ABSTRACT

Title of Dissertation: WIRELESS SENSING
FOR ACTIVITY MONITORING
AND DETECTION

Yuqian Hu
Doctor of Philosophy, 2021

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

In the era of the Internet-of-Things (IoT), billions of smart devices are deployed in indoor environments, connect, share data, and integrate information to fulfill users' needs. Wi-Fi is the ubiquitous communication interface in IoT networks. Inspired by the fact that the Wi-Fi signal can interact with the environment during the propagation, it can extend its role from a communication medium to a wireless sensing tool to perceive human activities in surrounding environments. By analyzing the dynamic components of the Wi-Fi radios introduced by human motion, many applications on activity monitoring and detection are enabled. To contribute to the novel applications of Wi-Fi, this dissertation mainly focuses on passive fall detection, indoor proximity detection, and virtual keyboard implementation for Wi-Fi sensing.

In the first part of this dissertation, we propose a novel Wi-Fi-based environment-independent indoor fall detection system by leveraging the features inherently associated with human falls — the patterns of speed and acceleration over time. The system consists of an offline template-generating stage and an online decision-making stage. In the offline stage, the speed of human falls is first estimated based on the statistical modeling about the channel state information (CSI). Dynamic time warping (DTW) based algorithms are applied to generate a representative template for typical human falls. Then fall event is detected in the online stage by evaluating the similarity between the patterns of real-time speed/acceleration estimates and the representative template. Results of extensive experiments demonstrate the proposed system can achieve consistently high accuracy in time-varying line-of-sight (LOS) and non-line-of-sight (NLOS) environments and can be generalized to new environments without re-training.

In the second and third parts, we investigate the feasibility of detecting motion in proximity robustly and responsively based on a single pair of commercial Wi-Fi devices. We establish the connection between the underlying radio propagation properties and the proposed features. Extensive experiments in various environments validate the efficiency of the devised feature-based detection scheme. Further, we generalize the system to a multi-device structure and conduct experiments under single-user and multi-user sensing scenarios. The results verify the responsive on-device proximity detection can be achieved by combining the information from different links, illustrating its potential for real-time home automation applications.

The last part of the dissertation considers the design of a universal virtual keyboard that reuses a commodity 60 GHz Wi-Fi radio as a radar. By leveraging the unique advantages of 60 GHz Wi-Fi signals, the proposed system can convert any flat surface into an effective typing media and support customized keyboard layouts. We devise a novel signal processing pipeline to detect, segment and separate, and finally recognize keystrokes. The proposed virtual keyboard system enables concurrent keystrokes and does not need any training except for a minimal one-time effort of only three keypresses for keyboard calibration upon the initial setup. Extensive experiments demonstrate a high recognition accuracy for both single-key and multi-key scenarios on different keyboards, presenting the proposed systems as a promising solution to future applications.

WIRELESS SENSING FOR ACTIVITY MONITORING AND DETECTION

by

Yuqian Hu

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
Professor Lawrence C. Washington

Dedication

To my family —

Daojun Hu, and Youlan Xiao

Acknowledgments

I owe my gratitude to all the people who have made this thesis possible and because of whom my graduate experience has been one that I will cherish forever.

First and foremost, I am deeply indebted to my advisor, Prof. K. J. Ray Liu, for his help, encouragement, guidance, and patience. Prof. Liu provided me with invaluable opportunities and constant support during my Ph.D. and acted as a role model to me not only academically but also in life. There are many things Prof. Liu has taught me, including the enthusiasm to push the boundaries of human knowledge, the pursuit of excellence in academics, the perseverance to solve challenging practical problems, and the dedication to make the research ideas a reality to further change the world. I was incredibly lucky to work with and learn from him.

I would like to thank all of my dissertation committee members. I am grateful to Dr. Beibei Wang for her continuous support for my research and kind guidance on my writing. I am also thankful to Prof. Min Wu and Prof. Furong Huang who have provided me with insightful research ideas and suggestions. My sincere thanks also go to Prof. Lawrence C. Washington for his precious time and efforts serving on my dissertation committee and reviewing my thesis.

I would also like to acknowledge help and support from all members of our Signals and Information Group. Thank Dr. Feng Zhang for providing me with so many enlightening suggestions on my research; Thank Dr. Chenshu Wu for his mentorship and collaboration on research. I also wish to give my thanks to my fellow students: Fengyu Wang, Xiaolu Zeng, Sai Deepika Regani, Ruomin Ba, Yusen Fan, Muhammed Zahid Ozturk, and Guozhen Zhu for various thoughtful discussions, support, and the wonderful time during the past few years.

My heartfelt thanks to Origin Wireless Inc. for offering summer internship opportunities to me. With the proprietary facilities and exclusive data at Origin Wireless Inc., I was able to participate in their research programs to develop results to be included in this dissertation. I am grateful to my colleagues, Dr. Hung-Quoc Lai, Dr. Hangfang Zhang, Dr. Zhung-Han Wu, and Dr. Yi Han for their precious support on hardware implementation.

I would like to thank all my family members, who have always encouraged me to explore new directions in life. I owe my deepest thanks to my parents Youlan Xiao and Daojun Hu. Without their unconditional love and strong support, I would not have been able to walk through my overseas graduate years to this important achievement. My particular thanks go to my uncle, Shuhai Xiao, who inspires my academic journey since my childhood, and my aunt, Ling Li for her selfless support all the time and weekly warm greeting calls. I am also grateful to my friends Jun Wang, Yiqi Li, Yexin Cao, Lei Chen, Chen Bai, and Jinjing Han for their friendship and support. Words cannot express the gratitude I owe them. This dissertation is dedicated to them.

Table of Contents

List of Tables

List of Figures

List of Abbreviations

ACF	Autocorrelation Function
AoA	Angle of Arrival
AP	Access Point
CDF	Cumulative Distribution Function
CFO	Channel Frequency Offset
CFR	Channel Frequency Response
CIR	Channel Impulse Response
CNN	Convolutional Neural Network
CSI	Channel State Information
DTW	Dynamic Time Warping
DFS	Doppler Frequency Shift
EM	Electromagnetic
FMCW	Frequency Modulated Continuous Wave
FFT	Fast Fourier Transform
FPR	False Positive Rate
GHz	Gigahertz
IoT	Internet of Things
LOS	Line of Sight
MIMO	Multiple Input Multiple Output
MHz	Megahertz
mmWave	Millimeter-Wave
MUSIC	Multiple Signal Classification
MVDR	Minimum Variance Distortionless Response
NLOS	Non Line of Sight

PCA	Principal Component Analysis
PDP	Power Delay Profile
PWR	Passive Wi-Fi Radar
RF	Radio Frequency
RSS	Received Signal Strength
RSSI	Received Signal Strength Indicator
Rx	Receiver
SFO	Sampling Frequency Offset
STFT	Short-Time Fourier Transform
STO	Symbol Timing Offset
ToA	Time of Arrival
ToF	Time of Flight
TPR	True Positive Rate
TRRS	Time Reversal Resonance Strength
Tx	Transmitter
UWB	Ultra-Wide Band

Chapter 1: Introduction

1.1 Motivation

The Internet of Things (IoT) is transforming our world in tremendous ways. By enabling everyday objects to connect, interact and exchange data with each other, we can automate the integration of the most valuable information from different devices to address specific needs.

Almost all IoT devices communicate through wireless modules such as Wi-Fi [9], Bluetooth [3], Zigbee [10], etc. In addition to providing connectivity among "things", these radio technologies also extend their role from a communication medium to a wireless sensing tool and enable IoT devices with the capability to decipher the physical world by leveraging pervasive wireless signals. This is because radio frequency (RF) signal can be altered by the environment during its propagation and thus contains the information of the surrounding environment.

Among different radio technologies, Wi-Fi is one of the most widely used wireless technologies today in indoor environments and could be the ideal infrastructure to accommodate a large number of IoT devices. It is simple to use, easy to install, and economical. Households in the US have on average 25 connected devices in 2021 and the majority of them use Wi-Fi technology [1]. It has been envisioned that Wi-Fi sensing will become a prominent solution to IoT applications due to its ubiquitous deployment and cost-efficiency [42, 76]. In fact, with the technology advancing, Wi-Fi sensing has supported a variety of features in real-world scenarios such as motion detection [97], vital sign monitoring [99], speed estimation [85, 96], indoor positioning [17, 18] and tracking [86, 95].

Nowadays, as individuals spend the majority of their time in their homes or workplaces, to provide services to occupants and fulfill their needs, understanding the human daily indoor activities becomes essential in modern security systems, smart homes, and health care. In this dissertation, we study activity monitoring and detection using Wi-Fi sensing. More specifically, we design three important IoT applications - *fall detection* [29, 30], *proximity detection* [26], and *virtual keyboard* [27, 28].

1. *Fall detection* is important to seniors who live alone in their homes. Fall is recognized as one of the most frequent accidents among elderly people, which can cause severe injuries and even death [4]. The damage caused by the falls is not only reflected in the immediate injury of the body, but also in all subsequent adverse effects caused by the lack of timely assistance. Therefore, a real-time indoor fall detection system with timely and automatic alarms is highly in need, which could potentially save lives by requesting external help without delay.

2. *Proximity detection* through motion sensing has gained much attention recently. It can not only detect the presence of the moving target but also provide

location information. It plays a fundamental role in many applications, including security surveillance, intruder monitoring, area-aware home automation, etc. In addition, proximity detection can provide important location-related context to support further indoor activity recognition.

3. *Virtual keyboard* has been greatly demanded as a handy substitute for ordinary physical keyboards. Keyboard, as the primary and most integrated computer peripheral, has become an indispensable part of our daily lives. However, physical keyboards have been suffering from poor portability issue. Additionally, as IoT devices go smaller, they are typically not feasible to have a bulky physical keyboard. Therefore, virtual keyboard has gained increasing popularity in recent years to enable typing experience for billions of IoT devices without a keyboard.

1.2 Related Works

The related works of this dissertation cover indoor activity recognition, motion detection and localization, and virtual keyboard implementation, which are reviewed in the following subsections, respectively.

1.2.1 Indoor Activity Recognition and Fall Detection

Existing activity recognition techniques can be roughly divided into two categories: active and passive systems. The active or device-based techniques require users to wear special devices, including ECG sensors, pressure sensors, accelerometers, gyroscopes, smartwatches, and smartphones, etc., to track the motion of their bodies. However, in addition to the potential false alarms of wearable systems [47], it needs users' continuous cooperation and is not suitable for home security applications. Also, it is cumbersome and sometimes impractical to ask users especially elder people to carry specialized sensors for healthcare applications [39], which motivates the development of passive or device-free systems. The most common passive systems are vision-based [45]. Typically, an array of cameras, infrared sensors, or depth cameras like Kinect need to be deployed to monitor an area of interest. While high accuracy could be achieved under favored settings of good lighting conditions and a clear field of view, vision-based systems are limited by the visibility requirement and also bring privacy concerns, especially in some specific environments such as the bathrooms and bedrooms. Acoustic-based methods have also been used to monitor activities [62], but they usually experience weak echo audio signal and ambient noise, and are not widely deployed in practical applications.

On the other hand, wireless sensing is an innovative approach to capture activities in a non-obtrusive way while overcoming the aforementioned limitations. Existing works on passive wireless sensing can be categorized into different groups based on the features extracted from the wireless channel as Table 1.1 shows: radar-based, received signal strength indicator (RSSI) based, and channel state information (CSI) based systems.

Radar-based systems detect events relying on specialized devices that are not readily available in homes. Many of them infer the motion of the reflector by evaluating the Doppler frequency shift and extract micro-Doppler signatures for recognizing

Methodology	Existing works	Limitations
Radar-based	Indoor activity recognition [11, 61, 67] Fall detection [13, 44, 49, 70]	Limited detection range Cannot be integrated with commercial Wi-Fi Specialized device required Doppler signature affected by moving direction
RSSI-based	Indoor activity recognition [23, 68] Fall detection [32]	Coarse granularity High sensitivity to environmental changes Multiple devices required
CSI-based	Indoor activity recognition [14, 81, 92] Fall detection [53, 78, 83, 101]	Re-training required in new environments Performance degrades with LOS path blocked
PWR	Indoor activity recognition [19] Human sensing [21, 37]	Extra peripherals and directional antennas required Doppler signature affected by moving direction Stable reference channel required

Table 1.1: Related works on activity recognition using wireless sensing

finer body movements, such as activity monitoring [11, 67], gesture recognition [34], and fall detectors [13, 70]. However, although radars may have higher range or frequency resolutions, they usually require line-of-sight (LOS) for reliable operation. Therefore they can only monitor activities in very limited coverage and are also limited by the requirement of extra specialized dedicated devices. Further, the speed estimation derived from Doppler shift by radars varies for different moving directions and the heading direction of the subject is usually predefined [85].

Some other works on activity recognition use commodity devices. Commercial wireless devices, such as Wi-Fi infrastructures, are available in most indoor spaces and allow more flexible low-cost deployment. Chetty *et al.* apply passive Wi-Fi radar (PWR) to realize through-the-wall human sensing [21] and overcome the coverage limitation of traditional radars. However, PWR, which also relies on the principle of radars, collects Doppler information and therefore is also direction/location-dependent and requires directional antennas [37].

Another widely-used method is using the RSSI to characterize indoor activ-

Reference	Features	Antenna	Detection	False alarm	Environment-independent
WiFall [83]	Variance	3Tx/3Rx	87%	18%	✗
RT-Fall [78]	Phase difference	1Tx/2Rx	91%	11%	✗
FallDeFi [53]	Spectrum	2Tx/2Rx	94.33%	14.92%	✗
TL-Fall [101]	Frequency	1Tx/3Rx	86.83%	15.29%	✗
DeFall	Speed	1Tx/1Rx	95.80%	1.47%	✓

Table 1.2: A brief summary of different CSI-based approaches to fall detection and the claimed performance.

ities, either on Wi-Fi or other wireless devices. For example, Gu *et al.* explore Wi-Fi ambient signals for RSSI fingerprint of different activities [23]. However, since RSSI measures the overall amplitude response of multiple superposed subcarriers, it loses the phase information as well as the detailed information of each frequency component. Therefore it suffers from dramatic performance degradation in complex situations due to multi-path fading and temporal dynamics [93], limiting its stability and reliability in practical applications.

CSI, which measures both amplitude and phase information on different frequency components, provides finer-grained information for a propagation environment and has become popular in the field of wireless sensing recently. Different from the mechanism of PWR, CSI is the standard information that can be extracted from commodity Wi-Fi devices. By analyzing CSI accessible on mainstream devices nowadays, one could monitor indoor activities and detect indoor events, such as Wi-chase [14], CRAM [81] and TRIEDS [92].

Fall detection gains increasing attention among various activity recognition applications [82]. Many CSI-based fall detection systems have been implemented [53, 60, 78, 83]. WiFall [83] detects falls by extracting features from CSI amplitude information, while RT-Fall [78] exploits the efficacy of phase difference for activity segmentation and fall detection. FallDeFi [53] performs time-frequency analysis using short-time Fourier transform (STFT) to detect falls. Unfortunately, as illustrated in Table. 1.2, since the features extracted in the existing Wi-Fi based fall detection systems mentioned above are environment-dependent, the trained classifiers in these works suffer from the impact of environmental changes and cannot be generalized well to new environments without performance degradation. Re-training is required in these systems when the environmental settings change, with the undesirable complication that users would be asked to fall and collect training data every time the placements of furniture or the deployments of the devices get changed. To address these challenges, in this dissertation we propose *DeFall* that explores the inherent environment-independent features of a fall and can be put into use once deployed in any new environment without any re-training or calibration [29, 30].

1.2.2 Indoor Motion Detection and Localization

Indoor motion detectors are an important component of IoT security and automation applications. Most of the existing CSI-based methods detect motion by capturing the temporal variations caused by the moving object and extract time-domain features, such as mean and variance of CSI amplitude [80], the variance distribution of CSI amplitude [107] and the variance of amplitude variance over subcarriers [43]. Other works [90, 96, 97] exploit the auto-correlation of the CSI over time. PILOT [90] decomposes the CSI amplitude correlation matrix using singular value decomposition (SVD) and monitors the variations of the singular vectors. WiDetect [97] explores the efficiency of autocorrelation function (ACF) of CSI power with a single time lag, while WiSpeed [96] makes use of more lags to detect motion and estimate speed. In addition, many other works detect motion by exploiting the subcarrier diversity in the frequency domain. For example, R-TTWD [106] extracts the first-order difference of eigenvector of CSI amplitudes across subcarriers, while PADS [57] uses the maximum eigenvalues of covariance matrices from normalized amplitude among subcarriers as one of the features to detect human movements.

The location-related information is also important in real-world applications for indoor motion sensing. However, most of the aforementioned motion detection works only aim at detecting the dynamic object in a predefined large area without being able to capture the location information of the motion, which inspires the development of CSI-based motion localization schemes. Many of the existing localization systems rely on a database storing the CSI fingerprints and the target is localized by matching the newly collected CSI features with the stored profiles [90, 104], which requires laborious training and is not easy to be generalized to new environments due to the environment-dependent and location-dependent fingerprints. Other works build geometric models to infer the angle-of-arrival (AoA) or time-of-flight (ToF) information [58, 102] and usually require dedicated calibration or multiple transceivers with a specific geometric arrangement, which prevents their practical usage in real-life scenarios.

Modality	Reference	Method	Hardware	Training
Vision	Murase *et al.* [50]	Contour extraction, Real-Adaboost	Camera	Yes
	Ji *et al.* [31]	Contour feature restriction	RGB camera	No
	CamK [94]	Shape-based fingertip tracking	Camera	Yes
	Su *et al.* [71]	Morphology processing, ellipse fitting	Image sensor	No
Acoustic	Zhuang *et al.* [109]	MFCC, HMM, linear classification	Microphone	Yes
	Zhu *et al.* [108]	Time difference of arrival (TDoA)	Smart phone	No
	UbiK [79]	Multipath fading of audio signals	Smart phone	Yes
Ambient	Marquardt *et al.* [46]	Accelerometer, FFT, MFCC	Smart phone	Yes
	VibKeyboard [41]	Power spectral density, SVM	Vibration sensor	Yes
Wearable	Zhao *et al.* [103]	Angle complementary filter, kNN	mIMU	Yes
	Wu *et al.* [89]	Velocity and acceleration measurement	Pressure sensor	Yes
	iKey [20]	MFCC, class-center classification	Vibration sensor	Yes
	Scherer *et al.* [64]	LDA, feedback training	EEG sensor	Yes
RF-based	WiKey [12]	PCA, DWT, DTW, kNN	2 × 3 transceivers	Yes
	Windtalker [48]	DWT, PCA, STFT, DTW	Directional antenna	Yes
	Chen *et al.* [16]	FIR filter, phase/amplitude matching	1 × 2 transceivers	Yes
	SpiderMon [40]	PCA, Wavelet decomposition, SVM, HMM	Directional antenna	Yes
	WiPass [100]	Symlet filter, DCASW, DTW	1 × 2 transceivers	Yes

Table 1.3: Summary of related works on virtual keyboard systems/keystroke recognition

1.2.3 Keystroke Recognition and Virtual Keyboard Implementation

As a portable alternative to physical keyboards, various virtual keyboard systems based on different modalities have been proposed as summarized in Table 1.3. The existing virtual keyboard systems can be implemented in an active or passive manner.

The active systems integrate the sensors into the specialized wearables such as glove [89], wristband [20], and ring [103] to track the motion of fingers. Considering user convenience, passive approaches have gained more attention over the years and great effort has been made to implement passive virtual keyboards in both academia and industry.

Vision-based methods usually employ cameras to detect and localize the keystroke by shape-based fingertip tracking [31, 94]. Although these approaches can achieve high accuracy with the significant advances of technology and algorithms in the

Modality	Limitation
Vision	Requirement of good visibility Specialized devices Privacy issue
Acoustic	Applicable for only single-key keystroke High sensitivity to ambient sounds Specialized devices
Ambient	Specialized devices Applicable for only single-key keystroke
Wearable	High false alarm rate Inconvenience to users Specialized devices
RF-based	Re-training required in new environments Applicable for only single-key keystroke Non-portable devices

Table 1.4: General limitations of the existing works on virtual keyboard/keystroke recognition.

field of computer vision, they are limited by their privacy invasion and the requirement for ambient light. Besides traditional RGB cameras, optics-based sensors such as LiDAR and depth sensors in Kinect have also been integrated for virtual input [15, 71, 72]. However, LiDAR is too expensive for home use and lacks strict international protocols that guide data collection and processing.

Acoustic- and ambient-based sensing have also been considered to enable passive virtual keyboard implementation. Some researchers [79, 108, 109] classify the acoustic signals when typing different keys for keystroke recognition while others [41, 46] use either the accelerometer in mobile phones or vibration sensor to capture and decode the vibrations from nearby keystrokes. Nevertheless, the sensitivity to ambient sounds or vibrations prevents these approaches from being widely deployed in practical applications.

Several recent studies [12, 16, 48] have demonstrated the potential of using the

2.4 GHz/5 GHz wireless radios to distinguish different keystrokes. More specifically, by analyzing the unique patterns of CSI when pressing different keys, Ali *et al.* [12] explore the feasibility of using 2.4 GHz Wi-Fi radios for keystroke recognition. Chen *et al.* [16] localizes the keystrokes by matching and canceling the signal amplitude/phase over different antennas. Extracting features in both time and frequency domains, Windtalker [48] utilizes network layer traffic information and physical layer CSI information to recognize keystrokes. Limited by the fundamental characteristics of 2.4 GHz/5 GHz signals, however, all of the implementations mentioned above can only work for a single keystroke, are vulnerable to surrounding motion interference, and require significant effort for training and learning, preventing them from broader generalization to new keyboards or new environments. Recently, 60 GHz Wi-Fi technology has emerged to enable fine-grained applications including keystroke recognition. In this dissertation, we reveal the possibility of using a single 60 GHz Wi-Fi radio for universal virtual keyboards with minimal one-time calibration effort.

1.3 Dissertation Outline and Contributions

Considering the limitations of current studies discussed in Section 1.2 and the significance of using Wi-Fi signals to accomplish IoT applications in activity monitoring, we are motivated to develop new Wi-Fi sensing techniques that not only fully utilize the information embedded in indoor multipath environments, but also support simple implementation with commercial Wi-Fi devices.

In this dissertation, we first introduce the primer of wireless sensing in Chapter

2, including the propagation characteristics of Wi-Fi signals at different frequency bands and the specific approaches to harness them. Then, we present different indoor monitoring systems, that is, a passive indoor fall detection system in Chapter 3, a proximity detector and its home automation application in Chapter 4 and Chapter 5 respectively, and a virtual keyboard system in Chapter 6. Chapter 7 concludes the dissertation.

The contributions and outline of Chapter 3-6 are described as follows.

1.3.1 Environment-Independent Passive Fall Detection using Wi-Fi (Chapter 3)

In this chapter, we propose *DeFall*, a Wi-Fi based passive fall detection system that is independent of the environment and free of prior training in new environments. Unlike previous environment-dependent works, our key insight is to probe the physiological features inherently associated with human falls, i.e., the distinctive patterns of speed and acceleration during a fall. *DeFall* consists of an offline template-generating stage and an online decision-making stage, both taking the speed estimates as input. In the offline stage, augmented dynamic time warping (DTW) algorithms are performed to generate a representative template of the speed and acceleration patterns for a typical human fall. In the online phase, we compare the patterns of the real-time speed/acceleration estimates against the template to detect falls. To evaluate the performance of *DeFall*, we build a prototype using commercial Wi-Fi devices and conduct experiments under different settings. The results demonstrate that *DeFall* achieves a detection rate above 95% with a false alarm rate lower than 1.50% under both line-of-sight (LOS) and non-LOS (NLOS) scenarios with one single pair of transceivers. The extensive comparison study verifies that *DeFall* can be generalized well to new environments without any new training.

1.3.2 Robust Passive Proximity Detection using Wi-Fi (Chapter 4)

In this chapter, we design a reliable and effective proximity detector to distinguish the nearby and faraway motions. To achieve this, we propose two robust proximity detection metrics, correlation, and covariance, over the adjacent subcarriers which are extracted from the CSI available on mainstream Wi-Fi devices. Rather than a data-driven scheme, our features are derived from in-depth insight into the fundamentals of RF propagation. Specifically, we explore the physics behind the practical indoor multipath propagation and investigate the relationship between the human-device distance and the correlation/covariance of CSI power response over subcarriers in the frequency domain. We then demonstrate the feasibility of the proposed features in detecting the motion in the proximity of devices with extensive experiments. Experimental results in various real-world scenarios show that the proposed scheme can achieve true positive rates (TPR) greater than 95% and 99% in distance-based and room-level proximity detection, respectively, while maintaining the corresponding false positive rates (FPR) less than 5% and 0.5%. The detection delays for a detection distance of 2 m are within 0.6 s, which verifies the responsiveness of the proposed scheme.

1.3.3 Home Automation Applications via On-Device Proximity Detection (Chapter 5)

In this chapter, we extend the structure of a single-receiver proximity detector to a multi-device architecture and facilitate the home automation application via on-device proximity detection. To address the problem of hardware diversity, we devise an offline training phase to adaptively determine the thresholds of proximity state and silent state, which are fed into real-time monitoring. In the online monitoring phase, the links between the central receiver(Rx) and different IoT devices are integrated for a final decision on the device action. The experiments verify that given the number of targets smaller than the number of devices of interest, the system can detect the motion of multiple users in the proximity of the devices responsively within 0.4s. Besides the real-time proximity detection on IoT devices, the corresponding location log can be generated to provide information for further home automation applications.

1.3.4 Universal Virtual Keyboard using A Single Millimeter Wave Radio (Chapter 6)

In this chapter, We design *mmKey*, the first virtual keyboard system using a single 60 GHz Wi-Fi radio. With minimal infrastructure support, *mmKey* can turn any flat surface, with a printed paper keyboard, into an effective interactive tool. We present a novel signal processing pipeline to detect, segment, and recognize both single keystroke and multi-finger concurrent keystrokes without any training. We prototype *mmKey* by reusing a commodity 60 GHz Wi-Fi radio as a millimeter-wave (mmWave) radar and validate the performance of *mmKey* by extensive experiments on three different virtual keyboards, including a computer keyboard, a piano keyboard, and a phone keypad. We conduct experiments at different locations in both home and office environments, with ten volunteers involved. Experimental results demonstrate a remarkable accuracy of $> 95\%$ for single-keystroke scenario and $> 90\%$ for multiple concurrent keystrokes. Furthermore, by feeding *mmKey*'s output to commercial text correction tools, we achieve a considerable word recognition accuracy of $> 97\%$ for natural typing on a printed computer keyboard. With the great performance, *mmKey* promises universal virtual keyboards for computers, mobiles, wearables, and IoT devices, should they be equipped with a mmWave radio.

Chapter 2: Primer of Wireless Sensing

Wireless sensing technology combines perception and communication together. It has three distinct characteristics: 1) *Sensorless* as no special sensors are required except the wireless signals; 2) *Wireless* as there is no need to deploy the physical wired connection; 3) *Contactless*, a.k.a. passive, as there is no need for users to wear any devices [105]. Due to the wide deployment of Wi-Fi infrastructure, Wi-Fi sensing has become particularly attractive and extensive research on passive Wi-Fi sensing has spawned. The IEEE 802.11 standard specifies various bands in the RF spectrum that can be used for Wi-Fi, including but not limited to 2.4 GHz, 5 GHz, and 60 GHz. The radio signals at different frequency bands exhibit distinct propagation characteristics, which enable them to capture the environmental dynamics at different scale levels and inspire different applications. In this chapter, we introduce the model of the propagation of radio signals in practical systems.

2.1 2.4 GHz/5 GHz Wi-Fi

The conventional 2.4 GHz or 5 GHz Wi-Fi is the most well-known and most widely-deployed wireless networking technology, given its widespread use in home and office settings. CSI, which is now accessible in mainstream Wi-Fi devices, has been used for different sensing purposes.

2.1.1 WiFi Sensing with CSI

During the wireless radio propagation indoors, radio waves arrive at the receiver (Rx) over multiple paths due to reflection, diffraction, and scattering caused by walls, furniture, and human bodies in the indoor environment. As the signal propagates through these multiple paths, the environment information is contained in the received signal. With $L(t)$ independent clusters of multipath components, the multipath channel $h(t)$ at time t, a.k.a. channel impulse response (CIR), is given by

$$h(t, \tau) = \sum_{l=1}^{L(t)} \beta_l(t)\delta(t - \tau_l(t)), \tag{2.1}$$

where β_l refers to the coefficient of the l-th multipath component (MPC) and τ_l is the time delay associated with β_l. $\delta(\cdot)$ denotes the Dirac delta function [73]. The propagation delay is a function of the propagation distance $\tau_l(t) = \frac{d_l(t)}{c}$, where c is the speed of light and $d_l(t)$ is the traveled distance of the l-th MPC.

In practice, to mitigate the inter-symbol-interference (ISI), 2.4 GHz/5 GHz WiFi system is built on the orthogonal frequency-division multiplexing (OFDM) communication scheme, in which data are transmitted on multiple subcarriers in parallel. Different subcarriers have different frequencies. Therefore, although the wireless channel generally uses CIR expressed in 2.1 to describe the multipath effect, the commercial Wi-Fi devices usually obtain the propagation information via estimating channel frequency response (CFR) in the frequency domain, which is the discrete Fourier transform (DFT) of CIR and takes the form of

$$H(t,f) = \sum_{l=1}^{L(t)} \beta_l(t) e^{-j2\pi f \tau_l(t)}, \tag{2.2}$$

where f, in the OFDM-based Wi-Fi system, denotes the particular frequency of each subcarrier where the channel is measured. The CSI refers to the CFR equivalently.

2.1.2 Decomposition of CSI

CSI describes the multipath propagation of the signals from a transmitter (Tx) to an Rx. The total of $L(t)$ MPCs of the signal propagation can be classified into two sets: $L_s(t)$ and $L_d(t)$. $L_s(t)$ is the set of the time-invariant MPCs that are reflected or scattered off the static objects, e.g., walls and furniture, while $L_d(t)$ denotes the time-varying MPCs with paths altered by the moving subjects, e.g., dynamic gestures and walking human. Based on the superposition principle, we then decompose $H(t,f)$ into a sum of signals contributed by different MPCs as

$$H(t,f) = \sum_{i \in L_s(t)} H_i(t,f) + \sum_{j \in L_d(t)} H_j(t,f) + \varepsilon(t,f), \tag{2.3}$$

where $\varepsilon(t,f)$ is the additive thermal noise. Within a sufficiently short period, it is reasonable to assume that the static component $H_i(t,f), i \in L_s(t)$ changes slowly in time. Then, defining $H_s(f) \triangleq \sum_{i \in L_s} H_i(f)$ and $H_d(t,f) \triangleq \sum_{j \in L_d(t)} H_j(t,f)$, we have the following approximation:

$$H(t, f) = H_s(f) + H_d(t, f) + \varepsilon(t, f). \tag{2.4}$$

Without the loss of generality, we can denote the combined time-varying components $H_d(t, f)$ and $\varepsilon(t, f)$ as

$$\tilde{H}_d(t, f) = H_d(t, f) + \varepsilon(t, f), \tag{2.5}$$

where $H_d(t, f)$ is the superimposed signal contributed by a large number of multipath dynamic rays with random phases, which, through the Central Limit Theorem, follows a circularly symmetric Gaussian distribution with zero mean and the same variance for both real and imaginary parts. Meanwhile, $H_d(t, f)$ has a Rayleigh distributed amplitude and uniformly distributed phase. In addition, the real and imaginary parts of $\varepsilon(t, f)$ can be assumed as white Gaussian. Since $H_d(t, f)$ and $\varepsilon(t, f)$ are from independent sources, it can be assumed that the overall dynamic $\tilde{H}_d(t, f)$ tends to be circularly symmetric Gaussian. Then the received CSI can be further represented as

$$H(t, f) = H_s(f) + \tilde{H}_d(t, f). \tag{2.6}$$

By extracting and analyzing the dynamic component $\tilde{H}_d(t, f)$, the indoor Wi-Fi sensing systems are capable of capturing activities in the environment through the changes in the received CSI.

In practice, however, the measured $H(t, f)$ suffers from severe synchronization errors, including carrier frequency offset (CFO), sampling frequency offset (SFO), symbol timing offset (STO) [17, 18, 91]. Both SFO and STO introduce linear phase offset which grows with the subcarrier index, and CFO introduces initial phase offset which is constant for each OFDM symbol. That is, the estimated raw CSI $H_{\text{raw}}(t, f_k)$ on the k-th subcarrier can be represented as

$$H_{\text{raw}}(t, f_k) = H(t, f_k)e^{-j(k\phi_{\text{linear}}(t)+\phi_{\text{ini}}(t))}, \tag{2.7}$$

where $\phi_{\text{linear}}(t)$ denotes the slope of the linear phase offset and $\phi_{\text{ini}}(t)$ is the initial phase offset, both of which are time-varying and different for each CSI. These synchronization errors significantly distort CSI phase, which causes difficulties in the extraction of the complex-valued $\tilde{H}_d(t, f)$ and thus limit its practical usage in real-world applications. To reduce or remove the effect of phase distortions, existing works usually rely on *phase calibration* or *phase elimination*.

2.1.3 CSI Phase Calibration and Elimination

Phase calibration of CSI In literature, there are two commonly used methods to calibrate the CSI phase: linear regression [35, 57, 66] and conjugate multiplication [36, 58, 59]. The linear regression based method aligns the phase by estimating the $\phi_{\text{linear}}(t)$ and $\phi_{\text{ini}}(t)$, while the initial phase $\phi_{\text{ini}}(t)$ cannot be completely removed as it is mixed with the original common phase. The conjugate multiplication method exploits the fact that different antennas on the same Wi-Fi card share the same RF oscillator and thus have the same time-varying random phase offsets. When the conjugate multiplication is applied between two antennas, the phase offsets are canceled out, which, however, also alters the signal phase and fails to recover the original CSI. In Chapter 4, we adopt the linear regression method to compensate the linear phase difference $\phi_{\text{linear}}(t)$ and cancel the $\phi_{\text{ini}}(t)$ for validation.

Phase elimination of CSI Since the impact of phase distortion on CSI amplitude is negligible, many works [96, 97] eliminate phase noises by calculating CSI power $G(t, f)$ of a CSI measurement, which is defined as the square of the CSI amplitude, taking the form of

$$\begin{aligned}
G_{\text{raw}}(t,f) &\triangleq |H_{\text{raw}}(t,f)|^2 \\
&= |H_s(f) + H_d(t,f) + \varepsilon(t,f)|^2 \\
&= |H_s(f) + H_d(t,f)|^2 + |\varepsilon(t,f)|^2 + 2\text{Re}\big\{\varepsilon^*(t,f)\big(H_s(f) + \tilde{H}_d(t,f)\big)\big\} \\
&= |H_s(f) + H_d(t,f)|^2 + \epsilon(t,f),
\end{aligned} \tag{2.8}$$

with the superscript * denoting the operator of complex conjugate. The operator $\text{Re}\{\cdot\}$ denotes the real part, and $\epsilon(t, f)$ is defined as the noise term. With the fact that the magnitude of thermal noise $\varepsilon(t, f)$ is usually much smaller than that of CSI $H(t, f)$, the term $\epsilon(t, f)$ can be approximated as additive white Gaussian noise (AWGN) with variance $\sigma(f)^2$ and is statistically independent of $H_s(f) + H_d(t, f)$.

Based on Eqn. (2.6), the power response can be also calculated as

$$
\begin{aligned}
G_{\text{raw}}(t,f) &\triangleq |H_{\text{raw}}(t,f)|^2 \\
&= |H_s(f) + \tilde{H}_d(t,f)|^2 \\
&= |H_s(f)|^2 + |\tilde{H}_d(t,f)|^2 + H_s^*(f)\tilde{H}_d(t,f) + H_s(f)\tilde{H}_d^*(t,f) \\
&= |H_s(f)|^2 + |\tilde{H}_d(t,f)|^2 + 2\text{Re}\{H_s^*(f)\tilde{H}_d(t,f)\}.
\end{aligned}
\tag{2.9}
$$

In both Chapter 3 and Chapter 4, we exploit the measured CSI power response sequences over time for sensing.

2.1.4 CSI Preprocessing

Normalization of CSI power response The raw CSI samples are collected by commercial Wi-Fi devices. As the automatic gain controller affects the reported CSI amplitude, we preprocess the CSI sequence on the k-th subcarrier by normalizing the sample power response over the whole frequency band as

$$
G(t,f_k) = \frac{|H_{\text{raw}}(t,f_k)|^2}{\sum_{k=1}^{N_s} |H_{\text{raw}}(t,f_k)|^2}, \text{ for all } k, \tag{2.10}
$$

where N_s is the number of subcarriers, and H_{raw} is the reported CSI.

Outlier removal After normalization, to sanitize the CSI power response sequence, the well-known Hampel filter is applied for adaptively detecting and removing the outliers [22]. Specifically, Hampel filter computes the median of a sliding window composed of the samples, recognizes the data points that are far enough from the median as outliers, and replaces them with the median, which could be formulated as

$$G(t, f_k) = \begin{cases} G(t, f_k), & |G(t, f_k) - m_{t,k}| \leq \mu S_{t,k}, \\ m_{t,k}, & |G(t, f_k) - m_{t,k}| > \mu S_{t,k}, \end{cases} \tag{2.11}$$

where $m_{t,k}$ is the median value from the moving data window $\{G(t-j, f_k), j \in [-T, T]\}$ and $S_{t,k}$ is the median absolute deviation (MAD) scale estimate, defined as

$$S_{t,k} = 1.4826 \times \text{median}_{j \in [-T,T]}\{|G(t-j, f_k) - m_{t,k}|\}. \tag{2.12}$$

In both Chapter 3 and Chapter 4, we utilize the same CSI preprocessing.

2.2 60 GHz Wi-Fi

Nowadays, the increase in bandwidth-hungry wireless applications such as high-definition uncompressed video streaming, large file transfer, and wireless display, has driven demand for technologies that can support wider bandwidth [75] and therefore leads to the emergence of 60 GHz networking radios (e.g., 802.11ad/ay [6]) as the next-generation wireless communication technique, which allows efficient high-speed throughput (up-to several gigabits) with very low latency. While 60 GHz Wi-Fi is originally standardized communication, it also offers distinct advantages for sensing applications compared with 2.4 GHz/5 GHz bands: millimeter-wavelength on a high-frequency band, finer range resolution by large bandwidth, and highly directional beamforming enabled with a large phased array. When combined with signal processing and machine learning techniques, 60 GHz radios can identify objects, movements, and precise locations. This capability can be harnessed for finer-resolution wireless sensing applications including target material sensing [88], multi-user tracking [87], hand-writing interaction [61], vital sign monitoring [77] and keystroke recognition [27, 28].

Recently, with the rapid development of technologies, 60 GHz radios have become available on commodity networking devices and are being integrated into smartphones [2, 7] and cars [8]. With the increasing deployment and the extraordinary sensing capability, 60 GHz radios hold enormous potential to take wireless beyond speeds and support a new class of user experiences.

2.2.1 CIR on 60 GHz Radio

As shown in Fig. 2.1, our experimental system is built upon a testbed provided by Qualcomm, which enables a radar mode on a commodity 802.11ad chipset by attaching an additional antenna array to the chipset. With this, the 60 GHz radio, under the radar mode, can be transmitted and received on a single networking device and capture the channel response for precise sensing [98]. The Tx and Rx arrays both have 32 antennas assembled in a 6 × 6 layout. We use a coordinate system as illustrated in Fig. 2.1, where the reflected signals impinge on the Rx antenna array with different azimuths φ and elevations θ.

