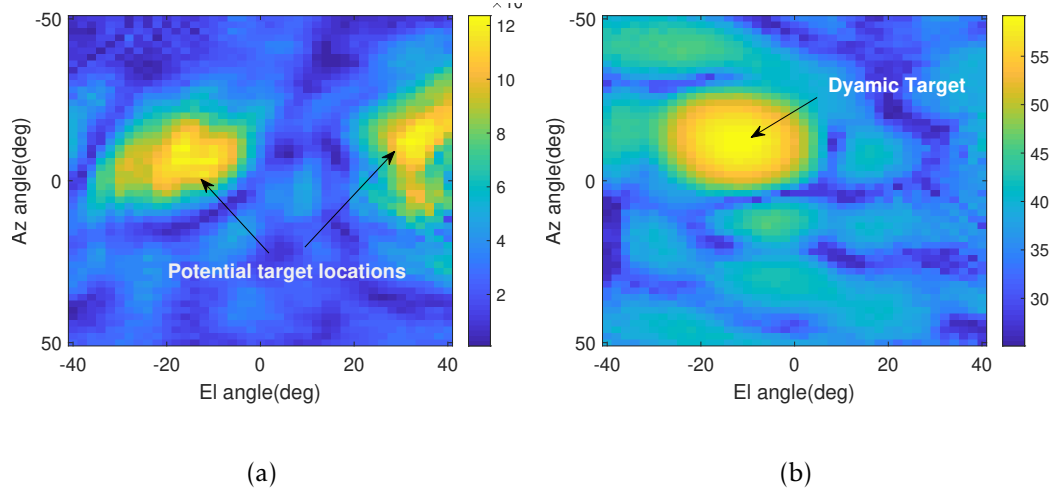


Figure 5.3: (a) Magnitude of the beamformed CIR, and (b) The maximum Doppler power matrix, $\mathbf{P}$, corresponding to the 8th CIR tap at one time instance, s . The matrix $\mathbf{P}$ retains the power contributed only by the dynamic targets. This resulted in suppression of one of the two potential targets observed in (a).

mined from the beamformed received signal. To differentiate the writing object from other static objects in the environment, *mmWrite* exploits its dynamic nature. Next, the beamformed CIR is further processed to extract the velocity information of the targets.

5.1.3 Doppler Velocity Extraction

The correspondence between the relative radial velocity of the target w.r.t the source and the observed frequency of the received signal, from the well-known Doppler effect, motivated us to transform the beamformed signal into the frequency domain. This transformation is achieved by performing a Short-Time Fourier Transform (STFT) along the *slow-time* dimension t . The Fourier trans-

form applied in a window of length w_1 and a step size of w_s . A smaller value of w_s increases the sampling frequency of points on the trajectory, reducing computational complexity. As w_s is halved the complexity doubles. As we have more points on the trajectory, it can capture more drastic changes in the writing. More discussion is provided in §5.2.2. Increasing w_1 improves the resolution in the frequency domain; however, the instantaneous information is smoothed out. A value of $w_1 = 192$ and $w_s = 32$ is used in this work unless mentioned otherwise. The transformation can be written as:

$$Y[\mathbf{T}, s, f] \xleftarrow{FFT} y[\mathbf{T}, t + (s - 1)w_s - w_1 + 1 : t + (s - 1)w_s], \quad (5.5)$$

where s is the window index which is also indicative of the *slow-time*. As the target moves, the observed frequency of the received signal is altered due to the Doppler effect. This change in frequency (Δf) depends on the direction and speed of the moving target and is given by:

$$\Delta f = \frac{2v}{c} f_c, \quad (5.6)$$

where v is the relative radial velocity of the target, f_c is the frequency of the carrier wave and c is the speed of light. This correspondence between the observed frequency, $f = f_c + \Delta f$, and the target radial velocity can be shown on the transformed CIR as:

$$Y[\mathbf{T}, s, f] \longleftrightarrow Y[\mathbf{T}, s, v]. \quad (5.7)$$

The reflected signals from the static objects in the environment contribute to the zero frequency bin of the transformed CIR, while those reflected from the

moving targets contribute to the non-zero frequency bins corresponding to their relative radial velocity. Thus, to detect the moving target, i.e., the writing object, the power in the non-zero frequency bins must be analyzed. Among the non-zero frequency bins, the maximum Doppler power ($\mathbf{P}$) is extracted for each spatial bin (azimuth, elevation and range) and time instance as follows:

$$v_{\mathbf{T},s}^* = \underset{v \neq 0}{\operatorname{argmax}} |Y[\mathbf{T}, s, v]|, \quad (5.8)$$

$$P[\mathbf{T}, s] = |Y[\mathbf{T}, s, v_{\mathbf{T},s}^*]|. \quad (5.9)$$

Recall that the non-zero frequency bins of the Doppler power capture the reflected signals corresponding to the target with non-zero radial velocity. However, during handwriting, there will be instances when the writing object has a zero radial velocity, and $\mathbf{P}$ does not capture the contribution from signals of interest. The zero radial velocity could be due to one of the following reasons: (a) The speed of the target is zero, or (b) the radial component of the velocity of the target is zero, i.e., the target is moving perpendicular to the range dimension. In such instances, the Doppler power corresponding to the zero frequency bin ($\mathbf{P}_0$) is analyzed to capture the reflected signals from the target of interest. Mathematically, this is written as,

$$P_0[\mathbf{T}, s] = |Y[\mathbf{T}, s, 0]|. \quad (5.10)$$

Fig. 5.3(b) shows an instance of the matrix $\mathbf{P}$ at a range tap of 8 for different azimuth and elevation angles. The four-dimensional matrices $\mathbf{P}$ and $\mathbf{P}_0$ indicate the Doppler power for the range dimension with a resolution of 4.26 cm, and

azimuth and elevation dimensions with a spacing of 2 degrees for each time instance, s . These matrices are analyzed to detect and localize the target.

5.1.4 Target Detection

The spatial bins consisting of dynamic targets have a higher Doppler power compared to the ones without a dynamic target. Therefore, the goal of target detection is to identify the bins with relatively higher Doppler power to obtain a rough estimate of the location of the dynamic target. For this, the Doppler power values need to be compared with a threshold. However, the spatial and temporal variation in the noise level prevents us from using a universal threshold for target detection. This problem has been addressed in the radar literature with an adaptive threshold. The most commonly used technique is the CA-CFAR target detection [59].

In this work, a three-dimensional CFAR window is used, corresponding to the range, azimuth, and elevation dimensions. A CFAR threshold map is computed for $P[\mathbf{T}, s]$, and the spatial bins with Doppler power above the CFAR threshold are extracted. Let the 3D CFAR window be denoted by $\mathbf{C}_{\text{win}}$ and C_{add} be the additional threshold. The overall CFAR threshold matrix $\mathbf{C}^{\text{th}}$ is given by:

$$\mathbf{C}^{\text{th}} = \mathbf{C}_{\text{win}} \star \mathbf{P}[:, s] + C_{\text{add}}, \quad (5.11)$$

where $\star$ is the convolution operation. The Doppler power matrix after CFAR

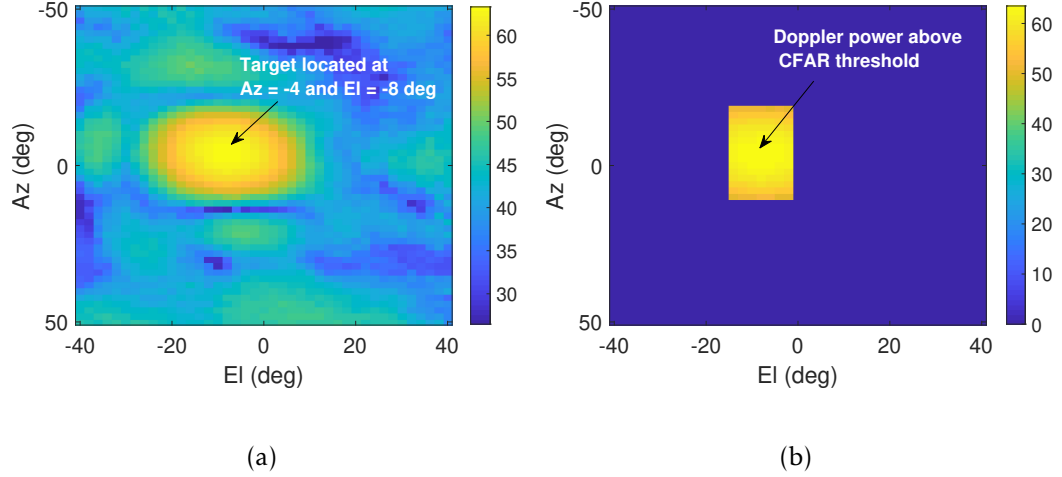


Figure 5.4: (a) Doppler power at range tap 7 for different azimuth and elevation angles, (b) Doppler power after CFAR thresholding. In this example, the target is detected at an elevation of -8° and an azimuth of -4° with a Doppler power of 63.5 dB.

thresholding, $\mathbf{P}^c$ is written as:

$$P^c[\mathbf{T}, s] = \begin{cases} P[\mathbf{T}, s], & \text{if } P[\mathbf{T}, s] > C^{\text{th}}[\mathbf{T}] \\ 0, & \text{otherwise.} \end{cases} \quad (5.12)$$

Fig. 5.4(a) shows the Doppler power for different azimuth and elevation angles and a fixed range tap of 7. The resulted Doppler power after CFAR thresholding is shown in Fig. 5.4(b). Before designing the target detection module using the 3D-CFAR technique, let us understand the two challenges in *mmWrite*: Multiple moving parts and zero radial velocity.

Multiple moving parts: During handwriting, at any given instance of time, different parts of the hand (fingers, wrist, arm) move in different directions and speeds, resulting in different radial velocities due to different inclinations. As a

result, the highest Doppler power does not necessarily correspond to the target of interest (the tip of the pen, for example) at all times. For instance, Fig. 5.5 shows the range, azimuth, and elevation corresponding to the two highest values in the Doppler power matrix $\mathbf{P}$ with time. The traces T_1 and T_2 indicate the spatial information corresponding to the first and second highest Doppler power bins. As shown, two distinct target traces can be observed, and the highest Doppler power keeps switching between the two. The Doppler power at a particular spatial bin and time depends on many factors such as the instantaneous radial velocity of the dynamic target if present, the radar cross-section, material, and location of the target. To avoid misdetection of the target of interest, N targets are detected at each instance instead of one. These N targets are identified iteratively in the decreasing order of Doppler power by comparing with the CFAR threshold and by nulling the region around the previously detected targets to avoid overlap of detections. The target of interest is then identified from amongst the N targets at each time instance using target tracking algorithm described in §5.1.6.