To extract CIR, each Tx antenna transmits a burst consisting of a group of 32 pulses, which are then received by 32 Rx antennas sequentially, and the correspond-

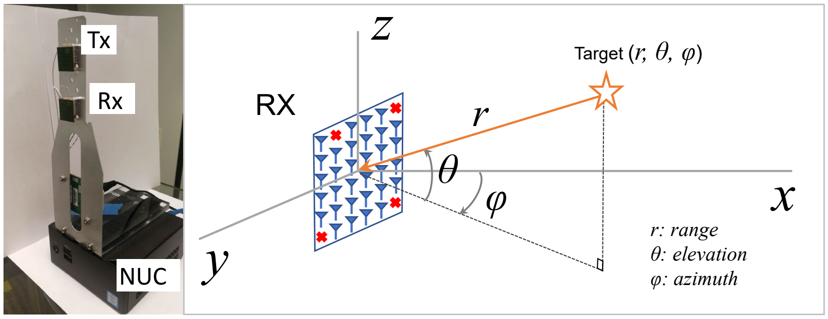


Figure 2.1: Device and coordinate system. The antenna array contains 32 elements in a 6 × 6 layout, with 4 reserved locations marked by red crosses.

ing CIR is recorded. The duration of each pulse is $T_p = 10\,\mu s$, and the duration of each burst is $T_b = 10\,\text{ms}$. By analyzing the received signals, we can monitor the activities including keystrokes and hand movements. The user can decide the number of Tx antennas to use. In the mode of N_{tx} working Tx antennas, each Tx antenna takes a burst T_b to transmit and it takes $N_{\text{tx}} \times T_b$ to finish one period of transmission. The concept of "tap" is adopted for differentiating targets at different ranges. To understand "tap", consider a pair of Tx and Rx antennas and two reflectors (target 1 and target 2) at different distances as Fig. 2.3 shows. For each pulse, the two reflections from target 1 and target 2 result in different time-of-arrival (ToA) due to different path lengths. A single impulse tap represents the smallest ToA difference that can be separated. Note that our device uses a bandwidth of 3.52 GHz, leading to a ToA resolution of 0.28 ns. That is, signals whose propagation delays differ by greater than 0.28 ns are recorded on different taps, which corresponds to a range resolution of 4.26 cm (for reflecting paths). Therefore, the CIR tap gives an estimate of the range of the reflector, and each tap is equivalent to a range difference of 4.26 cm.

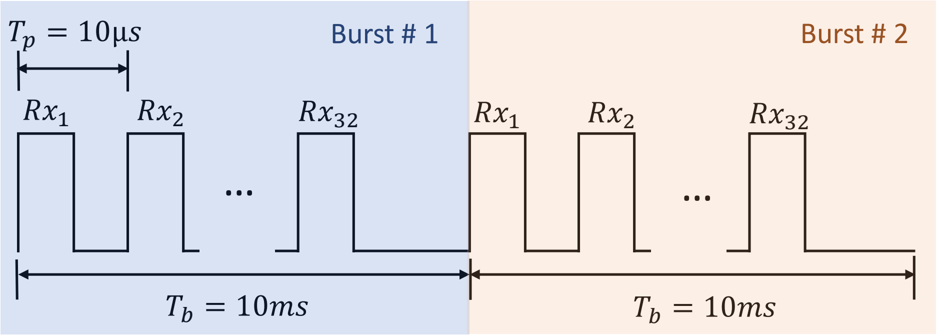


Figure 2.2: Frame structure in mmWave radio.

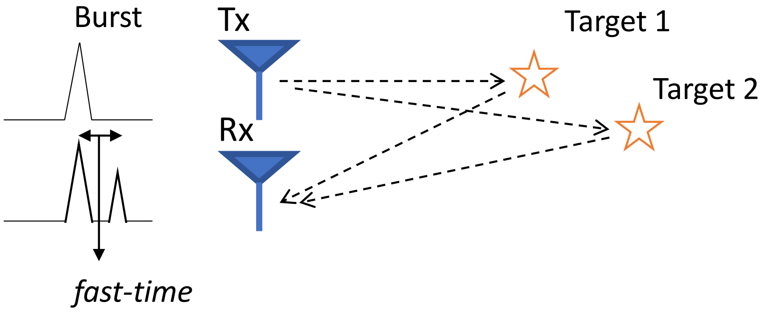


Figure 2.3: Concept of the range tap.

In our experiments, we use 1 Tx antenna and a total of 32 Rx antennas. The CIR reported by the n-th Rx antenna at time slot t can be expressed as

$$h_n(t) = \sum_{l=0}^{L-1} g_{n,l}(t)\delta(t - \tau_l), \tag{2.13}$$

where L is the number of range taps. $\delta(\cdot)$ is the Delta function which represents the unit impulse. $g_{n,l}$ and τ_l are complex channel gain and propagation delay of the l-th range tap, respectively. Denote the number of antennas as N, for each time slot t, the captured CIR is an $N \times L$ complex matrix.

Chapter 3: Environment-Independent Passive Fall Detection

3.0.1 Distinct Characteristics of Fall Events

Different types of falls might happen in our daily life. Some are assisted falls occurring where the subject is assisted by another person or other supporting objects during the falling process. Unassisted falls occur unexpectedly without any support due to extrinsic environmental factors such as spills on the floor or intrinsic risk factors such as impaired gait. Compared with assisted falls, unassisted falls are closer to free falls and the falling body has a larger speed at the moment of hitting the ground, leading to a higher risk of causing severe injuries or even death [69]. Therefore the system we propose focuses mainly on detecting unassisted falls and especially those when the subjects fall from a standing position which produces the largest speed.

Speed and acceleration are two characteristics that are usually used to describe motion. Intuitively, fall can be viewed as a type of abnormal indoor event with abnormal speed and acceleration, and therefore they are both considered as the unique characteristics that help distinguish falls from other daily activities. The uniqueness resides not only in the absolute values of speed and acceleration during a fall but also in how they change over time. More specifically, as a human falls

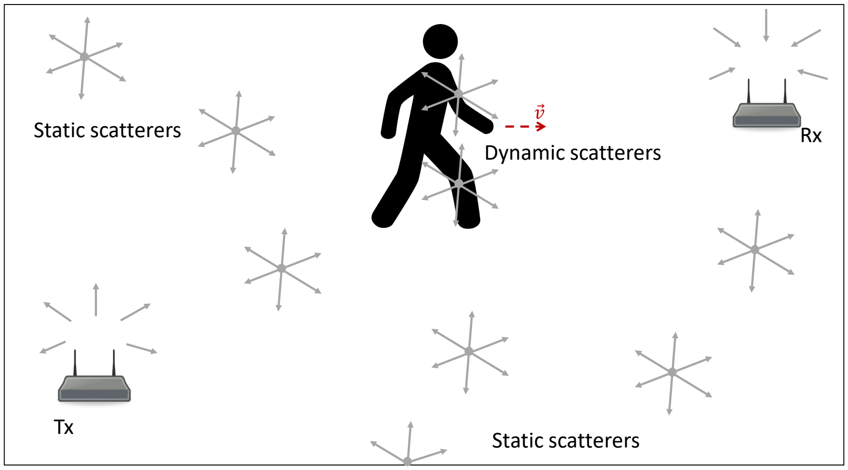


Figure 3.1: Indoor rich-scattering model.

to the ground, his/her body will experience a rapid acceleration first. Once the body hits the floor, the body speed reduces to nearly zero sharply. In fact, most of the unexpected falls exhibit a similar pattern and this implies the feasibility of developing an environment-independent system by monitoring the speed and acceleration variation, which is the foundation of *DeFall*.

3.0.2 Speed Estimation from WiFi CSI

We capture the human activities through CSI. Since the unique pattern of the series of speed is utilized, it is critical to have an accurate and reliable estimate of the speed based on WiFi CSI, which is not trivial due to the multi-path effects of the indoor propagation. Some device-free CSI-based speed estimators [38, 56, 58] have been proposed and most of them make use of the Doppler frequency shift (DFS) to calculate the speed of the human body, which have several limitations. First, DFS-based methods utilize the reflection model, assuming that the human body is simplified as a single reflector and produces only one dominant reflection path, which usually does not hold in a practical indoor environment with rich multi-path propagation. Second, to make sure the direct reflection path from human body has strong enough energy to be perceived, the existing works are limited to the LOS coverage since the moving body should be able to be “seen” by both Tx and Rx. Third, as indicated by [56], DFS induced by human motion is not only related to the motion speed but also depends on the relative location and direction with respect to the link. In addition, DFS-based speed estimators take CSI phase into account, while the phase of CSI on commercial WiFi devices cannot be measured accurately due to the phase synchronization errors between the WiFi Tx and Rx [18].

Inspired by WiSpeed [96], in this work we assume a practical rich-scattering environment, as shown in Fig. 3.1, and estimate the speed based on a statistical model of EM wave theory, which only makes use of the CSI magnitude information.

Specifically, the CSI magnitude can be measured through CSI power response $G(t, f)$, which, according to Eqn. (2.8), is defined as

$$G(t, f) \triangleq |H(t, f)|^2 = \xi(t, f) + \epsilon(t, f), \tag{3.1}$$

where $\xi(t, f) = \|\vec{E}_{\text{Rx}}(t, f)\|^2$, and $\vec{E}_{\text{Rx}}(t, f)$ denotes the propagated signals. $\epsilon(t, f)$ denotes the additive noise, and $\xi(t, f)$ and $\epsilon(t, f)$ are assumed to be independent of each other.

It has been shown in [96] that the speed of a moving object can be reliably estimated by evaluating the ACF of $G(t, f)$. The theoretical ACF of $G(t, f)$, $\rho_G(\tau, f)$, can be derived as

$$\rho_G(\tau, f) = \frac{\sigma_\xi^2(f)}{\sigma_\xi^2(f) + \sigma_\epsilon^2(f)}\rho_\xi(\tau, f) + \frac{\sigma_\epsilon^2(f)}{\sigma_\xi^2(f) + \sigma_\epsilon^2(f)}\delta(\tau), \tag{3.2}$$

where τ is the time lag of the ACF. $\sigma_\xi^2(f)$ and $\sigma_\epsilon^2(f)$ are the variances of $\xi(t, f)$ and $\epsilon(t, f)$, respectively. $\rho_\xi(\tau, f)$ and Dirac delta function $\delta(\cdot)$ are the ACFs of $\xi(t, f)$ and $\epsilon(t, f)$. When $\tau \neq 0$, we have $\delta(\tau) = 0$ and $\rho_G(\tau, f)$ can be further derived based on the statistical theory of EM waves [25] as

$$\rho_G(\tau, f) = \sum_{u \in \{x,y,z\}} (C_1(f)\rho_{E_u}(\tau, f) + C_2(f)\rho_{E_u}^2(\tau, f)), \tag{3.3}$$

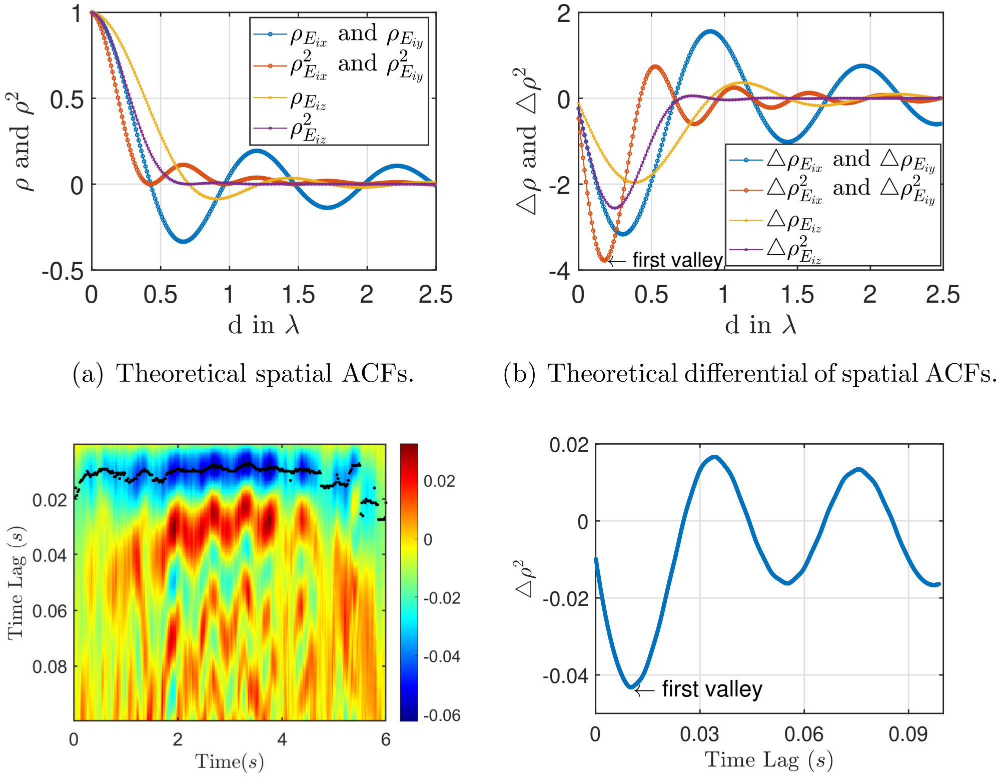
where $C_1(f)$ and $C_2(f)$ are scaling factors determined by the power reflected by all scatterers. $\rho_{E_u}(\tau, f)$ is the ACF of $\vec{E}_{\mathrm{Rx}}(t, f)$ in u-axis direction where $u \in \{x, y, z\}$.

For the i-th dynamic scatterer that moves at speed v_i along z-axis, the scattered signal is denoted as $\vec{E}_{iu}(t, f)$. Then the components of its ACF $\rho_{E_{iu}}(\tau, f)$ in $\{x, y, z\}$-axes can be expressed as the following closed-form equations, respectively:

$$\begin{aligned} \rho_{E_{ix}}(\tau, f) &= \rho_{E_{iy}}(\tau, f) \\ &= \frac{3}{2}\frac{\sin(kv_i\tau)}{kv_i\tau}(1 - \frac{1}{(kv_i\tau)^2}) + \frac{3}{2}\frac{\cos(kv_i\tau)}{(kv_i\tau)^2}, \end{aligned} \tag{3.4}$$

$$\rho_{E_{iz}}(\tau, f) = \frac{3}{(kv_i\tau)^2}(\frac{\sin(kv_i\tau)}{kv_i\tau} - \cos(kv_i\tau)), \tag{3.5}$$

where k denotes the wave number. Intuitively, the equations above have established a relationship between the ACF $\rho_G(\tau, f)$ and the presence of motion and also the



(a) Theoretical spatial ACFs.

(b) Theoretical differential of spatial ACFs.

(c) Differential ACF of response power and the valley locations.

(d) Instance of differential spatial ACF of CSI response power.

Figure 3.2: Spatial ACF and its differential for EM wave components.

moving speed.

- **The relationship between $\rho_G(\tau, f)$ and the presence of motion** From Eqn. (3.2), if motion is present in the propagation environment of WiFi signals, as $\tau \to 0$ we have $\delta(\tau) = 0$ and $\rho_\xi(\tau, f) \to 1$ due to the property of white noise and the continuity of motion [97]. Consequently, $\rho_G(\tau, f) \to \frac{\sigma_\xi^2(f)}{\sigma_\xi^2(f)+\sigma_\epsilon^2(f)} > 0$ as $\tau \to 0$. If there is no motion, the environment is static and the variance $\sigma_\xi^2(\tau, f) = 0$ and thus $\rho_G(\tau, f) = 0$ as $\tau \to 0$. Therefore the value of $\lim_{\tau\to 0} \rho_G(\tau, f)$ can indicate the presence of motion in the surrounding environment.

- **The relationship between $\rho_G(\tau, f)$ and the moving speed** For the simple case of all dynamic scatterers moving in the same speed and direction, without loss of generality we can assume the moving direction is in the z-axis and get the $\rho_G(\tau, f)$ as Eqn. (3.3) with its components expressed in Eqn. (3.4) and (3.5). Each component and its differential can be visualized in Fig. 3.2(a) and Fig. 3.2(b), respectively. Observing that the first local valley of $\triangle\rho^2_{E_u}(\tau, f)$, $\forall u \in \{x, y\}$, happens to be the first local valley of $\triangle\rho_G(\tau)$ as well, we can extract the speed information of the moving scatterers by locating the first local valley of $\triangle\rho_G(\tau, f)$. Fig. 3.2(c) shows an example of $\triangle\rho_G(\tau)$ over time for a "walking" event, in which the the first valley locations are marked by the black dots. Fig. 3.2(d) shows a snapshot of the $\triangle\rho_G(\tau)$ in Fig. 3.2(c).

In the case where a single subject, e.g., a human, moves within the coverage of the pair of Rx and Tx, the dynamic signals are dominated by the parts that are reflected by the human torso. Therefore it is reasonable to assume that in this case, all dynamic scatterers are moving at the same speed as well as in the same direction, and we can estimate the speed of the human using the proposed method to further detect a fall.

3.1 System Design

In this section, we depict the major modules in the *DeFall* system in detail. The system mainly consists of two stages as illustrated in Fig. 3.3. In the offline stage, the speed of a fall is estimated from the WiFi CSI by applying a statistical

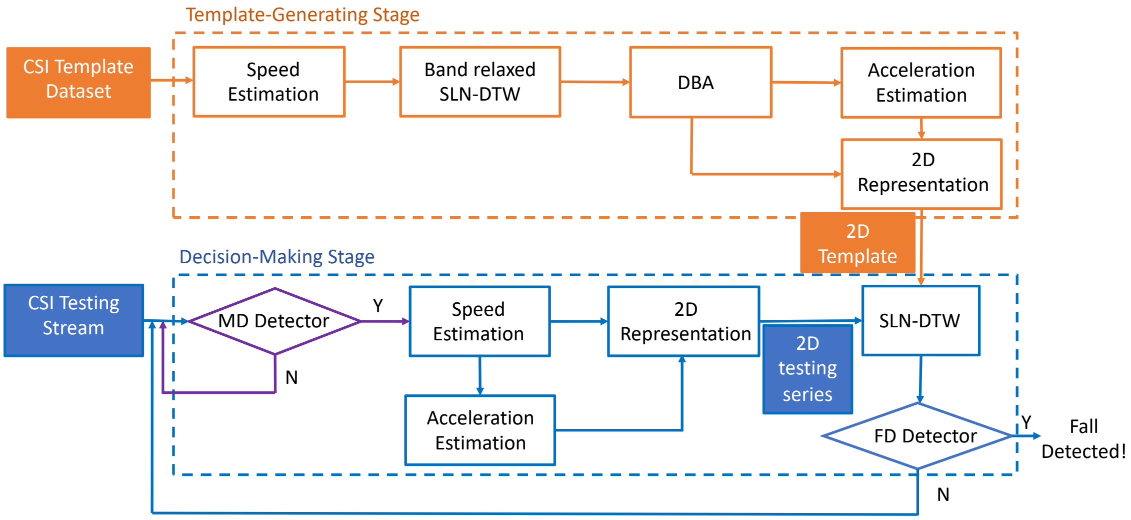


Figure 3.3: An overview of *DeFall*.

model on the radio propagation in an indoor rich-scattering environment. After that, DTW-based algorithms are performed to generate a representative template for a typical human fall. Then a fall event is detected in the online stage by evaluating the similarity between the patterns of real-time speed/acceleration estimates and the representative template. In addition, an online motion detection module is added before the fall detection module as a pre-judgment procedure.

3.1.1 Template-Generating Stage

In the offline template-generating stage, M CSI sequences of fall events are picked randomly and a "template database" $\mathbb{S} = \{S_1, S_2, ..., S_M\}$ is built based on the corresponding estimated speed series.

3.1.1.1 Challenges in Building a General Template

To construct a single representative template, we perform an "average" on the database. Since the collected data are all time sequences, the result by direct point-to-point matching and averaging will be easily affected by sequence shift and misalignment. Therefore, the operation of distance measurement, as well as series alignment, will be performed in the DTW space [63].

However, there may exist redundant speed segments of other activities before or after the fall event, and the classic DTW algorithm is sensitive to the endpoints of the sequences. Therefore, the endpoints of the series should be carefully defined and the template database cleaning is required.

3.1.1.2 Template Database Sanitization

To remove the redundancy while adapting to the possible variability in event instances, we resort to the band-relaxed segmental locally normalized DTW (SLN-DTW) [51]. The basic idea of SLN-DTW is to detect low-distortion local alignments between the objective series S_x and a series S_y from the rest sequences $\{S_1, S_2, ..., S_{x-1}, S_{x+1}, ..., S_M\}$ by dynamic programming [51]. The original SLN-DTW aims at matching objective series S_x in the testing stream S_y with the assumption that S_x coincides exactly with the target event, which is not suitable since any of the collected series in $\mathbb{S}$ may contain redundancy. Therefore band-relaxed SLN-DTW in [51] is applied. It relaxes the boundary constraints of SLN-DTW so that the starting and ending points of S_x can be aligned adaptively and the common parts can be retrieved reliably.

To be specific, let i and j represent the indices of the objective sequence S_x and the testing sequence S_y, respectively. We can construct a grid $[1,\ldots,i,\ldots,L_1] \times [1,\ldots,j,\ldots,L_2]$, where L_1 and L_2 denote the lengths of S_x and S_y. With relaxed boundaries, the starting point of the optimal warping path is allowed to be located in the starting band $\{(i,j)|i \in [1,B_s]\}$ while the ending point is selected in the ending band $\{(i,j)|i \in [B_e,L_1])$. Then the accumulative distance matrix D and the length matrix L can be generated, where the elements of the two matrices, $D(i,j)$ and $L(i,j)$, represent the total cumulative distance and path length from a starting point $(i_s,\ j_s)$ to (i,j). And the cost function is defined as the ratio $C(i,j) = \frac{D(i,j)}{L(i,j)}$. The procedure of the band-relaxed SLN-DTW applied for template database cleaning can be summarized as:

Step 1 Initializing distance matrix D and length matrix L:

For $\forall(i,j)$ where $1 \leq i \leq B_s,\ 1 \leq j \leq L_2$, we have

$$\begin{cases} D(i,j) = d(i,j) \\ L(i,j) = 1, \end{cases} \tag{3.6}$$

where $d(i,j)$ is the Euclidean distance between the i-th point in S_x and the j-th point in S_y.

Step 2 Iteration:

For $\forall(i,j)$ where $1 \leq i \leq B_s$ and $1 \leq j \leq L_2$, minimize $C(i,j) = \min\limits_{(u,v)} \frac{d(i,j)+D(u,v)}{L(u,v)+1}$ where $(u,v) \in \{(i,j),(i-1,j),(i,j-1),(i-1,j-1)\}$. For each iteration, the updates of the corresponding D and L are

$$D(i,j) = \begin{cases} d(i,j), \text{ if } (u,v) = (i,j) \\ D(u,v) + d(i,j), \text{ otherwise,} \end{cases} \tag{3.7}$$

$$L(i,j) = \begin{cases} 1, \text{ if } (u,v) = (i,j) \\ L(u,v) + 1, \text{ otherwise.} \end{cases} \tag{3.8}$$

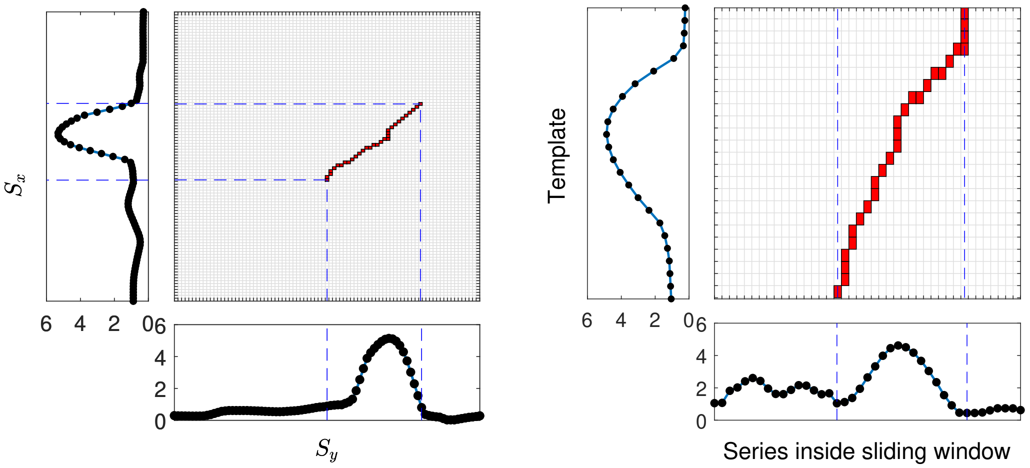
For $\forall(i,j)$ where $B_s < i \leq L_1$, minimize $C(i,j) = \min\limits_{(u,v)} \frac{d(i,j)+D(u,v)}{L(u,v)+1}$ where $(u,v) \in \{(i-1,j),(i,j-1),(i-1,j-1)\}$. For each iteration, update the corresponding D and L as

$$\begin{cases} D(i,j) = D(u,v) + d(i,j) \\ L(i,j) = L(u,v) + 1 \end{cases}. \tag{3.9}$$

Step 3 Trace back:

Find the minimum $C(k,j)$ for $k \in [B_e, L_1]$ and trace back along path (i,j) until $i = B_s$ to extract the optimum path across the central band $\{(i,j)|i \in [B_s, B_e]\}$. After that, if the cost to the next point is smaller than the current cost, i.e., $C(i_{\text{next}}, j_{\text{next}}) < C(i_{\text{now}}, j_{\text{now}})$, continue tracing back. Otherwise, stop and produce the optimum path.

Band-relaxed SLN-DTW is applied between every two speed sequences to extract their common parts. Therefore, for each objective series $S_x \in \{S_1, S_2, ..., S_M\}$, there are M-1 possible truncations with M-1 start indices $P_{x,s}$ and M-1 end indices $P_{x,e}$. And the part of S_x with indices lying in $[med(P_{x,s}), med(P_{x,e})]$ is regarded as the sanitized speed sequence of the fall event in sample S_x, where $med(P_{x,s})$ and



(a) Result of band-relaxed SLN-DTW to extract template.

(b) Result of a real-time detection by SLN-DTW.

Figure 3.4: Applying the principle of SLN-DTW in series sanitization and fall detection.

$med(P_{x,e})$ are medians of the start indices and end indices, respectively. In this way, the template database is refined to $\hat{\mathbb{S}} = \{\hat{S}_1, \hat{S}_2, \ldots, \hat{S}_M\}$. Fig. 3.4(a) illustrates an instance of the sanitized speed series by applying SLN-DTW.

3.1.1.3 Averaging in the DTW Measure Space

The M cleaned speed series in the refined database $\hat{\mathbb{S}}$ are then scaled to the same length and averaged in the DTW measure space to construct a single representative profile. The problem to find an optimal average can be formulated as an optimization problem that given a set of template time series $\hat{\mathbb{S}} = \{\hat{S}_1, \hat{S}_2, ..., \hat{S}_M\}$, the averaged series $\bar{S}$ is the series that minimizes the sum of squared DTW distances between $\bar{S}$ and all of sequences in $\hat{\mathbb{S}}$ as

$$\bar{S} = \arg\min_{S} \sum_{x=1}^{M} DTW^2(S, S_x). \tag{3.10}$$

The DTW distance of two sequences $DTW(A, B)$ is defined as the Euclidean distance between series A and series B along the optimal warping path as follows:

$$DTW(A, B) = \sqrt{\sum_{p^*=1}^{|P^*|} \|A[a_{p^*}] - B[b_{p^*}]\|^2}, \tag{3.11}$$

where P^* is the optimal warping path that minimizes the normalized distance as

$$P^* = \min_P \frac{1}{|P|} \sum_{p=1}^{|P|} \|A[a_p] - B[b_p]\|^2, \tag{3.12}$$

where a_p and b_p are indices of A and B associated with the p-th point on path P.

To solve the minimization problem (3.10) and get the optimal average series, DTW barycenter averaging (DBA) algorithm [54] is implemented. DBA is an iterative algorithm that refines an average sequence $\bar{S}$ on each iteration following an expectation-maximization scheme, whose convergence has been proved in [55]. The optimal speed time series $\bar{S}$, produced by DBA, is then considered as the speed template.

Besides speed, acceleration depicts the motion during a fall from another different point of view. To get a more comprehensive description of the fall events, we derive an acceleration series $\bar{S}'$ from the speed template $\bar{S}$ and combine them by point-to-point stitching to generate a 2-D template $\bar{S}_{2D}$. The efficacy of utilizing the 2-D combined template $\bar{S}_{2D}$ rather than a single 1-D template $\bar{S}$ or $\bar{S}'$ will be discussed in Section VI.

3.1.2 Decision-Making Stage

As mentioned in Section 3.0.1, fall events experience distinct speed and acceleration patterns which could be used for distinguishing falls from other indoor daily activities. However, a high sampling rate is needed for speed estimation [96]. To save energy and computation cost, in the decision-making stage, a low-rate motion detection (MD) module is included in addition to the fall detection (FD) module.

3.1.2.1 Motion Detection Module

As indicated in Section 3.0.2, $\lim_{\tau \to 0} \rho_G(\tau, f)$ could be utilized as a criterion for MD. In practice, due to the limitation of the sampling rate, we could only use $\rho_G(\tau = \frac{1}{F_s}, f)$ to approximate $\tau \to 0$.

For the purpose of efficient energy-saving, the MD module with a low sampling rate is added as a pre-detection of human motion prior to the FD module, and the FD module is triggered only in the presence of motion.

3.1.2.2 Fall Detection Module

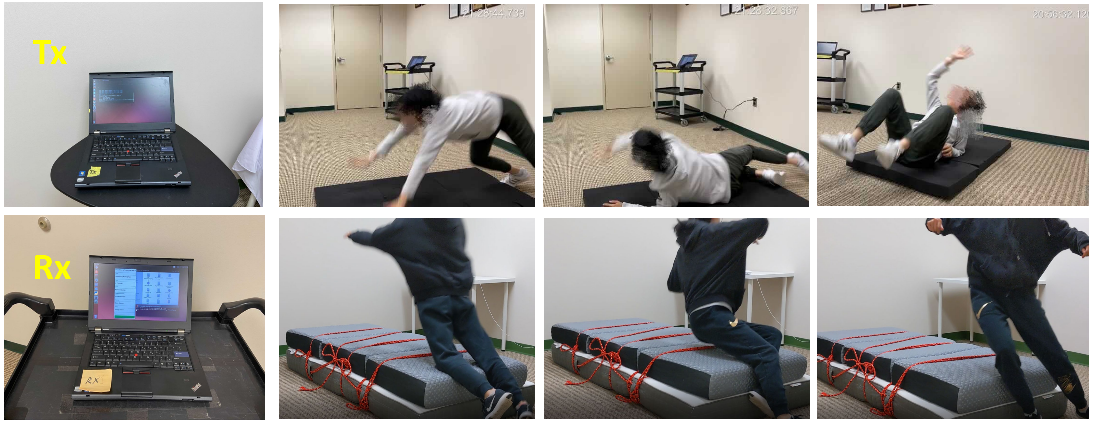
In the FD module, we apply a sliding window $\mathcal{W}$ on the incoming CSI stream. The testing speed sequence T is estimated from the CSI series in window $\mathcal{W}$. The acceleration sequence T' is further derived from T, followed by a combination operation to form a 2-D pattern T_{2D}.

Then fall events can be detected by comparing the testing time series T_{2D} with the template $\bar{S}_{2D}$. The corresponding similarity of the two series is evaluated in DTW space to adapt to the misalignment of the two sequences in the time domain.

Since the fall events involving different people may experience different duration, the series segmented by a length-fixed sliding window may also include other activities before or after the target event, which cannot be handled by the traditional DTW, and thus we adopt the SLN-DTW [51] again to localize the start and end instances of an event, as Fig. 3.4(b) illustrates. Regarding the template $\bar{S}_{2D}$ as the objective series and T_{2D} as the testing series, we set the lengths of starting and ending bands of S_{2D} to be 1 since the template $\bar{S}_{2D}$ is already sanitized.

By implementing SLN-DTW, the similarity of the testing stream T_{2D} and $\bar{S}_{2D}$ is evaluated. When the DTW distance between the testing series and the reference template is less than a preset empirical threshold γ, the testing sequence T_{2D} has a similar pattern to the reference fall template $\bar{S}_{2D}$ and the detector will alert that a fall occurs, where γ is empirically decided by experiments as well as the requirement of FAR and DR.

In the real-time monitoring, MD module keeps running with a lower sampling rate and as long as the motion is detected, the FD module starts working with a high sampling rate to estimate the speed and detect fall events. When the similarity between T_{2D} and $\bar{S}_{2D}$ stays lower than some threshold for a long enough time, it switches back to MD module to save power consumption and computation cost.



(a) Devices.

(b) Fall examples.

Figure 3.5: Illustrations for (a) devices for data collection and (b) examples for real falls including forward, lateral and backward falls.

3.2 Experimental Results

To build our *DeFall* system, we employ two laptops (Thinkpad T420) equipped with off-the-shelf WiFi network interface cards (Intel 5300) as the Tx and Rx, as shown in Fig. 3.5(a). We use the Linux 802.11n CSI Tool [24] to collect CSI measurements. Each of them is equipped with three omnidirectional antennas and the CSI stream over each pair of antennas has a total of 30 subcarriers. By default, the system works on WLAN channel 153 with a carrier frequency of 5.805 GHz and bandwidth of 40 MHz. In the MD module, the sampling rate is set to be 30 Hz. For FD module, to achieve a better speed estimation resolution to capture the high-speed motion, the Tx sends sounding frames with a channel sampling rate of 1500 Hz.

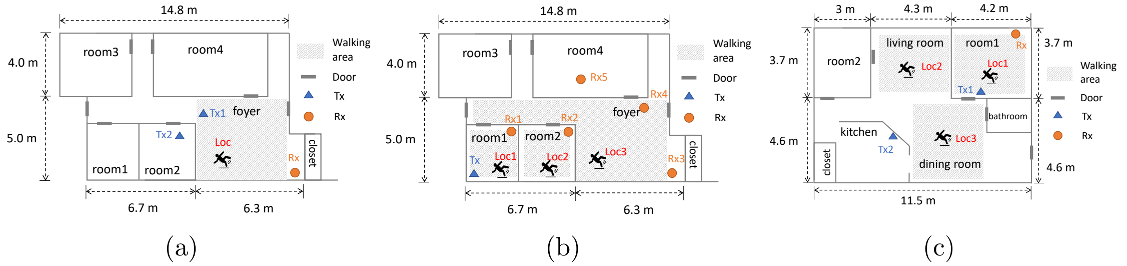
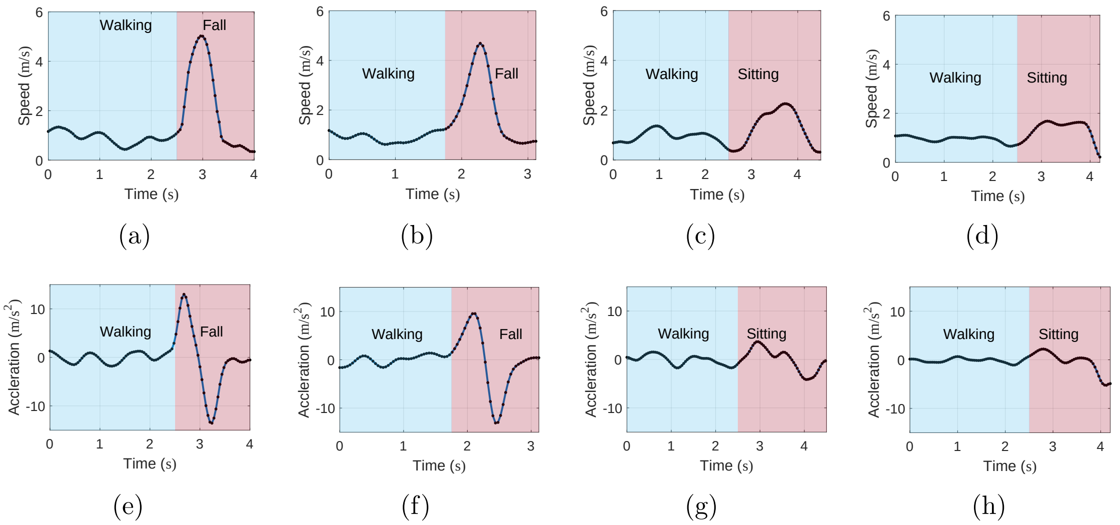


Figure 3.6: Experimental environments.(a) Environment 1: typical office environment. (b) Environment 2: different rooms and placements in the typical office. (c) Environment 3: typical apartment environment.

3.2.1 Experimental Environments

We evaluate *DeFall* with extensive experiments under various conditions (e.g., LOS and NLOS) at different locations in both office and home environments, with multiple volunteers involved. The detailed settings are shown in Fig. 3.6 with the locations of the Tx, the Rx, and the falling person marked. The data by a human-like dummy in environment 1 (Fig. 3.6(a)) is used for template-generating as well as detection algorithm verification. Then real fall/non-fall activities are performed by volunteers in all environments to further evaluate the impacts of environment diversity, user diversity and also types of falls. The ground truth is recorded by video.

In each environment, we change one of the Tx/Rx and conduct experiments with different placements under both the LOS and NLOS scenarios. Under the LOS scenario, Tx and Rx could both "see" the subject, while in the NLOS case, there does not exist any direct link between the subject and one or more devices, which is very common for an indoor environment. Specifically, in environment 1 (Fig. 3.6(a)) and environment 3 (Fig. 3.6(c)), the Tx is deployed on positions $\mathrm{Tx}_1/\mathrm{Tx}_2$ under



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 3.7: Instances of speed and acceleration patterns for "walk-then-fall" and "walk-then-sit". (a)(e): Speed and acceleration for "walk-then-fall" under LOS; (b)(f): Speed and acceleration for "walk-then-fall" under NLOS; (c)(g): Speed and acceleration for "walk-then-sit" under LOS; (d)(h): Speed and acceleration for "walk-then-sit" under NLOS.

LOS/NLOS conditions. In environment 2 (Fig. 3.6(b)), only Rx_1 is under the LOS scenario, while the other placements correspond to the NLOS cases.

3.2.2 Data Collection

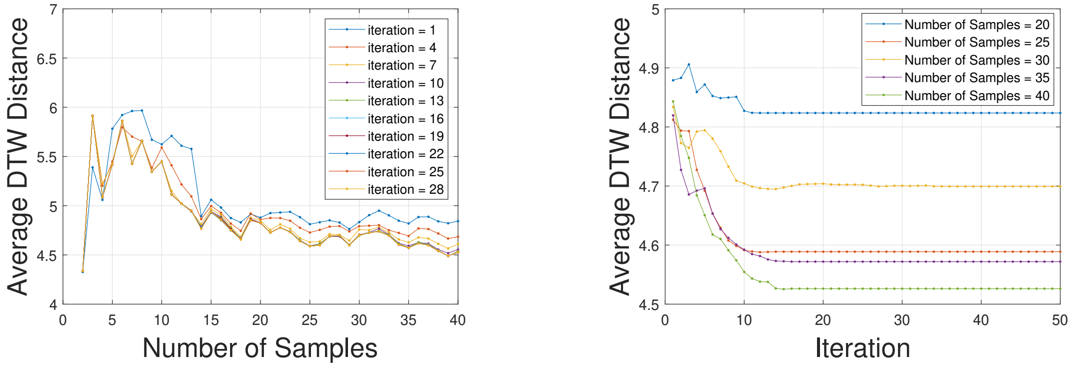
The data collection is carried out on different days lasting for more than three months, during which the surrounding propagation environment keeps changing, including the changes of the placements of furniture, the opening or closing of doors and windows, etc. To verify the feasibility of *DeFall*, we first use a human-like dummy to collect both the template data and testing data. After that, the samples from real human falls, as illustrated by Fig. 3.5(b), are further studied to evaluate the effectiveness of the system.

In the verification experiments, we consider both separate fall events and continuous motion followed by falls. “Stand-then-fall”, which represents falling from a stationary standing posture, is realized by first letting the dummy stand straight and then making it fall freely; while “walk-then-fall”, indicating the falls happen after the continuous walking motion, requires the experimenter to walk around the standing dummy at a normal speed and then make it fall. Instances of the speed/acceleration patterns for “walk-then-fall” and “walk-then-sit” under both LOS and NLOS scenarios are presented in Fig. 3.7, where we can observe the distinct patterns between falls and other activities such as sitting and walking. After the long-term data collection, there are 846 fall samples from the dummy and 814 non-fall samples for verification in total as Table 3.1 illustrates.

In order to prove that our system can work well in real world, we further evaluate its performance based on real human falls. We first involve three volunteers (1 female, 2 males) for multiple long-term experiments to study the impact of environment diversity, the presence of ambient motion, and the types of falls (forward, backward and lateral falls). To investigate the impact of user heterogeneity, we involve 7 more volunteers at different ages to perform different falls.

3.2.3 Generated Templates

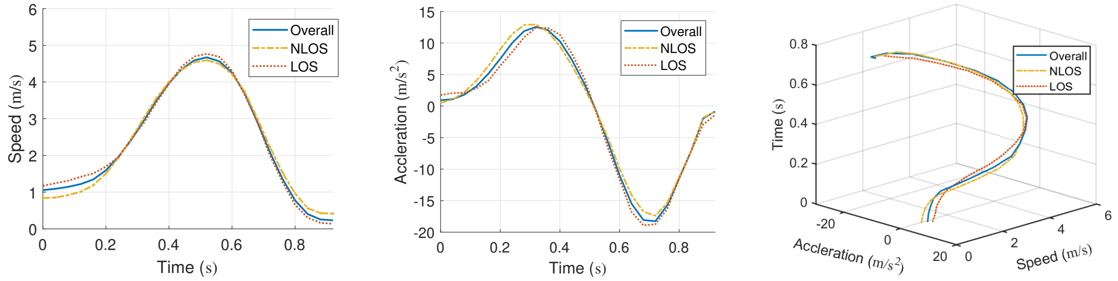
In the offline template-generating stage, we build the template dataset on the fall samples from the dummy. There are two factors to be selected in the DBA algorithm, i.e., the size of template database and the number of iterations. The investigation of these two factors can be seen in Fig. 3.8. In Fig. 3.8(a), the average



(a) Average DTW distance versus sample size in DBA algorithm.

(b) Average DTW distance versus number of iterations in DBA algorithm.

Figure 3.8: Investigation of DBA factors.



(a) Speed templates.

(b) Acceleration templates.

(c) 2D templates.

Figure 3.9: Templates.

DTW distance gets more stable as the size of template database increases to 25, while in Fig. 3.8(b) it converges after around 15 iterations. Therefore the number of iterations and the size of the template database are reasonably set to be 20 and 40, respectively.

The generated template after refinement and averaging is presented in Fig. 3.9. As we can observe, the template has the same tendency as expected. The speed rises to a peak value first and then drops, while the acceleration is positive first and then becomes negative. Also, it can be found that the templates of LOS and NLOS are highly consistent with each other. Since the speed estimation in

DeFall is based on the rich-scattering model, as long as the target is within the coverage of the radios, the system can capture the speed accurately using either the LOS or NLOS link, preserving not only the average speed but also the precise speed changes. Due to the high consistency between the LOS and NLOS scenarios, we use both LOS and NLOS data for template-generating and apply the overall template in the detecting phase.

3.2.4 Performance Evaluation

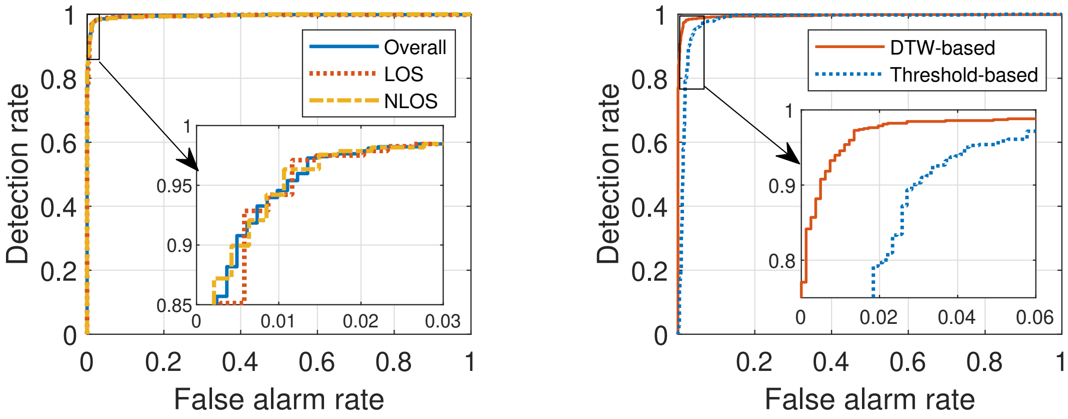
3.2.4.1 Evaluation Metrics

The evaluation metrics of the system performance are detection rate and false alarm rate. Detection rate, shorted as DR, is defined as the percentage of correctly detected falls among all falls:

$$\mathrm{DR} = \frac{\text{\# of detected falls}}{\text{\# of total falls}}, \tag{3.13}$$

while false alarm rate, simplified as FAR, is the percentage of non-falls that are mistaken as falls among all non-falls:

$$\mathrm{FAR} = \frac{\text{\# of wrongly detected nonfalls}}{\text{\# of total nonfalls}}. \tag{3.14}$$



(a) ROC curves for overall, LOS and NLOS cases.

(b) ROC curve of *DeFall* compared with threshold-based method in WiSpeed.

Figure 3.10: ROC curves for (a) different scenarios and (b) comparison with threshold-based method in WiSpeed.

3.2.4.2 Receiver Operating Characteristic (ROC) Curve

The threshold γ in the decision-making stage plays an important role in determining the boundary between fall and non-fall events, and therefore it has to be selected carefully. To evaluate the performance of *DeFall*, instead of proposing the specific threshold directly, we first calculate the DR and FAR with various thresholds and generate the overall ROC curve as illustrated by Fig. 3.10(a). We also investigate the ROC curves in LOS and NLOS scenarios, respectively. As seen, there exists a trade-off between DR and FAR. If the threshold is high, then there tends to be fewer speed sequences to reach the standard, i.e., the higher threshold is, the lower the DR is, while also getting a lower FAR. Note that the ROC curves of LOS and NLOS overlap with each other. Also, both of them are highly similar to the overall ROC trend, verifying the consistency of the proposed system in LOS/NLOS scenarios.

3.2.4.3 Effectiveness of the DTW-based Pattern Matching

In the previous work WiSpeed [96], a simple threshold-based method is applied to detect falls based on the maximum speed. In *DeFall*, we use the same CSI-based speed estimator as Wispeed but improve the detection performance by adding the DTW-based pattern matching for making a decision. To show the effectiveness of the DTW-based detection module, we also get the ROC curve of WiSpeed and compare it with *DeFall* as shown in Fig. 3.10(b). As Fig. 3.10(b) illustrates, at the same level of FAR, the DR of *DeFall* is higher than WiSpeed. The area under the curve (AUC) of the ROC of *DeFall* is larger as well, proving a better performance. In particular, when the FAR is less than 1.5%, *DeFall* can still achieve a high DR over 95% while the corresponding DR of WiSpeed drops to a level less than 75%.

3.2.5 Robustness to Indoor Activities

For a fall detection system, DR is very crucial due to the high risk of missed detection. On the other hand, FAR is also essential since other daily activities are performed most of the time in practice. Thus a system that yields a low FAR while keeping a reasonable DR is preferred. In this work, the threshold yielding an overall FAR 1.47% is selected for further system evaluation and the corresponding overall DR is 97.28%.

The results of DR and FAR for all types of events are summarized in Table 3.1. According to the results, *DeFall* succeeds in performing a high DR and low FAR under both LOS and NLOS scenarios. Comparing the results of different fall events,

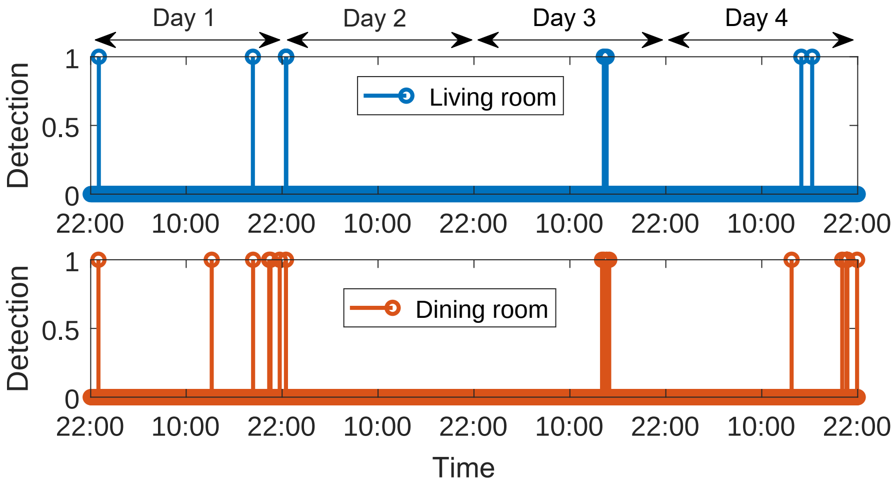


Figure 3.11: Long-term testing in an apartment environment.

Scenario	Events		Number	DR/FAR (*DeFall*)		DR/FAR (WiFall)		DR/FAR (FallDeFi)	
LOS	Fall	Stand-then-Fall	424	97.40%	97.10%	75.71%	74.32%	91.04%	89.58%
		Walk-then-Fall	94	95.74%		68.09%		82.98%	
	non-Fall	Walking	167	0.00%	1.45%	16.77%	17.73%	10.18%	12.79%
		Sitting down	177	2.82%		18.64%		15.25%	
NLOS	Fall	Stand-then-Fall	270	98.15%	97.56%	70.37%	69.82%	85.19%	84.45%
		Walk-then-Fall	58	94.83%		67.24%		81.03%	
	non-Fall	Walking	212	0.47%	1.49%	15.09%	17.23%	8.96%	10.43%
		Sitting down	258	2.33%		18.99%		11.63%	
Overall	-			97.28%/1.47%		72.58%/17.45%		87.59%/11.43%	

Table 3.1: Comparison results with WiFall and FallDeFi.

we can notice a higher DR on "stand-then-fall" events than "walk-then-fall" events since "walk-then-fall" may introduce interference to the speed estimation at the beginning of falls. Also, among the non-fall events, as "sitting-down" experiences an acceleration followed by a deceleration, which is more similar to the fall pattern than "walking", it is can be observed that FAR of "sitting-down" is slightly higher than that of "walking".

We implement WiFall [83] and FallDeFi [53] for the comparative study. WiFall extracts seven different features for classification based on the variation of CSI amplitude over time, while FallDeFi selects features from the STFT spectrogram and the power burst curve (PBC). To make a fair comparison, we optimize the parameters in these two works to adapt to our dataset. We apply 40 fall samples to generate the template in *DeFall*, while we use 80 samples (40 falls and 40 non-falls) to train the classifiers in WiFall and FallDeFi. The results in Table 3.1 demonstrate that *DeFall* outperforms both WiFall and FallDeFi with a higher overall DR (24.7% higher than WiFall and 9.69% higher than FallDeFi) and a lower FAR (15.98% lower than WiFall and 9.96% lower than FallDeFi). In addition, under both LOS and NLOS scenarios, *DeFall* also performs better in terms of the corresponding DR and FAR. The reasons for this performance enhancement can be attributed to the environment resilient speed information extracted by *DeFall* and the DTW-based pattern matching which adapts to the consecutive activities. More specifically, the features extracted in WiFall are based on the signal variation and could be different in different settings, while the speed estimated in *DeFall* is an inherent property of falls. Although compared with WiFall, FallDeFi devises more robust features using time-frequency analysis, its spectral features are partly dependent on the signal strength and do not take the detailed change pattern of falls into consideration. We also observe that the performance of WiFall and FallDeFi degrades especially for consecutive events such as “walk-then-fall”. This is because WiFall and FallDeFi either assume segmented activities or apply the “event duration” as a feature, which can easily lead to misclassifications if falls and other normal activities cannot be separated reliably, while *DeFall* employs a sliding window combined with the pattern matching approach which is more flexible to handle consecutive activities.

To take all possible daily activities into consideration and test the robustness of the system in practice, we further run the system in the same apartment (Fig. 3.6(c)) for four days. Specially, we deploy two pairs of transceivers to cover the main motion areas - living room and dining room, respectively. The participant stays in the apartment every day and may perform any daily activities in the monitored areas. No fall happens during the testing. We apply the same template and detection algorithm to the collected speed series. The experiment results for the long-term continuous test can be found in Fig. 3.11. The detection is output every second and the corresponding false alarms are counted. We have only 12 and 19 false alarms in total during the four-day testing in the living room and dining room. On average, we have 3 and 4.8 false alarms per day, which is acceptable considering the complicated daily activities.

3.2.6 Robustness to Falling Objects

There are fall-like events that may cause false alarms, such as falls of chairs and dropping a small object to the ground. To test the robustness of the system against the interference from these events, extensive fall experiments are conducted on objects with different sizes and different materials. For small objects, each of them is lifted up and then dropped from a height of 1m, which is repeated 50 times at various locations to evaluate a reliable FAR. We also repeat testing a falling wooden chair. The corresponding result listed in Table 3.2 presents that all the FARs are 0.0%, verifying the robustness of *DeFall*. This is because common objects which can be lifted up are usually much smaller than a human body and therefore, even if dropped from a high position, they produce fewer dynamic scatterers and cause less disturbance to the environment. In such a case, the speed or acceleration

Objects	Material	Size/Weight	FAR
Bottle	Plastic, water	0.5 kg	0.0%
Bag	Nylon	1 kg	0.0%
Plate	Plastic	Radius=12 cm	0.0%
Plate	Metal	Radius=10 cm	0.0%
Book	Paper	22 cm × 18 cm	0.0%
Box	Paper	17 cm × 17 cm × 25 cm, 0.8 kg	0.0%
Chair	Wood	50 cm × 40 cm × 58 cm, 3 kg	0.0%

Table 3.2: Impact of falling objects.

values cannot be detected or the values are not continuous. For falling chairs, they always fall with lower centers of gravity and have smaller speeds than human falls.

3.2.7 Types of Falls

Three volunteers are involved in the experiments of real falls. In this subsection, we study the impact of the orientation of a fall, i.e., forward, backward and lateral. Each of the three volunteers performs 5 forward falls, 5 backward falls and 5 lateral falls in all settings. To better mimic real-world falls, the subjects first perform normal activities, either walking or standing still, and then fall in different directions under each setting. The overall results are summarized in Table 3.3. Taking the experiments in all environments into account, we find that the falls in all the considered directions can achieve a detection rate above 95.00%. The results are as expected because the proposed approach is independent of the moving/falling directions.

	Forward	Backward	Lateral	Average
User 1	96.67%	95.56%	94.44%	95.56%
User 2	94.44%	96.67%	93.33%	94.81%
User 3	95.56%	97.78%	97.78%	97.04%
Average	95.56%	96.67%	95.19%	95.80%

Table 3.3: Detection rates on different fall types.

	Env. 1		Env. 2					Env. 3	
	LOS	NLOS	LOS	NLOS				LOS	NLOS
	Tx_1	Tx_2	Rx_1	Rx_2	Rx_3	Rx_4	Rx_5	Tx_1	Tx_2
Loc 1	97.78%	95.56%	100%	97.78%	91.11%	95.56%	93.33%	97.78%	95.56%
Loc 2	-	-	-	100%	93.33%	93.33%	91.11%	-	95.56%
Loc 3	-	-	-	-	97.78%	97.78%	93.33%	-	97.78%
Average	97.78%	95.56%	100%	98.89%	94.07%	95.56%	92.59%	97.78%	96.30%

Table 3.4: Detection rates in various environments.

3.2.8 Environment Diversity

To validate the robustness of *DeFall* to diverse environments, extensive experiments are carried out in both office and home environments. Also, the locations of Tx and Rx are changed. During the experiments, each subject performs 15 falls in different directions on different positions marked in the floorplan shown in Fig. 3.6. As reported in Table 3.4, a minimum detection rate of 91.11% can be achieved in different environments. The high detection rates are as expected because the proposed approach is environment-independent.

We also highlight such independence by comparing it with WiFall and FallDeFi. As Table 3.5 indicates, when we classify the data collected in a different environment from the training dataset, *DeFall* outperforms both of WiFall and FallDeFi with a higher detection rate.

Training/Template	Testing	DR (*DeFall*)	DR (WiFall)	DR (FallDeFi)
Env. 1	Env. 1	96.67%	72.22%	90.00%
LOS in Env. 1	NLOS in Env. 1	95.56%	53.33%	82.22%
Env. 1	Env. 2	95.37%	56.11%	81.30%
Env. 1	Env. 3	96.67%	52.78%	76.11%

Table 3.5: Impact of environmental changes.

	User 1	User 2	User 3	User 4	User 5
DR	95.56%	94.81%	97.04%	91.67%	97.22%
weight (kg)	77	50	84	70	62
height (cm)	174	166	168	172	175
	User 6	User 7	User 8	User 9	User 10
DR	94.44%	97.22%	100%	94.44%	97.22%
weight (kg)	60	90	70	91	53
height (cm)	169	178	171	170	160

Table 3.6: Detection rates of different users.

3.2.9 User Diversity

As different users have different heights, body shapes and gait styles, and a reliable fall detector should not be affected by the diversity of users, it is non-trivial to investigate the impact of the user heterogeneity on the performance. To do this, during a two-week experiment, 7 more volunteers with ages ranging from 23 to 59 are asked to perform falls at various locations in environment 2. Similar to the previous experimental procedure, the volunteers perform random activities before falling. Each volunteer falls three times (1 forward, 1 backward and 1 lateral) under each setting. Combining the experiments of different displacements and different locations, we focus on studying the impact of user diversity and summarize the results in Table 3.6. For Users 1-3, we have 270 samples each, while for Users 4-10, 36 samples are collected by each user. The users have weights varying from 50kg to 91kg and heights varying from 160 cm to 178 cm. Among all the ten subjects, User

	w/ other moving person				w/o other motion
	light motion		heavy motion		
	3-5 m	> 5 m	3-5 m	> 5 m	
LOS	12/15	15/15	8/15	15/15	15/15
NLOS	13/15	15/15	6/15	14/15	15/15

Table 3.7: Detection results with ambient motion.

7 is the tallest and User 9 is the heaviest while both User 7 and User 9 experience missed detection. Therefore, although a greater height and weight indicate more dynamic scatterers on the torso for signal propagation, we cannot conclude that there is an obvious monotonic relationship between users' heights or weights and the detection performance based on the samples from ten subjects.

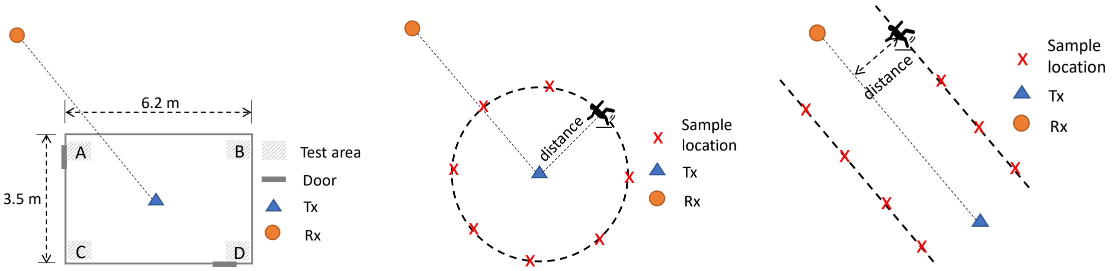
3.2.10 Impact of Ambient Motion

In this part, we investigate the robustness of the *DeFall* system when a second subject is moving in the vicinity of the first falling subject, with both LOS and NLOS cases considered. Intuitively, two factors may affect the performance, the distance between the two subjects ("distance") and the motion strength of the second subject. To study their impacts, we conduct experiments with a distance of 3-5 m and >5 m between the two subjects while for the motion strength, we consider heavy motion (walking) and light motion (reading and typing). Under each setting, 15 fall events occurred with different fall orientations, and the results are presented as the ratio of the number of detected falls to a total of 15 falls in Table 3.7. As we can see, when the second person is walking, the detection rate degrades as the distance decreases due to the interference from the other person. Nevertheless, some falls can still be detected even when the walking person is close. This is because the system estimates speed by localizing the first valley as Fig. 3.2(d) shows, which always corresponds to the highest speed of the massive scatterers. Therefore, the high speed associated with a fall can still be captured. Also, the motion strength of the interfering person has a significant impact on the performance, and a light motion by the second person has much less interference than a strong motion.

In fact, fall detection mainly aims at protecting the elderly people who live alone by sending alarms to remote caregivers when the subject being monitored falls. If there is another person nearby, timely assistance is available and the harm to the subject experiencing a fall will be greatly reduced. Therefore, the slight degradation in the multi-user has little impact on the applicability of *DeFall* as it ensures great performance for a single-user case and the cases when another user is relatively still or far away.

3.2.11 Coverage

It is important to provide a large coverage for a real-world fall detection system. For this purpose, we have conducted experiments in a large empty hall to evaluate if falls that are far away from the devices can be detected. As shown in Fig. 3.12(a), for the NLOS setting, the Tx is deployed at the center of a room with size 6.2 m × 3.5 m, the Rx is in the hall and 10m away from the Tx, and falls are performed at four corners with the corresponding DR and FAR shown in Table 3.8 given the preselected threshold in Section 3.2.5. For the LOS setting, in order to find the impact of the distance between the subject and devices, we first sample uniformly on positions



(a) Setting 1: NLOS scenario. (b) Setting 2: LOS scenario. (c) Setting 3: LOS scenario.

Figure 3.12: Settings for analyzing coverage.

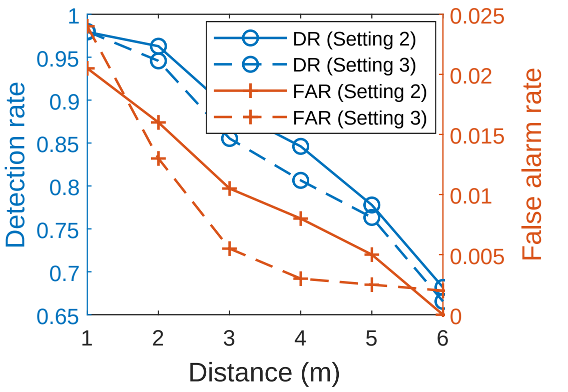


Figure 3.13: DR and FAR under setting 2 & 3.

	DR	FAR
A	95.45%	1.28%
B	91.30%	1.33%
C	95.45%	2.56%
D	90.91%	0.0%

Table 3.8: DR and FAR under setting 1.

marked with "X" along the circle centered around Tx with a changing radius as shown in Fig. 3.12(b). In the second LOS setting shown in Fig. 3.12(c), falls occur along lines parallel to the direct link. Massive fall samples are collected using the human-like dummy and a real human performs non-fall activities. Illustrated by Table 3.8, the system under setting 1 can cover an area as large as a normal office. Also, revealed by Fig. 3.13, DR and FAR both decrease as the distance between the subject and devices increases.

3.3 Discussion

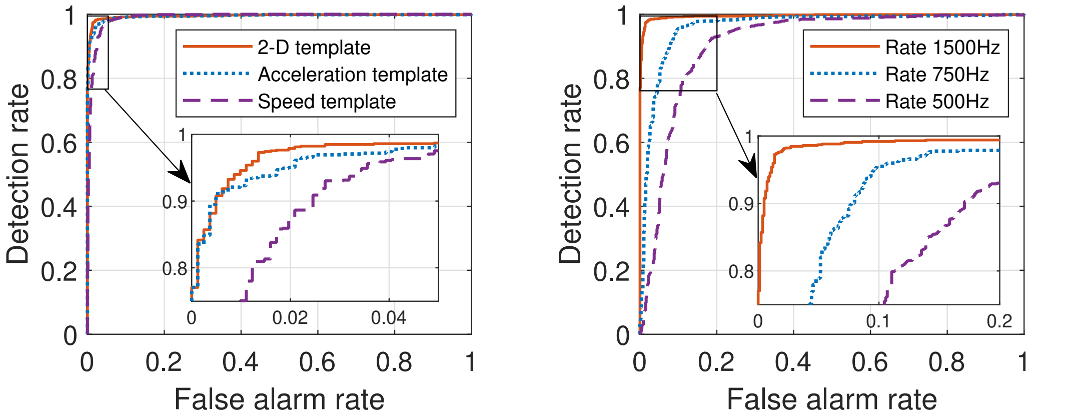
In this section, we will discuss the result of applying 2-D template comparing with that of using only 1-D speed template and 1-D acceleration template, which demonstrates the necessity of combining the features into 2-D space. Also, we will study the impact of the sampling rate and investigate the speed distribution of activities. In addition, we evaluate the computation cost.

3.3.1 Necessity of 2-D Space

The 2-D combining step integrates the information contributed by speed and acceleration. The combination is based on different weights α and realized by $\bar{S}_{2D} = ((1-\alpha)\bar{S}, \alpha\bar{S}')$. By setting $\alpha = 0$, we can get the single speed template while the single acceleration template can be obtained through setting $\alpha = 1$. For 2-D space, we set α to be 0.5. The performance of the separate 1-D templates and combined 2-D template can be found in Fig. 3.14(a). As shown, the 2-D combined template outperforms any single template as it provides a more comprehensive description of the events.

3.3.2 Impact of Sampling Rate

As described in Section 3.1, the sampling rate is a critical factor affecting the performance of our system. We now revisit the experimental results in Section 3.2 and evaluate the performance with different sampling rate F_s. The DR and FAR are studied under different sampling rates 1500 Hz, 750 Hz and 500 Hz, and their



(a) ROC curves for 2-D template and 1-D separate templates.

(b) ROC curves for different sampling rates.

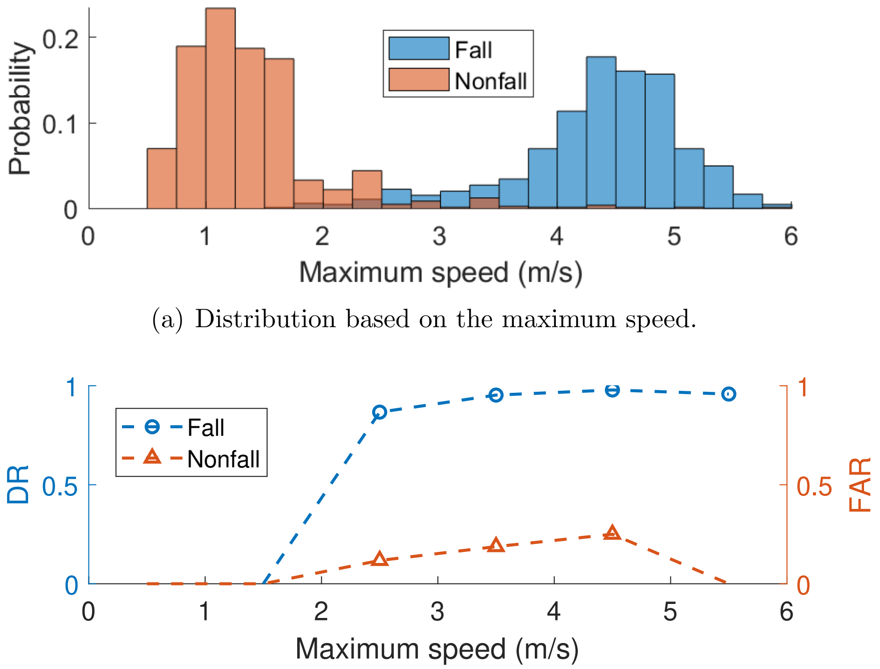
Figure 3.14: ROC curves for (a) comparison on 2-D template and 1-D separate templates and (b) different sampling rates.

corresponding ROC curves are plotted in Fig. 3.14(b). As illustrated, the overall trends for all the cases are the same, in which FAR increases as DR increases. However, the decrease of the sampling rate leads to a degradation in the performance of *DeFall*. This is because as the sampling rate gets reduced, the resolution of the ACF-based speed estimator degrades correspondingly, which will introduce more estimation errors and harm the detection accuracy.

3.3.3 Speed Distribution of Activities

Although the maximum speed of the template can be up to around 5 m/s, it does not mean the system can only detect falls with speeds reaching as high as 5 m/s. This is because the proposed template-based method not only depends on the single maximum speed value but also relies on the trend of speed, which relaxes the decision boundary compared with a hard-thresholding method.

To understand the system capability better, we divide our real fall/non-fall



(a) Distribution based on the maximum speed.

(b) DR/FAR in each maximum speed interval.

Figure 3.15: Maximum speed distribution and DR/FAR for each maximum speed interval.

samples into different intervals (0 1 m/s], (1 m/s 2 m/s], $\cdots$, (5 m/s 6 m/s] based on their maximum speeds. The corresponding distribution is shown in Fig. 3.15(a). Then we apply the same overall template and decision boundary and evaluate DR/FAR in each individual interval, where the DR/FAR is the ratio of detection to the total falls/non-falls in that specific interval. As we can see, the maximum speeds of fall and non-fall series samples are distinct with a small overlapped part. The FAR and DR both increase with the speed, as illustrated by Fig. 3.15(b). However, even if the falls and non-falls have the maximum speeds in the same interval, our system can still distinguish most of them with reasonable DR/FAR because of their distinct patterns over time.

Stage	Module	CPU time	
		(s)	(%)
Offline template generating (dataset size: 40)	Speed estimation	8.571322	96.23
	Database sanitization	0.168743	1.895
	DBA for average	0.165920	1.863
	2D representation	0.001251	0.014
	Total	8.9072	
Online computation for each output	Motion detection	0.000535	2.355
	Speed estimation	0.004661	20.52
	SLN-DTW & Detection	0.017519	77.13
	Total	0.0227	

Table 3.9: Processing time of *DeFall*.

3.3.4 Computation Cost and Real-Time Realization

In the real-time scenario, as the extracted speed feature is independent of the environments, the offline stage can be completed ahead of time. Then the pre-trained template can be applied directly in the decision-making stage.

To evaluate the system computation effort, we repeat the process using MATLAB on a desktop with Intel Core i7-9700K processor and 32 GB memory. We record the average processing time. Table 3.9 illustrates the CPU time for each module in the offline and online stages. Note that we do not require a massive amount of data for template generating, which significantly reduces the computation cost. The total pre-training process only takes an average CPU time of 8.9 s. In the online stage, to produce a decision output, it costs only 0.0227 s with a speed estimation module and a simple DTW-based similarity calculation, which is short enough for real-time applications.

3.3.5 Limitation

Since we only focus on the hard falls which may cause serious injuries, our approach only deals with unexpected falls such as stumbles or slips due to weak gait. Other types of falls such as falls slowly from a lower height, may result in a different speed pattern from what is described in the sections above. To solve this problem, we need more analysis on the action decomposition to get a better understanding of the entire process of various falls. In this case, multiple templates and deep learning techniques may be required.

Another limitation is that since the algorithm can only estimate the approximate average speed of the moving objects in the environment, our system works well when there is a single person or no strong ambient motion close to the subject. However, this is fully compliant with the goal of our system to protect the elderly who live alone.

3.4 Summary

In this chapter, we propose *DeFall*, a novel environment-independent indoor fall detection system using commercial WiFi devices. The system extracts speed information to detect falls even through the walls with a single pair of transceivers. A real prototype is built to validate the feasibility and evaluate the performance in various environments. The results show that *DeFall* achieves a detection rate higher than 95% on real falls while maintaining a false alarm rate lower than 1.50% under both LOS and NLOS scenarios, without any scenario-tailored prior training.

Chapter 4: Robust Passive Proximity Detection using WiFi

4.1 Statistical Proximity Sensing

We first analyze the statistical behavior of each component of the received CSI power response under certain statistical assumptions and link them with the distance of motions to antennas.

4.1.1 Characteristics of CSI Power Decomposition

To eliminate the phase distortion of CSI on commercial Wi-Fi devices, we take the CSI power which, according to Eqn. (2.9), is defined as

$$G(t,f) \triangleq |H_s(f)|^2 + |\tilde{H}_d(t,f)|^2 + H_s^*(f)\tilde{H}_d(t,f) + H_s(f)\tilde{H}_d^*(t,f). \tag{4.1}$$

Among them, the component of the static signal $H_s(f)$ keeps constant and so as its power $|H_s(f)|^2$. Based on the assumption of circularly symmetric Gaussian distribution of $\tilde{H}_d(t,f)$, $H_s^*(f)\tilde{H}_d(t,f)$ and $H_s(f)\tilde{H}_d^*(t,f)$ both have zero means. To further investigate the statistical characteristics of $G(t,f)$, we have three key observations:

Remark 1 *Compared with the amplitude of the static signal $|H_s(f)|$ which is the* *sum of reflections from large static objects, the amplitude of dynamic signals* $|\tilde{H}_d(t,f)|$ *is very small, thus the component* $|\tilde{H}_d(t,f)|^2$ *is negligible [36, 59].*

Remark 2 *Denote* $\tilde{H}_d(t,f)$ *as*

$$\tilde{H}_d(t,f) = \tilde{A}_d(t,f)e^{j\tilde{\phi}_d(t,f)}, \tag{4.2}$$

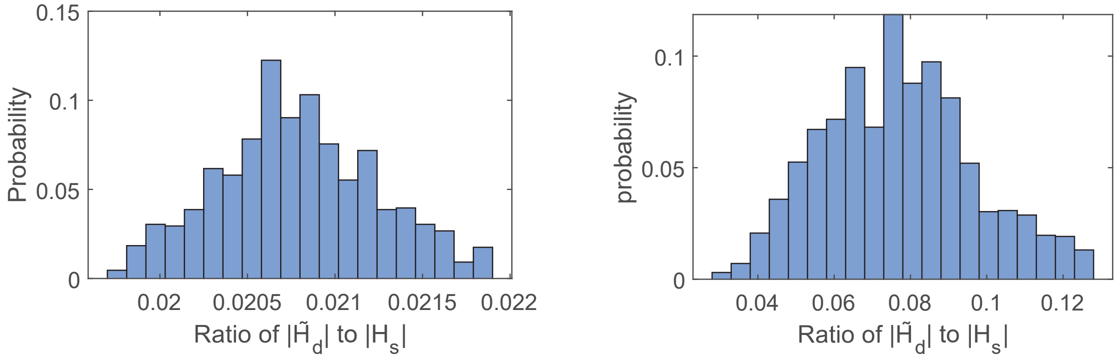
where $\tilde{A}_d(t,f)$ *and* $\tilde{\phi}_d(t,f)$ *are the amplitude and phase of* $\tilde{H}_d(t,f)$*, respectively. At two frequencies* f_1 *and* f_2 *with* $f_2 \to f_1$*, we have* $\tilde{H}_d(t,f_2) \to \tilde{H}_d(t,f_1)$ *due to the continuity of CSI in frequency domain, i.e.,* $\tilde{A}_d(t,f_1) \to \tilde{A}_d(t,f_2)$ *and* $\tilde{\phi}_d(t,f_1) \to \tilde{\phi}_d(t,f_2)$ *for dynamic signal components. With the assumption that* $\tilde{\phi}_d(t_n,f)$ *is uniformly distributed over* $[0,2\pi)$ *in time domain at any fixed* f*, the summation* $\tilde{\phi}_d(t_n,f_1) + \tilde{\phi}_d(t_n,f_2)$ *also follows a uniform distribution over* $[0,2\pi)$ *after phase wrapping.*

Remark 3 *Denote* $H_s(f)$ *as*

$$H_s(f) = A_s(f)e^{j\phi_s(f)}, \tag{4.3}$$

where $A_s(t,f)$ *and* $\phi_s(t,f)$ *are the amplitude and phase of* $H_s(t,f)$*, respectively. At two frequencies* f_1 *and* f_2 *with* $f_2 \to f_1$*, we have* $H_s(f_2) \to H_s(f_1)$ *due to continuity and thus* $\phi_s(f_1) \to \phi_s(f_2)$ *for static signal components.*

We validate these remarks via CSI data collected in a real-world environment. The validation requires separate complex-valued H_s and $\tilde{H}_d$. However, due to the



(a)

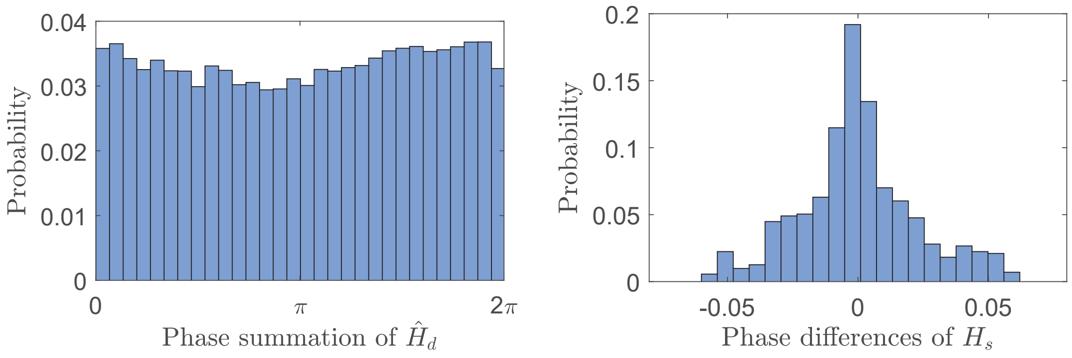
(b)

Figure 4.1: $\frac{|\tilde{H}_d|}{|H_s|}$ (a) in static environment and (b) with nearby motion.

significant CSI phase distortion, it is difficult to perform a perfect phase sanitization to obtain the true H_s and $\tilde{H}_d$. To eliminate the phase offset as much as possible, we adopt a linear-fitting method for calibration [57, 66]. Then H_s is estimated by taking the average of phase-cleaned CSI, while $\tilde{H}_d$ is the residual after mean subtraction. Although the cleaned phase still contains noise, it does support the validation of the aforementioned remarks in the following analysis and derivation.

To verify Remark 1, we collect data in an empty environment and in an environment with motion, and compute the ratio of $\frac{|\tilde{H}_d|}{|H_s|}$, respectively. Fig. 4.1 shows the distributions of $\frac{|\tilde{H}_d|}{|H_s|}$. As we can see, in the static scenario, $|\tilde{H}_d|$ only contains noise component and $\frac{|\tilde{H}_d|}{|H_s|}$ is smaller than 0.022. Further, even when there is nearby motion, most of the ratios of $\frac{|\tilde{H}_d|}{|H_s|}$ remains lower than 0.12 as Fig. 4.1(b) presents, implying that the component of $|\tilde{H}_d(t, f)|^2$ in Eqn. (4.1) is negligible.

Remarks 2 and 3 need to be verified under the condition of $f_2 \to f_1$. In practice, however, different frequency components of CSI are recorded over subcarriers and $f_2 \to f_1$ is not a practical condition to achieve. Nevertheless, we consider the



(a)

(b)

Figure 4.2: (a) Distribution of phase summation of $\tilde{H}_d$ over adjacent subcarriers; (b) Distribution of phase difference of H_s between adjacent subcarriers.