Zero radial velocity: As mentioned before, to detect a target with zero radial velocity, the Doppler power corresponding to the zero frequency bin $\mathbf{P}_0$ needs to be analyzed. This is because, at low/zero radial velocity, the received power contributed by the target is shifted to the zero Doppler velocity bin. Let us consider an example. In Fig. 5.6, the Doppler power corresponding to one spatial bin, $\mathbf{T}$ is shown. Specifically, the time series $P_0[\mathbf{T}, s]$ and $P[\mathbf{T}, s]$ are shown with respect to s . It can be observed that when the target reaches a low radial velocity at the time indices highlighted in the figure, there is a shift in the Doppler power from

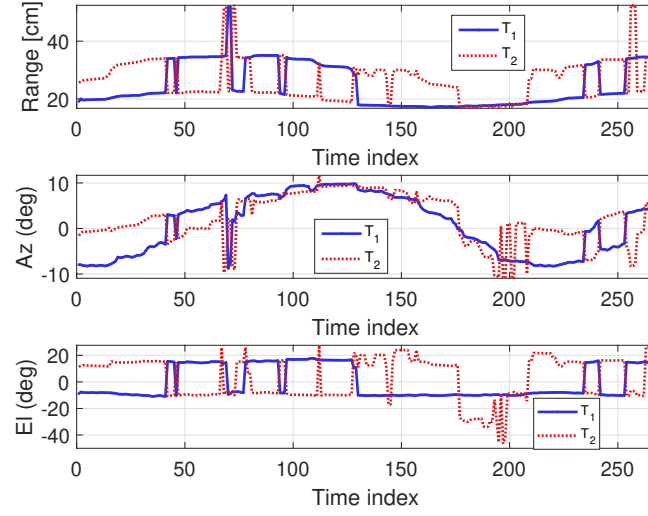


Figure 5.5: Estimated target locations with time using the two highest Doppler power bins. The traces are marked as T_1 and T_2 , corresponding to the first and second highest Doppler power respectively.

$\mathbf{P}$ to $\mathbf{P}_0$. However, application of CFAR target detection on $\mathbf{P}_0$ is not effective due to reflections from irrelevant static objects in the environment, usually referred to as clutter in the radar literature [59]. The noise level or the clutter can be estimated from the Doppler power of a particular spatial bin when the target is not present. Therefore, instead of estimating the noise from the adjacent spatial locations as in the CFAR technique, we estimate it from the median power of a particular bin over time. Let C_0^{th} denote the threshold for target detection on $\mathbf{P}_0$. It can be written as, $C_0^{\text{th}}[\mathbf{T}] = \text{median}(P_0[\mathbf{T},.])$. The Doppler power matrix after

CFAR thresholding, P_0^c is written as:

$$P_0^c[\mathbf{T}, s] = \begin{cases} P_0[\mathbf{T}, s], & \text{if } P_0[\mathbf{T}, s] > C_0^{\text{th}}[\mathbf{T}] \\ 0, & \text{otherwise.} \end{cases} \quad (5.13)$$

Based on the above discussions, the target detection module is designed as follows. For each time instance, we identify a maximum of N non-overlapping probable locations of the target corresponding to the decreasing Doppler power from P_0^c and P^c . The location coordinates corresponding to the spatial bins of the targets are stored in $\mathbf{T}_0^*$ and $\mathbf{T}^*$ respectively as described in Algorithm 1.

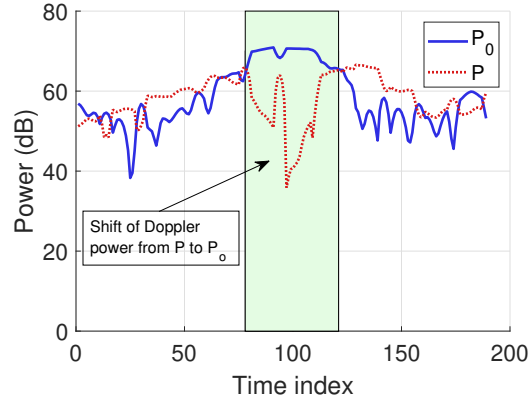


Figure 5.6: Demonstration of the Doppler power shift from non-zero frequency bin to the zero frequency bin when the target has low or zero radial velocity. Observe that when the target reaches a low radial velocity at time index of about 75, the Doppler power shifts from the non-zero frequency bin to the zero frequency bin and then switches back when the target gains some radial velocity at a time index of about 120.

Algorithm 3 Target detection algorithm for *mmWrite*

```
1: for each time instance  $s$  do,  
  
2:   for  $i = 1 : N$  do                                 $\triangleright N$  is the number of targets detected  
  
3:      $\mathbf{T}_{\max} = \underset{\mathbf{T}}{\operatorname{argmax}}(P^c[\mathbf{T}, s])$   
4:     if  $P^c[\mathbf{T}_{\max}, s] > 0$  then  
5:        $\mathbf{T}^*[i, s] = \mathbf{T}_{\max}$   
6:     end if  
7:      $P^c[\mathbf{T}_{\max} - \Delta : \mathbf{T}_{\max} + \Delta, s] \rightarrow \text{Null}$   
8:   end for  
9:   Repeat Step 2 to 8 with  $\mathbf{P}_0^c$  to determine  $\mathbf{T}_0^*$   
10: end for
```

5.1.5 Target Localization

From the previous stage, the location bin of the target can be determined from $\mathbf{T}^*[:, s]$ at every time step. Since the accuracy of the estimates is limited to 2 degrees in the azimuth and elevation dimensions and 4.25 cm in the range dimension from the design of the radar, a discontinuous/quantized trajectory is obtained for characters spanning only a few centimeters as shown in Fig. 5.8(a). However, due to the smooth and continuous nature of handwriting, it has been observed that the Doppler power “flows” gradually from one bin to another. This fact is leveraged to obtain finer location estimates. The observation can be demonstrated by a simple experiment, as explained below.

A straight line is drawn away from the device keeping the azimuth and el-

Algorithm 4 Target tracking algorithm for *mmWrite*

```
1: for each time instance  $s$  do,

2:    $W[s] \leftarrow$  lowest elevation target in CTL with  $CS > CS_{th}$ 

3:   for  $j = 1 : \text{length}(CTL)$  do                                 $\triangleright$  Update current entries of  $CTL$ .

4:     if  $\text{any}(\text{cont}(CTL(j), L[:, s])) = 1$  then

5:        $CTL(j).CS ++$ 

6:     else

7:       if  $\text{cont}(CTL(j), W(s))$  then                                 $\triangleright$  Target missed from  $L$ .

8:         if  $\text{any}(\text{cont}(CTL(j), L_0[:, s])) = 1$  then  $\triangleright$  Search for missed target

          location in  $L_0$ .

9:          $CTL(j).CS ++$ 

10:      else

11:         $CTL(j) \rightarrow []$        $\triangleright$  Target missed from  $L$  and  $L_0 \implies$  remove

          from CTL.

12:      end if

13:    end if

14:  end if

15: end for

16: for  $k=1:N$  do                                                 $\triangleright$  Add new entries to  $CTL$ .

17:   if  $\text{any}(\text{cont}(L[k, s], CTL(:))) \neq 1$  then       $\triangleright$  New location discontinuous

    with CTL  $\implies$  add to CTL.

18:      $loc.L = L[k, s]; loc.CS = 1;$ 

19:      $CTL(end + 1) = loc$ 

20:   end if

21: end for
```

evation nearly zero and the corresponding Doppler power in the range taps is analyzed. As shown in Fig. 5.7a, as the target moves away from the radar, the Doppler power shifts gradually from tap 11 (46.46 cm) to tap 13 (55.38 cm). According to our proposed approach, the tap corresponding to the maximum Doppler power is considered as the range of the target. Therefore, we get the discrete range estimates as shown in black. This observation motivated us to combine the information in adjacent taps to obtain an improved location estimate. However, using adjacent four values near the peak is not very useful. For example, during the period when the power corresponding to tap 13 is the maximum (orange line), the power corresponding to tap 11 (purple line) is already close to the noise level. This observation motivated us to use the SPI technique discussed in §2.2.3 to improve the location estimates. In this work, SPI is applied to the range, azimuth, and elevation dimension independently. Fig. 5.7b shows the range estimates before and after SPI. Interpolation using the combined information from the three dimensions could improve the estimation, and such an analysis is left for future work.

The SPI operation has a significant effect on the performance of handwriting tracking as demonstrated in Fig. 5.8 since it largely boosts the spatial resolution. The trajectory of the word “beam” in Fig. 5.8(a) is formed by many overlapping points because of the discrete location taps (range, azimuth and elevation) while the trajectory formed after SPI in Fig. 5.8(b) is readable by the naked eye. Let $\mathbf{T}_{a,b,c}^*[i,s] = [az^*[i,s] + a, el^*[i,s] + b, r^*[i,s] + c]$ where a, b, c are the index offsets in the corresponding dimensions. The final estimates of azimuth, elevation and

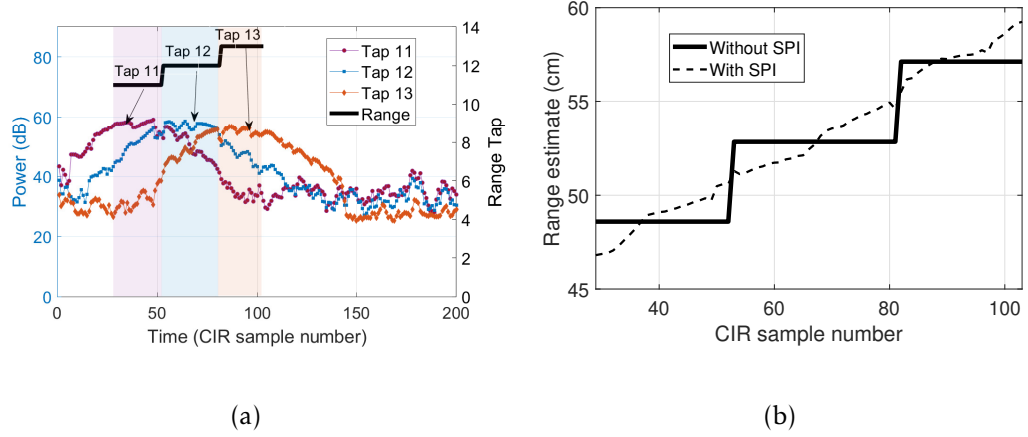


Figure 5.7: (a) Doppler power for different range taps with time for a straight line drawn along the range away from the device, (b) Range estimates before and after SPI.

range coordinates are obtained from the SPI as:

$$az_{SI}^*[i, s] = \text{SPI}\{P[\mathbf{T}_{-1,0,0}^*[i, s], s], P[\mathbf{T}_{0,0,0}^*[i, s], s], P[\mathbf{T}_{1,0,0}^*[i, s], s]\}. \quad (5.14)$$

$$el_{SI}^*[i, s] = \text{SPI}\{P[\mathbf{T}_{0,-1,0}^*[i, s], s], P[\mathbf{T}_{0,0,0}^*[i, s], s], P[\mathbf{T}_{0,1,0}^*[i, s], s]\}. \quad (5.15)$$

$$r_{SI}^*[i, s] = \text{SPI}\{P[\mathbf{T}_{0,0,-1}^*[i, s], s], P[\mathbf{T}_{0,0,0}^*[i, s], s], P[\mathbf{T}_{0,0,1}^*[i, s], s]\}. \quad (5.16)$$

At time instant s , the location of the i^{th} moving target is given by,

$$\mathbf{L}[i, s] = [az_{SI}^*[i, s], el_{SI}^*[i, s], r_{SI}^*[i, s]]. \quad (5.17)$$

Let the location estimates obtained from $\mathbf{T}_0^*[i, s]$ be denoted by $\mathbf{L}_0[i, s]$. The ultimate trajectory is constructed using $\mathbf{L}$ and $\mathbf{L}_0$ in the next section.