CSI power response on neighboring subcarriers as a weak alternative to the requirement $f_2 \to f_1$. To verify Remark 2, Fig. 4.2(a) shows the distribution of the phase summation of $\tilde{H}_d$ over adjacent subcarriers with frequencies f_1 and f_2, which is wrapped into $[0, 2\pi)$. As shown, the samples are nearly uniformly distributed in $[0, 2\pi)$ and satisfy the assumption in Remark 2. At last, to verify Remark 3, the distribution of phase difference of H_s between adjacent subcarriers is presented in Fig. 4.2(b). As shown, the phase difference, corresponding to $\phi_s(f_1) - \phi_s(f_2)$ in Eqn. (4.6), is small between adjacent subcarriers and consistent with the Remark 3.

4.1.2 Statistical Proximity Metrics

Based on Remark 1, the covariance of $G(t,f)$ between subcarriers, denoted as $\gamma_G(f_1,f_2)$, is calculated as

$$\begin{aligned}\gamma_G(f_1,f_2)=&\Big\langle G(t,f_1)-\big\langle G(t,f_1)\big\rangle, G(t,f_2)-\big\langle G(t,f_2)\big\rangle\Big\rangle\\ =&\big\langle H_s^*(f_1)\tilde{H}_d(t,f_1)+H_s(f_1)\tilde{H}_d^*(t,f_1), H_s^*(f_2)\tilde{H}_d(t,f_2)+H_s(f_2)\tilde{H}_d^*(t,f_2)\big\rangle\\ =&H_s^*(f_1)H_s^*(f_2)\big\langle\tilde{H}_d(t,f_1),\tilde{H}_d(t,f_2)\big\rangle+H_s(f_1)H_s(f_2)\big\langle\tilde{H}_d^*(t,f_1),\tilde{H}_d^*(t,f_2)\big\rangle+\\ &H_s^*(f_1)H_s(f_2)\big\langle\tilde{H}_d(t,f_1),\tilde{H}_d^*(t,f_2)\big\rangle+H_s(f_1)H_s^*(f_2)\big\langle\tilde{H}_d^*(t,f_1),\tilde{H}_d(t,f_2)\big\rangle\\ =&2\mathrm{Re}\big\{H_s^*(f_1)H_s^*(f_2)\big\langle\tilde{H}_d(t,f_1),\tilde{H}_d(t,f_2)\big\rangle\big\}+\\ &2\mathrm{Re}\big\{H_s(f_1)H_s^*(f_2)\big\langle\tilde{H}_d^*(t,f_1),\tilde{H}_d(t,f_2)\big\rangle\big\},\end{aligned}\tag{4.4}$$

where $\langle\rangle$ stands for the ensemble average over all realizations, and $\mathrm{Re}\{\cdot\}$ denotes the operation of taking the real part of a complex number.

Based on Eqn. (4.2), component $\big\langle\tilde{H}_d(t,f_1),\tilde{H}_d(t,f_2)\big\rangle$ can also be represented as

$$\big\langle\tilde{H}_d(t,f_1),\tilde{H}_d(t,f_2)\big\rangle=\frac{1}{N}\sum_{n=1}^{N}\tilde{A}_d(t_n,f_1)\tilde{A}_d(t_n,f_2)e^{j\left(\tilde{\phi}_d(t_n,f_1)+\tilde{\phi}_d(t_n,f_2)\right)},\tag{4.5}$$

where N is the number of samples for ensemble average calculation. According to Remark 2, wrapped $\tilde{\phi}_d(t_n,f_1)+\tilde{\phi}_d(t_n,f_2)$ in Eqn. (4.5) follows a uniform distribution over $[0,2\pi)$ which is independent of amplitudes. That is, we have $\big\langle\tilde{H}_d(t,f_1),\tilde{H}_d(t,f_2)\big\rangle\approx 0$ for adjacent subcarriers with frequencies denoted as f_1

and f_2 respectively. In such a case, Eqn. (4.4) can be approximated as

$$\begin{aligned}\gamma_G(f_1,f_2) &\approx 2\mathrm{Re}\big\{H_s(f_1)H_s^*(f_2)\big\langle\tilde{H}_d^*(t,f_1),\tilde{H}_d(t,f_2)\big\rangle\big\}\\ &= 2A_s(f_1)A_s(f_2)\mathrm{Re}\big\{e^{j(\phi_s(f_1)-\phi_s(f_2))}\big\langle\tilde{H}_d^*(t,f_1),\tilde{H}_d(t,f_2)\big\rangle\big\}\\ &= 2A_s(f_1)A_s(f_2)\Big[\cos\big(\phi_s(f_1)-\phi_s(f_2)\big)\mathrm{Re}\big\{\big\langle\tilde{H}_d^*(t,f_1),\tilde{H}_d(t,f_2)\big\rangle\big\}\\ &\quad+\sin\big(\phi_s(f_1)-\phi_s(f_2)\big)\mathrm{Im}\big\{\big\langle\tilde{H}_d^*(t,f_1),\tilde{H}_d(t,f_2)\big\rangle\big\}\Big],\end{aligned}\tag{4.6}$$

where $\mathrm{Im}\{\cdot\}$ denotes the operation of taking the imaginary part of a complex number. The coefficient $2A_s(f_1)A_s(f_2)$ is a positive constant coefficient related to the amplitudes of static signals at frequencies f_1 and f_2. According to Remark 3, on adjacent subcarriers $f_2 \to f_1$, we have $\phi_s(f_1)-\phi_s(f_2)\to 0$ and $\sin\big(\phi_s(f_1)-\phi_s(f_2)\big)\to 0$ while $\cos\big(\phi_s(f_1)-\phi_s(f_2)\big)$ has a positive value. Thus, we further simplify Eqn. (4.6) as

$$\gamma_G(f_1,f_2)\approx C(f_1,f_2)\mathrm{Re}\big\{\big\langle\tilde{H}_d^*(t,f_1),\tilde{H}_d(t,f_2)\big\rangle\big\},\tag{4.7}$$

where $C(f_1,f_2)\triangleq 2A_s(f_1)A_s(f_2)\cos\big(\phi_s(f_1)-\phi_s(f_2)\big)$ is a positive constant, and $\big\langle\tilde{H}_d^*(t,f_1),\tilde{H}_d(t,f_2)\big\rangle$ calculates the covariance between $\tilde{H}_d(t,f_1)$ and $\tilde{H}_d(t,f_2)$. By separating the components of dynamic signals $H_d(t,f)$ and noise $\varepsilon(t,f)$ and employing the relation between the covariance and correlation, we further represent $\gamma_G(f_1,f_2)$ as

$$\gamma_G(f_1,f_2)\approx C(f_1,f_2)\big[\sigma_d(f_1)\sigma_d(f_2)\mathrm{Re}\big\{\rho_d(f_1,f_2)\big\}+\sigma_\varepsilon(f_1)\sigma_\varepsilon(f_2)\delta(f_1,f_2)\big],\tag{4.8}$$

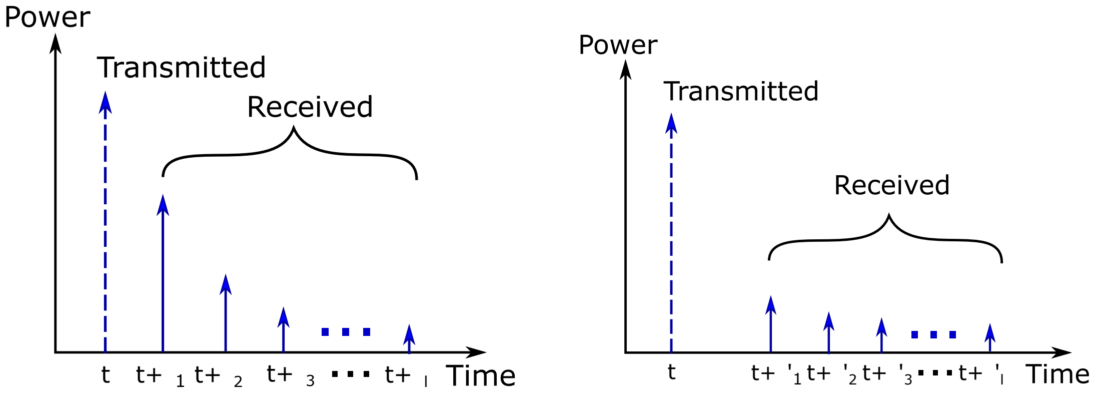
with $\sigma_d^2(f)$ and $\sigma_\varepsilon^2(f)$ denoting the variance of the dynamic signals and noise, respectively. $\rho_d(f_1, f_2)$ is the correlation coefficient between $H_d(t, f_1)$ and $H_d(t, f_2)$, while $\delta(f_1, f_2)$ is the correlation between $\varepsilon(t, f_1)$ and $\varepsilon(t, f_2)$. The corresponding correlation coefficient $\rho_G(f_1, f_2)$ between $G(f_1)$ and $G(f_2)$ can thus be written as

$$\begin{aligned}\rho_G(f_1, f_2) =& \frac{\gamma_G(f_1, f_2)}{\sqrt{\gamma_G(f_1, f_1)}\sqrt{\gamma_G(f_2, f_2)}} \\ \approx& C'(f_1, f_2)\Big[\frac{\sigma_d(f_1)\sigma_d(f_2)\mathrm{Re}\big\{\rho_d(f_1, f_2)\big\}}{\sqrt{\sigma_d^2(f_1) + \sigma_\varepsilon^2(f_1)}\sqrt{\sigma_d^2(f_2) + \sigma_\varepsilon^2(f_2)}} + \\ & \frac{\sigma_\varepsilon(f_1)\sigma_\varepsilon(f_2)\delta(f_1, f_2)}{\sqrt{\sigma_d^2(f_1) + \sigma_\varepsilon^2(f_1)}\sqrt{\sigma_d^2(f_2) + \sigma_\varepsilon^2(f_2)}}\Big],\end{aligned} \tag{4.9}$$

where $C'(f_1, f_2) \triangleq \cos(\phi_s(f_1) - \phi_s(f_2))$ is a positive constant and $C'(f_1, f_2) \to 1$ as $f_1 \to f_2$. In order to realize the proximity detection, we need to investigate how $\gamma_G(f_1, f_2)$ and $\rho_G(f_1, f_2)$ change with the distance between motion and antennas. To do this, we look into each component in Eqn. (4.8) and (4.9). It is known that term $\sigma_d(\cdot)$ is the reflected dynamic energy from the human body, which decreases as the distance between motion and devices increases. To explore how the component $\mathrm{Re}\{\rho_d(f_1, f_2)\}$ behaves, we then resort to the multipath model in communication field.

4.1.3 Understanding Proximity Features

To investigate how the CSI correlation over frequencies is impacted by the surrounding motions, we first look into the concept of power delay profile (PDP) because the frequency-domain correlation and the PDP behaves as a Fourier transform pair. PDP evaluates the statistical characteristics of multipath propagation



(a) PDP with a small Tx-Rx distance.

(b) PDP with a large Tx-Rx distance.

Figure 4.3: Power delay profiles.

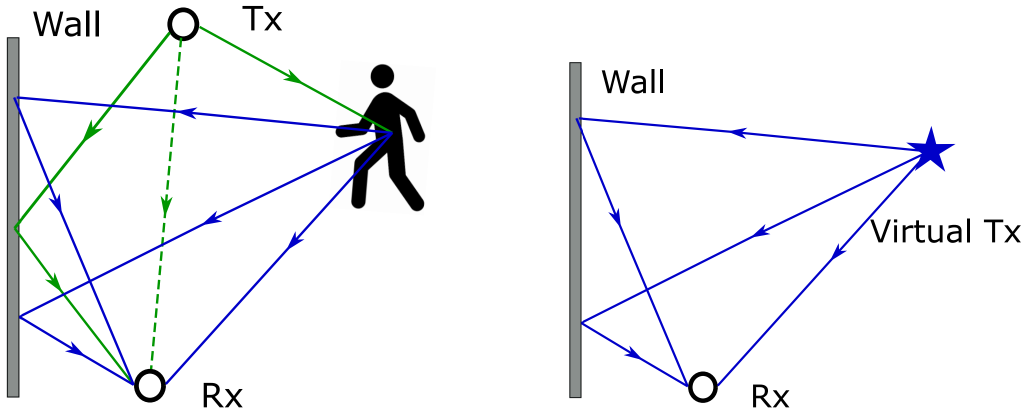
and is given by the square of CIR as

$$P_h(t,\tau) = |h(t,\tau)|^2, \tag{4.10}$$

which characterizes the connection between transmission delay and channel gain.

Without loss of generality, we assume the energy of MPCs at a specific time t follows the exponential envelope with decay rate α for simplicity yielding $\beta_i^2(t) = \beta_1^2(t)\text{exp}(-\alpha\tau_i)$, where β_1^2 is the energy of the first tap corresponding to the direct path between Tx and Rx. The Fourier transform of the PDP gives the frequency correlation function of $H(f)$, denoted as $R_H(t,\triangle f) = \sum_{i=1}^{\infty} P_h(t,\tau_i)e^{-j2\pi\triangle f\tau_i}$, where $\triangle f$ is the frequency difference. The normalized correlation coefficient ρ_H can then be evaluated as

$$\begin{aligned}\rho_H(\triangle f) = \frac{R_H(t,\triangle f)}{R_H(t,0)} &= \frac{\sum_{i=1}^{\infty}\beta_i^2(t)e^{-j2\pi\triangle f\tau_i}}{\sum_{i=1}^{\infty}\beta_i^2(t)} \\ &= \frac{1-e^{-\alpha\triangle\tau}}{1-e^{-(\alpha+j2\pi\triangle f)\triangle\tau}},\end{aligned} \tag{4.11}$$



(a) Multipath propagation.

(b) Illustration of virtual Tx.

Figure 4.4: Generalized multipath model.

whose real part increases with the positive decay rate α at fixed $\triangle f$ and $\triangle \tau$.

When the Tx is close to the Rx, the LOS signal along the direct link between the Tx and the Rx dominates the propagation as a primary component, while the power of the other NLOS components decays very fast, producing a larger α as Fig. 4.3(a) shows and thus a larger $\mathrm{Re}\{\rho_H(\triangle f)\}$ for a fixed $\triangle f$. As the Tx moves away from the Rx, the influence of the primary components decreases at the Rx side due to amplitude attenuation, and the amplitudes of the reflected and scattered signals relative to the direct path become larger, which leads to the decrease of the decay rate α as Fig. 4.3(b) illustrates [74]. Therefore, $\mathrm{Re}\{\rho_H(\triangle f)\}$ decreases at a larger Tx-Rx distance. In other words, $\mathrm{Re}\{\rho_H(\triangle f)\}$ indicates the Tx-Rx distance.

As Fig. 4.4(a) shows, when a person moves near an Rx, attenuated and delayed copies of the original signal are generated by the reflections from the body, and transmitted through different paths. Therefore, the moving human body can be viewed as a dynamic virtual transmitter, denoted by Tx$'$. Then the term $\mathrm{Re}\{\rho_H(\triangle f)\}$ in Eqn. (4.11) can be replaced with $\mathrm{Re}\{\rho_d(\triangle f)\}$, the frequency correlation function

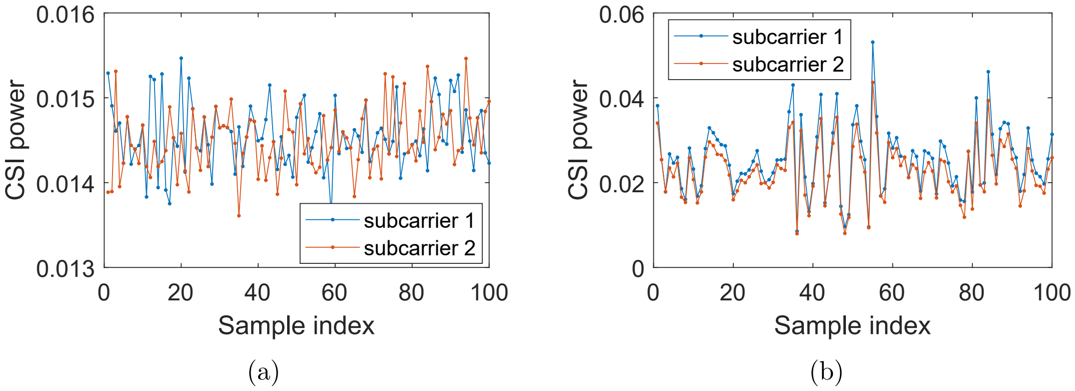


Figure 4.5: CSI power responses of two adjacent subcarriers over time (a) in empty case and (b) when motion is close to the Rx.

between different frequencies of CSI dynamic component $H_d(t, f)$. By evaluating $\mathrm{Re}\{\rho_d(\triangle f)\}$, we can know if the human body is far away from the Rx. A larger $\mathrm{Re}\{\rho_d(\triangle f)\}$ at a fixed $\triangle f$ indicates a closer moving human body. Therefore, based on the relationship in Eqn. (4.9) and (4.8), $\rho_G(\triangle f)$ and $\gamma_G(\triangle f)$ also increase with the decrease of human-Rx distance with a small enough $\triangle f$ while the $\triangle f$ is measured as the frequency difference of the adjacent subcarriers. To explicitly show such a monotonic relationship, the static and dynamic cases of $G(t)$ of two adjacent subcarriers are illustrated by Fig. 4.5(a) and Fig. 4.5(b), respectively. As expected, when there is no motion in proximity, the fluctuations of $G(t)$ on two subcarriers are random, while human movements nearby affect the CSI power responses on successive subcarriers similarly, implying a high positive correlation and covariance of $G(t)$ between these two subcarriers.

Further, Fig. 4.6 shows the correlation matrices among subcarriers when the person is moving at different distances to the Rx. As expected, $\rho_G(\triangle f)$ degrades as the human body moves away especially for the near-diagonal elements that reflect

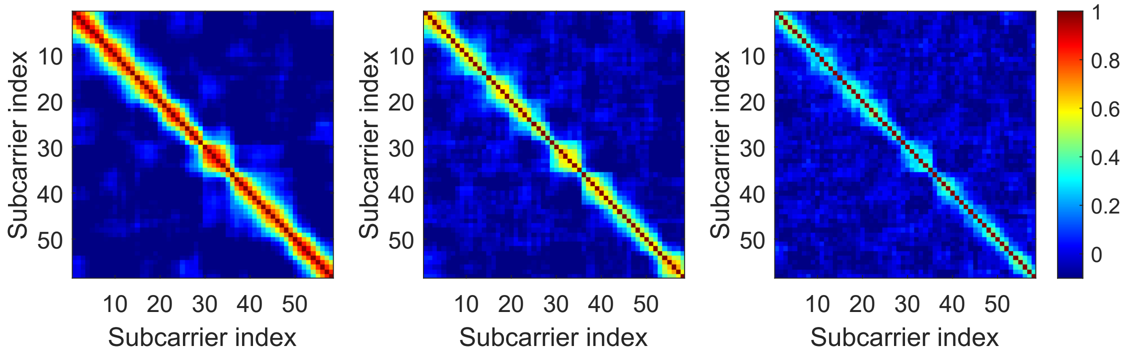


Figure 4.6: Correlation matrices when the human body moves at distance 1 m (left), 3 m (middle) and 5 m (right).

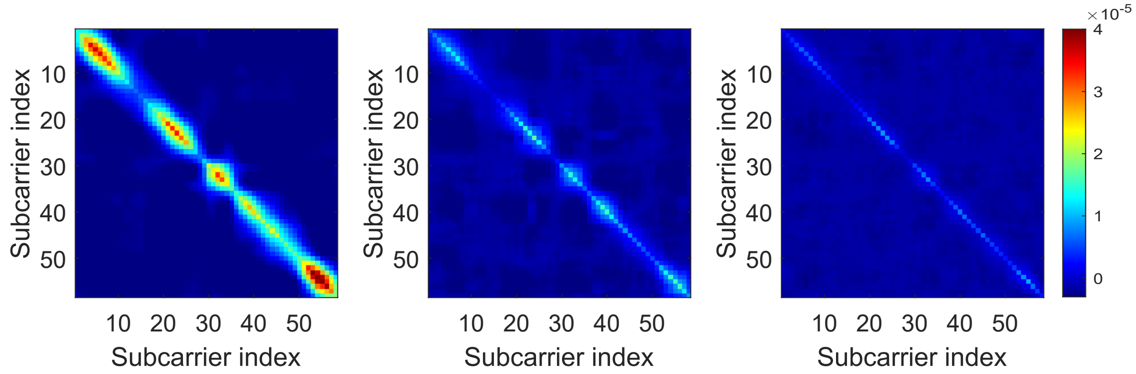


Figure 4.7: Covariance matrices when the human body moves at distance 1 m (left), 3 m (middle) and 5 m (right).

the correlation between neighboring subcarriers. Similarly, based on Eqn. (4.8) and the analysis in Section 4.1.1, the covariance between subcarriers $\gamma_G(\triangle f)$ can also indicate Tx′-Rx distance. As Fig. 4.7 shows, the smaller the Tx′-Rx distance, the larger the diagonal and near-diagonal elements of the frequency covariance matrix. That is, $\gamma_G(\triangle f)$ indicates the nearby motion, especially with a small $\triangle f$.

Besides the distance-based proximity detection, features ρ_G and γ_G can also be used in detecting proximity in the same room. When motion occurs in the same room as the Rx, the LOS path between the moving target and the Rx will dominate the propagated signals, leading to a higher correlation between neighboring subcarriers

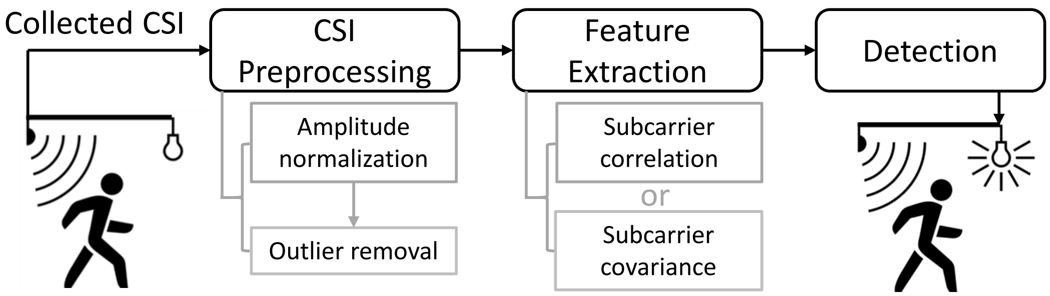


Figure 4.8: Processing pipeline.

compared with the motions that have no direct link to the devices.

Note that although frequency correlation and covariance have been exploited in other works [52, 56, 57, 84, 106], they are used for either subcarrier selection or general motion detection with sophisticated eigen-decomposition and none of these approaches explores the effectiveness of these features for proximity detection. In addition, they are merely data-driven and lack insight of how frequency correlation reflects the physics of motion. Differently, in this work, we explore the underlying RF characteristics and link the frequency correlation/covariance with detecting the proximity of motion theoretically [26].

4.2 Proximity Detector

The proximity detector follows an end-to-end pipeline to detect motion in proximity, as Fig. 4.8 illustrates. In pre-processing module, CSI is normalized over power response. Then the correlation and covariance across adjacent subcarriers are extracted as the detection features, followed by a threshold-based detection rule to make final decisions.

4.2.1 CSI Preprocessing

4.2.2 Feature Extraction

Subcarrier correlation. The correlation between adjacent subcarriers increase as the subject gets closer, as illustrated by Fig. 4.6. Inspired by such patterns, we average the correlation coefficients over adjacent subcarriers to detect motion in proximity, which is calculated as

$$M_\rho = \frac{1}{N_s - 1} \sum_{k=1}^{N_s-1} \rho_G(f_k, f_{k+1}), \tag{4.12}$$

where k is the index of subcarrier and f_k is corresponding subcarrier frequency.

Subcarrier covariance. As previously shown in Fig. 4.7, the Tx′-Rx distance information is embedded in the covariance between neighboring subcarriers. Similar to Eqn. (4.12), to capture the joint variability of CSI in different subcarriers, we take the near-diagonal elements in γ_G into account and average them as

$$M_\gamma = \frac{1}{N_s - 1} \sum_{k=1}^{N_s-1} \gamma_G(f_k, f_{k+1}), \tag{4.13}$$

which is the second feature for motion proximity detection.

4.2.3 Detection Rule

With the monotonic relationship between the human-Rx distance and features M_ρ and M_γ, thresholds can be applied on M_ρ and M_γ, respectively, to detect the proximity of motion. To select proper thresholds, an offline step can be involved to construct a database of the features M_ρ and M_γ, which includes the samples collected with a single nearby motion, faraway motion as well as the empty case. Then the corresponding ROC curve can be obtained to illustrate the diagnostic ability. At the selected detection rate/false alarm level, we can determine the corresponding discrimination thresholds. In this work, we mainly focus on the feasibility study and characterize the features through the overall behavior of the ROC curves.

Since these two metrics are highly related, both measuring of the joint variability of CSI power at neighboring subcarriers, we use them separately and generate the detection based on either M_ρ or $M\gamma$. Both metrics perform well in the experimental evaluation and we also make a comparison on them in the next section.

4.3 Experimental Evaluation

4.3.1 Benchmark Studies

In this section, we conduct an in-depth investigation of the capability of the proposed features in detecting the proximity of motion. We prototype our scheme using off-the-shelf Wi-Fi devices at a carrier frequency of 5.808 GHz with a 40 MHz bandwidth. The default sampling rate is set to 30 Hz. In the following, we first study the effect of human (Tx′)-Rx distance and compare the proposed features with other baseline features. Then, we study the impact of the directions, user diversity, motion speed, sampling rate and reveal the channel reciprocity in the same Tx/Rx setting. Later, we also investigate the performance in different settings

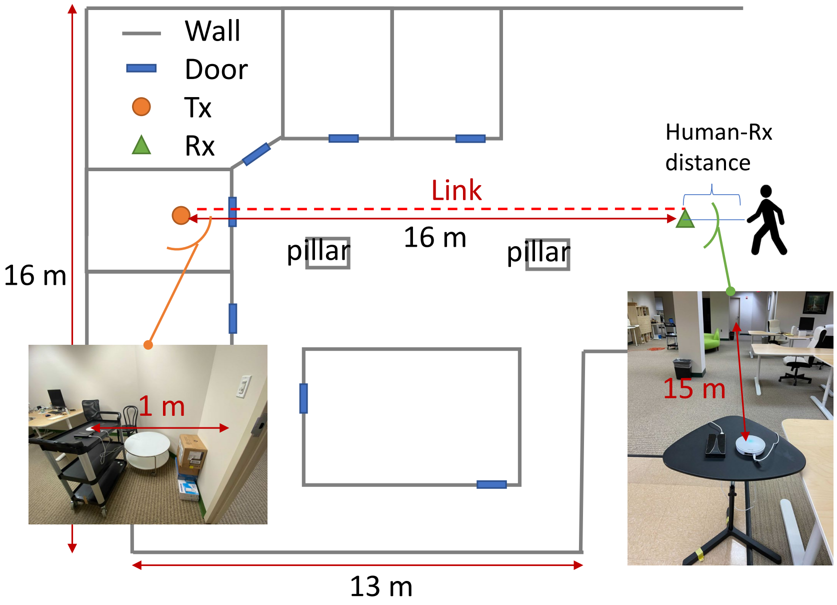
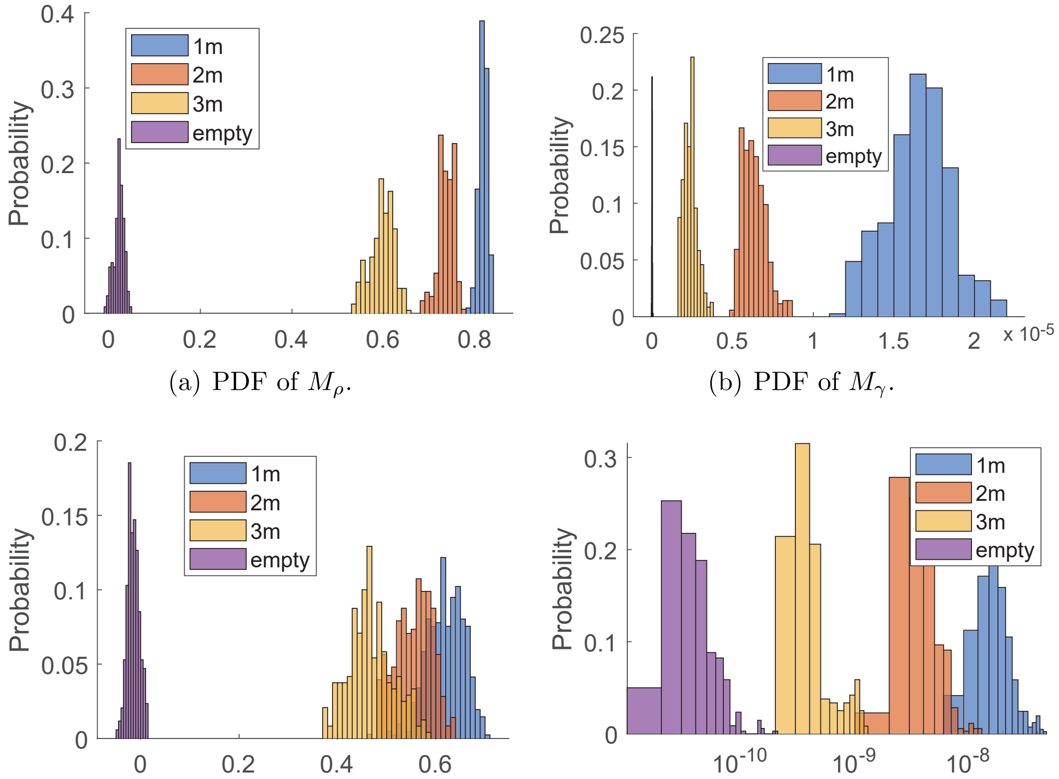


Figure 4.9: Experimental setting.

by changing the Tx-Rx locations and environments.

4.3.1.1 Human Movement at Different Human-Device Distances

As shown in Fig. 4.9, we first place a pair of Wi-Fi transceivers in a typical office environment. The transceivers are placed 16m apart from each other, with 2 antennas on both Tx and Rx. To quantitatively investigate the motion at different Tx′-Rx distances, a volunteer is asked to perform normal walking on the spot at pre-fixed distances 1 m, 2 m and 3 m. We use a sliding window of 2 s to calculate M_ρ and M_γ from CSI. Fig. 4.10a-b show the distributions of the feature values obtained at each location, respectively. Consistent with the analysis in Section 4.1, it is observed that the smaller the distance between the human and the Rx, the larger the values of M_ρ and M_γ.



(a) PDF of M_ρ.

(b) PDF of M_γ.

(c) PDF of the metric in WiDetect.

(d) PDF of the metric in SIED.

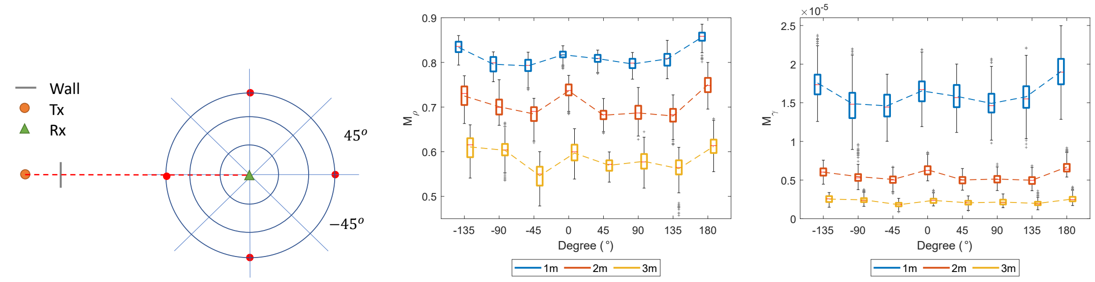
Figure 4.10: Probability distribution function (PDF) of M_ρ, M_γ, motion statistic proposed in WiDetect [97] and the variance of variances of amplitudes of each subcarrier proposed in SIED [43].

4.3.1.2 Comparative Study

Previous works have studied human motion sensing by extracting features in the temporal domain. Here, we compare the proposed metrics with features in two state-of-art systems, WiDetect [97] and SIED [43], using the same dataset described above. WiDetect devises a "motion statistic" based on the observation that the sample ACF of CSI power response increases in presence of motion; SIED captures the variance of variances of amplitudes of each subcarrier. We use the same parameter setting including sampling rate and window size for feature calculation. Fig. 4.10 shows an illustration of the behavior of M_ρ, M_γ and the motion metrics proposed in WiDetect and SIED under different scenarios. We can observe the distinct gap between empty (i.e., no motion) and motion cases for all features in Fig. 4.10a-d, which validates their ability to sense human motion. However, when it comes to the same motion at different distances, there does not exist such a stable gap to discriminate between them in Fig. 4.10c-d. This is because the ACF of the CSI power in the time domain and the variance of variances of amplitudes of each subcarrier have no clear relationship with the human-Rx distance and therefore they are insensitive to the human-Rx distance.

4.3.1.3 Difference between M_γ and M_ρ

According to the definitions of correlation and covariance, the two metrics M_γ and M_ρ are closely related. Both of them are sensitive to the proximity of motion. However, since M_ρ has been normalized to the range [-1,1], it is easier to devise a universal threshold for M_ρ and generalize it to different environments, while M_γ does not have a range limit and requires more careful tuning in various environments. On the other hand, M_γ is the product of M_ρ and $\sqrt{\sigma_d^2(f_2) + \sigma_\varepsilon^2(f_2)}$, both increasing as human-Rx distance decreases, therefore M_γ amplifies the impact of human-Rx distance and is particularly sensitive as Fig. 4.10(b) indicates, especially at a shorter distance.



(a) Setup with different directions.

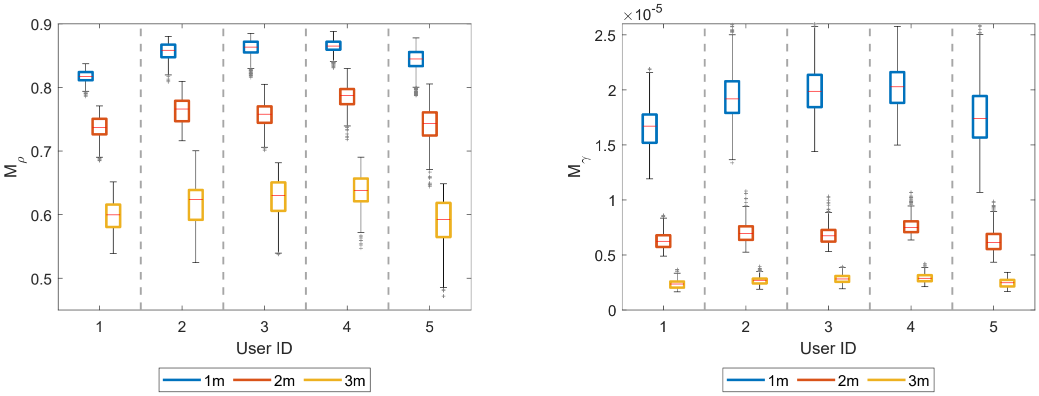
(b) Spread of M_ρ.

(c) Spread of M_γ.

Figure 4.11: Impact of direction.

4.3.1.4 Impact of Direction

Since users can approach the Rx from different directions, it is necessary to investigate the impact of direction. Therefore, as Fig. 4.11(a) shows, we conduct experiments with the user moving at different angles to the Tx-Rx link, ranging from −135° to 180° with an increment of 45°. The obtained distributions of the metrics are illustrated in Fig. 4.11, where we can see that values of the metrics decrease with the increasing human-Rx at each angle. It is worth noting that when the user is on the side closer to the direct link, M_γ becomes larger due to a larger σ_d in Eqn. (4.13), while the M_ρ is more robust. As we can also observe, the gaps between the features at different human-Rx distances are clear for us to apply a threshold-based method for proximity detection, although the optimal thresholds in different directions may slightly vary.



(a) Spread of M_ρ.

(b) Spread of M_γ.

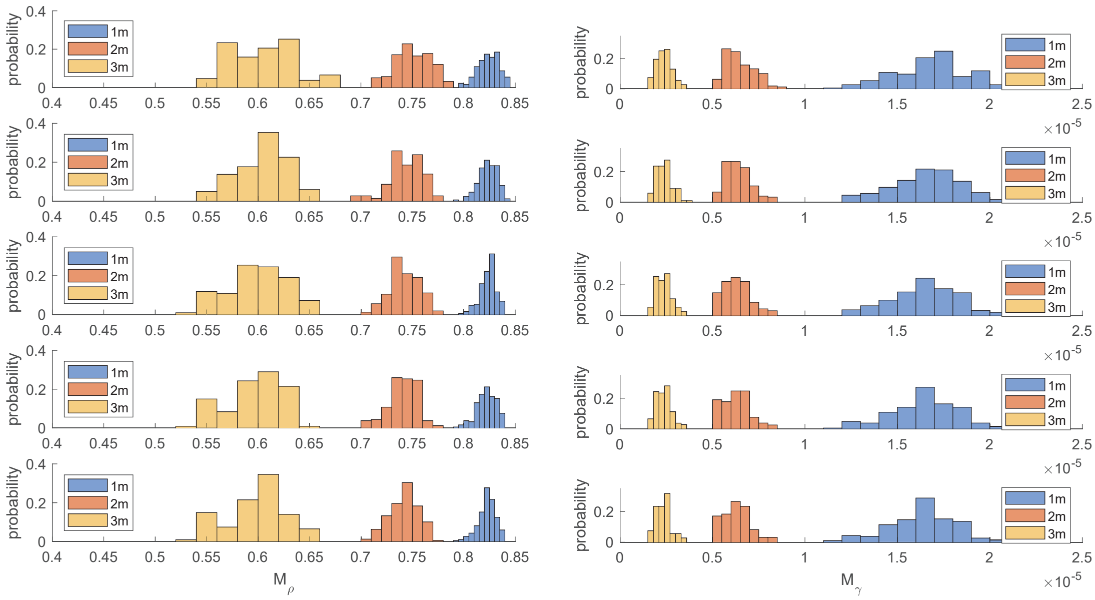
Figure 4.12: Impact of user diversity.

4.3.1.5 Impact of User Diversity

Since different people have different shapes and gait styles, and the way they approach the Rx could be different, we also evaluate the robustness of our system against user diversity. In this set of experiments, we recruit 5 volunteers and ask them to walk at distances 1 m, 2 m and 3 m to the Rx and repeat multiple times. These volunteers include 3 females and 2 males with a height range of 160 cm to 183 cm, and a weight range of 50 kg to 76 kg. The obtained metric values are shown in Fig. 4.12. As we can see, the proposed metrics work well for different subjects with similar values and clear boundaries between the motions at different distances, demonstrating the generality of M_γ and M_ρ for proximity detection.

4.3.1.6 Impact of Sampling Rate

Sampling rate affects the power consumption and computation cost. To evaluate the impact of sampling rate on the system performance, we repeat the above-



(a) Distribution of M_ρ.

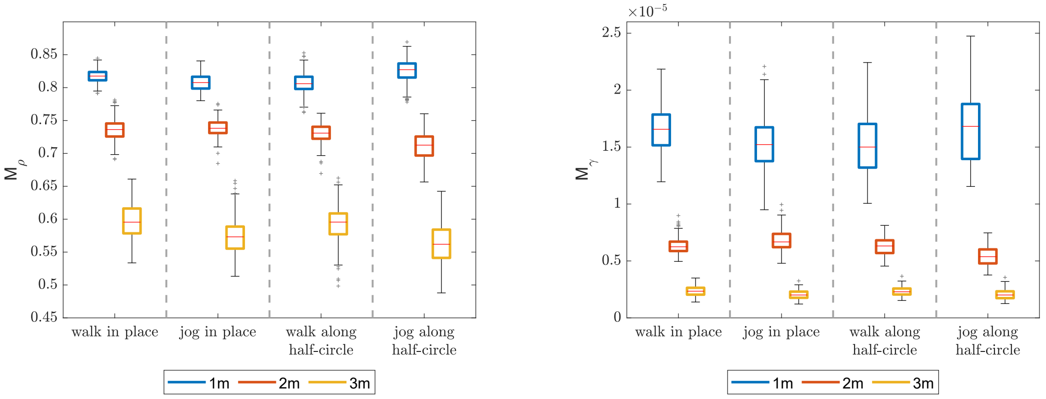
(b) Distribution of M_γ.