5.1.6 Trajectory Construction

In the previous module, multiple target locations (N) have been detected at each time instance, s and stored in $\mathbf{L}$ and $\mathbf{L}_0$. In this module, a trajectory tracking

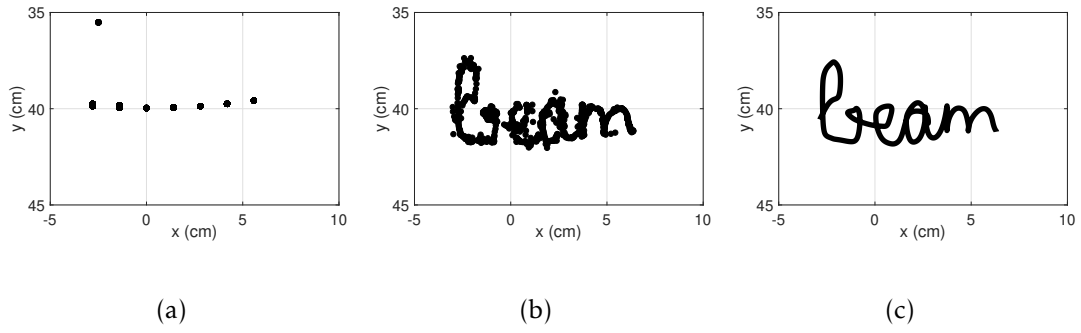


Figure 5.8: Illustration of the target localization and trajectory construction for the word “beam”. (a) Discrete points of the trajectory from target detection, (b) Finer location estimates obtained from SPI and (c) Trajectory after DCT-based smoothing.

algorithm is designed to extract the raw trajectory of the target of interest from $\mathbf{L}$ and $\mathbf{L}_0$. Following are some observations that will be used to build the algorithm.

1. Assuming the handwriting is on a flat surface and is smooth and continuous, the elevation angle of the writing object does not change abruptly. This fact is used to define continuity of target location at two successive time instances.
2. At any given time instance, the elevation angle corresponding to the target of interest, i.e., the tip/lower part of the writing tool is the least among all the other detected target locations.
3. If a particular location is being continuously detected for many time instances in $\mathbf{L}$ and suddenly disappears, there is a possibility that the target has reached a low radial velocity and is detected in $\mathbf{L}_0$.

The target tracking algorithm is described in Algorithm 2. The target locations are read from $\mathbf{L}$ over time and the final target locations are stored in $W[s]$. The continuous trajectories are tracked and stored in a list. If a trajectory segment is continuous for more than a predefined number of time instances, then it is considered as a potential trajectory of the target of interest. If there are multiple potential trajectories, the lowest elevation trajectory is considered as the *mmWrite* trajectory. Additionally, if a trajectory suddenly becomes discontinuous after being detected as the *mmWrite* trajectory for a while, then we also check if the target location is detected in $\mathbf{L}_0$ to determine continuity. Some of the terms used are summarized below:

- CS (Continuity Score): Each detected target location is associated with a score abbreviated as CS. It is a scalar that indicates the length of the continuous trajectory detected in the previous instances of which the current location is a part.
- CS_{th} : Threshold on the continuity score
- CTL (Continuous Target List): List of target locations detected whose CS is greater than 0.
- $Cont(a,b)$: Function to check if two locations a and b are continuous. Recall that two locations detected at successive time instances are said to be continuous if the elevation angle does not change significantly.

The trajectory tracking algorithm also removes outliers that result from

sudden changes in the point of reflection or missed targets. The tracked target locations with time are stored in the vector $\mathbf{W}$. A raw trajectory can be formed by combining all the estimates. Such a trajectory is noisy and an example is shown in Fig. 5.8(b). To obtain a smoother trajectory, we propose a smoothing technique based on the Discrete Cosine Transform (DCT). The idea is to compute the DCT of the noisy data and inverse transform to the data domain while preserving only the significant coefficients. By doing so, we discard the insignificant coefficients that account for the noisy and irrelevant part of the signal [26], thereby obtaining a smooth trajectory. Mathematically, the recovered writing trajectory, $\hat{\mathbf{W}}$ is given by,

$$\hat{\mathbf{W}} = \text{IDCT}(\tau \text{DCT}(\mathbf{W})). \quad (5.18)$$

Here IDCT is the inverse discrete cosine transform, and τ is a diagonal matrix whose entries are given by,

$$\tau_{i,j} = \begin{cases} [1 + s(2 - 2\cos(\frac{(i-1)\pi}{n}))^2]^{-1} & i = j, \\ 0 & i \neq j, \end{cases}$$

where n is the number of data points and s is the smoothing factor. The DCT coefficients of the data points are scaled by the matrix τ in the transform domain. The scaled coefficients are converted back to the data domain using the IDCT. The trajectory smoothing is achieved by the scaling matrix τ with the smoothing factor s and the cosine factor which imposes decreasing weights to the DCT coefficients. An example of the smoothed trajectory after DCT-based smoothing ($\hat{\mathbf{W}}$) is shown in Fig. 5.8(c).

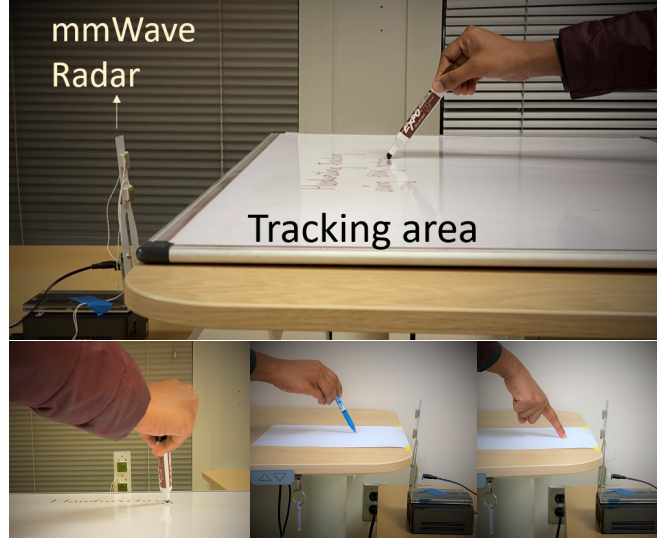


Figure 5.9: Experimental setup for *mmWrite*. (Top) Writing on a whiteboard using a marker, (Bottom left to right) Front view of the setup showing the writing surface at nearly zero elevation, writing using a pen and hand.

5.2 Experimental Evaluation

The performance of the proposed *mmWrite* system is evaluated in this section. Moreover, the effect of different parameters and experimental settings is discussed in detail. All the experiments are performed in a busy office environment using a similar setup as shown in Fig. 5.9, in which the writing surface and the radar lie on a flat surface. In the default setup, the height of the writing surface is adjusted to be around the zero elevation angle of the radar. Handwriting is performed so that the writing tool faces the radar and is the closest moving target to the radar. The writing tool is held at its upper portion to facilitate differentiation of the reflections from the writing tool and the hand/fingers. The

handwriting itself is performed with natural writing speed using a marker on a paper/board. The CIR time series is recorded during the writing and is processed later using *mmWrite*. The recovered trajectories are subjected to different types of evaluations, which are described in detail below.

5.2.1 Overall Performance

The performance of *mmWrite* is evaluated using three different approaches. Unless otherwise mentioned, the radar is placed at the origin $(0, 0)$.

Visual inspection and Shape matching: A valid handwriting tracking system should be able to preserve the relative shape of the handwriting trajectory that is legible to humans or recognizable by any standard handwriting recognition software. We assess this by visually comparing the ground-truth handwritten trajectories and the recovered trajectories by *mmWrite*. In this experiment, randomly selected English alphabets from the EMINST dataset [17] are printed and traced within an area about $5\text{ cm} \times 5\text{ cm}$ on an A4 paper with a marker. Fig. 5.10 shows two versions of different English alphabets with the first, third and the fifth row showing the ground truth character images and the second, fourth and the sixth rows showing the recovered characters. It can be confirmed that the relative shapes of the alphabets are well preserved by *mmWrite*, which are visually similar to the ground-truth handwriting and can be recognized by humans confidently. Additionally, we also evaluated the shape similarity using shape context as discussed in [10]. Classification using shape context achieved an accuracy of

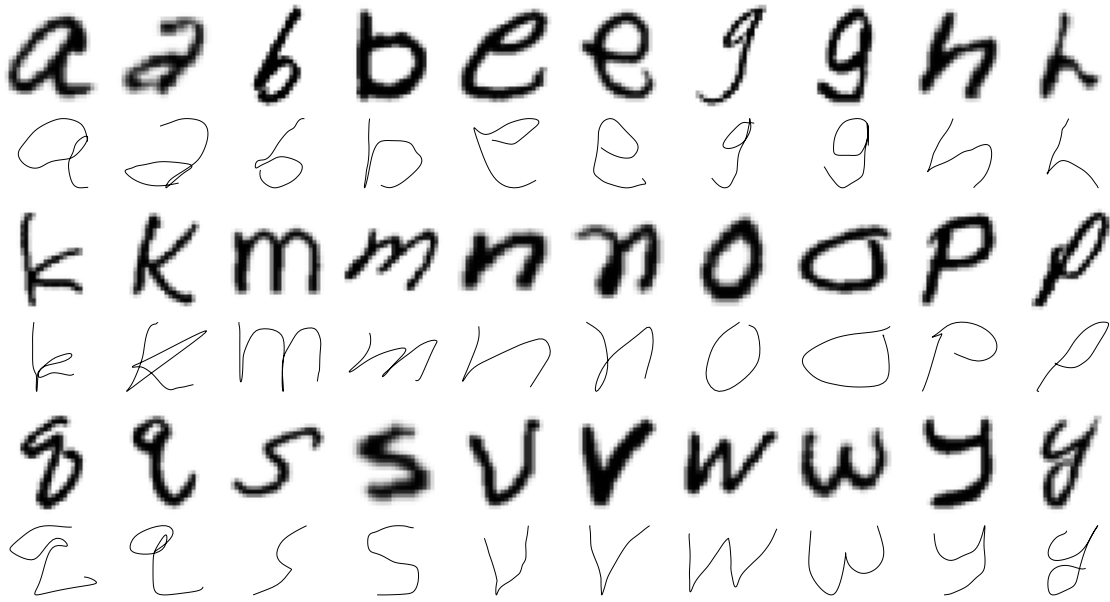


Figure 5.10: Visual comparison of ground truth characters and recovered characters. The characters in bold are picked from EMINST dataset and serve as the ground truth (row 1, 3, and 5). The corresponding recovered characters are shown in row 2, 4, and 6.