Figure 4.13: Impact of the sampling rate. The sampling rate from top to bottom is set to 20 Hz, 30 Hz, 60 Hz, 90 Hz and 180 Hz, respectively.

mentioned experiments with varying channel sampling rates. As we can observe from Fig. 4.13, the averages of both M_ρ and M_γ keep the same level as the sampling rate increases when there exists motion. This is as expected since the two features are extracted in the frequency domain with the assumption that the motion is a stationary random process over time. Therefore, for the proposed proximity detector, a lower channel sampling rate suffices for the sake of better energy efficiency.

4.3.1.7 Impact of Motion Speed

In this section, we study the impact of different body motion speeds. The experiments are carried out in two scenarios. In the first setting, the volunteer performs two types of motions - walking and jogging in place at different human-Rx distances. In the other setting, the user is asked to walk and jog along a circle while



(a) Spread of M_ρ.

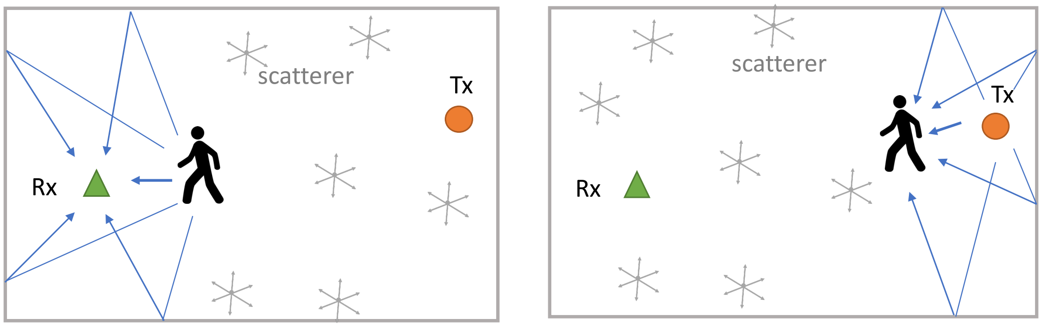
(b) Spread of M_γ.

Figure 4.14: Impact of motion speed.

keeping at a fixed distance to the Rx. The corresponding results are reported in Fig. 4.14(a) and Fig. 4.14(b), respectively. Compared with walking, although jogging produces higher speeds on the torso and limbs including swinging arms, moving legs and feet, it still shows similar distributions of both M_ρ and M_γ as walking.

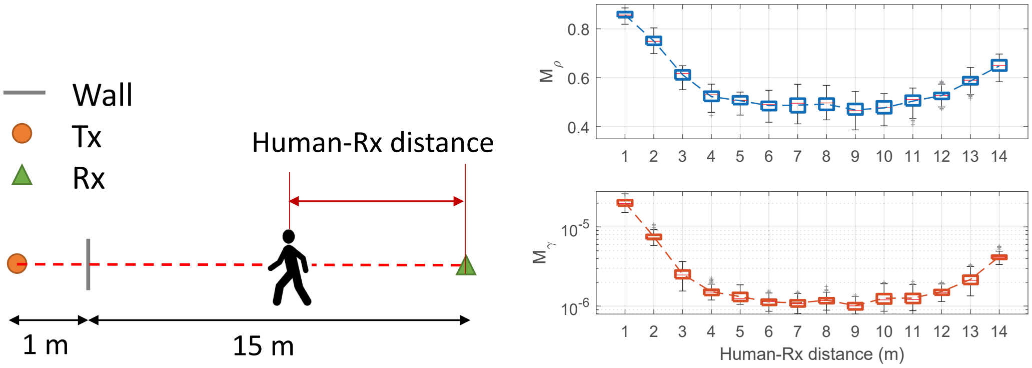
4.3.1.8 Motion on the Link

Due to the channel reciprocity, motion near either the Tx or the Rx impacts the signal propagation. Note that the proposed statistical model considers the moving human body as a collection of multiple virtual antennas. As Fig. 4.15 shows, when the human is in the proximity of the Tx/Rx, the motion can be viewed as a set of virtual Rx/Tx antennas correspondingly and the statistical model still fits. Based on such reciprocity, we can conclude that the proximity features detect the nearest motion to either Tx or Rx. An example is shown in Fig. 4.16, which illustrates the trends of M_ρ and M_γ when the user moves from 1 m away to the Rx to 2 m away to the Tx along the direct link. As expected, the values of M_ρ and M_γ decrease first



(a) (b)

Figure 4.15: Understanding the symmetric behaviors of M_ρ and M_γ using channel reciprocity. (a) Illustration of wave propagation with many scatterers when human is close the Rx. (a) Illustration of wave propagation with many scatterers when human is close the Tx.

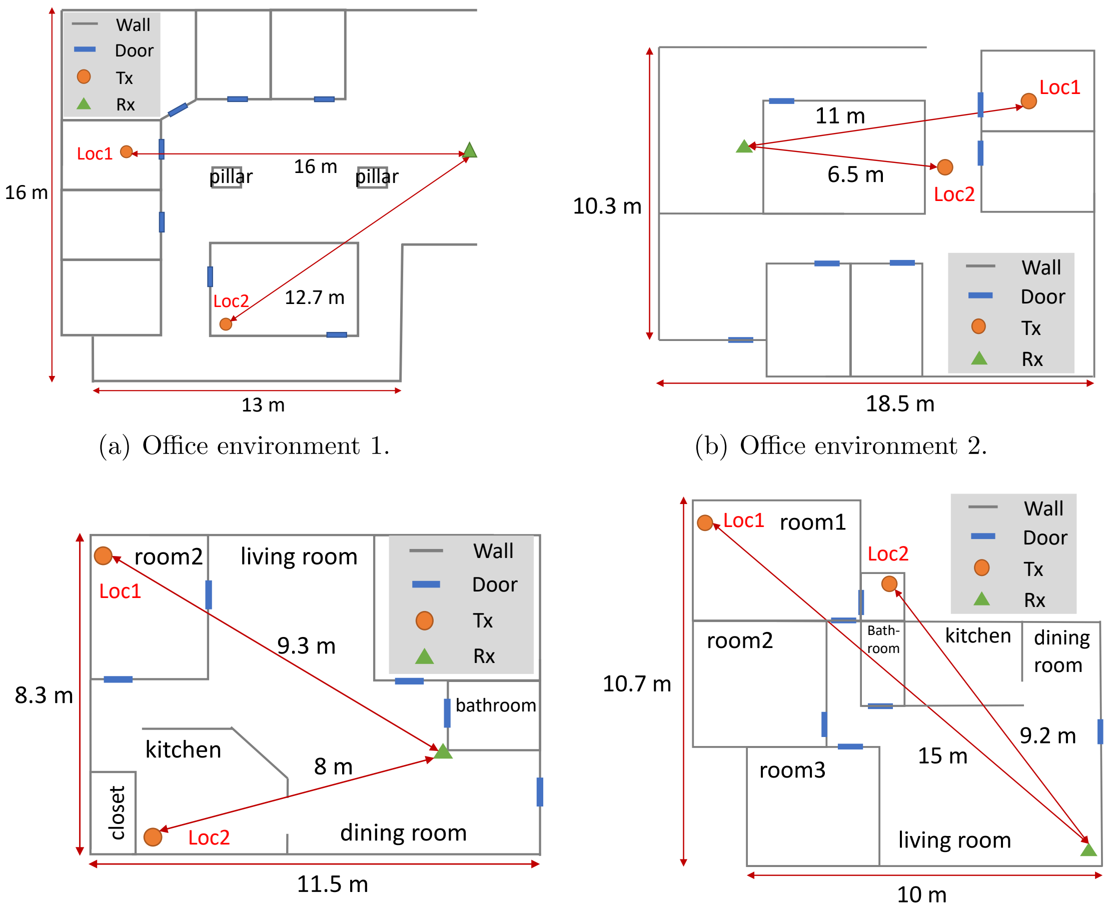


(a) Setup when motion is on the link.

(b) Spread of M_ρ and M_γ.

Figure 4.16: Motion on the link.

as the human moves away from the Rx but increase as the motion gets closer to the Tx. Note that the values ofM_ρ and M_γ are not completely symmetric at the same distances to the Rx and Tx due to the blockage of a wall in front of the Tx. This also shows that the proximity feature can detect motion in proximity in both LOS and NLOS scenarios.



(a) Office environment 1.

(b) Office environment 2.

(c) Home environment 1.

(d) Home environment 2.

Figure 4.17: Illustration of devices placements in different environments.

4.3.1.9 Performance in Different Settings

To examine the robustness of the proposed metrics, we evaluated the performance in various Tx/Rx locations and distances, for both office (Fig. 4.17a-b) and home environments (Fig. 4.17c-d). The accessible area within 2 m from the Rx is considered as the area in proximity. During the experiments, the user is asked to walk in the proximity area and the rest areas (excluding the area near the Tx), respectively. Define TPR as the ratio of correctly detected proximate motions among all movements close to the Rx and FPR as the percentage of faraway mo-

Environment		Office 1	Office 2	Home 1	Home 2	Average
M_ρ	TPR	96.20%	99.07%	96.38%	98.53%	TPR:
Location 1	FPR	4.98%	5.00%	4.96%	4.97%	97.31%
M_ρ	TPR	95.56%	98.65%	95.08%	99.03%	FPR:
Location 2	FPR	4.97%	4.98%	4.99%	4.99%	4.98%
M_γ	TPR	95.89%	99.07%	97.49%	99.12%	TPR:
Location 1	FPR	4.98%	5.00%	4.96%	4.97%	97.90%
M_γ	TPR	95.04%	100%	96.92%	99.68%	FPR:
Location 2	FPR	4.97%	4.57%	4.99%	4.99%	4.93%
WiDetect	TPR	65.58%	75.93%	65.18%	68.33%	TPR:
Location 1	FPR	4.98%	5.00%	4.96%	4.97%	68.29%
WiDetect	TPR	67.62%	68.92%	58.46%	76.05%	FPR:
Location 2	FPR	4.97%	4.98%	4.99%	4.99%	4.98%
SIED	TPR	87.97%	95.06%	91.09%	87.10%	TPR:
Location 1	FPR	4.98%	5.00%	4.96%	4.97%	90.14%
SIED	TPR	88.77%	92.57%	88.62%	89.97%	FPR:
Location 2	FPR	4.97%	4.98%	4.99%	4.99%	4.98%

Table 4.1: Performance in different environments.

tions which are mistaken as nearby motions. Table 4.1 summarizes the performance of all metrics (M_γ, M_ρ and metrics proposed in WiDetect and SIED) in different settings. As shown, M_ρ and M_γ outperforms WiDetect and SIED with the average TPR of 97.31% (29.02% higher than WiDetect and 7.17% higher than SIED) and 97.90% (29.61% higher than WiDetect and 7.76% higher than SIED), respectively at a similar level of FPR smaller than 5%.

4.3.2 Case Studies on Single Pair of Transceivers

To validate the feasibility and the efficiency of the proposed features, we conduct extensive experiments under various settings to emulate real-world applications. Multiple users are involved in the data collection. Based on the monotonic relationship between M_ρ/M_γ and human (Tx$'$)-Rx distance, a threshold-based approach can be applied for detection. To evaluate the proposed metrics, we do not

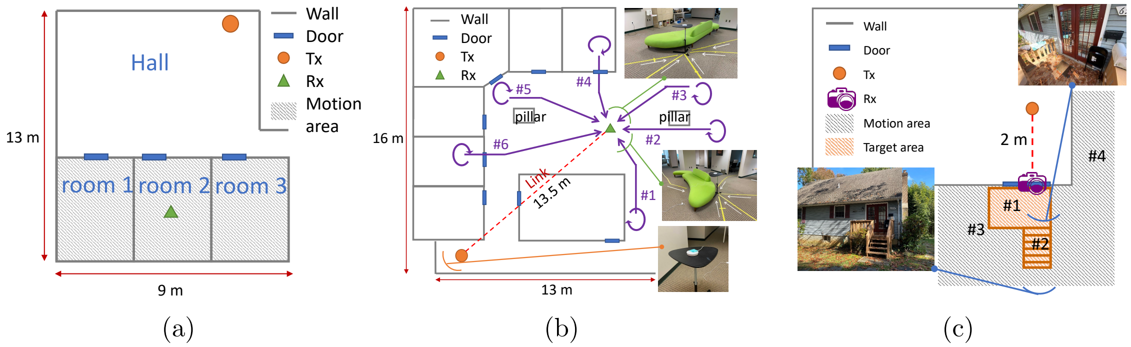


Figure 4.18: Settings for case studies. (a) Room identification; (b) Smart panel; (c) Security camera.

propose specific threshold values but calculate the TPR as well as the FPR with various thresholds and generate the ROC curves for M_ρ and M_γ respectively. Also, we evaluate the responsiveness of the metrics by comparing with the timestamps recorded by cameras at millisecond level.

4.3.2.1 Case 1: Room Identification

Room identification is very useful in home automation. For example, a light or heater inside a room can be automatically turned on once a user is detected to enter that room, and a smart home can localize a robot through tracking the motion at room level. In this set of experiments, there are three adjacent rooms, and the Rx is deployed at the center of the middle room as Fig. 4.18(a) shows. The user is asked to walk randomly inside three rooms, respectively. The motions located in the same room as the Rx are considered as the proximate ones, while those in the other rooms are considered as distant movements. The ROC curves for M_ρ and M_γ to classify proximate and distant motions are illustrated in Fig. 4.19. As seen, with a FPR smaller than 0.5%, the highest TPRs of both M_ρ and M_γ can still reach

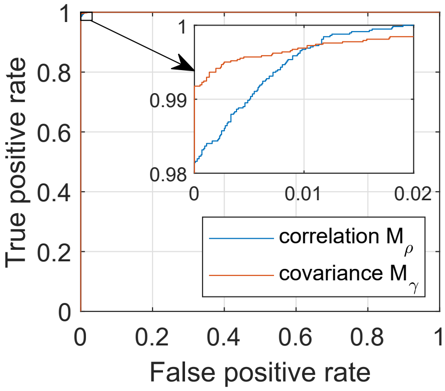


Figure 4.19: ROC curves.

99.11% and 99.62%.

It can be noticed that better performance is achieved in Setting 2 than in Setting 1. This is because for room environments, the motions in Room 1/Room 3 are under NLOS scenarios to the Rx and produce much more dispersed PDPs than the motions in Room 2, which increases the distinction between the derived features. In the corridor setting, the motions to be classified are all under LOS scenarios with the only difference in distance and thus are not as easy to differentiate.

4.3.2.2 Case 2: Approaching the Rx along Different Paths

In this experiment, we consider the motion approaches the Rx from different directions with different starting points. There are six walking paths as indicated in Fig. 4.18(b). The volunteer is asked to walk following the paths at a normal walking speed and multiple cameras are deployed to record the ground truth. Fig. 4.20 shows the increases in both M_ρ and M_γ as the person walks towards the Rx along different paths, conforming to the previous theoretical analysis. We also observe that M_γ is more sensitive to the Tx′-Rx distance with a larger relative increment on the trend

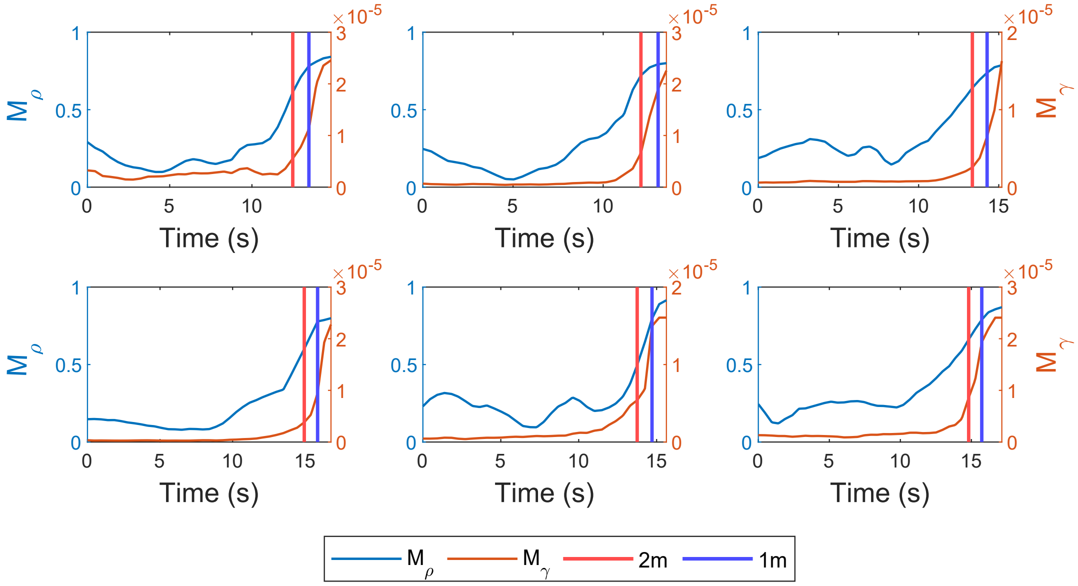


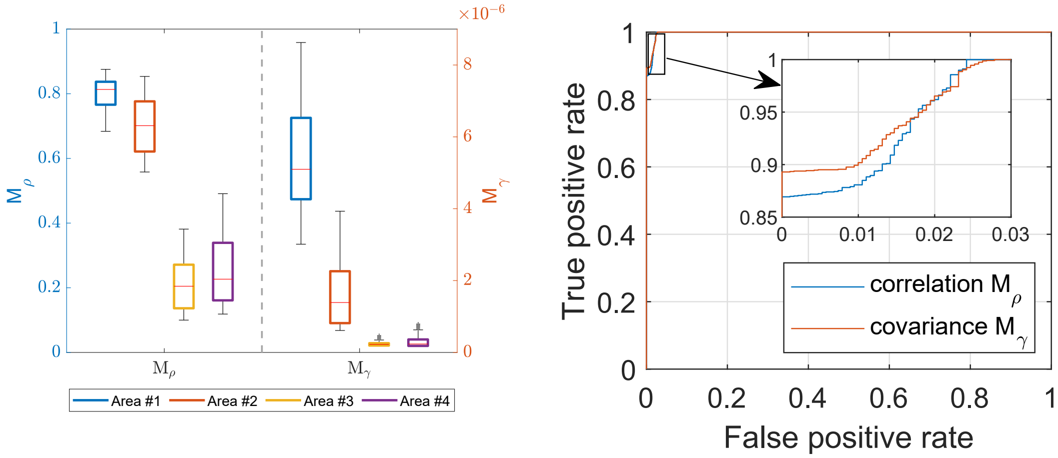
Figure 4.20: M_ρ and M_γ over time while approaching the Rx.

Feature	Path 1	Path 2	Path 3	Path 4	Path 5	Path 6
M_ρ	-0.083s	-0.543s	-0.371s	-0.024s	0.312s	-0.446s
M_γ	-0.175s	-0.357s	0.550s	0.256s	-0.423s	-0.538s

Table 4.2: Detection delay while approaching the Rx.

while M_ρ is more stable as it has been normalized to [-1,1].

To investigate the real-time system latency, we mark the ground truth at distances 1 m and 2 m to the Rx as shown in the pictures in Fig. 4.18(b). Empirical thresholds are applied on M_ρ and M_γ for proximity detection, respectively. By comparing with the timestamps recorded by cameras, we evaluate the delays for detecting the motion within 2 m and summarize them in Table 4.2, where "-" represents the detection that occurs before the person arrives at a 2-meter distance. According to Table 4.2, we can conclude that both metrics can responsively detect the proximity of the human body 2 m away within 0.6 seconds.



(a) Spread of M_ρ and M_γ.

(b) ROC curves.

Figure 4.21: Spread and ROC curves of M_ρ and M_γ in camera control application.

4.3.2.3 Case 3: Home Security Camera

Existing home security systems mostly rely on video-based solutions, such as ring cameras on the front door, and detect motions by either PIR sensors or analyzing sequential frames of live video for differences, which usually requires extra equipment or suffers from high computation and reduces the battery life [5]. To reduce the power consumption, we implement the proximity detector in the home surveillance system so that the camera is turned on only when there is motion in proximity.

More specifically, we consider the wireless camera as the Rx and enable it to sense the surrounding motion through capturing and analyzing the variations of CSI. We attach the camera to the front door of a house and the Tx is placed in the living room. As Fig. 4.18(c) shows, we consider the 1 m × 2 m deck and the stairs at the entrance as the target area (brown shadow) and movements in the target area are regarded as proximate motions, while the motions in the other areas (gray shadow) are recognized as faraway motions. Samples are collected when the volunteer walks in the target area and the surrounding area, respectively. The corresponding M_ρ and M_γ are shown in Fig. 4.21(a), and the performance of the detector is evaluated by the ROC curves in Fig. 4.21(b).

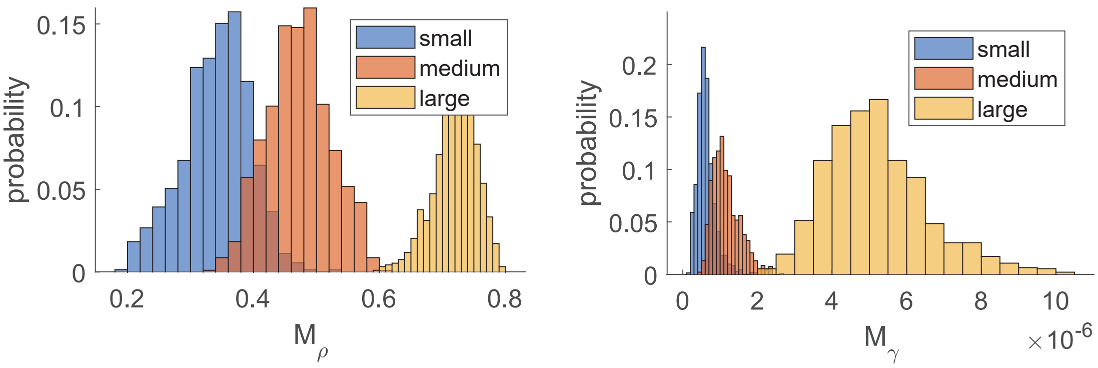
Compared with the previous indoor cases, the experiments on the ring camera application are conducted in the semi-outdoor scenario, i.e., the user comes from outside and therefore produces fewer propagation paths. However, even in such a semi-outdoor environment, the proposed features work effectively in distinguishing motions nearby and far away with a TPR of 95% while the corresponding FPR is lower than 2%.

4.4 Discussion

4.4.1 Impact of Motion Strength

In this work, we demonstrate that the correlation and covariance of CSI power between neighboring subcarriers act as effective metrics for sensing motion in proximity. Since we focus on detecting a user approaching one sensing device, only body-level motion (e.g., walking) is taken into consideration. When the user is in light motion such as reading or typing, the fluctuations caused by motion are small or even comparable to noise.

As seen in Eqn. (4.8) and (4.9), both features M_ρ and M_γ are impacted by σ_d, which reflects the dynamic energy level, i.e., the motion strength. The motion



(a) Distribution of M_ρ. (b) Distribution of M_γ.

Figure 4.22: Impact of motion strength.

strength can be affected by the dynamic reflection area. The larger the dynamic reflection area, the higher the motion strength. To investigate its impact, the user performs motions at a distance of 2 m to the Rx with different dynamic levels including single-hand waving, double-hand waving and full-body motion, which have increasing dynamic reflection areas and thus, increasing motion strengths. Fig. 4.22 presents the distributions of M_ρ and M_γ for these three motions. As illustrated, larger M_ρ and M_γ are observed with a higher motion strength.

4.4.2 Multi-Person Case

Currently, we only focus on the cases when there is a single person or no strong ambient motion near the Rx. With the presence of multiple persons, the dynamic signals reflected by different targets can be superimposed together at the Rx side. Due to the limited Wi-Fi bandwidth (40 MHz), it is difficult for commercial Wi-Fi devices to separate the reflected signals and derive the real number of moving targets accurately. For the proposed proximity detection, it can distinguish whether motion exists around the devices, but cannot tell how many moving persons there

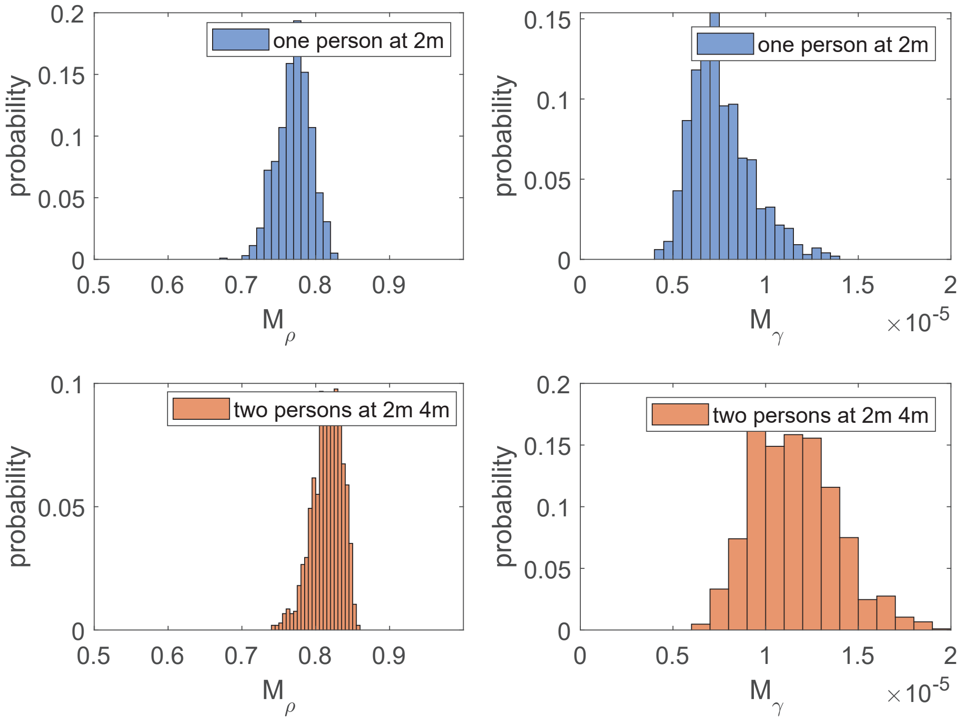


Figure 4.23: Distributions of M_ρ and M_γ when there are a single person at 2 m and two persons at 2 m and 4 m distances to the Rx.

are. However, as Fig. 4.23 shows, we observe that when more persons walk around the Rx, the distributions of M_ρ and M_γ are centered at larger values than the single-person case, which can give clues for the solution to multi-person sensing and could be an important direction for our future work.

4.4.3 Other Metrics Derived from Correlation across Subcarriers

4.4.3.1 The Correlation of Complex-Valued $\tilde{H}_d$

Inspired by the model in Section 4.1, the correlation coefficients of complex-valued $\tilde{H}_d$ across adjacent subcarriers can be used for detecting motion in the proximity of the devices. The covariance of $\tilde{H}_d$ across adjacent subcarriers has its real

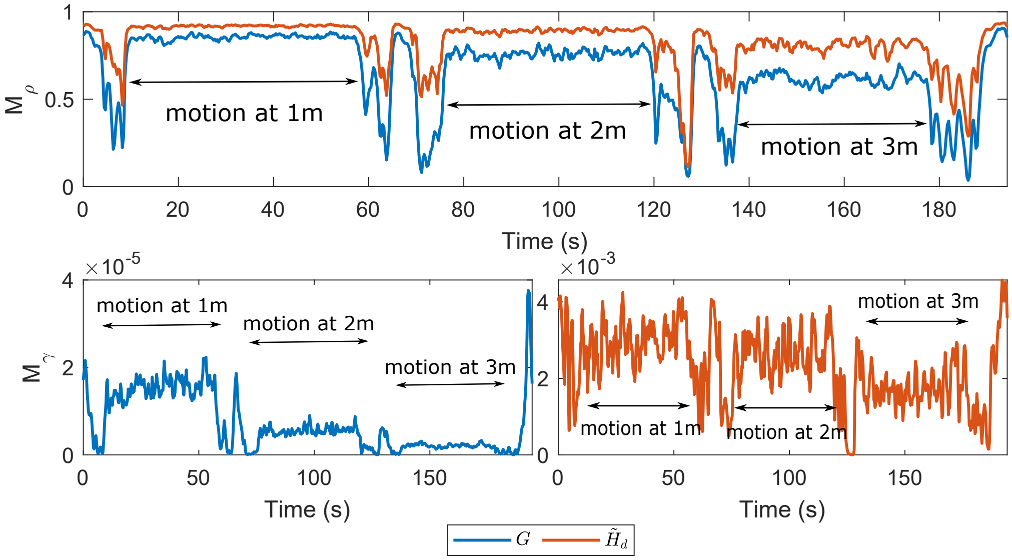


Figure 4.24: Correlation and covariance of $G(t, f)$ and $\tilde{H}_d(t, f)$ between adjacent subcarriers.

part expressed as

$$\mathrm{Re}\{\gamma_{\tilde{d}}(f_1, f_2)\} = \sigma_d(f_1)\sigma_d(f_2)\mathrm{Re}\{\rho_d(f_1, f_2)\} + \sigma_\varepsilon(f_1)\sigma_\varepsilon(f_2)\delta(f_1, f_2). \tag{4.14}$$

The corresponding correlation is

$$\begin{aligned}\mathrm{Re}\{\rho_{\tilde{d}}(f_1, f_2)\} = &\frac{\sigma_d(f_1)\sigma_d(f_2)}{\sqrt{\sigma_d^2(f_1) + \sigma_\varepsilon^2(f_1)}\sqrt{\sigma_d^2(f_2) + \sigma_\varepsilon^2(f_2)}}\mathrm{Re}\{\rho_d(f_1, f_2)\} + \\ &\frac{\sigma_\varepsilon(f_1)\sigma_\varepsilon(f_2)}{\sqrt{\sigma_d^2(f_1) + \sigma_\varepsilon^2(f_1)}\sqrt{\sigma_d^2(f_2) + \sigma_\varepsilon^2(f_2)}}\delta(f_1, f_2).\end{aligned} \tag{4.15}$$

As implied, $\mathrm{Re}\{\rho_{\tilde{d}}(f_1, f_2)\}$ directly indicates the distance between the moving target and the device as both the coefficient $\frac{\sigma_d(f_1)\sigma_d(f_2)}{\sqrt{\sigma_d^2(f_1)+\sigma_\varepsilon^2(f_1)}\sqrt{\sigma_d^2(f_2)+\sigma_\varepsilon^2(f_2)}}$ and the term $\mathrm{Re}\{\rho_d(f_1, f_2)\}$ have a monotonic relationship with human-device distance. Fig. 4.24 shows the trends of correlation and covariance of both $G(t, f)$ and $\tilde{H}_d(t, f)$, respectively. As seen, $\mathrm{Re}\{\gamma_{\tilde{d}}(f_1, f_2)\}$ and $\mathrm{Re}\{\rho_{\tilde{d}}(f_1, f_2)\}$ can also differentiate the motion

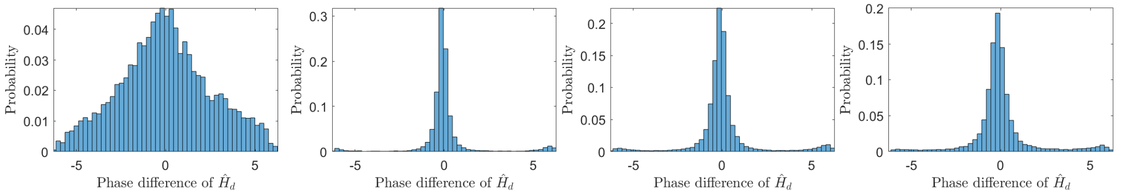


Figure 4.25: Distributions of $\phi_{\tilde{d}}(f_1) - \phi_{\tilde{d}}(f_2)$ with f_1 and f_2 representing the frequencies of adjacent subcarriers. The figures from left to right correspond to "empty","motion at 1 m","motion at 2 m" and "motion at 3 m".

Case	Empty	Motion at 1 m	Motion at 2 m	Motion at 3 m
Kurtosis	2.645	11.6577	8.9844	7.8528

Table 4.3: Detection delay while the human approaches the Rx.

at different distances but are not as distinct as $M_\gamma(f_1, f_2)$ and $M_\rho(f_1, f_2)$. This is because the estimation of complex $H_d(t, f)$ requires dedicated phase cleaning which introduces noise easily.

4.4.3.2 The Distribution of Phase Differences of $\tilde{H}_d$ between Adjacent Subcarriers

The correlation between complex $\tilde{H}_d$ on adjacent subcarriers can be reflected by their phase difference. When there is no motion in the surrounding, only noise contributes to the phase difference between adjacent subcarriers with frequencies f_1 and f_2, which exhibits a triangle distribution as the convolution of two uncorrelated uniform density functions. In contrast, when there exists nearby motion, $\tilde{H}_d(f_1)$ and $\tilde{H}_d(f_2)$ are highly correlated with each other and their phase difference will concentrate at zero. We use kurtosis to evaluate the sharpness and concentration of distribution of the phase difference. Fig. 4.25 presents the distributions of phase difference $\phi_{\tilde{d}}(f_2) - \phi_{\tilde{d}}(f_2)$ with motion at different distances. As illustrated, as human movements get closer to the device, the distribution becomes centralized around zero. Quantitatively, the kurtosis, as Table 4.3 shows, increases as the human-device distance decreases. However, the acquisition of phase $\phi_{\tilde{d}}(t, f)$ involve the phase calibration preprocessing and can be noisy. In addition, the accumulation of samples is needed to form the distribution and causes delays in the real-time monitoring.

4.4.3.3 The Coefficients on Motion Strength

As analyzed in Section 4.1, with human-device distance increases, the interference from the dynamic target gets smaller, i.e., σ_d^2 decreases. Therefore, the ratio coefficient in Eqn. (4.15) can also reflect the distance between the subject and devices, which, denoted as ζ, is represented by

$$\zeta(f_1, f_2) = \frac{\sigma_d(f_1)\sigma_d(f_2)}{\sqrt{\sigma_d^2(f_1) + \sigma_\varepsilon^2(f_1)}\sqrt{\sigma_d^2(f_2) + \sigma_\varepsilon^2(f_2)}} \tag{4.16}$$

Implied by Eqn. (4.16), ζ is confined to the range $[0, 1)$. Compared with $M_\rho(f_1, f_2)$ and $M_\gamma(f_1, f_2)$, $\zeta(f_1, f_2)$ does not require the continuity of subcarriers, i.e., f_1 and f_2 are not necessarily to be frequencies of adjacent subcarriers. However, to calculate $\zeta(f_1, f_2)$, phase calibration is required to estimate the complex-valued H_s and $\tilde{H}_d$. Then with the static component H_s subtracted, the ratio $\frac{\sigma_d}{\sigma_\varepsilon}$ can be estimated by comparing the H_d under empty and motion scenarios. Fig. 4.26 shows the distributions of ζ with different human-device distances. As expected, ζ decreases as the human-device distance increases, which, however, is not as distinguishable as M_ρ

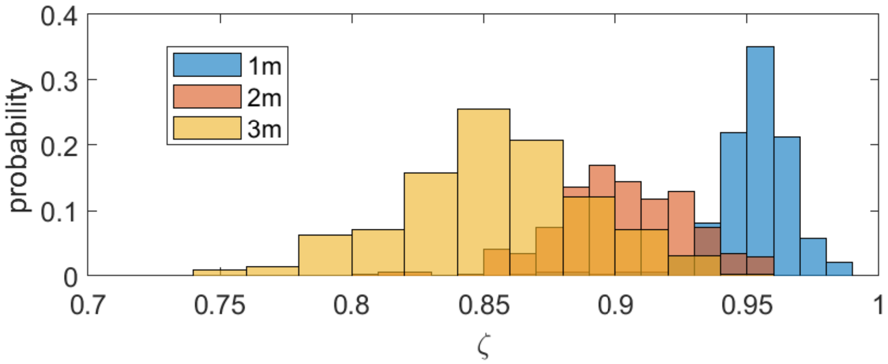


Figure 4.26: The distribution of ratio coefficient ζ at different distances.

and M_γ due to the calibration error in the processing of CSI phase calibration.

4.5 Summary

In this chapter, we propose two robust and responsive features, the correlation and covariance of CSI power on adjacent subcarriers, for detecting motion in proximity based on a single pair of commercial Wi-Fi devices and explore the underlying radio propagation properties. Extensive experiments in various environments validate the effectiveness of the proposed feature-based detection scheme.

Chapter 5: Home Automation Application via On-Device Proximity Detection

5.1 Background

It has been a few years since the home automation technology spawned. Home automation is exactly what it sounds like: automating the ability to control items around the house. Instead of going up to the devices and manually taking actions, the devices themselves can monitor and even predict the activities and react accordingly.

Home automation is powered by IoT. As Fig. 5.1 shows, through the adoption of the IoT into homes, all the devices in our life could connect to Wi-Fi, not only computers and smartphones but everything from small appliances such as lights and coffee makers to a large refrigerator and even a whole security system.

Many energy-efficient home automation applications involve location-based services, especially the on-device proximity of motion. For example, smart light turns on automatically when users get close; security cameras offer benefits through alarming and opening real-time video when detecting the presence of motion in the area of interest; a baby monitor near the bed can track babies' movements

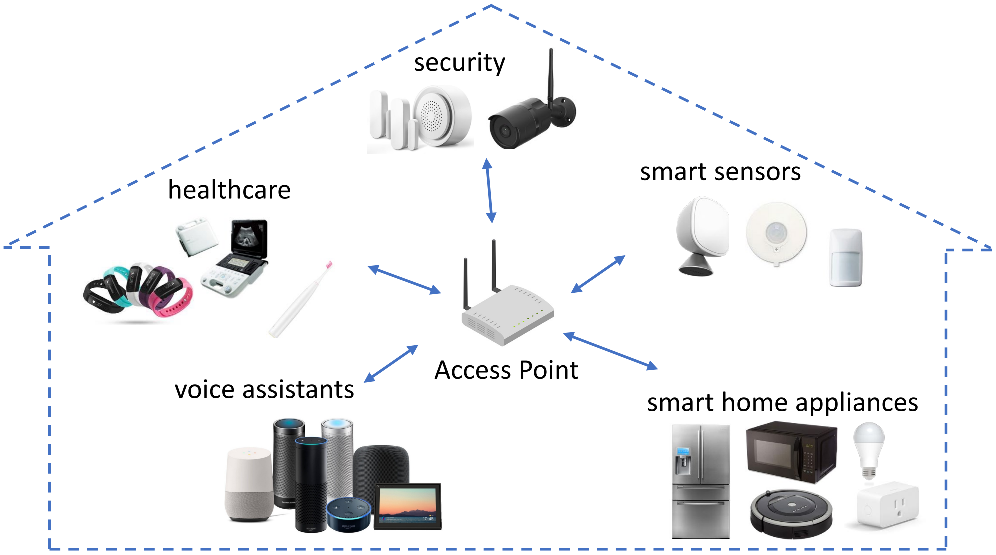


Figure 5.1: An illustration of IoT-based home automation system.