96.6% on the characters shown in Fig. 5.10.

Character recognition accuracy: Given the encouraging results as in Fig. 5.10, it would be interesting to see how good the recovered characters could be recognized by standard handwriting recognition tools. To evaluate that, we wrote English alphabets in different sized grids and at different distances from the device. For each scenario, we randomly picked 50 alphabets from the EMINST dataset and traced their trajectories on a paper. The trajectories obtained by *mmWrite* are then exported from MATLAB to control the mouse pointer and are then fed to a standard handwriting recognition software (myScript). The character recognition accuracy reported by the software is recorded and shown in Fig. 5.11. It can

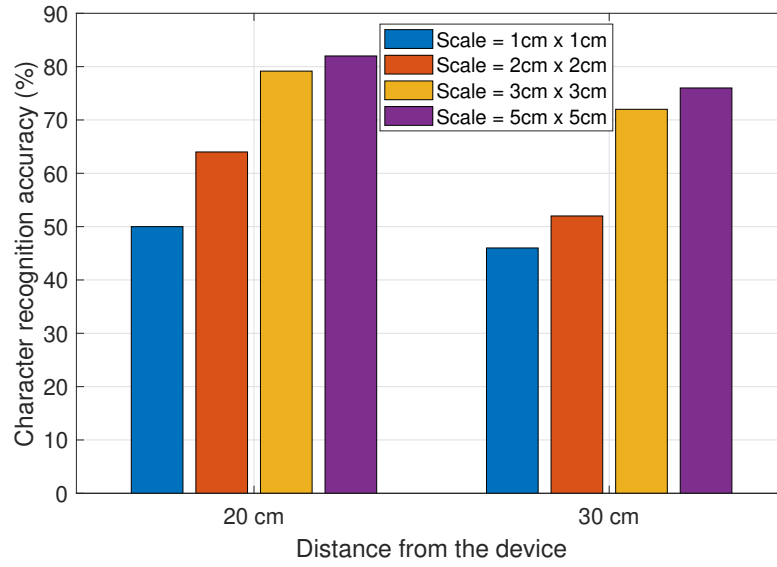


Figure 5.11: Character recognition accuracy at different distances from the device with characters written in different scales.

be observed from the figure that the accuracy decreases with distance from the device and increases with the scale of handwriting, as expected. This is because of the decreasing cross-range accuracy. For instance, the character recognition accuracy for characters written within 3 cm x 3 cm is 80% and 72% at distances 20 cm and 30 cm respectively. The accuracy for the same distance of 20 cm is 80% and 82% for scales of 3 cm x 3cm and 5 cm x 5 cm respectively. As will be shown later in Fig. 5.17, the tracked characters, even written in really small scales (1 cm x 1 cm), can still be easily read by human eyes, although they might not be accurately recognized by standard software.

Tracking accuracy: Now we quantitatively evaluate the tracking accuracy of *mmWrite*. To do so, we trace a circle and a triangle (both predefined with known size and shape which serve as the ground truth trajectory) multiple times, and

reconstruct the trajectories. Due to lack of time synchronization between the handwritten trajectories and the *mmWrite* estimates, we report the median of the minimum projection errors between the two versions of trajectories as the tracking error. Two instances of the ground truth and the projected trajectories are shown in Fig. 5.12(a) and Fig. 5.12(b).

It is known that the distance from the device affects the tracking performance due to the reduced cross-range accuracy. This can be observed from the CDF of the tracking errors shown in Fig. 5.12(c) where the tracking error increases with distance from the radar (R). To further gauge the influence of range and azimuth angles on the tracking error, we repeat the above experiment for different ranges and azimuth angles. Fig. 5.13 shows the resultant error map. The median tracking error increases from about 3 mm at a range of 20 cm to 40 mm at a range of 3 m. The tracking error also slightly increases with the azimuth angle at any given range. The reflected power from the tiny writing object is too weak to be detected by the radar, beyond a range of 3 m. Also, the target cannot be detected beyond an azimuth angle of 50-55 degrees due to the hardware limitations. Nevertheless, Fig. 5.13 has demonstrated that the *mmWrite* system can already cover an area of about 8 m^2 , about 10 $\times$ larger than the state-of-the-art system using mmWave radio [78].

5.2.2 Micro-Benchmarks

Handwriting tracking on different surfaces: *mmWrite* can be used to track hand-

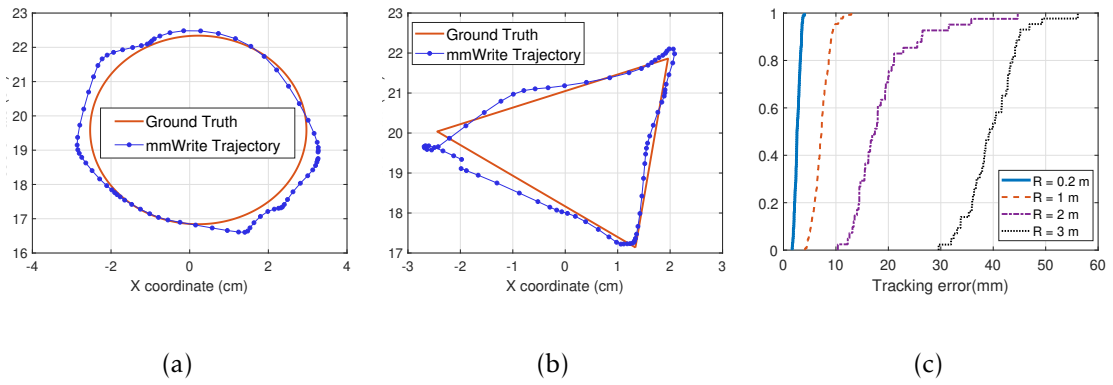


Figure 5.12: Ground truth and the reconstructed trajectories of a (a) circle and (b) triangle.(c) CDF of tracking error.

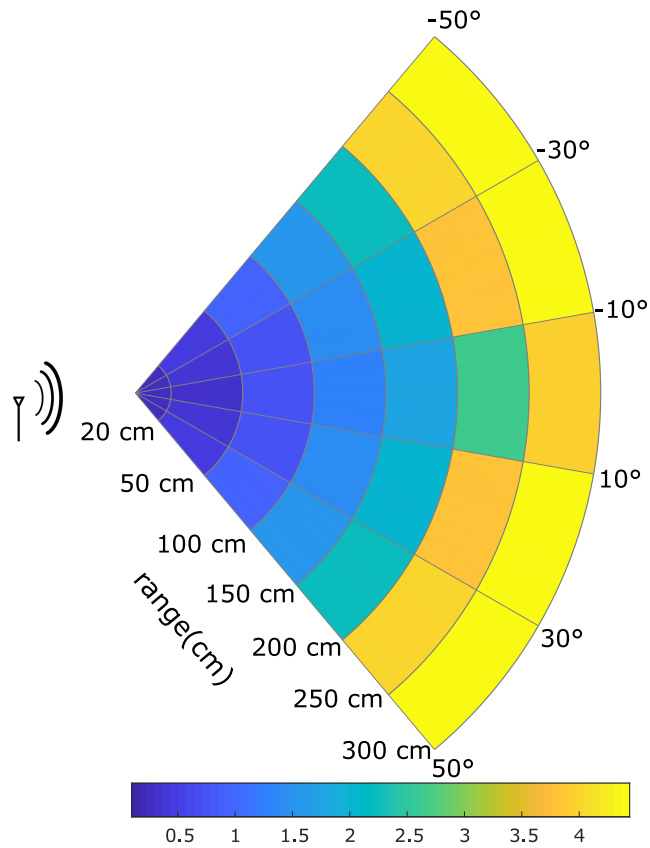


Figure 5.13: Tracking error at different range and azimuth angles from the radar.

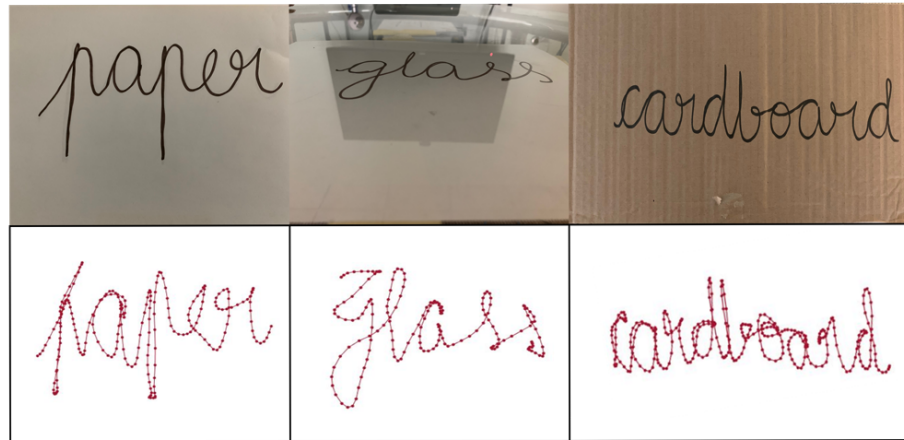


Figure 5.14: Handwriting tracking on different materials

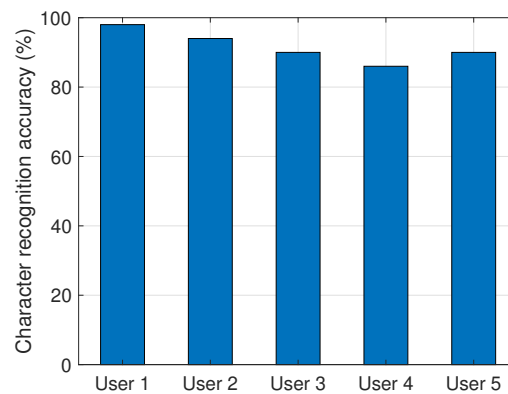


Figure 5.15: Character recognition accuracy for different users.

writing on flat surfaces of different materials. This is because only the signals reflected off the writing object are analyzed by *mmWrite*, and the material of the writing surface is irrelevant. *mmWrite* can, therefore, transform any flat surface to a potential writing surface. Fig. 5.14 shows some instances of handwriting tracked on different materials such as paper, glass, and cardboard.