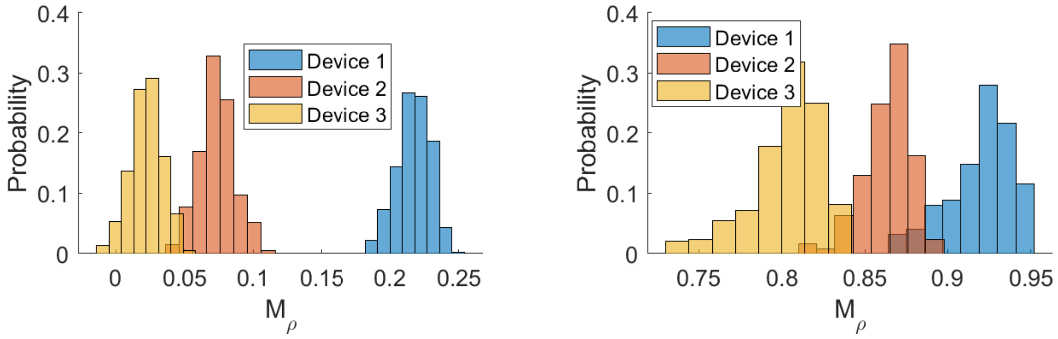
while they're sleeping and alert parents to any stillness that may signal a problem. Further, it is well known that activities are closely related to areas, e.g., cooking in the kitchen and showering in the bathroom. Through a fusion of proximity detection on multiple devices, location-based information can be extracted and contribute to the recognition of activities of daily living (ADL) as well as the customized design of home routines for whole-home automation. Therefore, on-device proximity detection is useful in home automation systems.

As discussed in Chapter 1, Wi-Fi sensing plays an important role in IoT application realizations thanks to the ubiquitous wireless signals in homes. Due to the rich information introduced into the Wi-Fi signals by human activities, Wi-Fi embedded IoT devices can perceive the surrounding motion automatically, which further reduces manual intervention and make the home "smarter". Thus Wi-Fi based on-device proximity detection is particularly attractive to facilitate various home automation.

5.2 Challenges

Enabling the on-device proximity detection on multiple devices for real-time home automation using Wi-Fi signals is not an easy task. Multiple challenges need to be addressed. First, although the capability of penetrating walls and other obstacles of Wi-Fi radios facilitate the whole-home coverage of Wi-Fi, it is not a benefit for the purpose of proximity detection. Therefore, it is essential to develop a Wi-Fi-based feature that is sensitive to the motion near the device within a particular small range. However, the existing Wi-Fi-based on-device motion detection systems usually aim at detecting the motions in a large coverage without knowing the distance of the target motion from the device [43, 97, 107], while existing localization techniques require high deployment efforts and/or do not work robustly [35, 58, 90, 102]. Therefore, they are not practically useful for motion proximity detection. The metric for proximity detection should ideally satisfy the following three requirements to be utilized in the real-world home automation systems: 1) *High sensitivity* to the distance between motion and the target device; 2) *Robustness* under various sampling rates to accommodate different IoT devices at different sounding frequencies; 3) *Low complexity* to reduce the computation cost and enable the system to run in a real-time manner. To address this challenge, we follow our prior work in Chapter 4, implement the feature M_ρ from the correlation coefficients between adjacent subcarriers as the metric to evaluate if the motion is in the proximity of devices or not.

However, a home automation system may contain a wide range of various smart



(a) Distribution of M_ρ under empty scenario.

(b) Distribution of M_ρ with nearby motion.

Figure 5.2: Impact of hardware diversity.

home devices. The chipsets in different devices, produced by different manufactures could be diverse in the sensitivity to dynamics and the number of subcarriers, which will affect the values of M_ρ and also the specific detection boundaries. In addition, different devices could be customized and assigned with different monitored areas. Fig. 5.2 shows the distributions of correlation coefficients of 3 different co-located devices connected to the same CSI Rx. As we can see, under empty and nearby motion scenarios, different devices may have distinct values of M_ρ. To combat the hardware diversity, a simple training phase is involved to adaptively learn the detection boundaries for individual devices.

Third, given the detection of each device, it is still difficult to tell where the moving target is, either at the Tx or Rx side, due to the channel reciprocity. Fortunately, in a home automation system, we usually have multiple IoT devices connected to the same home router and therefore, the Rx will receive CSI simultaneously from multiple Tx. To overcome the challenge of channel reciprocity, we integrate the information from individual devices and make the decision based on the link fusion.

5.3 Range-aware Home Automation using Multiple Devices

Topology of device placements A home automation system typically connects controlled devices to a central hub with a user interface. The most typical and realistic topology of the IoT devices placements is taken into consideration, in which the CSI Rx (i.e., the home router) is placed in the center of the home, while the IoT devices are placed randomly in the house, connected to but far away from the central Rx.

Metric Based on the analysis in Chapter 4, two metrics, M_ρ and M_γ are effective in detecting proximity detection while keeping robust under different sampling rates. Extensive experiments show the similar performance of M_ρ and M_γ in different environments and scenarios. We select M_ρ as the on-device proximity detection feature to implement in the home automation applications. Then the threshold-based detection rule is applied on the calculated M_ρ.

System overview Due to the aforementioned hardware diversity, the large number of various devices connected to the home router and the different detection areas of different devices, it is difficult to find a universal detection boundary for all devices. Therefore we apply a simple training step to adaptively find the detection thresholds. With the obtained boundaries for each individual device, we further combine all their detection results to localize the target user. Fig. 5.3 shows the system overview.

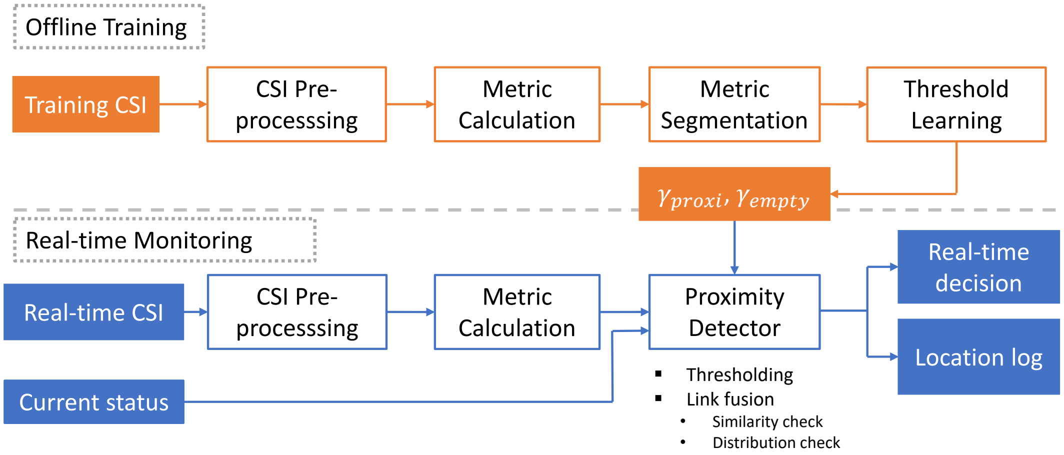


Figure 5.3: System diagram.

5.3.1 Offline Training

A simple offline training phase is involved to determine the boundary of the proximity detection on each device. The detailed steps are listed as follows.

- **Data Acquisition** The user walks inside the monitored area around each device for several minutes (e.g., 2 minutes in our experiments). The CSI is collected continuously in time.
- **Metric Calculation** After time sequences of CSI measurements are obtained, the proximity is evaluated by calculating the corresponding M_ρ. As illustrated by Section 4.2.3 in Chapter 4, we first preprocess the CSI by applying the amplitude normalization and Hampel outlier removal. Then following Eqn. (4.12), we get the M_ρ values over time. Denote the metric measured on link Tx$_k$-Rx as $M_{\rho,k}$. For a total of K IoT devices, we can get K sequences of M_ρ.
- **Metric Segmentation** Fig. 5.4 shows an instance of the continuous M_ρ sequence when the user walks in different areas, which presents distinct seg-

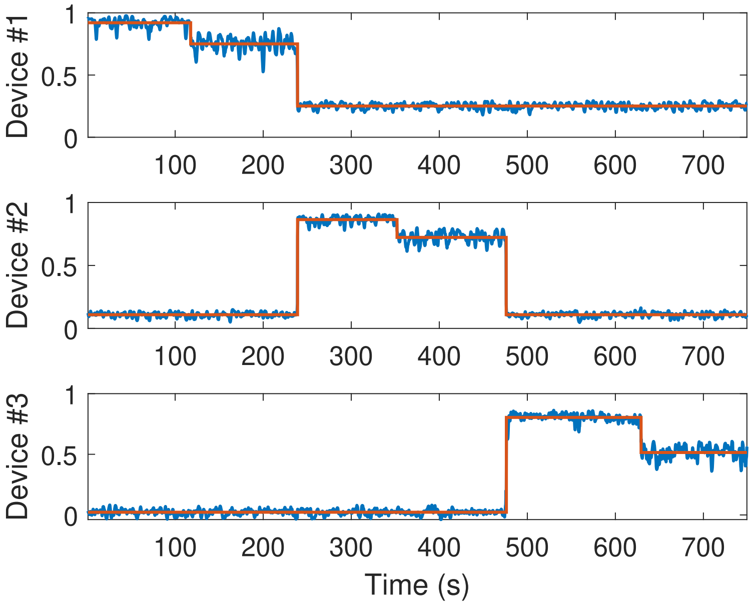


Figure 5.4: Metric segmentation via change point detection.

ments. With the prior knowledge that the motion in the proximity of the target Tx will produce the largest M_ρ, we can separate different segments through change point detection and extract the segment with the largest mean values. Given an ordered sequence of data, $y_{1:n} = (y_1, \cdots, y_n)$, the change point analysis aims to identify a number of points, m, together with their positions, $\tau_{1:m} = (\tau_1, \cdots \tau_m)$ where the statistical property (i.e., the mean of M_ρ in our system) changes. Define $\tau_0 = 0$ and $\tau_{m+1} = n$ and assume that the change points are ordered as $\tau_0 < \tau_i < \cdots < \tau_m < \tau_{m+1}$. Consequently the m change points will split the data into $m + 1$ segments. The problem of change point identification can be formulated as the minimization of

$$\sum_{i=1}^{m+1} \left[C(y_{(\tau_{i-1}+1):\tau_i}) \right] + \mu m, \tag{5.1}$$

where C is a cost function for a segment. μm is a penalty to guard against overfitting and increases linearly with the number of change points [33]. We segment data based on the mean and therefore set $C(y_{(\tau_{i-1}+1):\tau_i}) = \bar{y}_{(\tau_{i-1}+1):\tau_i}$.

- **Threshold Learning** On the k-th device, the threshold for differentiating between the nearby and faraway motion is determined by

$$\gamma_{\text{proxi}}(k) = \alpha \text{med}_t M_\rho^1(t,k) + (1-\alpha)\text{med}_t M_\rho^0(t,k), \tag{5.2}$$

where $M_\rho^1(t)$ denotes the M_ρ sequence with nearby motion while $M_\rho^0(t)$ denotes the M_ρ series with motion farther away. α, $0 \leq \alpha \leq 1$, is a sensitivity coefficient for proximity detection in that the sensitivity of the proposed motion detector increases as α decreases. In addition, given the M_ρ segment corresponding to empty case, the threshold for differentiating between the faraway motion and empty on the k-th device is determined by

$$\gamma_{\text{empty}}(k) = \beta \text{med}_t M_\rho^{-1}(t,k) + (1-\beta)\text{med}_t M_\rho^0(t,k), \tag{5.3}$$

where $M_\rho^{-1}(t)$ denotes the M_ρ sequence labeled empty. β, $0 \leq \beta \leq 1$, is a sensitivity coefficient to detect the silent period of the k-th device.

5.3.2 Online Monitoring

Timely detection During the online monitoring phase, the IoT devices act based on the proximity detection on each individual device and their fusion. The transition between the states of the k-th device at time t can be written as

$$\mathrm{S_{nonactive}(t,k)} \underset{\mathrm{D_{empty}(t,k)=1}}{\overset{\mathrm{D_{proxi}(t,k)=1}}{\rightleftharpoons}} \mathrm{S_{active}(t,k)} \tag{5.4}$$

where $D_{\text{proxi}}(t,k)$ and $D_{\text{empty}}(t,k)$ indicate the detection of proximity and empty around the k-th IoT devices, respectively.

However, when motion happens near the central Rx, all links will have large proximity features due to channel reciprocity. In addition, when more human subjects move around different IoT devices simultaneously, the corresponding links will also become active. To accommodate the motion near different IoT devices, we make an assumption that # of users $\leq$ # of IoT devices, which is reasonable as there are usually a large number of IoT devices connected in the same home environment. Motion in the proximity of the k-th IoT device is recognized by comparing the real-time $M_\rho(t,k)$ with its own γ_{proxi} and integrating multiple links as

$$D_{\text{proxi}}(t,k) = \begin{cases} 1, & \text{conditions \#1\&2\&3 are satisfied,} \\ 0, & \text{otherwise,} \end{cases} \tag{5.5}$$

where $D_{\text{proxi}}(t,k) = 1$ indicates the existence of motion nearby, i.e., someone is moving in the monitoring area, while $D_{\text{proxi}}(t,k) = 0$ means the environment is static or the motion is out of the target area. The three conditions for decision rules are as following:

Condition #1: Thresholding check. At time t_0, the proximity metric value on the k_1-th device exceeds its detection boundary, i.e., $M_\rho(t,k_1) \geq \gamma_{\text{proxi}}(k_1)$. With

Condition #1 satisfied at moment t_0, *Condition #2* and *#3* are further checked.

Condition #2: Distribution check. Intuitively, with the motion in the proximity of the k_1-th device, detection by the other devices far away from the target motion will not be triggered. Under the assumption that # of users $\leq$ # of IoT devices, there is at least one link having the majority of its $M_\rho(t)$ below γ_{empty} during the past duration $[t_0 - T, t_0]$. Therefore, we formulate this requirement as $\exists k_2 \neq k_1$ with $\frac{\sum_{t=t_0-T}^{t_0} \mathbb{1}\left(M_\rho(t,k_2)<\gamma_{\text{empty}}\right)}{T} > \eta_1$, where η_1 is the threshold for distribution check.

Condition #3: Similarity check. If there is motion approaching the central Rx, M_ρ on all devices will increase similarly and thus the correlation coefficients across the M_ρ of different links in a past duration T will be high. With the assumption of # of users $\leq$ # of IoT devices, at least one of the correlation coefficients between links is very small if no motion is around the Rx, which is formulated as $\exists m \neq n$ with $\rho\Big(M_\rho(t,k_m), M_\rho(t,k_n)\Big) < \eta_2$ for $t = t_0 - T, t_0 - T + 1, \cdots, t_0$.

The decision-making rule of obtaining $D_{\text{empty}}(t,k)$ is expressed as

$$D_{\text{empty}}(t,k) = \begin{cases} 1, & M_\rho(t,k) \leq \gamma_{\text{empty}}(k), \\ 0, & \text{otherwise}, \end{cases} \tag{5.6}$$

where $D_{\text{empty}}(t,k) = 1$ indicates the absence of motion in the surrounding environment, while $D_{\text{empty}}(t,k) = 0$ means the environment is not completely static. Based on the transition between $S_{\text{nonactive}}$ and S_{active}, responsive proximity detection can be achieved during online monitoring.

Location log generation Further, given the state transition over time and the location of each device, the location log can be generated accordingly.

The timely detector captures the moment the target enters the area of the proximity of the device but has delays when the state transfers from "S_{active}" to "$S_{\text{nonactive}}$" in order to reduce the false negative error of wrongly deactivating the target Tx during the real-time monitoring. The log generation based on the post-processing, however, allows more delicate state transition and can even detect the motion around the central Rx. The states transition of the k-th device in the location log can be expressed as

$$\mathrm{L}_{\text{nonactive}}(\mathrm{t},\mathrm{k}) \xrightleftharpoons[\mathrm{D}_{\text{nonproxi}}(\mathrm{t},\mathrm{k})=1]{\mathrm{D}_{\text{proxi}}(\mathrm{t},\mathrm{k})=1} \mathrm{L}_{\text{active}}(\mathrm{t},\mathrm{k}), \tag{5.7}$$

where the input $D_{\text{proxi}}(t,k)$ is evaluated by Eqn. (5.5), while $D_{\text{nonproxi}}(t,k)$ is the non-proximity decision, which is written as

$$D_{\text{nonproxi}}(t,k) = \begin{cases} 1, & \text{conditions \#4\&5\&6 are satisfied,} \\ 0, & \text{otherwise,} \end{cases} \tag{5.8}$$

and has its three conditions as the reverse of conditions #1, #2, and #3 over time, respectively, that is,

Condition #4: Thresholding check. If the current state of the k_1-th device is $S_{\text{nonactive}}$ switched from S_{active}, then trace back along $t, t-1, t-2, \cdots$, until at t_0 we have $M_\rho(t_0,k_1) \geq \gamma_{\text{proxi}}(k_1)$. With *Condition #4* satisfied at moment t_0, *Condition #5* and *#6* are further checked.

Condition #5: Distribution check. $\exists k_2 \neq k_1$ with $\frac{\sum_{t=t_0}^{t_0+T} \mathbb{1}\left(M_\rho(t,k_2)<\gamma_{\text{empty}}\right)}{T} > \eta_1$.

Condition #6: Similarity check. $\exists m \neq n$ with $\rho\Big(M_\rho(t, k_m), M_\rho(t, k_n)\Big) < \eta_2$ for $t = t_0, t_0 + 1, \cdots, t_0 + T$.

We also have the log of motion surrounding central Rx as

$$\mathrm{L_{nonactive}(t, Rx)} \xrightleftharpoons[\mathrm{D_{nonproxi}(t, Rx)=1}]{\mathrm{D_{proxi}(t, Rx)=1}} \mathrm{L_{active}(t, Rx)}, \tag{5.9}$$

where the input decisions $D_{\text{proxi}}(t, \text{Rx})$ and $D_{\text{nonproxi}}(t, \text{Rx})$ are written as

$$D_{\text{proxi}}(t, \text{Rx}) = \begin{cases} 1, & \text{for } \forall k,\ M_\rho(t, k) \geq \gamma_{\text{proxi}}(k), \\ 0, & \text{otherwise}, \end{cases} \tag{5.10}$$

and

$$D_{\text{nonproxi}}(t, \text{Rx}) = \begin{cases} 1, & \text{for } \forall k,\ M_\rho(t, k) \leq \gamma_{\text{proxi}}(k), \\ 0, & \text{otherwise}. \end{cases} \tag{5.11}$$

5.3.3 Experimental Results

5.3.3.1 Experimental Setup

We build a prototype of the home automation system on commercial devices with 5 GHz carrier frequency and 40 MHz bandwidth. The devices are deployed in an office environment as Fig. 5.5 presents. There are three Tx connected to the same home router, i.e., the central Rx, each having its own customized monitored area. During the testing, the user approaches each Tx and the Rx following different paths and walks around in the target areas.

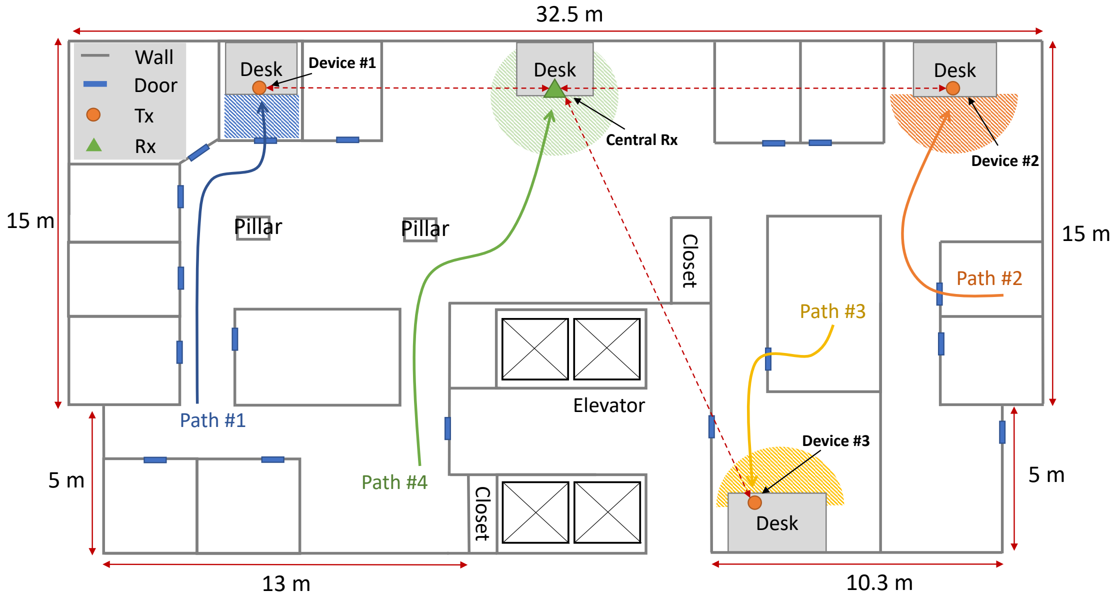


Figure 5.5: Experimental environments.

In order to verify the feasibility and evaluate the performance of the proposed system in indoor monitoring, extensive experiments have been conducted. To evaluate the system responsiveness, we measure the detection delays compared with the recorded timestamps by cameras. We also plot the generated location log together with the ground truth for comparison.

5.3.3.2 Single-User Case

Under the presumption of a single person, we can detect if the user enters the monitored area of each individual Tx using Eqn. (5.5). However, as Section 4.3.1.8 in Chapter 5 indicates, the motion around the Rx will also produce a large M_ρ.

Luckily, with information gathered from multiple links, the system can react based on the fusion decision. Fig. 5.6 shows the distinct patterns when the user is approaching a specific Tx and the Rx. As illustrated, when the user approaches the

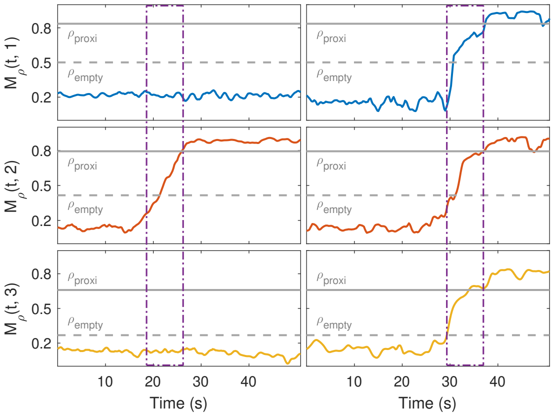


Figure 5.6: Proximity patterns when the user approaches Tx, more specifically, Device #2 (left) and central Rx (right). From top to bottom are the trends of M_ρ on Device #1, #2 and #3, respectively.

Output	+ Tx	+ Rx
$\rho(M_{\rho,1}, M_{\rho,2})$	-0.2690	0.8770
$\rho(M_{\rho,1}, M_{\rho,3})$	0.0309	0.9791
$\rho(M_{\rho,2}, M_{\rho,3})$	-0.0174	0.9063
R_1	1.00	0.175
R_2	0.04	0.225
R_3	1.00	0.00

Table 5.1: Corresponding correlation across links and distribution on each link, where "+" represents "approaching". ρ and R are the outputs of similarity and distribution checks.

Rx, values of $M_\rho(t, k)$ on all three devices increase at the same time and present similar trends. Note that the M_ρ on different devices may not be triggered at the same time which may easily cause a false alarm. To reliably decide which Rx is active, once one of the devices has M_ρ exceeding its threshold, we examine the similarity of M_ρ trends in a past duration between all devices. As shown in Table 5.1, when the user approaches the Rx, the correlations between all pairs of links are high due to the consistent increase of all links. In addition, we also check the distribution of M_ρ on each device as Table 5.1 summarizes. When the user is getting closer to the single Device #2, all of M_ρ on Device #1 and #3 maintain below empty boundary γ_{empty}, while the majority of them exceed γ_{empty} as the user approaches the central Rx.

Fig. 5.7 illustrates the trends of M_ρ on multiple devices when the user approaches different devices as Table 5.2 describes. Table 5.3 records the detection delays. As shown, each individual device reacts responsively to the motion in prox-

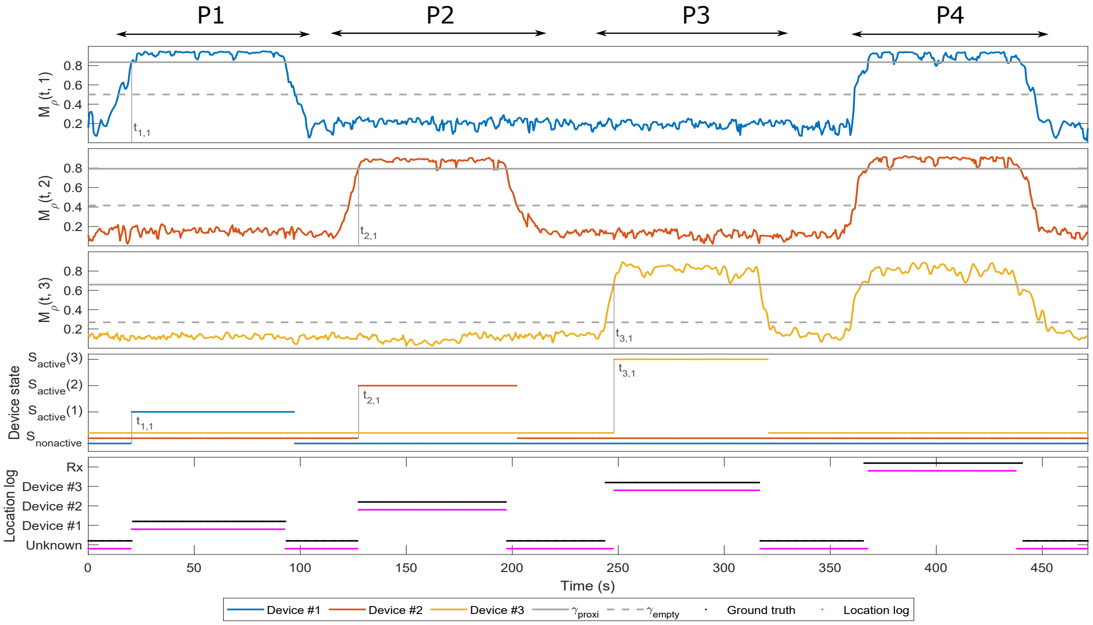


Figure 5.7: The real-time states of IoT devices and the corresponding location log under the single-user scenario. From top to bottom the figures show: $M_\rho(t, 1)$, $M_\rho(t, 2)$, $M_\rho(t, 3)$, the states S and the location log.

Period	Description
P_1	Person A + Device #1 → Person A - Device #1
P_2	Person A + Device #2 → Person A - Device #2
P_3	Person A + Device #3 → Person A - Device #3
P_4	Person A + Rx → Person A - Rx

Table 5.2: Events with a single user (Person A) at home, where "+" stands for "approaching" and "-" means "walking away from".

imity with a delay of less than 0.3 s. For the motion near the central Rx, the generated location log can recognize it accurately, verifying the effectiveness of the proposed system.

5.3.3.3 Multi-User Case

With the assumption of # of users ≤ # of IoT devices, while different users approach different devices, the proposed scheme can detect the users in a timely manner and generate the location log. However, if the target user approaches the

Device	P_1	P_2	P_3	P_4
#1	-0.666s	-	-	-
#2	-	0.256s	-	-
#3	-	-	-0.362s	-

Table 5.3: Detection delays in 1-user scenarios.

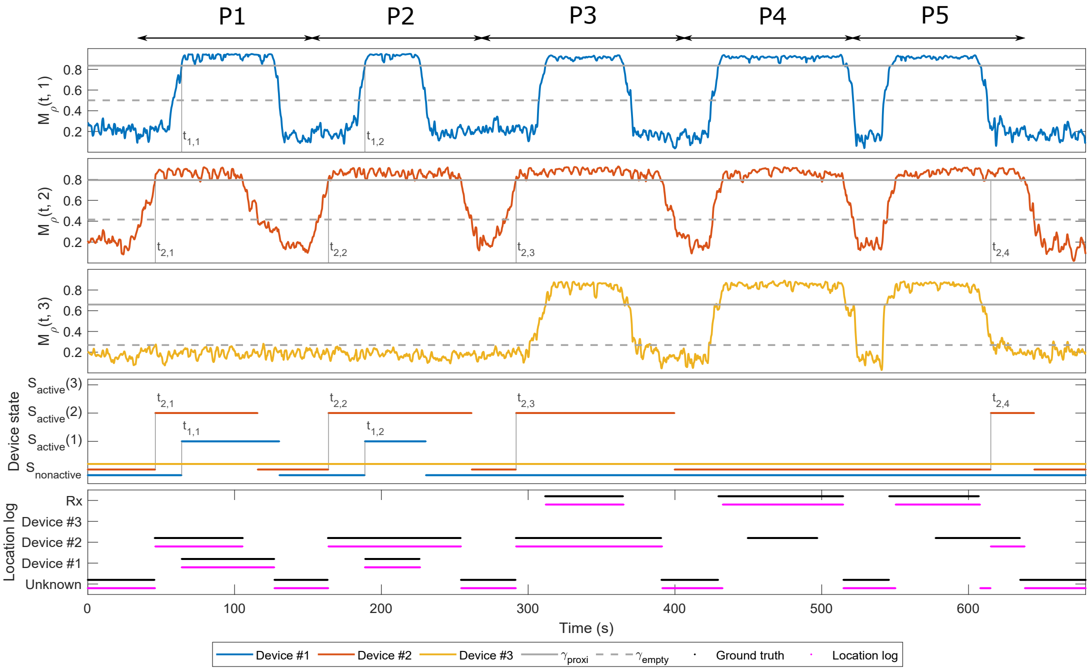


Figure 5.8: The real-time states of IoT devices and the corresponding location log under the multi-user scenario. From top to bottom the figures show: $M_\rho(t, 1)$, $M_\rho(t, 2)$, $M_\rho(t, 3)$, the states S and the location log.

device when there is an existing moving human around the Rx, the motion produced by the target user will be masked and responsive detection will not be achieved. Fig. 5.8 shows the patterns of M_ρ of different IoT devices when two users enter different areas. There are 5 instances in 5 periods from P_1 to P_5 with the descriptions listed in Table 5.4, in which P_1 and P_2 are the cases when users approach individual IoT devices, while events in P_3 to P_5 involve the periods when one of the users approaches the central Rx. As shown, when there is no motion around the Rx, the system can still detect the proximity of each IoT device timely with the delay reported in Table

Period	Description
P_1	Person B + Device #2 → Person A + Device #1 → Person B - Device #2 → Person A - Device #1
P_2	Person B + Device #2 → Person A + Device #1 → Person A - Device #1 → Person B - Device #2
P_3	Person B + Device #2 → Person A + Rx → Person A - Rx → Person B - Device #2
P_4	Person A + Rx → Person B + Device #2 → Person B - Device #2 → Person A - Rx
P_5	Person A + Rx → Person B + Device #2 → Person A - Rx → Person B - Device #2

Table 5.4: Events with two users (Person A and B) at home, where "+" stands for "approaching" and "-" means "walking away from".

Device	P_1	P_2	P_3	P_4	P_5
#1	-0.030s	-0.246s	-	-	-
#2	0.322s	0.276s	0.042s	-	-

Table 5.5: Detection delays in 2-user scenarios.

5.5. However, when large motion happens around the Rx and produces large M_ρ on all IoT devices, the increasing pattern of $M_\rho(t,k)$ on the k-th device will be masked by the existing large values of $M_\rho(t,k)$ and cannot be observed as P_4 and P_5 indicates. Nevertheless, the responsive detection can still be achieved as long as the "approaching the IoT devices" happens ahead of "approaching Rx" as the event in P_3 illustrates.

With the real-time proximity detection of $M_\rho(t,k)$ on individual devices and the prediction of motion around Rx, the location log of multi-user cases can be generated as Fig. 5.8 presents. Compared with the ground truth (black points), the location log (pink points) can indicate the active areas under multi-user scenarios reliably except the case with simultaneous motion at the Rx side, as the motion around Rx conceals the motion around individual devices.

5.4 Discussion

5.4.1 Impact on Network Communication

Since the proposed system is built on commercial networking infrastructure, it is non-trivial to investigate the impact of sensing on the communication functionalities. We measure and compare the throughput and CPU load before and after enabling the CSI estimation in the integrated system at a sounding rate of 30 Hz. The results demonstrate that the CSI capture on each individual device results in little impact with a decrease of 0.33% on throughput rate and an increase of 5.85% on CPU load, which reveals the potential of putting such co-resident sensing and communication systems into practical usage.

5.4.2 Multi-User Sensing

As the results reveal above, the proposed scheme may fail to achieve timely detection and generate an accurate location log when there is motion around the Rx under multi-user scenarios. In fact, the passive sensing of multiple targets is a well-known challenging problem, especially for commercial Wi-Fi based sensing due to the narrow bandwidth. In addition, continuous tracking of different targets is also challenging. In the future, we would like to employ a multi-way sensing structure to obtain responsive on-device proximity detection and utilize the finite state machine to match the trajectory of each target.

5.4.3 Motion Strength

As illustrated in Section 4.4.1 in Chapter 4, motion strength affects the values of M_ρ. In the proposed on-device proximity detection scheme, only the large body-level motion is considered. With various types of daily activities considered, motion strength and the distance work together on M_ρ of the home automation system components. More metrics should be involved in the future to accommodate diverse activities. For example, to handle the micro motions with weak motion strength, we may need to exploit vital sign monitoring to detect the human presence.

5.4.4 Device Grouping

In real-life scenarios, the IoT devices are placed randomly. With an increasing number of IoT devices at homes, it is very likely that multiple devices are co-located or have overlapped monitored areas. In such a case, the user-involved labeling is required in the offline training phase as the change point analysis may not automatically detect and recognize the target segment. In the future, we would like to add a device-grouping module to deal with the closely located devices and make thorough decisions.

5.5 Summary

In this chapter, we extend the single-Tx-single-Rx structure of the proximity detector in Chapter 4 to a multiple-Tx-single-Rx scheme to facilitate the home automation application. Combining the proximity features from multiple devices, the system enables timely on-device proximity detection. Given the locations of devices, the location log can be further generated over time.

Chapter 6: Universal Virtual Keyboard using A Single Millimeter Wave Radio

6.1 Challenges

mmKey overcomes multiple challenges to deliver a practical system on commodity mmWave radio. First, before keystroke recognition is possible, it is critical to design a robust motion detector that can capture the micro motions on the keyboard. To address this challenge, *mmKey* applies an anomaly detection on the differential amplitude of the CIR to sense the signal fluctuations and infer the presence of motions. Due to the high carrier frequency, signals attenuate rapidly over the propagation distance, and therefore the thresholds for keystroke detection should adapt to the distances. We achieve an adaptive z-score detector by referring to the empty CIR measured in absence of targets. We further leverage multiple antennas and different ranges to improve the robustness.

Second, since keystrokes involve not only the movements of fingers but also the shifts of palms and potentially arms, it is difficult to distinguish between the keystrokes (finger motions) and other motions. In addition, there are also irrelevant reflections from the background objects, which are mixed together with the keystroke motions. To overcome this challenge, we first devise a novel motion filter by leveraging the sensitivity of CIR phase as well as the differences in the spatial distribution of dynamic signals between keystrokes and other types of motions. Then we further employ adaptive background cancellation to extract only the dynamic reflections by tracking the CIR changes.

Last but most importantly, despite the many antennas of the mmWave device, the spatial resolution is physically limited due to the small effective aperture of the receive antenna array. Specifically, the on-chip analog beamforming provides an angular resolution of 15° on our experimental device with an array size of 1.8 cm × 1.8 cm, which is inadequate to localize and recognize a keystroke, especially when the key size is very small or simultaneous keystrokes are close to each other. To boost the spatial resolution, *mmKey* sysname performs MUltiple Signal Classification (MUSIC) algorithm on the received CIR and enables precise localization of the keystroke. In addition, by initial finger localization we only know the location of motions relative to the device. To determine the keys pressed by a user, we employ a low-effort one-time calibration stage upon initial setup, which involves as simple as three key-presses, so that the estimated locations by MUSIC can be mapped onto the corresponding keys of the keyboard.

6.2 Virtual Keyboard System Design

This section describe the implementation of *mmKey* in detail. The system aims to promptly and robustly recognize the keystrokes from the RF signals reflected from

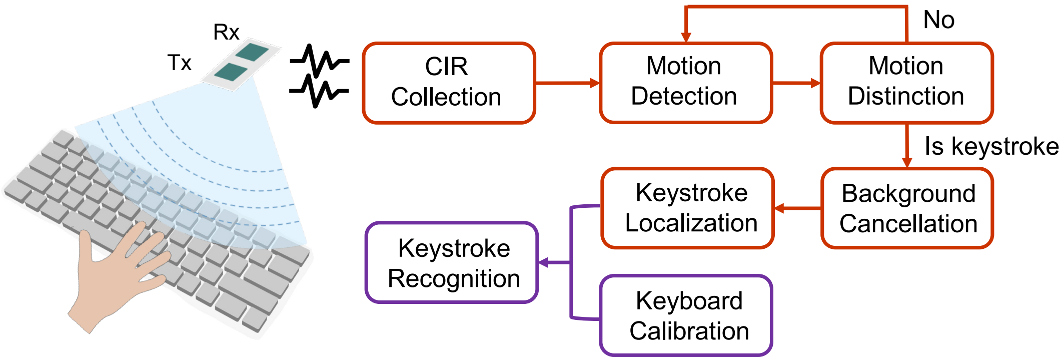


Figure 6.1: An overview of *mmKey*.

not only the fingers but also the hands and other static objects. As illustrated by Fig. 6.1, *mmKey* addresses the aforementioned challenges by the following procedures: 1) *Motion detection* that detects the presence of motions adaptively and robustly; 2) *Motion distinction* that distinguishes keystrokes by fingers from non-interested motions caused by hands, arms, and others; 3) *Adaptive background cancellation* that extracts dynamic reflections from the mixture of the superimposed reflected signals; and 4) *Keystroke localization* that localizes the keystrokes with high resolution. Note that a one-time calibration is used for key-location mapping upon initial setup, yet the effort is minimized as only three key presses.

6.2.1 Motion Detection

We capture the real-time motion by observing the fluctuations of the CIR amplitude $|h_{n,l}(t)|$. The CIR amplitude can be modeled as

$$|h_{n,l}(t)| = |h_{n,l}(t-1)| + a_{n,l}^{d}(t) + a_{n,l}^{noise}(t), \tag{6.1}$$

where $a^{d}_{n,l}(t)$ reflects the variation of amplitude contribution from dynamic signals and $a^{noise}_{n,l}(t)$ is due to measurement noise. Therefore, the differential CIR amplitude can be calculated as

$$\triangle|h_{n,l}(t)| = |h_{n,l}(t)| - |h_{n,l}(t-1)| = a^{d}_{n,l}(t) + a^{noise}_{n,l}(t). \tag{6.2}$$

When there is no motion, i.e., $a^{d}_{n,l}(t) = 0$ in Eqn. (6.1) and we have $\triangle|h_{n,l}(t)| = a^{noise}_{n,l}(t)$ in Eqn. (6.2). Without loss of generality, the amplitude change caused by measurement noise $a^{noise}_{n,l}(t)$ only can be assumed to follow a Gaussian distribution. Then by collecting a sequence of $\triangle|h_{n,l}(t)|$ in "no motion" scenario, we are able to construct a "stationary" frame denoted as $\triangle|\mathbf{h}_{ref,n,l}|$ and employ the z-score anomaly detection method to detect motion in real time by comparing the incoming differential CIR amplitudes $\triangle|h_{n,l}(t)|$ with $\triangle|\mathbf{h}_{ref,n,l}|$.

More specifically, by centering and normalizing $\triangle|h_{n,l}(t)|$ with the sample mean and standard deviation of $\triangle|\mathbf{h}_{ref,n,l}|$, we evaluate the z-score of $\triangle|h_{n,l}(t)|$ as

$$Z_{n,l}(t) = \frac{|\triangle|h_{n,l}(t)| - \hat{\mu}_{ref,n,l}|}{\hat{s}_{ref,n,l}}, \tag{6.3}$$

where $\hat{\mu}_{ref}$ and $\hat{s}_{ref}$ are the sample mean and standard deviation of $\triangle|\mathbf{h}_{ref,n,l}|$. The larger the value of $Z_{n,l}(t)$ is, the more the sample diverges from the reference frame, and the higher chance a motion occurs at time t.

Fig. 6.2(a) presents an instance of the reference frame $\triangle|\mathbf{h}_{ref,n,l}|$. z-score based anomaly detection assumes the reference sample sequence follows a Gaussian

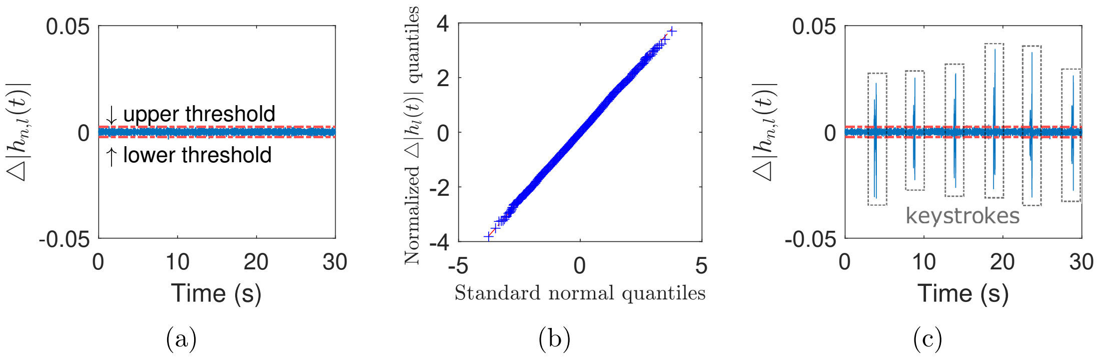


Figure 6.2: (a) Differential CIR amplitude of stationary reference frame; (b) Q-Q plot of differential CIR amplitude of reference frame; (c) Differential CIR amplitude of frame involving keystrokes.

distribution. Thus, we examine the quantile-quantile (Q-Q) plot of the normalized samples in $\triangle|\mathbf{h}_{ref,n,l}|$, as shown in Fig. 6.2(b). As seen, the distribution of normalized $\triangle|\mathbf{h}_{ref,n,l}|$ is very close to a normal distribution and satisfies the requirements of z-score computation. Fig. 6.2(c) shows $\triangle|h_{n,l}(t)|$ including six keystroke motions. Every time there is a keystroke, $\triangle|h_{n,l}(t)|$ experiences obvious fluctuations, which can be captured by evaluating $\triangle|h_{n,l}(t)|$ with a threshold calculated by Eqn. (6.3) as indicated by the red dotted line.

Motion detection aims to detect the start and end time of a keystroke and its corresponding range. Instead of relying on the z-score calculated from one single antenna, we leverage all available antennas and range taps to improve the robustness. Specifically, we apply a sliding window, with length W, to the incoming CIR stream and obtain the CIR for each window as a $N \times L \times W$ complex-valued matrix. To reduce the false alarms, we employ majority voting to the corresponding $\triangle|h_{n,l}|$ values and construct an indicator matrix $I(t)$ with dimension $N \times L \times W$, in which each element $I_{n,l}(t) = \mathbb{1}\{Z_{n,l}(t) > \upsilon\}$ where $\mathbb{1}$ is the indicator function and $\upsilon = 3$ is the commonly used value for z-score anomaly detection. Then motion is detected if the majority of the elements of $I(t)$ are ones. We further estimate the range tap of motion as the one that satisfies $\hat{l} = \arg\max_{l} \sum_{n=1}^{N} \sum_{t=t_0}^{t_0+W} Z_{n,l}(t)$. The start and end points of motion can be determined by searching the first and last anomaly time slot over consecutive windows on the $\hat{l}$-th tap.

6.2.2 Keystroke Distinction

Although the motion detector can identify which range tap is affected by motion, it cannot distinguish whether the motion is caused by a keystroke or by hand movements. Our key idea to distinguish keystrokes from hand motions is inspired by two observations: 1) Hand movements usually involve shifts of hand location while finger keystrokes do not, and 2) hand movements impact a much larger reflection area than finger motions. Accordingly, we devise two features for motion distinction: *CIR phase* and *dynamic level*.

6.2.2.1 Raw CIR Phase

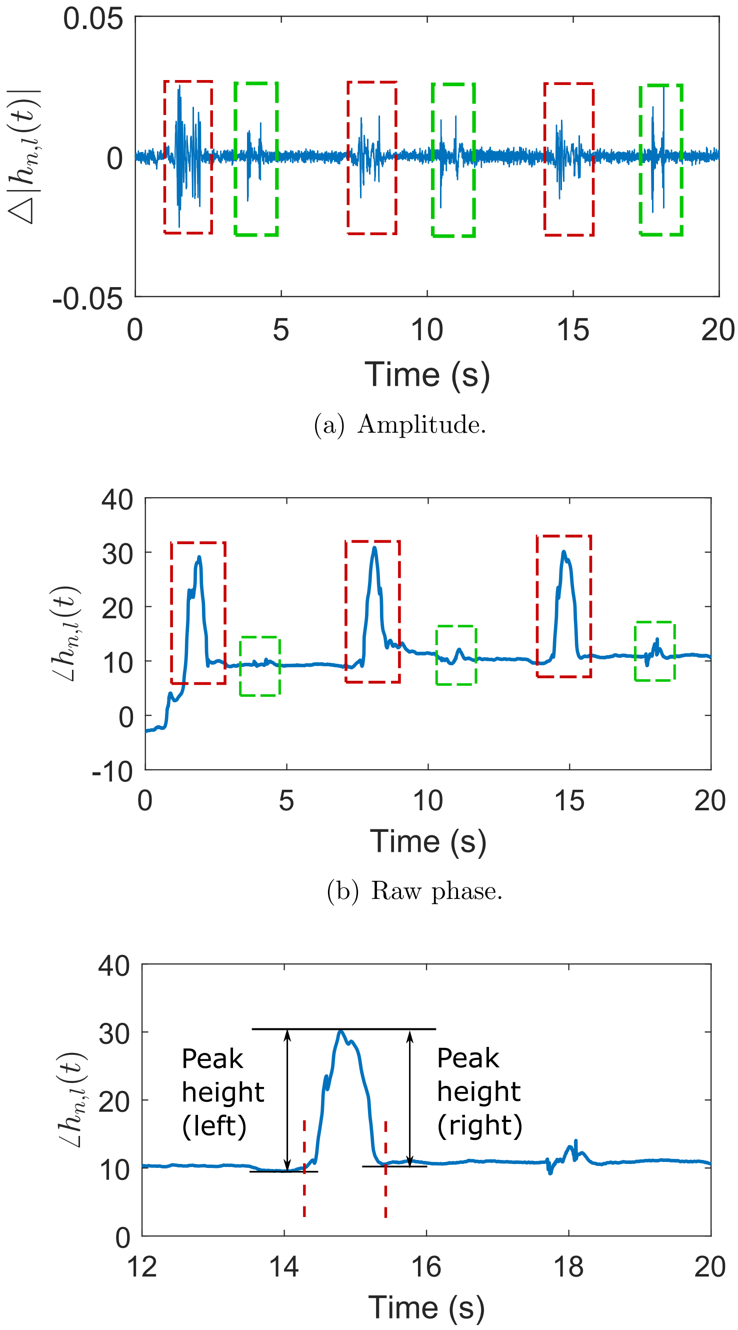
Compared with CIR amplitude, CIR phase is more sensitive to tiny location shifts of reflectors. Note that the CIR phase is already synchronized between all antennas and all samples. With the carrier frequency operated at 60.48 GHz, the wavelength is $\lambda = \frac{c}{f} = 5\,\mathrm{mm}$, meaning that a tiny shift of the reflector of 2.5 mm towards/away from the radio in the radial direction will produce a change of 2π in the CIR phase, underpinning accurate classification of large (e.g., hand) and micro (e.g., fingertips) motions.

Fig. 6.3 shows the CIR amplitude differential $\triangle|h_{n,l}(t)|$ as well as the CIR phase $\angle h_{n,l}(t)$ from a sequence of CIR involving three palm movements indicated by the red rectangles, each followed by a single finger keystroke indicated by the green rectangle. As Fig. 6.3(a) shows, based on the evident fluctuations of $\triangle|h_{n,l}(t)|$, all the six motions can be detected. However, from $\triangle|h_{n,l}(t)|$ it is hard to tell whether the motion is a finger keystroke or not, which could be more distinguishable by measuring $\angle h_{n,l}(t)$. As shown in Fig. 6.3(b), hand motions produce much higher peaks due to larger location changes than finger keystrokes. Therefore, the peak height acts as a promising feature for distinguishing these two motions. As Fig. 6.3(c) illustrates, we define the peak height as the average of the heights on both sides of a peak.

Since hand shifts impact more antennas and may cross multiple taps, we integrate the CIR phase $\angle h_{n,l}(t)$ over all antennas and three neighbour taps (corresponding to a range of about 13 cm) centered at the target tap, i.e., $[\hat{l}-1, \hat{l}, \hat{l}+1]$.

6.2.2.2 Dynamic Level

Observing that hand shifts also impact a larger reflection area than finger keystroke, we develop a novel feature of dynamic level to describe such a difference. Dynamic level is defined as the ratio of non-DC power to the total power of the



(a) Amplitude.

(b) Raw phase.

(c) Peak height of CIR raw phase.

Figure 6.3: Features for motion distinction.

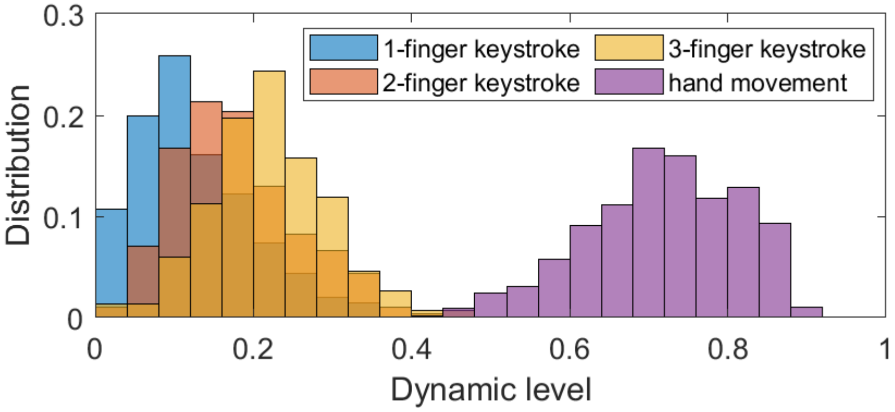


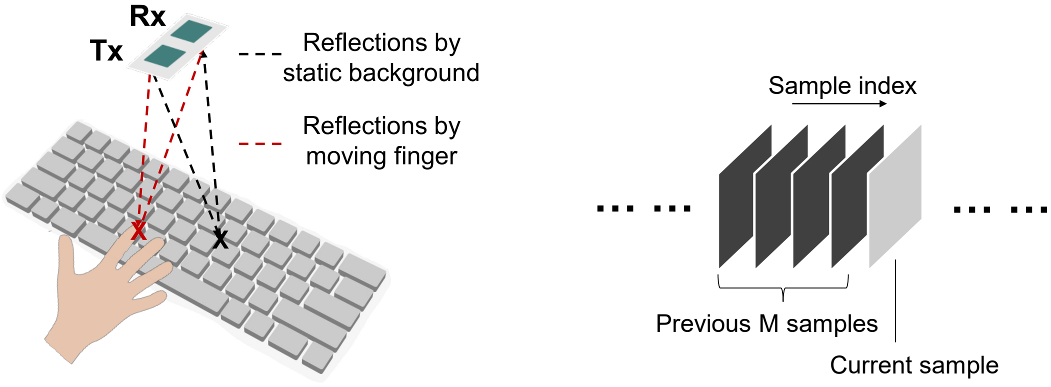
Figure 6.4: Dynamic level distribution.

CIR. Denoted as γ, it can be calculated by

$$\gamma = \frac{\sum_{l=\hat{l}-1}^{\hat{l}+1} \sum_{n=1}^{N} \sum_{f=1}^{K} |H_{l,n}(f)|^2}{\sum_{l=\hat{l}-1}^{\hat{l}+1} \sum_{n=1}^{N} \sum_{f=0}^{K} |H_{l,n}(f)|^2}, \tag{6.4}$$

where $H_{l,n}(f) = FFT(h_{l,n}(t))$. The denominator is the total power of signals reflected from both static background and dynamic hands/fingers, while the numerator is the power reflected only by the moving objects (with the DC components excluded). Therefore, dynamic level increases as the size of the reflection area increases. In other words, hand movements should yield higher dynamic levels than finger motions. Fig. 6.4 shows the distributions of the dynamic levels for one-finger keystroke, two-finger keystroke, three-finger keystroke and hand shift, respectively. As illustrated, three types of keystrokes share similar dynamic levels while hand motions experience much larger values, rendering it an effective metric to distinguish hand and finger motions.

Combining two features together, we distinct the motions with a simple two-step verification. More specifically, once the motion is detected and segmented,



(a) Mixed signals from object and background.

(b) Example of adaptive background cancellation.

Figure 6.5: Adaptive background cancellation.

the CIR frames are evaluated by thresholding both the peak height of raw CIR phase and dynamic level, and only the motions with both low peak heights and small dynamic levels are considered as finger keystrokes. In our experiments, this conservative decision rule can perfectly filter out interference motions by hands with empirical preset thresholds, but may also cause miss detection of finger keystrokes, which is measured by detection accuracy and evaluated in Section 6.3.

6.2.3 Keystroke Localization

6.2.3.1 Adaptive Background Cancellation

As Fig. 6.5(a) shows, the received signals are a mixture of the reflections from all the dynamic and static objects. Hence, we need to eliminate the background reflections and extract only the dynamic components associated with keystrokes.

For each time slot t, the CIR $h_{n,l}(t)$ can be expressed as the combination of the CIR $h_{n,l}(t-1)$ and their differential. From $t-1$ to t, the reflections from the static background are embedded in $h_{n,l}(t-1)$, while the change of CIR consists of the components due to the new dynamic reflection $h^d_{n,l}(t)$ and that due to noise $\varepsilon_{n,l}(t)$. Therefore, we can cancel the impact of background reflections by subtracting the term $h_{n,l}(t-1)$. Assuming $h^d_{n,l}(t)$ does not experience significant change for M successive samples, as illustrated in Fig. 6.5(b), $h^d_{n,l}(t)$ can be estimated as

$$\hat{h}^d_{n,l}(t) = h_{n,l}(t) - \frac{1}{M}\sum_{k=1}^{M} h_{n,l}(t-k), \tag{6.5}$$

where M denotes the number of samples used for background cancellation.

6.2.3.2 Localization with MUSIC

After extracting the dynamic signals contributed by finger keystrokes, we would like to get the 3-D coordinates of the keystroke locations, which will then translate into actual keys as detailed in the next section. The spatial resolution is greatly limited by the small effective aperture of the receive antenna array. To enhance the spatial resolution and thus accurately localize the keystrokes, *mmKey* performs digital beamforming on the received CIR based on the widely adopted MUSIC algorithm [65]. The basic idea of the MUSIC algorithm is to perform an eigen-decomposition for the covariance matrix of CIR, resulting in a signal subspace orthogonal to a noise subspace. MUSIC is typically used for reconstructing the spatial spectrum of sparse signals, which is in line with the goal of localizing less than 10 keystrokes.

We focus on the targeted $\hat{l}$-th range tap estimated in the previous modules.

Assume that there are D reflected signals impinging on the receive antenna array with different azimuths φ and elevations θ in the coordinate system shown in Fig. 2.1. Then, the CIR $\mathbf{h}$ can be formulated as

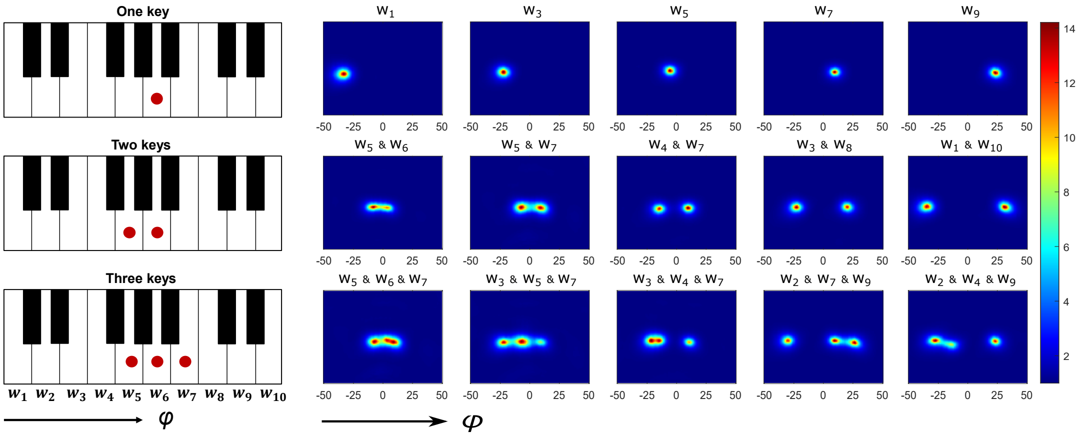
$$\mathbf{h} = \begin{bmatrix} \mathbf{s}(\theta_1, \varphi_1), \cdots, \mathbf{s}(\theta_D, \varphi_D) \end{bmatrix} \begin{bmatrix} x_1 \\ \vdots \\ x_D \end{bmatrix} + \begin{bmatrix} \epsilon_1 \\ \vdots \\ \epsilon_N \end{bmatrix}, \tag{6.6}$$

where $\mathbf{s}(\theta_i, \varphi_i)$ is the steering vector pointing to (θ_i, φ_i), corresponding to the direction of the i-th reflected signal, i.e., the normalized phase response of the antenna array for a signal coming from the direction (θ_i, φ_i). x_i denotes the complex value of the i-th reflected signal and ϵ_j stands for additive thermal noise by j-th antenna, which is assumed to be a Gaussian random variable with zero mean and independent and identically distributed (I.I.D.) for different receive antennas. A more concise matrix representation of Eqn. (6.6) can be written accordingly as $\mathbf{h} = S\mathbf{x} + \boldsymbol{\epsilon}$, where S is defined as the steering matrix. Then, the covariance of $\mathbf{h}$ can be evaluated as

$$R_h = \mathbb{E}[\hat{\mathbf{h}}\hat{\mathbf{h}}^H] = S\mathbb{E}[\hat{\mathbf{x}}\hat{\mathbf{x}}^H]S^H + \mathbb{E}[\epsilon\epsilon^H] = R_s + R_\epsilon, \tag{6.7}$$

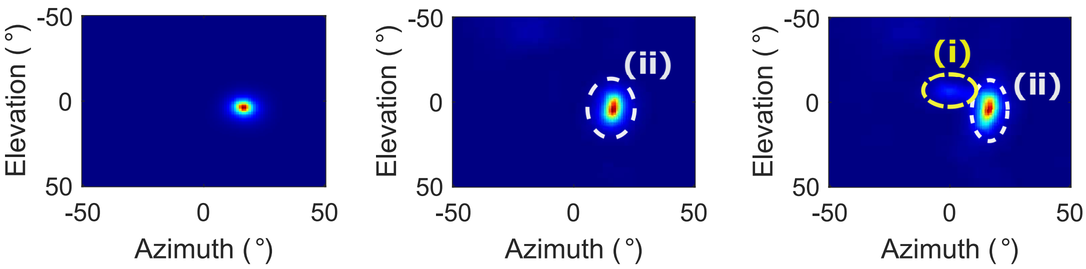
where $\hat{\mathbf{h}} = \mathbf{h} - \mathbb{E}[\mathbf{h}]$, and R_s and R_ϵ are the covariance matrices of the signal and noise components, respectively. Then, the eigen-decomposition can be represented as

$$R_h = \begin{bmatrix} U_s & U_\epsilon \end{bmatrix} \begin{bmatrix} \Lambda_s & \\ & \Lambda_\epsilon \end{bmatrix} \begin{bmatrix} U_s \\ U_\epsilon \end{bmatrix}, \tag{6.8}$$



(a) (b)

Figure 6.6: Illustration of one-key, two-key and three-key keystrokes and spatial spectrum.



(a) K = 1. (b) K = 3. (c) K = 5.

Figure 6.7: The MUSIC spectrum for $K = 1$, 3, 5, respectively.

where U_s is signal space while U_ϵ is noise space. The MUSIC spatial spectrum is expressed as

$$P(\theta, \varphi) = \left[\mathbf{s}^H(\theta, \varphi) U_\epsilon U_\epsilon^H \mathbf{s}(\theta, \varphi) \right]^{-1}. \tag{6.9}$$

Fig. 6.7(a) shows the pseudo-spectrum of a single keystroke motion. Peaks of the spatial spectrum P indicate the presence of reflected signals due to finger keystrokes, while low values of P indicate the absence of such reflections. Later we will also evaluate other spectrum estimation methods in Section 6.3, which shows MUSIC performs the best.

6.2.3.3 Location Refinement

Although MUSIC algorithm can achieve a high resolution in localizing the sources of motion, it requires a-priori knowledge of the number of sources, which is usually unknown in practice. To handle this problem, we apply a peak selection module before the target localization. A preset number of targets K is fed into the MUSIC algorithm to obtain the initial pseudo-spectrum. K peaks will be extracted from the pseudo-spectrum regardless of the actual number of targets present. A peak selection module is then designed to remove the false peaks.

Fig. 6.7 shows the pseudo-spectrum of a single keystroke motion with different preset K. As K increases, more and more outlier peaks present, including (i) *lower peaks in the background* (marked in yellow circle (i)) and (ii) *higher peaks diffused from the target peak* (marked in white circle (ii)).

To remove the false peaks and determine the number of true targets, we follow two criteria:

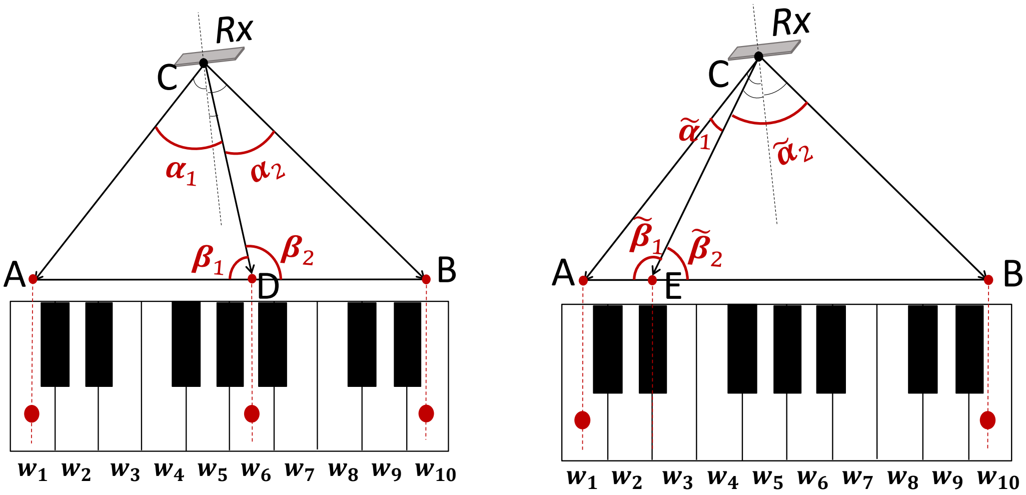
- The peaks with heights lower than a preset adaptive threshold th_1 are considered as the noise peaks and will be filtered out. To be generic, th_1 is a proportional function of the height of the highest peak, i.e., $th_1 = c \cdot max(p_1, ..., p_K)$, which is determined in the calibration phase.

- For the neighboring diffused peaks, an agglomerative hierarchical clustering is applied to merge them if the spatial angle distance between these peaks is within a threshold d_{th}. Observing that the peaks tend to expand more in the elevation direction as illustrated in Fig. 6.7bc due to the signals reflected by the upper parts of fingers, we adopt a relatively smaller weight in the elevation direction. More specifically, the distance between two peaks $(\triangle\theta, \triangle\varphi)$ is weighted with (a, b) respectively where $a < b$ to tolerate more expansion of the peak blurring in θ. d_{th} is an adaptive threshold indicating the size of clusters and is determined by the maximum spatial distance of diffused peaks in the calibration phase.

After filtering and clustering the detected peaks, the number of keystrokes is then estimated as the number of clusters, and the highest peak in each cluster is considered as the representative of the cluster, whose estimated location denoted as $(\hat{\theta}, \hat{\varphi})$ will be fed in the keystroke recognition module described in the next section.

6.2.4 Keystroke Recognition

The location of the finger keystroke $(\hat{\theta}, \hat{\varphi})$ estimated by the super-resolution MUSIC algorithm can only reflect the relative position of the keystroke with respect to the Rx. To map the keystroke location onto the keyboard and infer which key is pressed, we need the knowledge of the location of the keyboard relative to the Rx, with which a keystroke at location $(\hat{\theta}, \hat{\varphi})$ can be translated to a specific key. We employ a simply calibration step to obtain such mapping relationships, which only needs to be done once upon the initial setup of a keyboard. As *mmKey* can be compatible with multiple types of keyboards, such as piano keyboard and computer keyboard, we start with the 1-D case using the white keys of a piano keyboard as


Rx
C
α_1
α_2
β_1
β_2
A
D
B
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10}
Rx
C
$\tilde{\alpha}_1$
$\tilde{\alpha}_2$
$\tilde{\beta}_1$
$\tilde{\beta}_2$
A
E
B
w_1 w_2 w_3 w_4 w_5 w_6 w_7 w_8 w_9 w_{10}

(a) Model for piano keyboard.

(b) 1-D model during detection.

Figure 6.8: Geometrical models for keyboard calibration. The user only needs to press three known keys, as indicated by the red dots in (a).

an example, and extend it to the general 2-D case for the computer keyboard and phone keypad later.

6.2.4.1 1-D Case

To complete the keyboard calibration with the least effort, the user can randomly pick and press three keys. As seen in Fig. 6.8(a), assuming key w_1, w_6 and w_{10} are pressed during the calibration and the corresponding estimated azimuths by the MUSIC algorithm are represented as $\hat{\varphi}_1$, $\hat{\varphi}_6$ and $\hat{\varphi}_{10}$, we have $\alpha_1 = \hat{\varphi}_6 - \hat{\varphi}_1$ and $\alpha_2 = \hat{\varphi}_{10} - \hat{\varphi}_6$. According to the law of sines, we have

$$\begin{aligned} &\frac{|AD|}{\sin\alpha_1} = \frac{|AC|}{\sin\beta_1}, \text{ in } \triangle ACD\ ; \\ &\frac{|BD|}{\sin\alpha_2} = \frac{|BC|}{\sin\beta_2}, \text{ in } \triangle BCD\ ; \\ &\sin\beta_1 = \sin\beta_2, \end{aligned} \tag{6.10}$$

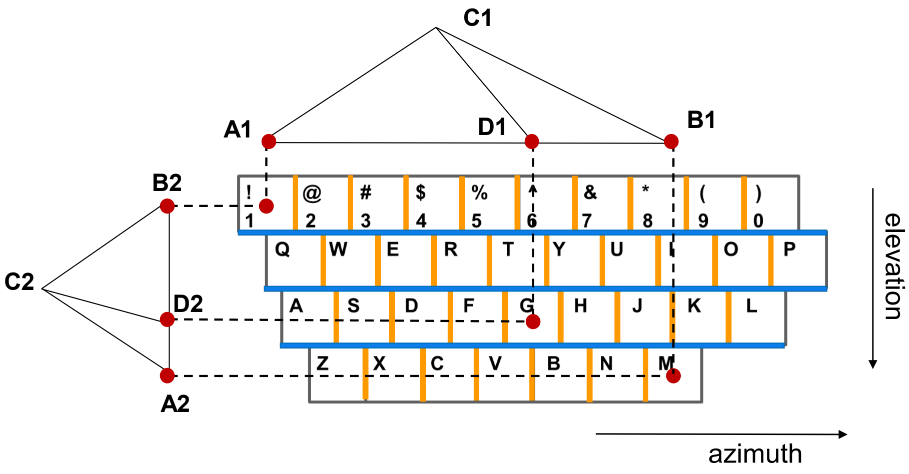


Figure 6.9: Generalized 2-D case for keyboard calibration.

where β_1 and β_2 are two unknown angles belonging to two adjacent triangles and forming a straight angle. Denoting the ratio of $|AC|$ to $|BC|$ as η, from Eqn. (6.10) we have

$$\eta = \frac{|AC|}{|BC|} = \frac{|AD|}{|BD|}\frac{\sin\alpha_2}{\sin\alpha_1}. \tag{6.11}$$

Since the ratio $\frac{|AD|}{|BD|}$ is already known as $\frac{5}{4}$ in this example. Assuming all of the keystrokes occur at the center of the key, we can derive the value of η. Further, the azimuth boundary of every two adjacent keys can also be derived. For example, as indicated in Fig. 6.8(b), to calculate the boundary between key w_2 and w_3, we apply the law of sines again as

$$\begin{aligned}
&\frac{|AE|}{\sin\widetilde{\alpha}_1} = \frac{|AC|}{\sin\widetilde{\beta}_1}, \text{ in } \triangle ACE \ ; \\
&\frac{|BE|}{\sin\widetilde{\alpha}_2} = \frac{|BC|}{\sin\widetilde{\beta}_2}, \text{ in } \triangle BCE \ ; \\
&\sin\widetilde{\beta}_1 = \sin\widetilde{\beta}_2,
\end{aligned} \tag{6.12}$$

where $\widetilde{\alpha}_1$, $\widetilde{\alpha}_2$, $\widetilde{\beta}_1$ and $\widetilde{\beta}_2$ are the angles corresponding to those in Fig. 6.8(a). Then, we get

$$\begin{aligned} &\frac{\sin \widetilde{\alpha}_1}{\sin \widetilde{\alpha}_2} = \frac{|BC|}{|AC|}\frac{|AE|}{|BE|} = \frac{1}{\eta}\frac{|AE|}{|BE|}, \\ &\widetilde{\alpha}_1 + \widetilde{\alpha}_2 = \alpha_1 + \alpha_2. \end{aligned} \tag{6.13}$$

Based on Eqn. (6.13), the exact values of $\widetilde{\alpha}_1$ and $\widetilde{\alpha}_2$ can be obtained since the ratio $\frac{|AE|}{|BE|}$ is known as $\frac{1}{5}$. Similarly, the boundaries between the other adjacent keys such as (w_1, w_2), ... , (w_9, w_{10}) can be derived. By subtracting the absolute azimuth of w_1, the boundary azimuths can be calculated for keystroke recognition.

6.2.4.2 Generalized 2-D Case

The geometrical model of the 1-D case can be easily extended to 2-D, where both the elevation and azimuth angles are used for keystroke recognition. As illustrated in Fig. 6.9, three keys “1”, “G” and “M” are pressed for calibration. In the horizontal azimuth direction, we have $\triangle A_1B_1C_1$, from which we can derive all the azimuth boundaries of keys (orange lines), while in the vertical elevation dimension we have $\triangle A_2B_2C_2$ and use it to calculate the elevation boundaries (blue lines). Here C_1 and C_2 denote the same location of the device, viewing from the azimuth and elevation dimensions, respectively.

Given the values of the boundary azimuth and elevation angles of each key, real-time keystrokes can be easily recognized by mapping the estimated keystroke position to the target key on the keyboard with a known layout.

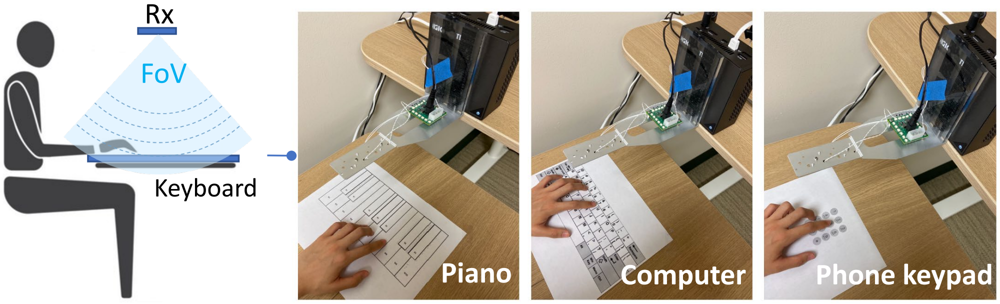


Figure 6.10: Setup examples for three difference keyboards.

6.3 Experimental Evaluation

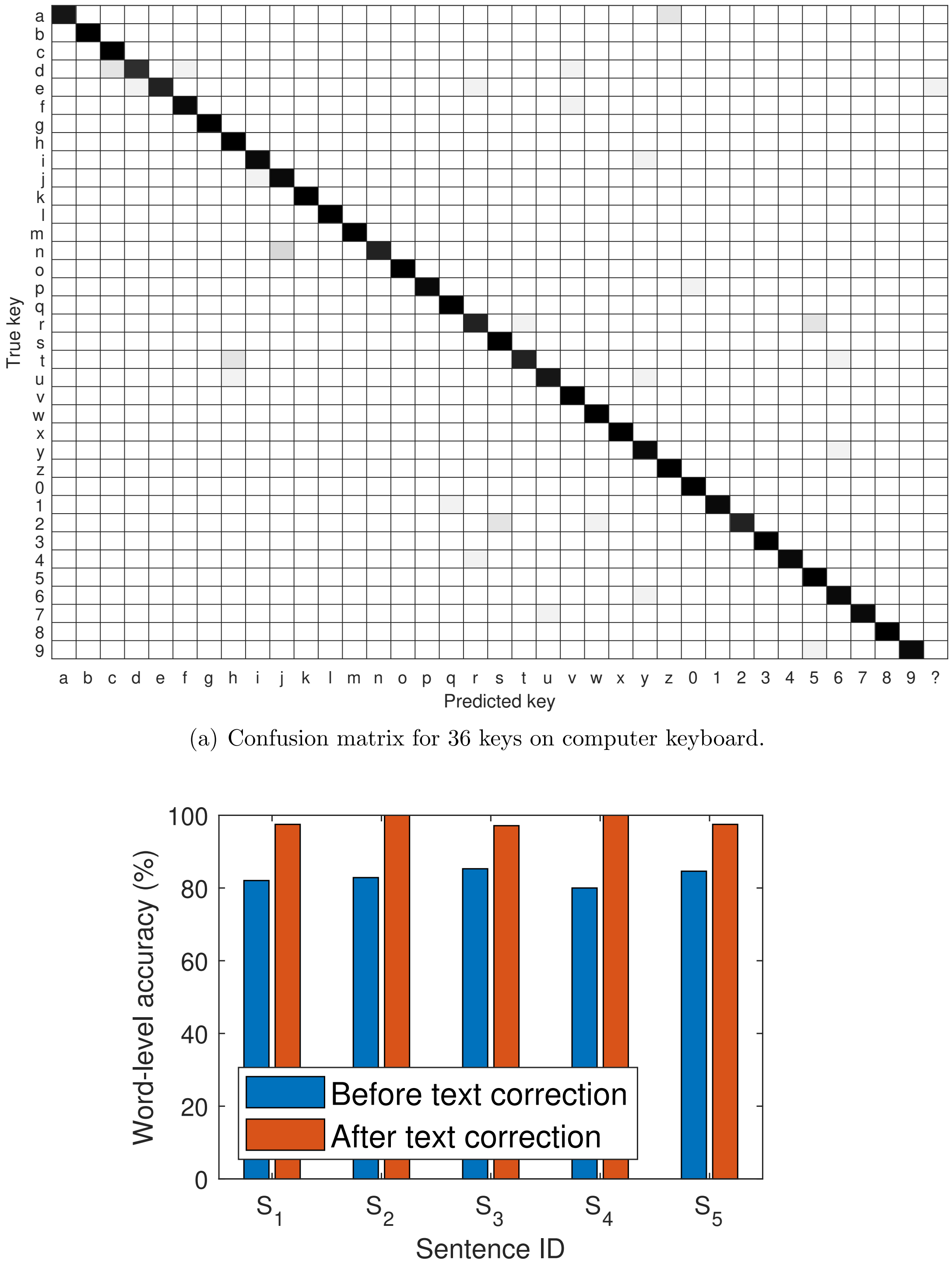
We prototype *mmKey* and conduct real-world experiments using a Qualcomm sponsored testbed, which reuses a 802.11ad/ay chipset as a radar-like platform. The default setup is presented in Fig. 6.10, where the device is put down to cover a flat surface that upholds a printed virtual keyboard. We consider different types of keyboards, including QWERTY computer keyboard, piano keyboard, and smartphone keypad. For each keyboard, we print the layout on a paper, maintaining the same physical size such that users would keep the most familiar typing feeling as on a real keyboard as Fig. 6.10 shows. Note that with a simple calibration mechanism and the high directionality of mmWave, *mmKey* can easily adapt to any keyboards including user-customized layouts, as long as the geometric arrangement is known. By default, we set the distance between the keyboard and the device to be around 20 cm such that the keyboard will be confidently within the field-of-view (FoV) of our device, which is 100° for both azimuth and elevation directions. The default sampling rate is $f_s = \frac{1}{T_b} = 100\,\text{Hz}$, where T_b is the burst duration as shown in Fig. 2.2. The selection of different parameters is further studied in Section 6.3.4.

We conduct experiments at different locations in both office and home environments, with 10 volunteers involved, including 4 females and 6 males aging from 23 to 32. To obtain the ground truth, we perform the experiments in two different ways: 1) For each keyboard, we generate a random key list covering each key once, and the participants are asked to press the virtual keys by following the list. Each participant will repeat multiple times for each key list. 2) The user is asked to type following a sequence of words/sentences or digits/music scores. Note that participants press the keys in a nearly natural way. Therefore, the keystroke speed is not controlled and it may vary over time for different users. During experiments, we collect CIR series from the testbed when a user is typing and send the data through an Ethernet cable to a computer for processing in MATLAB.

We mainly use three metrics for evaluation. We use detection accuracy (DA) and recognition accuracy (RA) to quantify how *mmKey* correctly detects the keystrokes and how it localizes and recognizes them, respectively. Based on DA and RA, we calculate the overall accuracy (OA) as OA = DA × RA. DA and RA are defined as follows:

$$\begin{aligned} \text{DA} &= \frac{\text{\# of detected keystrokes}}{\text{\# of total keystrokes}}; \\ \text{RA} &= \frac{\text{\# of recognized keystrokes}}{\text{\# of detected keystrokes}}. \end{aligned} \tag{6.14}$$

Below, we first evaluate the overall performance of three different types of keyboards and then report the parameter study in Section 6.3.4.



(a) Confusion matrix for 36 keys on computer keyboard.

(b) Word recovery accuracy in sentence input testing.

Figure 6.11: Performance on virtual computer keyboard.

6.3.1 Virtual Computer Keyboard

6.3.1.1 Performance on Individual Keys

We first investigate the performance of *mmKey* for a virtual computer keyboard. The printed standard alphanumeric keyboard has a common QWERTY-based layout with the distance between neighboring keys 19 mm. We involve the keys of letter and digit in the experiments and select “1”, “G” and “M” as the landmark keys for calibration. The OA confusion matrix for recognizing 36 keys (26 letters plus 10 digits) on a virtual computer keyboard is shown in Fig. 6.11(a). As we can see, *mmKey* achieves remarkable keystroke recognition with an average OA of 95.42% for a computer keyboard. As illustrated, there exist some detection errors. Some samples of certain keys are recognized as the neighboring keys, especially the one below the real key. This is because there exist reflections from the knuckles leading to the estimation error in the elevation direction. In real applications when users are typing typical texts, we believe these errors can be easily recovered by the mature spell check techniques, as demonstrated in the next section.