Handwriting tracking with different writing objects: Intuitively, we expect the writing object’s properties to have a more direct effect on the tracking perfor-

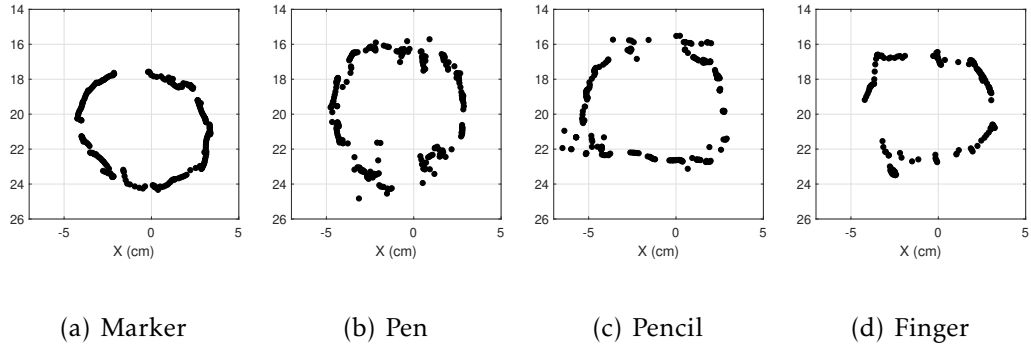


Figure 5.16: Trajectory points obtained for different writing objects.

mance than the writing surface. To analyze this, we have performed tracking with different writing objects: a marker, a pen, a pencil, and a human finger. We have traced a circle with these objects, and the raw trajectory points, $\mathbf{W}$ are shown in Fig. 5.16. It can be observed that a smooth and continuous trajectory is obtained in the case of a marker, while the trajectory points obtained in the case of a hand/finger gave noisy estimates. This behavior can be associated with different incidence points on the target and the non-rigid nature of the hand. The great performance with a marker will be particularly useful by transforming traditional whiteboards into interactive ones using *mmWrite*.

Handwriting recognition with different users: To study the user dependency on the performance of *mmWrite*, we have requested 5 different users to draw 50 characters at a range of about 30 cm and in an area spanning about 7 cm x 7 cm. The corresponding recognition accuracy is reported as shown in Fig. 5.15. It can be observed that the average character recognition is about 91% over different users.

Scale of handwriting: Apparently, the performance of *mmWrite* depends on the

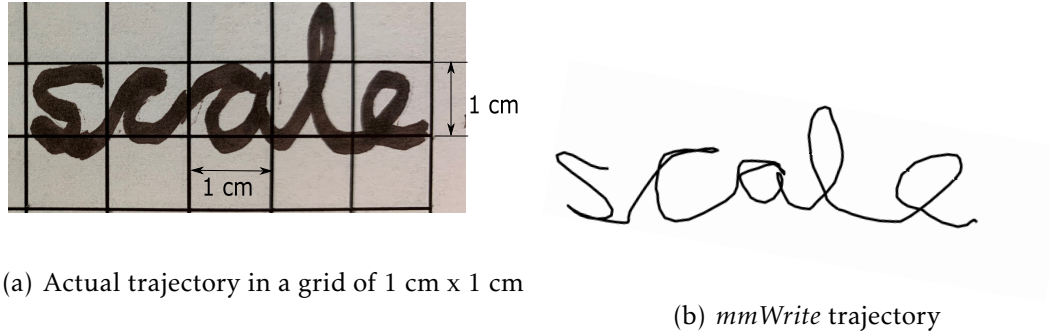


Figure 5.17: Demonstration of handwriting tracking for characters within 1 cm.

scale of handwriting. The higher the scale, the easier it is to identify the relative shape of the trajectory. In Fig. 5.11, the character recognition accuracy for characters written in different scales is shown. To explore the limit of the scale of handwriting that can be retrieved by *mmWrite*, experiments were performed with characters spanning a few millimeters at a distance of 20 cm from the radar. Fig. 5.17(a) shows the ground truth trajectory of the word “scale” with the grid size of 1 cm x 1 cm and Fig. 5.17(b) shows the obtained trajectory using *mmWrite*. It has been observed that characters written in a grid of size as small as 1 cm x 1 cm can be recovered and the resultant trajectories can be easily recognized by humans.

Repeatability of the trajectories: The absolute localization of the target enables *mmWrite* to generate precise estimates of the handwriting trajectories. To demonstrate the repeatability of *mmWrite* trajectories, the word “repeat” is traced several times on a paper and the trajectories are reconstructed. Fig. 5.18a shows the picture of the traced word on a paper and Fig. 5.18b shows the corresponding *mmWrite* trajectories. It can be observed that the trajectories overlap on each other, demonstrating the consistent precision of the estimated trajectories.



(a) Ground truth of the word “repeat”

(b) *mmWrite* result

Figure 5.18: Demonstration of *mmWrite* precision

Number of points on the trajectory: *mmWrite* reconstructs the handwriting trajectory by estimating the location of the moving/writing target at different time instances and by connecting all the location estimates. For a given length of the trajectory, if the target localization is performed at shorter intervals, a greater number of points are obtained, capturing more abrupt changes in the trajectory. However, this performance enhancement is achieved at the cost of higher computational complexity. The number of points on the recovered trajectory is dependent on two factors:

- **The speed of handwriting:** With a fixed sampling frequency of the CIR time series, faster handwriting results in fewer points on the trajectory compared to slower handwriting. Fig. 5.19 shows the recovered trajectories for letter ‘m’ written with two extreme speeds. Fig. 5.19(a) shows handwriting with very slow speed while Fig. 5.19(b) shows the trajectory corresponding to fast handwriting. Observe that for the faster speed, the number of points on the trajectory is less and the trajectory becomes smoother. However,

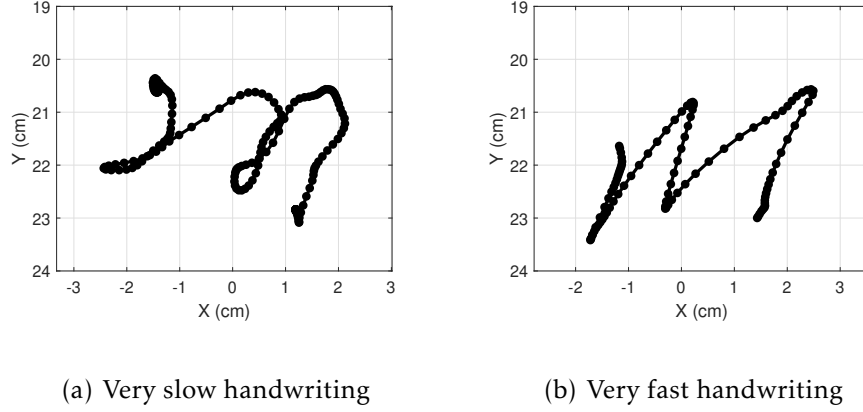


Figure 5.19: Recovered handwriting trajectories with extreme speeds of handwriting.

faster speed may not capture the minute details of the trajectories.

- **Step size in the windowed Fourier transform (w_s):** From the beamformed CIR time series, *mmWrite* computes the STFT along the slow-time dimension. As the step size of the moving window (w_s) decreases, target localization is performed more frequently, resulting in a larger number of points on the trajectory. For a step size of w_s , the sampling frequency of the points on the trajectory is $1/(w_s \times T_b)$, where T_b is the duration of the burst as discussed in §2.2.1. For instance, if the step size, $w_s = 20$ and $T_b = 400 \mu s$, the sampling frequency is 125 Hz. For a fixed $T_b = 400 \mu s$ the trajectories of writing letter ‘n’ with different w_s are shown in Fig. 5.20. As w_s increases, the sampling frequency decreases and lesser number of points are obtained on the trajectory. This causes smoothening of minute details in the trajectory.

Degree of smoothness: The degree of smoothness achieved by the DCT-based

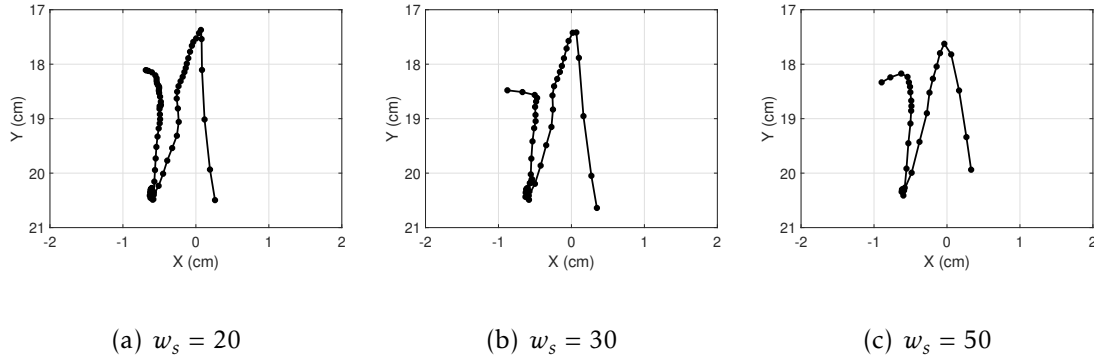


Figure 5.20: Recovered handwriting trajectories with different w_s .

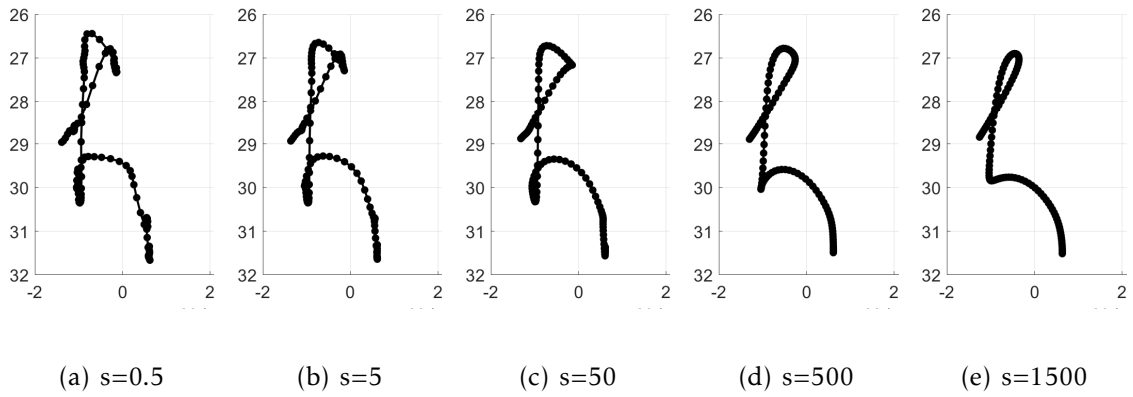


Figure 5.21: Smoothed trajectories for different smoothing parameter values.

smoothing depends on the smoothing parameter s . For a given number of points on the trajectory, the higher the value of s , the smoother the trajectory. However, if the value of s is too large, necessary details in the relative shape of the trajectory can be smoothed out, which is undesired. For instance, in Fig. 5.21, the smoothed trajectories with different values of the smoothing parameter s are shown. In this example, for $s = 1500$, the letter “h” is barely recognizable. *mmWrite* defaults with $s = 2$ in this work.

Complexity and Latency: Let N be the total number of points on a trajectory. The complexity for each of the modules in *mmWrite* is shown in Table.5.1. Most

modules include multiple linear-complexity operations and have an overall complexity of $O(N)$. Although the Doppler velocity extraction module uses the FFT operation, which has a complexity of $O(N\log N)$, the transformation is performed along the *slow-time* dimension with a duration of w_l resulting in $O(1)$ operations for each point. The trajectory construction module uses the DCT operation, which has a complexity of $O(N\log N)$. Overall, *mmWrite* has a complexity of $O(N\log N)$. This can be further optimized by smoothening discontinuous strokes separately, which is part of future work.

Table 5.1: Complexity for different modules in *mmWrite*.

Module	Complexity
Background Subtraction	$O(N)$
Digital Beamforming	$O(N)$
Doppler Velocity Extraction	$O(N)$
Target Detection	$O(N)$
Target Localization	$O(N)$
Trajectory Construction	$O(N\log N)$

The current version of *mmWrite* uses offline processing and records the complete CIR time series before producing the handwriting trajectory. If the static background is recorded beforehand, the estimated time in MATLAB to generate one raw trajectory point is about 0.4 seconds on a Windows machine with an Intel Core i7 processor. We can obtain a substantial reduction in the computation time by using parallel processing and GPUs.

5.3 Summary

In this Chapter, we present *mmWrite*, the first handwriting tracking system by reusing a single commodity 60GHz WiFi radio as a mmWave radar. With minimal infrastructure support, *mmWave* can transform any flat surface, be it a conventional whiteboard, a table, or a wall, into an interactive writing area. We present a complete pipeline to track a writing object at millimeter accuracy and thus retrieve handwritten characters as small as 1 cm x 1 cm close to the device and cover up to about 8 m^2 and a maximum range of 3 m. We perform numerous experiments to validate the performance of *mmWrite* in different conditions and environments, including varying distances, surfaces, writing speeds, and device parameters.

Chapter 6: Conclusion and Future Work

6.1 Concluding Remarks

In this dissertation, we first present the motivation and a primer of wireless and RF sensing. Later, we present three HCI systems which are summarized below.

1. *Driver Authentication*. In Chapter 3, we introduced the idea of in-car driver authentication to make the automobiles smarter and more user-friendly. To evaluate the feasibility and performance of the proposed system, we conducted long-term experiments in a car. We built the first long-term driver radio biometric database with data collected from several people, and utilized techniques from ML literature to solve this problem. Furthermore, we have implemented frequency hopping and used MIMO systems to exploit the frequency and multi-antenna diversities respectively. Experimental results show that the proposed system is practically feasible with good accuracy for two or three driver authentication, which is a typical use case for a smart car.
2. *Handwriting Tracking*. In Chapter 5, we proposed *mmWrite*, a handwrit-

ing tracking system using a single commercial 60 GHz radar. A complete pipeline of the handwriting tracking system from detecting the writing object to the post-processing steps is presented. It has been proved that the relative shape of the handwritten characters/words is preserved, which can be read by a human or fed to a handwriting recognition software. The performance of *mmWrite* is evaluated and discussed under different experimental conditions. With *mmWrite*, we envision smart environments in which any flat surface can potentially be converted to a writing surface and can be a writing surface, and the space for HCI can be extended beyond the touch screens.

3. *Gesture Writing Recognition*. In Chapter 4, we have proposed *GWrite*, a device-free gesture recognition system using commercial WiFi devices that can work in through-the-wall scenarios and over a broad set of gestures. Using a gesture model and embracing the intense multipath in a typical indoor environment, we have proved that the TRRS between CSIs decreases monotonically with the distance moved by the hand. This observation allowed us to extract features describing geometric relationships in gesture shapes consisting of straight-line segments. We have evaluated *GWrite* on a set of 15 uppercase English characters and achieved a recognition accuracy of 92%. As more bandwidth becomes available with WiFi 6/7, the gestures comply more closely with the theoretical model and can achieve higher recognition accuracies, playing an integral role in the future of smart

indoor environments and HCI applications.

6.2 Future Work

A proof-of-concept is presented for three different HCI systems using prototypes built on commercial WiFi devices. To make these systems practically usable and work in real-time, several issues need to be studied further.

6.2.1 Driver Authentication

The proposed driver authentication system is the first generation of such an effort where we focused on key enabling issues and carried out a proof-of-concept development. Several limitations and issues deserve additional attention, which can be addressed in the future.

1. In this work, we attempted to address the problem of “changing in-car environments” using learning methods for a restricted environment such as a car. For more general environments such as indoor, more advanced techniques will be necessary. Also, environment independent radio biometrics cannot be obtained by direct subtraction of the CSI of the empty environment. As future work, the pattern of the indoor multipath channel change and the dependence of human radio biometrics on the multipath channel can be studied.
2. We have used a simple NN in this work. Other NNs with a more complicated architecture involve a larger number of learnable parameters and

thus require a much larger number of samples for training. Gathering more radio biometric data for different people in different environments can enable the usage of a more complex NN architecture and provides a better understanding of human radio biometrics. Although this increases complexity, modern cars are being manufactured with more computational power, capable of handling autopilot, adaptive cruise and even face recognition. Hence we believe that the future cars will be well equipped to carry out the task of model updating in a NN. An alternate approach can be cloud computing, by sending gathered data to the cloud and fetching the recurrent training computed on the cloud.

3. In a practical scenario, there will be at most two or three authenticated drivers for a car. In such cases, the similarity in all the physical attributes is highly unlikely. As future work, more data need to be gathered to study the performance of the proposed system for people with similar physical characteristics like twins. The variability in the accuracy of different pairs of people can also be understood with more data and by including more testers.

6.2.2 Gesture Recognition

With *GWrite*, we recognized a specific set of gesture shapes consisting of straight-line segments. More information about the gesture shapes can be inferred as more bandwidth becomes available, as discussed below.

1. With higher bandwidths, the observed behavior of the TRRS similarity of CSIs converges to its theoretical counterpart. Consequently, the classification accuracy of the turn angles and the intersection points will improve. Further, gestures with minor differences in their trajectory shapes could also be reliably recognized.
2. Alongside gesture writing recognition, more information in the time domain in terms of the ToF and precise estimates of the relative speed and the distance moved could facilitate handwriting tracking in the air without the need for antenna arrays. Such a technology would move us one step closer to achieving “vision through walls.”

6.2.3 Handwriting Tracking

In this work, we built a prototype of a handwriting tracking system and used offline processing to recover the trajectory. At this stage, *mmWrite* can be further improved to robustly accommodate natural handwriting and build a real-time system. Possible extensions of the current work are mentioned below.

1. We assumed a single moving target in this work. The algorithm can be extended to multiple moving targets by repeating the target detection, localization, and tracking steps at each time instance for the desired number of targets. Spatial resolution plays a crucial role in the performance of multiple target tracking. The tracking error could be further reduced using multiple radios, for instance, by using a triangulation method.

2. We do not deal with multiple/discontinuous strokes in this work. With the current version of *mmWrite*, the recovered trajectories are projected onto a 2D plane. Therefore, discontinuous writing strokes would still appear continuous. One way to handle multiple strokes is to exploit the elevation information of the target, i.e., by discarding the trajectory points above a pre-defined threshold.

6.3 General Future Work

Given the proliferating number of WiFi-connected devices around us, wireless sensing will be an intuitive and natural way to realize HCI applications in the future.

With the ever-increasing demand for faster data rates, more RF spectrum is being made accessible with higher bandwidths. As we have seen in this dissertation, more bandwidth yields more diversity, information, and convergence to the theoretical radio analytic models. For instance, WiFi 6/7 brings in much room to enhance wireless sensing systems with more bandwidth. While the infrastructure for WiFi 6 is now being included in the next generation of smart devices, its sensing potential is yet to be studied.

We have designed wireless sensing applications using radio signals with distinct properties. While the 2.4/5 GHz band radio signals penetrate through walls and have broader coverage, they can only provide coarse information about the physical perturbations. On the other hand, 60 GHz radio signals suffer heavy

attenuation and cannot traverse through walls easily, and have limited coverage. However, they can provide much finer resolution, which helped us design the handwriting tracking application. It is interesting to study the sensing potential of the radio signals between these two bands that can have through-the-wall propagation with finer resolution than the 5 GHz band.

Deep learning has achieved great success in many fields, including computer vision, natural language processing, and genetic engineering, and has solved many complex problems. Although few works in wireless sensing use these approaches, the limited data hinders the widespread adoption of learning approaches. It is essential to develop standard shared data sets that can speed up the research on wireless sensing and help us build the future of smart environments together.

Bibliography

- [1] Heba Abdelnasser, Khaled Harras, and Moustafa Youssef. A Ubiquitous WiFi-Based Fine-Grained Gesture Recognition System. *IEEE Transactions on Mobile Computing*, 18(11):2474–2487, 2018.
- [2] Heba Abdelnasser, Khaled A Harras, and Moustafa Youssef. UbiBreathe: A Ubiquitous non-Invasive WiFi-based Breathing Estimator. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 277–286, 2015.
- [3] Heba Abdelnasser, Moustafa Youssef, and Khaled A Harras. WiGest: A Ubiquitous WiFi-based Gesture Recognition System. In *2015 IEEE Conference on Computer Communications (INFOCOM)*, pages 1472–1480. IEEE, 2015.
- [4] Sandip Agrawal, Ionut Constandache, Shravan Gaonkar, Romit Roy Choudhury, Kevin Caves, and Frank DeRuyter. Using Mobile Phones to Write in Air. In *Proceedings of the 9th international conference on Mobile systems, applications, and services*, pages 15–28. ACM, 2011.
- [5] David W Aha, Dennis Kibler, and Marc K Albert. Instance-Based Learning Algorithms. *Machine learning*, 6(1):37–66, 1991.
- [6] Mohammed Abdulaziz Aide Al-qaness and Fangmin Li. WiGeR: WiFi-based gesture recognition system. *ISPRS International Journal of Geo-Information*, 5(6):92, 2016.
- [7] Kamran Ali, Alex X Liu, Wei Wang, and Muhammad Shahzad. Keystroke Recognition Using WiFi Signals. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 90–102. ACM, 2015.
- [8] Nafiz Arica and Fatos T Yarman-Vural. An Overview of Character Recognition Focused on Off-Line Handwriting. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 31(2):216–233, 2001.

- [9] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The Internet of Things: A Survey. *Computer networks*, 54(15):2787–2805, 2010.
- [10] Serge Belongie, Jitendra Malik, and Jan Puzicha. Shape Matching and Object Recognition Using Shape Contexts. *IEEE transactions on pattern analysis and machine intelligence*, 24(4):509–522, 2002.
- [11] Debnath Bhattacharyya, Rahul Ranjan, Farkhod Alisherov, Minkyu Choi, et al. Biometric Authentication: A Review. *International Journal of u-and e-Service, Science and Technology*, 2(3):13–28, 2009.
- [12] Avrim L Blum and Pat Langley. Selection of relevant features and examples in machine learning. *Artificial intelligence*, 97(1-2):245–271, 1997.
- [13] Chen Chen, Yan Chen, Yi Han, Hung-Quoc Lai, and K.J Ray Liu. Achieving Centimeter-Accuracy Indoor Localization on WiFi Platforms: A Frequency Hopping Approach. *IEEE Internet of Things Journal*, 4(1):111–121, 2016.
- [14] Chen Chen, Yan Chen, Yi Han, Hung-Quoc Lai, Feng Zhang, and K. J. Ray Liu. Achieving Centimeter-Accuracy Indoor Localization on WiFi Platforms: A Multi-Antenna Approach. *IEEE Internet of Things Journal*, 4(1):122–134, 2016.
- [15] Chen Chen, Yi Han, Yan Chen, Hung-Quoc Lai, Feng Zhang, Beibei Wang, and K. J. Ray Liu. TR-BREATH: Time-Reversal Breathing Rate Estimation and Detection. *IEEE Transactions on Biomedical Engineering*, 65(3):489–501, 2017.
- [16] Mingshi Chen, Panlong Yang, Jie Xiong, Maotian Zhang, Youngki Lee, Chaocan Xiang, and Chang Tian. Your Table Can Be an Input Panel: Acoustic-based Device-Free Interaction Recognition. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(1):3, 2019.
- [17] Gregory Cohen, Saeed Afshar, Jonathan Tapson, and André van Schaik. EM-NIST: Extending MNIST to handwritten letters. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 2921–2926. IEEE, 2017.
- [18] Nello Cristianini, John Shawe-Taylor, et al. *An introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge university press, 2000.
- [19] Ilka Dove. Analysis of Radio Propagation Inside the Human Body for in-Body Localization Purposes. Master’s thesis, University of Twente, 2014.
- [20] Shihong Duan, Tianqing Yu, and Jie He. WiDriver: Driver Activity Recognition System Based on WiFi CSI. *International Journal of Wireless Information Networks*, 25(2):146–156, 2018.

- [21] Bin Fang, Cheung H Leung, Yuan Yan Tang, KW Tse, Paul CK Kwok, and YK Wong. Off-line signature verification by the tracking of feature and stroke positions. *Pattern recognition*, 36(1):91–101, 2003.
- [22] Andrew Fort, Julien Ryckaert, Claude Desset, Philippe De Doncker, Piet Wambacq, and Leo Van Biesen. Ultra-Wideband Channel Model for Communication Around the Human Body. *IEEE Journal on Selected Areas in Communications*, 24(4):927–933, 2006.
- [23] Camelia Gabriel, Sami Gabriel, and y E Corthout. The dielectric properties of biological tissues: I. Literature survey. *Physics in medicine & biology*, 41(11):2231, 1996.
- [24] Sami Gabriel, RW Lau, and Camelia Gabriel. The dielectric properties of biological tissues: II. Measurements in the frequency range 10 hz to 20 GHz. *Physics in medicine & biology*, 41(11):2251, 1996.
- [25] Sami Gabriel, RW Lau, and Camelia Gabriel. The dielectric properties of biological tissues: III. Parametric models for the dielectric spectrum of tissues. *Physics in Medicine & Biology*, 41(11):2271, 1996.
- [26] Damien Garcia. Robust smoothing of gridded data in one and higher dimensions with missing values. *Computational statistics & data analysis*, 54(4):1167–1178, 2010.
- [27] Dolores Garcia, Jesus Omar Lacruz, Pablo Jiménez Mateo, and Joerg Widmer. POLAR: Passive object localization with IEEE 802.11 ad using phased antenna arrays. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 1838–1847. IEEE, 2020.
- [28] Zhengxin Guo, Fu Xiao, Biyun Sheng, Huan Fei, and Shui Yu. WiReader: Adaptive Air Handwriting Recognition Based on Commercial Wi-Fi Signal. *IEEE Internet of Things Journal*, 2020.
- [29] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM Computer Communication Review*, 41(1):53–53, 2011.
- [30] Zijun Han, Zhaoming Lu, Xiangming Wen, Jingbo Zhao, Lingchao Guo, and Yue Liu. In-Air Handwriting by Passive Gesture Tracking using Commodity WiFi. *IEEE Communications Letters*, 2020.
- [31] Wenfeng He, Kaishun Wu, Yongpan Zou, and Zhong Ming. WiG: WiFi-based gesture recognition system. In *2015 24th International Conference on Computer Communication and Networks (ICCCN)*, pages 1–7. IEEE, 2015.

- [32] Ashesh Jain, Avi Singh, Hema S Koppula, Shane Soh, and Ashutosh Saxena. Recurrent Neural Networks for Driver Activity Anticipation via Sensory-Fusion Architecture. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3118–3125. IEEE, 2016.
- [33] Lianwen Jin, Duanduan Yang, Li-Xin Zhen, and Jian-Cheng Huang. A Novel Vision-based Finger-Writing Character Recognition System. *Journal of Circuits, Systems, and Computers*, 16(03):421–436, 2007.
- [34] M Keertikumar, M Shubham, and RM Banakar. Evolution of IoT in Smart Vehicles: An Overview. In *2015 International Conference on Green Computing and Internet of Things (ICGCIoT)*, pages 804–809. IEEE, 2015.
- [35] Ahmed E Kosba, Ahmed Saeed, and Moustafa Youssef. Robust WLAN Device-free Passive Motion Detection. In *2012 IEEE Wireless Communications and Networking Conference (WCNC)*, pages 3284–3289. IEEE, 2012.
- [36] Xiaoming Lai and Hans Torp. Interpolation Methods for Time-Delay Estimation Using Cross-Correlation Method for Blood Velocity Measurement. *IEEE transactions on ultrasonics, ferroelectrics, and frequency control*, 46(2):277–290, 1999.
- [37] Hong Li, Wei Yang, Jianxin Wang, Yang Xu, and Liusheng Huang. WiFinger: Talk to your smart devices with finger-grained gesture. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 250–261, 2016.
- [38] Xiang Li, Daqing Zhang, Qin Lv, Jie Xiong, Shengjie Li, Yue Zhang, and Hong Mei. IndoTrack: Device-Free Indoor Human Tracking with Commodity Wi-Fi. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(3):1–22, 2017.
- [39] Jaime Lien, Nicholas Gillian, M Emre Karagozler, Patrick Amihoud, Carsten Schwesig, Erik Olson, Hakim Raja, and Ivan Poupyrev. Soli: Ubiquitous Gesture Sensing with Millimeter Wave Radar. *ACM Transactions on Graphics (TOG)*, 35(4):1–19, 2016.
- [40] Hui Liu, Houshang Darabi, Pat Banerjee, and Jing Liu. Survey of Wireless Indoor Positioning Techniques and Systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 37(6):1067–1080, 2007.
- [41] Jian Liu, Yan Wang, Yingying Chen, Jie Yang, Xu Chen, and Jerry Cheng. Tracking Vital Signs During Sleep Leveraging Off-the-shelf WiFi. In *Proceedings of the 16th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 267–276. ACM, 2015.

- [42] Jiao Liu, Guanlong Teng, and Feng Hong. Human Activity Sensing with Wireless Signals: A Survey. *Sensors*, 20(4):1210, 2020.
- [43] K. J. Ray Liu and Beibei Wang. *Wireless AI: Wireless Sensing, Positioning, IoT, and Communications*. Cambridge University Press, 2019.
- [44] Marcus Liwicki, Alex Graves, Santiago Fernández, Horst Bunke, and Jürgen Schmidhuber. A Novel Approach to On-Line Handwriting Recognition Based on Bidirectional Long Short-Term Memory Networks. In *Proceedings of the 9th International Conference on Document Analysis and Recognition, ICDAR 2007*, 2007.
- [45] Yongsan Ma, Gang Zhou, and Shuangquan Wang. WiFi Sensing with Channel State Information: A Survey. *ACM Computing Surveys (CSUR)*, 52(3):1–36, 2019.
- [46] Eladio Martin, Oriol Vinyals, Gerald Friedland, and Ruzena Bajcsy. Precise Indoor Localization Using Smart Phones. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 787–790. ACM, 2010.
- [47] Gregory Melia. *Electromagnetic Absorption by the Human Body from 1-15 GHz*. PhD thesis, University of York, 2013.
- [48] Kavitha Muthukrishnan, Maria Lijding, Nirvana Meratnia, and Paul Havinga. Sensing motion using spectral and spatial analysis of WLAN RSSI. In *European Conference on Smart Sensing and Context*, pages 62–76. Springer, 2007.
- [49] Rajalakshmi Nandakumar, Vikram Iyer, Desney Tan, and Shyamnath Gollakota. FingerIO: Using Active Sonar for Fine-Grained Finger Tracking. In *Proceedings of the 2016 CHI Conference on Human Factors in Computing Systems*, pages 1515–1525, 2016.
- [50] Santosh Nannuru, Yunpeng Li, Yan Zeng, Mark Coates, and Bo Yang. Radio-Frequency Tomography for Passive Indoor Multitarget Tracking. *IEEE Transactions on Mobile Computing*, 12(12):2322–2333, 2012.
- [51] Eshed Ohn-Bar, Sujitha Martin, Ashish Tawari, and Mohan M Trivedi. Head, Eye, and Hand Patterns for Driver Activity Recognition. In *2014 22nd International Conference on Pattern Recognition*, pages 660–665. IEEE, 2014.
- [52] Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, and Shwetak Patel. Whole-Home Gesture Recognition Using Wireless Signals. In *Proceedings of the 19th annual international conference on Mobile computing & networking*, pages 27–38. ACM, 2013.
- [53] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Kyle Jamieson. Widar: Decimeter-Level Passive Tracking via Velocity Monitoring with

- Commodity Wi-Fi. In *Proceedings of the 18th ACM International Symposium on Mobile Ad Hoc Networking and Computing*, pages 1–10, 2017.
- [54] Kun Qian, Chenshu Wu, Yi Zhang, Guidong Zhang, Zheng Yang, and Yunhao Liu. Widar2. 0: Passive Human Tracking with a Single Wi-Fi Link. In *Proceedings of the 16th Annual International Conference on Mobile Systems, Applications, and Services*, pages 350–361, 2018.
 - [55] Sai Deepika Regani, Beibei Wang, and K. J. Ray Liu. WIFI-Based Device-Free Gesture Recognition Through-the-Wall. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021.
 - [56] Sai Deepika Regani, Beibei Wang, Min Wu, and K. J. Ray Liu. Time Reversal Based Robust Gesture Recognition Using WiFi. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8309–8313. IEEE, 2020.
 - [57] Sai Deepika Regani, Chenshu Wu, Feng Zhang, Beibei Wang, Min Wu, and K. J. Ray Liu. Handwriting Tracking Using 60 GHz mmWave Radar. In *2020 IEEE 6th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE, 2020.
 - [58] Sai Deepika Regani, Qinyi Xu, Beibei Wang, Min Wu, and K. J. Ray Liu. Driver Authentication for Smart Car Using Wireless Sensing. *IEEE Internet of Things Journal*, 7(3):2235–2246, 2019.
 - [59] Mark A Richards. *Fundamentals of Radar Signal Processing*. Tata McGraw-Hill Education, 2005.
 - [60] Alexander Schick, Daniel Morlock, Christoph Amma, Tanja Schultz, and Rainer Stiefelwagen. Vision-Based Handwriting Recognition for Unrestricted Text Input in Mid-Air. In *Proceedings of the 14th ACM international conference on Multimodal interaction*, pages 217–220, 2012.
 - [61] Mervyn Stone. Cross-validatory Choice and Assessment of Statistical Predictions. *Journal of the Royal Statistical Society: Series B (Methodological)*, 36(2):111–133, 1974.
 - [62] Li Sun, Souvik Sen, Dimitrios Koutsonikolas, and Kyu-Han Kim. WiDraw: Enabling Hands-free Drawing in the Air on Commodity WiFi Devices. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 77–89, 2015.
 - [63] L Svilainis and V Dumbrava. Analysis of the interpolation techniques for time-of-flight estimation. *Ultragaras" Ultrasound"*, 63(4):25–29, 2008.

- [64] Sheng Tan and Jie Yang. WiFinger: Leveraging Commodity WiFi for Fine-grained Finger Gesture Recognition. In *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*, pages 201–210, 2016.
- [65] Charles C. Tappert, Ching Y. Suen, and Toru Wakahara. The State of the Art in On-Line Handwriting Recognition. *IEEE Transactions on pattern analysis and machine intelligence*, 12(8):787–808, 1990.
- [66] Steven Thoen, Liesbet Van der Perre, and Marc Engels. Modeling the Channel Time-Variance for Fixed Wireless Communications. *IEEE Communications letters*, 6(8):331–333, 2002.
- [67] Zengshan Tian, Jiacheng Wang, Xiaolong Yang, and Mu Zhou. WiCatch: A Wi-Fi Based Hand Gesture Recognition System. *IEEE Access*, 6:16911–16923, 2018.
- [68] Barry D Van Veen and Kevin M Buckley. Beamforming: A Versatile Approach to Spatial Filtering. *IEEE assp magazine*, 5(2):4–24, 1988.
- [69] Vladimir Vapnik. *Estimation of Dependences Based on Empirical Data*. Springer Science & Business Media, 2006.
- [70] Deepak Vasisht, Swarun Kumar, and Dina Katabi. Decimeter-Level Localization with a Single WiFi Access Point. In *13th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 16)*, pages 165–178, 2016.
- [71] Beibei Wang, Qinyi Xu, Chen Chen, Feng Zhang, and K. J. Ray Liu. The Promise of Radio Analytics: A future paradigm of wireless positioning, tracking, and sensing. *IEEE Signal Processing Magazine*, 35(3):59–80, 2018.
- [72] Chuyu Wang, Jian Liu, Yingying Chen, Hongbo Liu, Lei Xie, Wei Wang, Bingbing He, and Sanglu Lu. Multi-Touch in the Air: Device-Free Finger Tracking and Gesture Recognition via COTS RFID. In *IEEE INFOCOM 2018-IEEE Conference on Computer Communications*, pages 1691–1699. IEEE, 2018.
- [73] Fengyu Wang, Feng Zhang, Chenshu Wu, Beibei Wang, and K. J. Ray Liu. ViMo: Multi-person Vital Sign Monitoring using Commodity Millimeter Wave Radio. *IEEE Internet of Things Journal*, 2020.
- [74] Jue Wang, Deepak Vasisht, and Dina Katabi. RF-IDraw: Virtual Touch Screen in the Air Using RF Signals. In *ACM SIGCOMM Computer Communication Review*, volume 44, pages 235–246. ACM, 2014.
- [75] Wei Wang, Alex X Liu, and Muhammad Shahzad. Gait Recognition Using WiFi Signals. In *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, pages 363–373. ACM, 2016.

- [76] Wei Wang, Alex X Liu, and Ke Sun. Device-Free Gesture Tracking Using Acoustic Signals. In *Proceedings of the 226nd Annual International Conference on Mobile Computing and Networking*, pages 82–94. ACM, 2016.
- [77] Yan Wang, Jian Liu, Yingying Chen, Marco Gruteser, Jie Yang, and Hongbo Liu. E-eyes: Device-free Location-oriented Activity Identification Using Fine-grained WiFi Signatures. In *Proceedings of the 20th annual international conference on Mobile computing and networking*, pages 617–628, 2014.
- [78] Teng Wei and Xinyu Zhang. mTrack: High-Precision Passive Tracking Using Millimeter Wave Radios. In *Proceedings of the 21st Annual International Conference on Mobile Computing and Networking*, pages 117–129. ACM, 2015.
- [79] Chenshu Wu, Feng Zhang, Yuqian Hu, and K. J. Ray Liu. GaitWay: Monitoring and Recognizing Gait Speed Through the Walls. *IEEE Transactions on Mobile Computing*, 2020.
- [80] Chenshu Wu, Feng Zhang, Beibei Wang, and K. J. Ray Liu. mmTrack: Passive Multi-Person Localization Using Commodity Millimeter Wave Radio. In *IEEE INFOCOM*, 2020.
- [81] Chenshu Wu, Feng Zhang, Beibei Wang, and K. J. Ray Liu. mSense: Towards Mobile Material Sensing with a Single Millimeter-Wave Radio. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(3):1–20, 2020.
- [82] Dan Wu, Ruiyang Gao, Youwei Zeng, Jinyi Liu, Leye Wang, Tao Gu, and Daqing Zhang. FingerDraw: Sub-wavelength Level Finger Motion Tracking with WiFi Signals. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 4(1):1–27, 2020.
- [83] Jiang Xiao, Kaishun Wu, Youwen Yi, and Lionel M Ni. FIFS: Fine-grained Indoor Fingerprinting System. In *2012 21st international conference on computer communications and networks (ICCCN)*, pages 1–7. IEEE, 2012.
- [84] Qinyi Xu, Yan Chen, BeiBei Wang, and K. J. Ray Liu. Radio Biometrics: Human Recognition Through a Wall. *IEEE Transactions on Information Forensics and Security*, 12(5):1141–1155, 2017.
- [85] Qinyi Xu, Zoltan Safar, Yi Han, Beibei Wang, and K. J. Ray Liu. Statistical Learning Over Time-Reversal Space for Indoor Monitoring System. *IEEE Internet of Things Journal*, 5(2):970–983, 2018.
- [86] Qinyi Xu, Beibei Wang, Feng Zhang, Deepika Sai Regani, Fengyu Wang, and K.J. Ray Liu. Wireless AI in Smart Car: How Smart a Car Can Be? *IEEE Access*, 8:55091–55112, 2020.

- [87] Moustafa Youssef, Matthew Mah, and Ashok Agrawala. Challenges: Device-free passive localization for wireless environments. In *Proceedings of the 13th annual ACM international conference on Mobile computing and networking*, pages 222–229, 2007.
- [88] Sangki Yun, Yi-Chao Chen, Huihuang Zheng, Lili Qiu, and Wenguang Mao. Strata: Fine-Grained Acoustic-based Device-Free Tracking. In *Proceedings of the 15th annual international conference on mobile systems, applications, and services*, pages 15–28, 2017.
- [89] Feng Zhang, Chen Chen, Beibei Wang, Hung-Quoc Lai, Yi Han, and K. J. Ray Liu. WiBall: A Time-Reversal Focusing Ball Method for Decimeter-Accuracy Indoor Tracking. *IEEE Internet of Things Journal*, 5(5):4031–4041, 2018.
- [90] Feng Zhang, Chen Chen, Beibei Wang, and K. J. Ray Liu. WiSpeed: A Statistical Electromagnetic Approach for Device-Free Indoor Speed Estimation. *IEEE Internet of Things Journal*, 5(3):2163–2177, 2018.
- [91] Feng Zhang, Chenshu Wu, Beibei Wang, Hung-Quoc Lai, Yi Han, and K. J. Ray Liu. WiDetect: Robust Motion Detection with a Statistical Electromagnetic Model. volume 3, page 122. ACM, 2019.
- [92] Jie Zhang, Zhanyong Tang, Meng Li, Dingyi Fang, Petteri Nurmi, and Zheng Wang. CrossSense: Towards Cross-Site and Large-Scale WiFi Sensing. In *Proceedings of the 24th Annual International Conference on Mobile Computing and Networking*, pages 305–320, 2018.
- [93] Yue Zheng, Yi Zhang, Kun Qian, Guidong Zhang, Yunhao Liu, Chenshu Wu, and Zheng Yang. Zero-Effort Cross-Domain Gesture Recognition with Wi-Fi. In *Proceedings of the 17th Annual International Conference on Mobile Systems, Applications, and Services*, pages 313–325, 2019.