6.3.1.2 Word Recovery

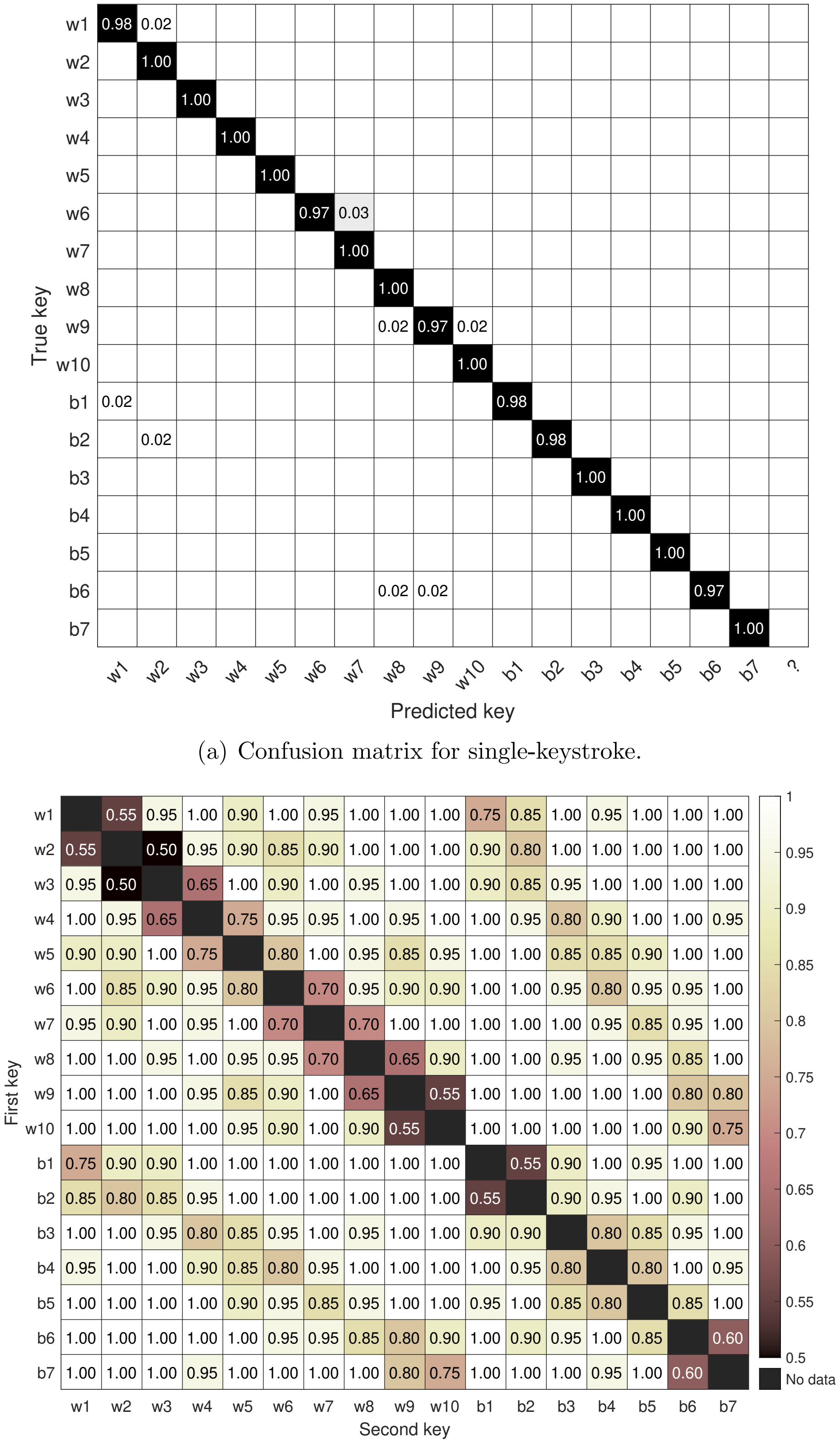
We further explore the capabilities of *mmKey* on recovering the input sentences and evaluate the accuracy in word level. Adopting the same methods in WiKey [12] to collect sentences samples, we ask the user to type each of the following sentences 5 times on the printed computer keyboard: S_1 = “the quick brown fox jumps over the lazy dog”, S_2 = “nobody knew why the candles blew out”, S_3 = “the autumn leaves look like golden snow”, S_4 = “nothing is as profound as the imagination”, and S_5 = “my small pet mouse escaped from his cage”.

We first run *mmKey* on the CIR data and obtain the direct outputs, i.e., a sequence of recognized keys. Then we feed the outputs into Grammarly[1] for correction, which is a popular commercial English writing tool. Here we calculate the word-level accuracy (WA) by $\mathrm{WA} = \frac{\#\text{ of correct words}}{\#\text{ of total words}}$ and illustrate the results in Fig. 6.11(b). It is as expected the WA on the direct outputs of *mmKey*, which is about 80%, is not as high as its OA since a single mis-recognized letter will lead to a wrongly recognized word. With the help of spell check/text correction, the word-level mis-recognition can be easily corrected with a considerable accuracy greater than 97%. With the high accuracy, *mmKey* could promise a ubiquitous virtual keyboard for mobile and portable usage everywhere in practice.

6.3.2 Virtual Piano Keyboard

In this part, we report the overall performance of *mmKey* in the recognition of keystroke on a piano keyboard. To adapt to the system capacity of limited FoV, we employ a segment of piano keyboard consisting of ten white keys and seven black keys in between. We denote the white keys and black keys from left to right as w_1, w_2, ..., w_{10} and b_1, b_2, ..., b_7, respectively. Three keys w_1, b_4 and w_{10} are used for calibration. Note that as piano only have two rows of keys, we use the weighted average of elevation angles of w_1, b_4, w_{10} as the elevation boundary, i.e.,

[1] https://www.grammarly.com



(a) Confusion matrix for single-keystroke.

(b) Accuracy when pressing two concurrent keys.

Figure 6.12: Performance of virtual piano keyboard.

$\frac{1}{2}(\frac{\theta_{w_1}+\theta_{w_{10}}}{2}+\theta_{b_4})$. Different from computer keyboards, users may press multiple keys concurrently when playing the piano. Thus we conduct experiments with two scenarios: *single keystroke* and *simultaneous multiple keystrokes*.

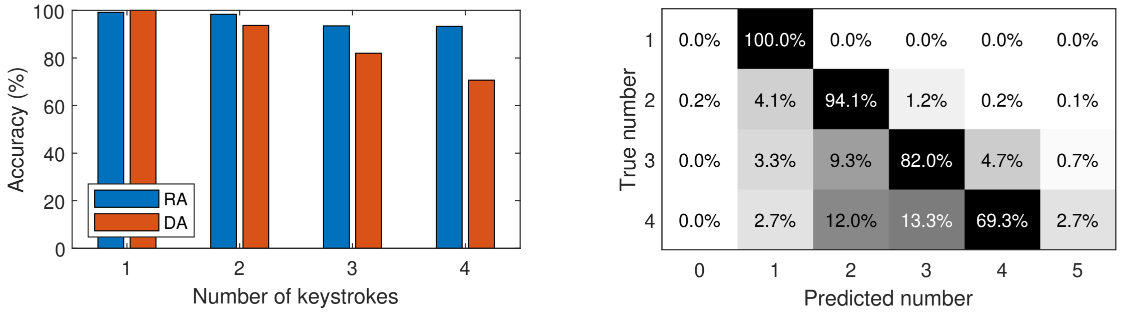
6.3.2.1 Single-Key Keystroke

For single-key keystroke case, the experimenter is asked to press each key 60 times. The pressing order follows a random sequence generated from MATLAB as stated before.

Accounting for both the white keys and black keys, Fig. 6.12(a) shows the confusion matrix for single-key keystrokes, where "?" means the miss of detection. As seen, *mmKey* achieves a high OA of 99.12%.

6.3.2.2 Simultaneous Multiple Keystrokes

Users may need to press multiple keys simultaneously to play the piano. We first look at the two-key case. The experimenter is asked to press two keys at the same time, and each combination of keys is repeated 20 times. The instances of two-key keystroke spectrum can be found in Fig. 6.6 while the OA for all combinations can be seen in Fig. 6.12(b). As shown, *mmKey* recognizes the keystrokes accurately when two keys are located far enough. However, when two pressed keys get closer especially for the adjacent keys, the accuracy may decrease due to the co-located fingers. The overall accuracy for double-key keystroke recognition is 92.54% for all cases, and the accuracy becomes 96.93% for non-adjacent keys. Also, observing the



	0	1	2	3	4	5
1	0.0%	100.0%	0.0%	0.0%	0.0%	0.0%
2	0.2%	4.1%	94.1%	1.2%	0.2%	0.1%
3	0.0%	3.3%	9.3%	82.0%	4.7%	0.7%
4	0.0%	2.7%	12.0%	13.3%	69.3%	2.7%

(a) Accuracy vs number of keystrokes.

(b) Confusion matrix for detection.

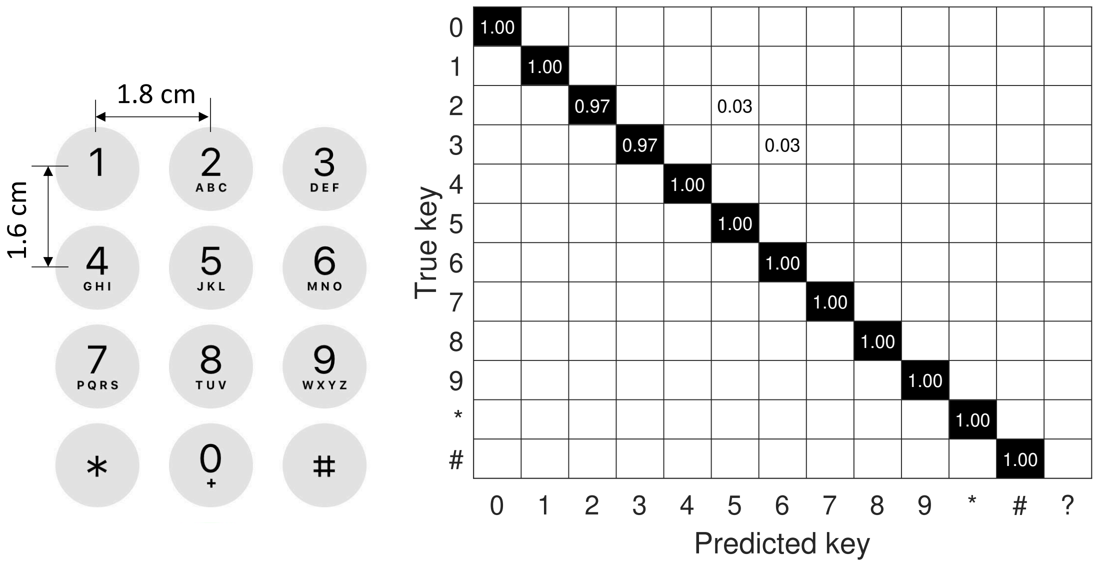
Figure 6.13: Multi-keystroke accuracy.

accuracy along the diagonal, we can find that the OA decreases near the edge of the keyboard due to the effects of inter-finger blockage at the edge locations.

Now we extend to the cases of > 2 concurrent keys by investigating the OA with respect to the number of concurrent keystrokes. The results show that the OA decreases to 76.67% when there are three keys being pressed and further decreases to 65.94% for four keys. We further examine the detection accuracy, as illustrated in Fig. 6.13(b). As seen, more keystrokes will lead to more miss detection due to the blockage between multiple fingers, but do not affect much the recognition accuracy as illustrated by Fig. 6.13(a). Once a keystroke is detected, *mmKey* can recognize it accurately.

6.3.3 Virtual Phone Keypad

We also test *mmKey* against a virtual phone keypad. Phone keypad is a widely-used keyboard in daily life due to the proliferation of mobile devices. As shown in Fig. 6.14(a), the applied keypad has 12 keys including digits 0 to 9 and two special characters "*" and "#". The keypad is printed on a paper with the same size as



(a) Phone keypad layout.

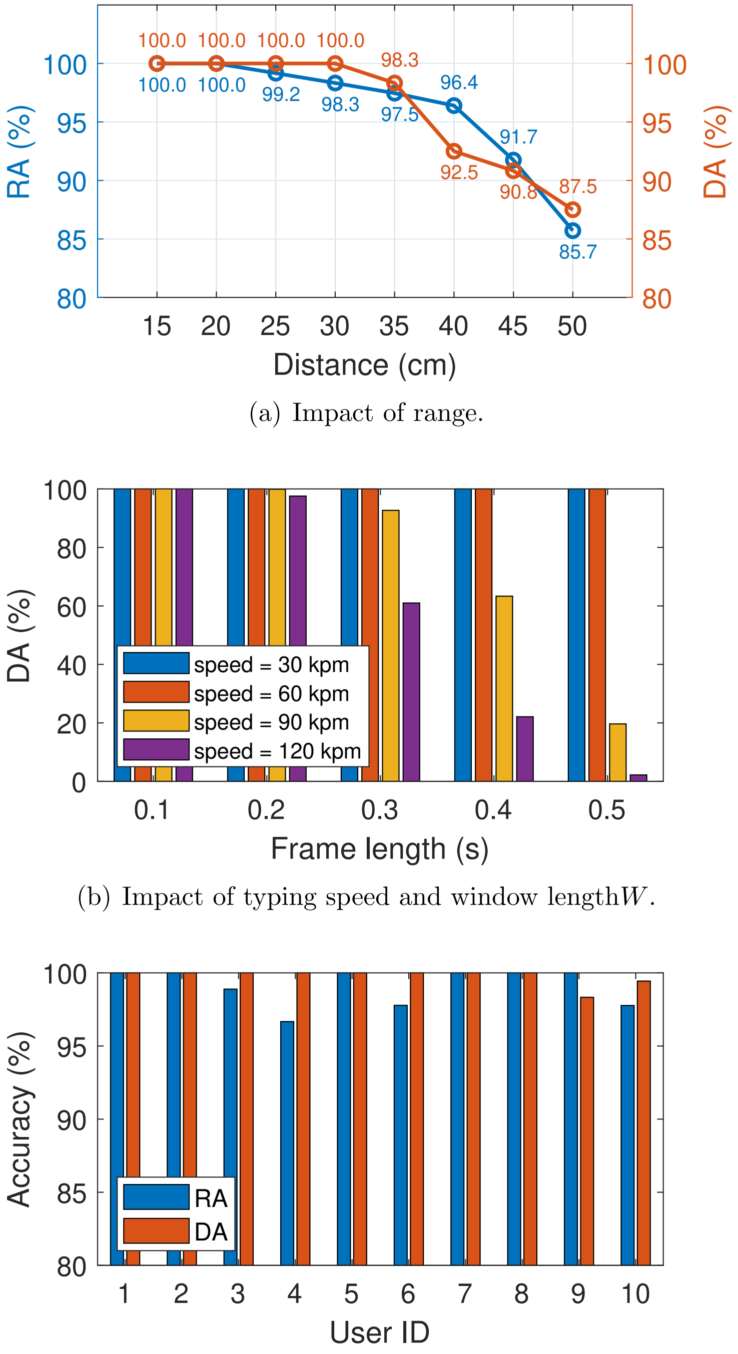
(b) Confusion matrix for phone keypad.

Figure 6.14: Performance on virtual phone keypad.

the real phone keypad with an inter-key separation of 1.8 cm in horizontal direction and 1.6 cm in vertical dimension. In the keyboard calibration phase, key "1", "8" and "#" are pressed to estimate the boundaries. In the real-time recognition phase, the user is asked to press different keys following the random-order list. There are total 40 samples for each key. The confusion matrix of OA is shown in Fig. 6.14(b). As we can see, although the keys are close to each other (< 2 cm), the keystrokes can be recognized accurately and reliably with an OA 98.33%.

6.3.4 Parameter Study

In this section, we benchmark *mmKey* by studying the impacts of different parameters. Without loss of generality, we evaluate with the piano keyboard unless otherwise specified.



(a) Impact of range.

(b) Impact of typing speed and window length$W$.

(c) Impact of user diversity.

Figure 6.15: Impact of factors on *mmKey* performance: (a) range, (b) speed and window length and (c) user heterogeneity.

6.3.4.1 Impact of Range

The performance of *mmKey* may vary over the distance from the target to the device. To investigate the impact, we perform experiments with eight different keyboard-radio distances, ranging from 15 cm to 50 cm with an increment of 5 cm. As depicted in Fig. 6.15(a), the increase of distance leads to a degradation in both DA and RA as expected. This is because both the spatial resolution and the reflection strengths decrease over range. In other words, the keys become relatively narrower from the view of the radar and the reflected signals become weaker at larger distances, which lead to detection and recognition errors.

6.3.4.2 Impact of Keystroke Speed and Window Length

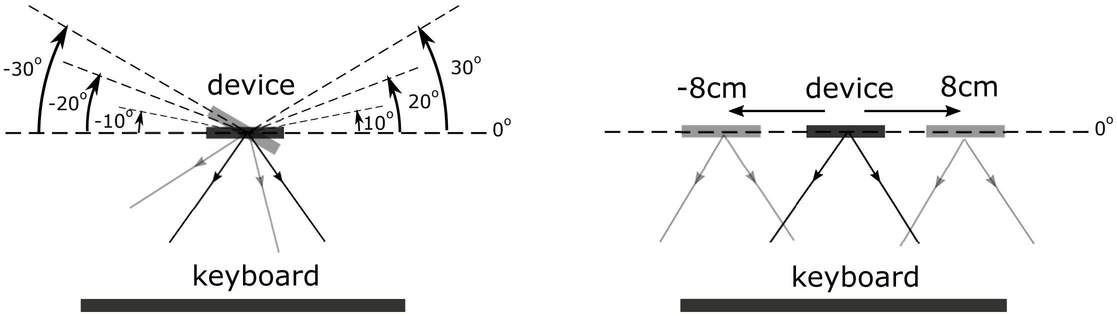
Keystroke speed is another important property that may raise users' concern. Since the motion detection and segmentation are applied with the assumption that finger motions do not overlap with each other, there are two factors that may impact the DA of *mmKey*: *keystroke speed* and *length of sliding window* W. We benchmark these two factors by performing experiments with different typing speeds, quantified by keystroke per minute (kpm). The DA of different combinations of speed and W are presented in Fig. 6.15(b). As seen, *mmKey* achieves consistently high accuracy for 30 kpm and 60 kpm, regardless of the W values. For faster typing speeds, the detection rate remains high if a short window is used, but it decreases quickly with larger W since two adjacent keystrokes are easily mistaken as a merged single keystroke. For performance evaluation, the default W is set to be 0.1s, which can handle various typing speeds.

6.3.4.3 Impact of User Heterogeneity

In this part, we study the impact of user heterogeneity. During the experiments, We enroll 10 participants labeled as “User 1” to “User 10”. We use a piano keyboard for evaluation without loss of generality. Among all the 10 participants, only one (User 9) is familiar with the piano keyboard and good at playing the piano, and two of them (User 2 and 3) have some basic knowledge of the piano. Also, during the data collection, there is no restriction on using one or two hands. Two of them (User 1 and 9) put both of their hands on the keyboard and type different keys using different hands while others use one hand. Therefore, these 10 users provide a reasonable level of diversity in terms of different typing behaviors. We calibrate the keyboard once at the beginning, and then apply it to all the participants. The results in Fig. 6.15(c) show a near 100% accuracy for both keystroke detection and recognition, implying that *mmKey* can support diverse users with only one-time calibration pre-processing.

6.3.4.4 Impact of Device Placement

As the typical experimental setup presented in Fig. 6.10, the device is put “parallel” to the keyboard and the Rx locates at almost the middle of the keyboard, which looks symmetric. However, in practice, the placement of device is not always perfect, but with potential distortions due to rotation and translation. Therefore,



(a) Rotation of device.

(b) Translation of device.

Figure 6.16: Setup of various placements with distortions (front view).

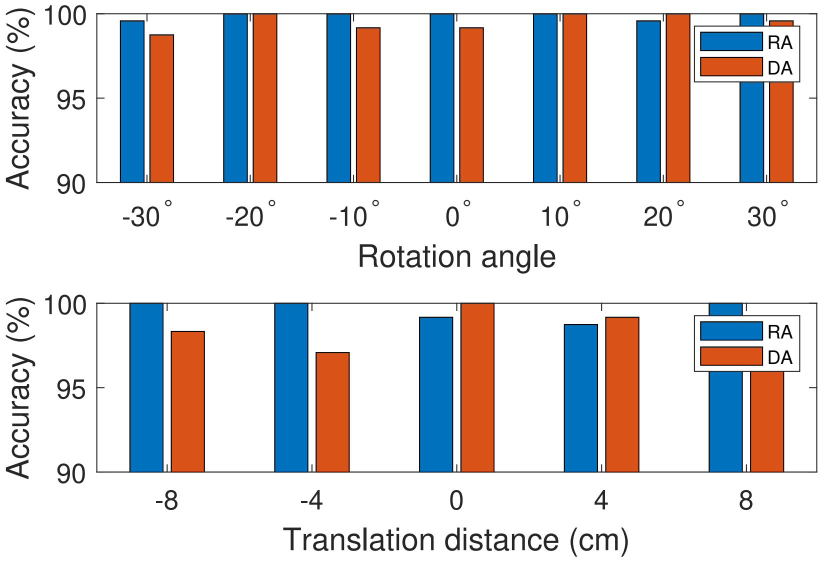


Figure 6.17: Performance for various placements with distortions.

we study the impacts by placing the device with different orientations and offsets, as detailed below.

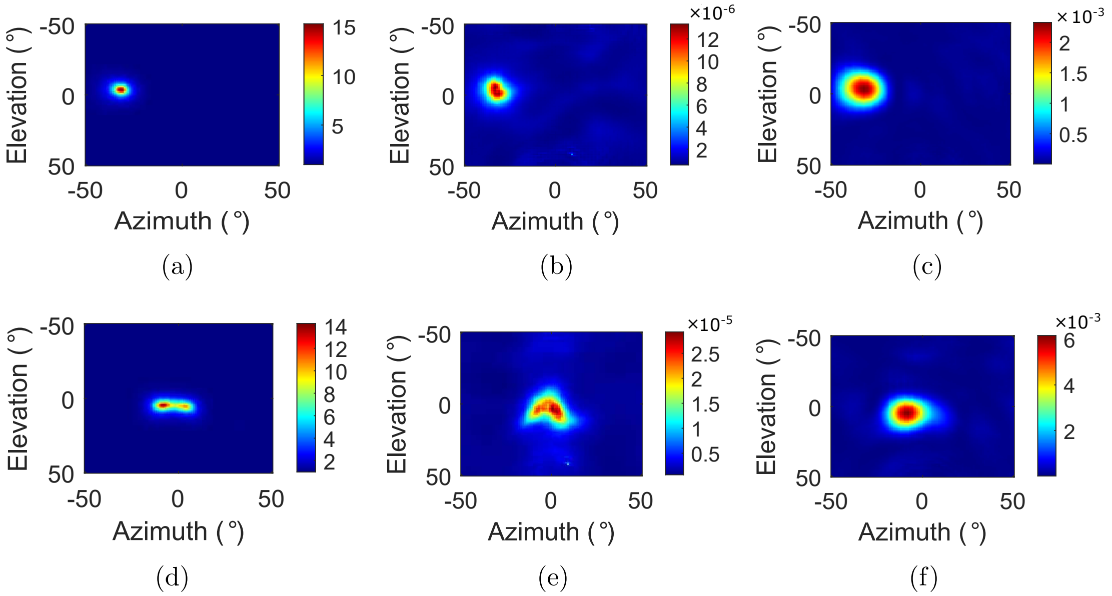
Rotation. In this experiment, the device is not perfectly parallel to the keyboard but deviates with some angle as Fig. 6.16(a) presents. Here we test deviation angles ranging from $-30°$ to $30°$ with a step of $10°$, where "-" represents the clockwise rotation and "+" is anticlockwise from the front view of users. The results are shown in Fig. 6.17, which shows remarkable robustness to the orientation distortions with consistently high DA and RA retained.

Translation. For the translation case, the device is not aligned with the center of the keyboard but is moved by a certain distance. As Fig. 6.16(b) illustrates, we shift the device from the center of the keyboard by different distances. Specifically, we tested at -8 cm, -4 cm, 0 cm, 4 cm, 8 cm, where "-" and "+" represent the left and right directions from the front view, respectively. As depicted in Fig. 6.17, the results show no significant difference among the tested distances in both DA and RA, indicating that *mmKey* can adapt to various translation placements.

6.3.4.5 Comparing Spatial Spectrum Estimators

We employ MUSIC to achieve super resolution in space on our device. In this section, we compare *mmKey* with existing beamforming techniques, including conventional beamforming (CBF) and well-known minimum variance distortionless response (MVDR) beamforming (a.k.a. Capon beamforming).

Fig. 6.18a-c are the spectrum of pressing a single key w_1 generated by the three spatial spectrum estimators. As we can visually compare, for single-finger keystroke case, the MUSIC algorithm can produce the finest spectrum due to its super-resolution property. For the double-finger keystroke case, MUSIC also performs much better than the other two. Fig. 6.18d-f illustrate the spectrum of keystrokes of two adjacent keys w_5 and w_6. As we can see, only MUSIC algorithm can detect two neighboring sources while it becomes hard to distinguish the multiple keystrokes by MVDR or CBF. MVDR is slightly better than CBF in this case, but



(a) (b) (c)

(d) (e) (f)

Figure 6.18: Examples of obtained spectrum by three different spatial spectrum estimators: MUSIC, MVDR and CBF. (a)-(c): Single-key case by MUSIC, MVDR and CBF, respectively. (d)-(f): Double-key case by MUSIC, MVDR and CBF, respectively.

is not as focusing as MUSIC.

To quantitatively compare their performance, we apply three estimators on the same dataset in different scenarios, respectively. Fig. 6.19 shows the OA for each estimator under each setting. It can be observed that for all the different keyboards, *mmKey* based on MUSIC achieves the highest accuracy. The OA drops with MVDR and CBF due to their limited resolution, especially for the double-key case, where the MUSIC-based approach outperforms MVDR and CBF significantly with an over 90% OA.

6.3.5 Subjective Evaluation with User Study

Finally, we carry out a user study on all the participants for their feedback on user experience. We collect subjective measurements from the ten volunteers

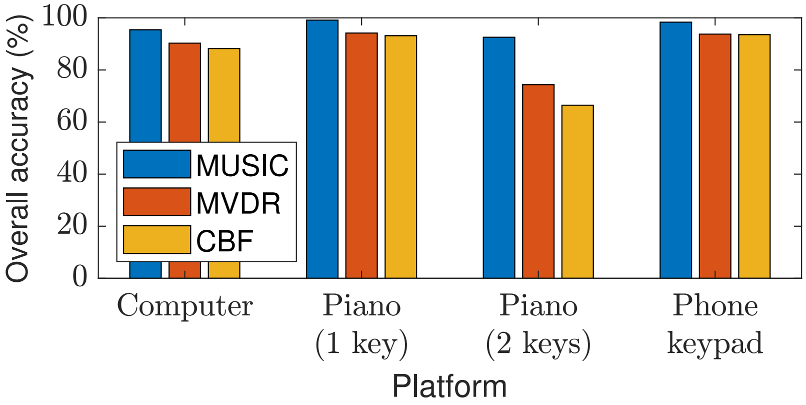


Figure 6.19: Overall accuracy on different platforms using three spatial spectrum estimators: MUSIC, MVDR and CBF.

through an online questionnaire. The users are asked about the perceived experience of using *mmKey* as an input tool, including the setup complexity, ease of use, coverage, portability and so on. The responses from real users are summarized in Fig. 6.20. All responses are scaled from 1 to 5 where 5 is the most positive rating. As we see, *mmKey* is rated positively with the average responses to most questions greater than 4.0. We note that each volunteer is also asked to give an overall rating on how he/she likes *mmKey*. On average, ten volunteers give a positively 4.0+ average rating and two of them express their willingness to experience it in real applications.

6.4 Discussion

6.4.1 Typing Speed

The average typing speed on a physical computer keyboard is about 37∼40 words per minute, which translates to about 185∼200 characters (keys) per minute.

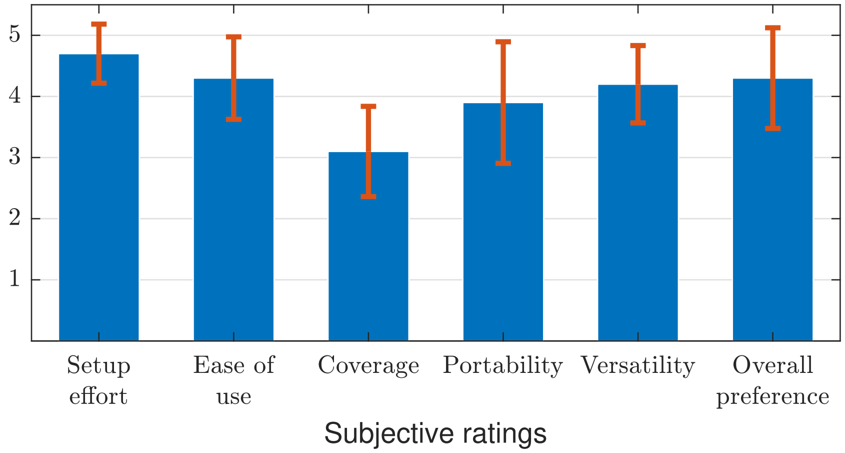


Figure 6.20: Means and standard deviations of the subjective ratings, all on 5-level scales where 5 is the most positive rating.

mmKey supports a reasonably good speed of 120 kpm, as people generally type more slowly on a virtual printed keyboard due to the lack of keypress feedback. However, the performance of *mmKey* may deteriorate for fast typing (>120 kpm), during which the finger motion and hand motion overlap with each other. It is worthwhile to study the segmentation of the CIR time series and explore new features in order to support reliable recognition of high-speed keystrokes.

6.4.2 Detection Range

While *mmKey* can support accurate keystroke recognition ($> 90\%$) at a range up to 45 cm, which is large enough to cover a computer keyboard given the common FoV of mmWave radios, the detection range still needs to be improved so as to support the implementation of a full piano keyboard. As the device-keyboard distance increases, the reflected signals become weaker and the keys become relatively smaller from the view of the radar, which makes the keystroke detection and recognition harder. We keep it as future work to investigate the antenna diversity for a better resolution and thus larger ranges.

6.4.3 Keyboard Calibration

mmKey needs minimal calibration, i.e., only three key-strokes, to associate the key locations relative to the device with the actual keys. Except for that, *mmKey* does not need any other training, making it deployable anywhere for a ubiquitous virtual keyboard. The calibration only needs to be done once for a specific setup. However, it is recommended not to change the relative location between the device and the printed keyboard; Otherwise, a re-calibration would be needed to associate the new mapping.

6.4.4 Cost and Device Readiness

mmKey is implemented on a mmWave platform sponsored by Qualcomm. *mmKey* itself does not introduce any extra hardware cost. The Qualcomm platform does need some modifications and is admittedly bulky for its current form. Nevertheless, the platform only uses a single commodity 60 GHz WiFi chipset with an additional antenna array and thus would be fairly low-cost and tiny once mass production. And we sincerely hope Qualcomm would publicly release the testbed soon. In the meanwhile, we plan to extend *mmKey*, as a software solution, to other mmWave platforms such as TI mmWave radars.

6.4.5 Portability

Although current *mmKey* prototype still requires additional hardware and is not as portable as wearables and multi-functional keyboards, it enables a virtual keyboard by reusing a mmWave device wherever it is already available. With the miniaturization of antennas and chips, it is expected that the mmWave hardware will become lighter, portable, cheaper and energy-saving as a tiny chipset that will be widely available on home routers, smartphones [2, 7] and vehicles [8]. Then *mmKey* could immediately enable virtual typing around those devices in the integrated IoT system. Although wearables and hand-equipped sensors allow typing anywhere, they could not easily input information to the targeted IoT devices in the above scenarios since the wearables are unlikely connected to the IoT devices.

6.4.6 Potential Applications

As the first virtual keyboard using a single mmWave radio, the core contribution of *mmKey* is the processing pipeline that enables accurate localization of micro motion, which can also enable in-air finger tracking/gesture recognition and similar interactive applications.

6.5 Summary

This chapter presents *mmKey*, the first universal virtual keyboard system using a single mmWave device. *mmKey* achieves accurate multi-finger keystroke detection and recognition and supports various keyboard layouts. It employs a novel pipeline of signal processing to detect, segment, and recognize keystrokes, without requiring any training. We evaluate the performance of *mmKey* on various keyboards, including virtual piano keyboard, virtual phone keypad and virtual computer keyboard. The results demonstrate an overall accuracy over 95% for single-key case on different keyboard layouts and a recognition accuracy over 90% for multi-key scenario, which translates to a word recognition accuracy above 97%.

Chapter 7: Conclusion and Future Work

7.1 Concluding Remarks

In this dissertation, we first introduce the primer of wireless sensing, including the techniques to process the CSI estimated by 2.4 GHz/5 GHz Wi-Fi and the basics of 60 GHz Wi-Fi. Then, we present the following CSI-based indoor wireless sensing systems:

1. *Fall Detection.* In Chapter 3, we propose *DeFall*, a Wi-Fi based passive fall detection system that is independent of the environment and free of prior training in new environments. Unlike previous works, our key insight is to probe the physiological features inherently associated with human falls, i.e., the distinctive patterns of speed and acceleration during a fall. *DeFall* consists of an offline template-generating stage and an online decision-making stage, both taking the speed estimates as the input. In the offline stage, augmented DTW algorithms are performed to generate a representative template of the speed and acceleration patterns for a typical human fall. In the online phase, we compare the patterns of the real-time speed/acceleration estimates against the template to detect falls. To evaluate the performance of *DeFall*, we built a

prototype using commercial Wi-Fi devices and conducted experiments under different settings. The results demonstrate that *DeFall* achieves a detection rate above 95% with a false alarm rate lower than 1.50% under both LOS and NLOS scenarios with one link measurement. The extensive comparison study verifies that *DeFall* can be generalized well to new environments without any new training.

2. *Proximity Detection.* In Chapter 4, we present a novel technique for detecting motion in proximity by exploring the physics behind the indoor RF multipath propagation. We discover that motion in the proximity of the Rx/Tx produces distinct time dispersion over the radio channel at the Rx/Tx side. By exploring two novel metrics and linking them with the distance of the motions to antennas, we are able to precisely distinguish motions in nearby proximity from the motions far away. Extensive experiments in various real-world scenarios demonstrate that the proposed scheme can achieve TPR greater than 95% and 99% in distance-based and room-level proximity detection, respectively while maintaining the corresponding FPR less than 5% and 0.5%. The detection delays for a detection distance of 2 m are within 0.6 s, which verifies the responsiveness of the proposed scheme.

3. *Home Automation via On-Device Proximity Detection.* In Chapter 5, we present the design, implementation, and evaluation of a home automation system via on-device proximity detection. With IoT enabled across the home, the devices can respond to the actions users take. While each individual

device can react to the proximity of motion respectively, the network of multiple devices can enable comprehensive decision-making to localize the motion precisely. We believe the proposed scheme takes a promising step towards practical technology for ubiquitous proximity detection on multiple connected devices, allowing for a range of critical applications for smart life.

4. *Virtual Keyboard Implementation.* In Chapter 6, we present *mmKey*, the first universal virtual keyboard system using a single commercial 60 GHz Wi-Fi radio. By leveraging the unique advantages of millimeter-wave signals, *mmKey* converts any flat surface, with a printed paper keyboard, into an effective typing medium. *mmKey* enables concurrent keystrokes and supports multiple keyboard layouts (e.g., computer keyboard, piano keyboard, or phone keypad). We design a novel signal processing pipeline to detect, segment, and separate, and finally recognize keystrokes. *mmKey* does not need any training except for a minimal one-time effort of only three key-presses for keyboard calibration upon the initial setup. We prototype *mmKey* using a commodity 802.11ad/ay chipset, customized to support radar-like operations, and evaluate it with different keyboard layouts under various settings. Experimental results with 10 participants demonstrate a keystroke recognition accuracy of $> 95\%$ for a single-key case and $> 90\%$ for the multi-key scenario, which leads to a word recognition accuracy of $> 97\%$.

7.2 Future Works

In this dissertation, we have demonstrated the capability of wireless sensing for activity monitoring and detection using prototypes built on commercial Wi-Fi devices. However, to make the proposed IoT applications more versatile and useful in real-life scenarios, several critical issues remain to be explored and resolved.

- In our indoor fall detection research, we have demonstrated that falls in different environments at different realizations can be detected by extracting the inherent speed information. However, we only focused on the unexpected hard falls that will cause severe injuries. Such hard falls possess a distinct but consistent speed pattern. In real-life scenarios, there are many other types of falls that may not follow similar patterns. Therefore, in the future, we would like to collect more fall events data and incorporate multiple templates to comprehensively describe fall events.

- In the proposed proximity detection system, we only considered the scenario with a single moving human. However, in real-life scenarios, multiple users may move around the devices at the same time, which will introduce more dynamics to the radio propagation, leading to larger values of proximity features as Section 4.4.2 in Chapter 4 illustrates. Therefore, to successfully deploy indoor monitoring systems, we need to study and analyze the decoupling of multi-user impact in proximity features and develop the corresponding algorithms.

- In addition, in our study of proximity detection, we investigated the radio propagation characteristics with the presumption of consistent motion strength and validate the effectiveness using a single type of motion, walking. However, in practice, users perform various daily activities indoors including working, typing, and reading, which have less strength than walking and bring less interference to the environment. As the discussion in Section 4.4.1 of Chapter 4 indicates, motion strength affects the values of proximity features and the feature values of light motion are generally smaller than heavy motion at the same distance to the devices. In the future, it is necessary to devise finer features to characterize motion strength and differentiate motion types.

- In our virtual keyboard system, we validate the effectiveness of the proposed signal processing pipeline on detecting, localizing, and recognizing the micro finger motion including keystrokes. In the future, we would like to apply the same algorithm to explore other similar interactive approaches such as in-air finger tracking, gesture recognition eyeblink detection, etc., which will enable more interesting IoT applications. In addition, it would be of interest to extract the radio biometrics through measuring "keystroke dynamic", the typing manner and rhythm of users for future recognition and authentication.

- Lastly, in all of the IoT applications in this dissertation, we deal with time series using traditional signal processing techniques and only involve light training due to the limited data. Therefore, it is urgent to collect more data from human groups of various diversity, especially for fall detection. Meanwhile, with the surge of super computers with high computation powers in the era of big data, deep-learning models, such as recurrent neural network (RNN), have successfully been applied to address time series forecasting problems, which is a very important topic in data mining. They have proved to be an effective solution given their capacity to learn and automatically extract the temporal dependencies present in time series such as video and audio streams. In the future, we would like to develop novel deep learning-based algorithms dedicated to radio propagation over time and analyze if it will help improve the robustness in more complicated indoor environments. It is our belief that wireless sensing based systems can achieve comparable identification accuracy with the existing passive vision/audio-based approach, but outperforms them in terms of privacy and restriction of deployment, e.g., no requirement of LOS. We hope that one day the wireless sensing-based IoT application will become prevalent in our everyday life.

Bibliography

[1] 2021 Connectivity and Mobile Trends Survey. [Online]. Available: https://www2.deloitte.com/us/en/insights/industry/telecommunications/connectivity-mobile-trends-survey.html. Accessed: 2021.

[2] Apple U1 TMKA75 Ultra Wideband (UWB) Chip Analysis. [Online]. Available: https://www.techinsights.com/blog/apple-u1-tmka75-ultra-wideband-uwb-chip-analysis. Accessed: 2020.

[3] Bluetooth Technology. [Online]. Available: https://www.bluetooth.com/. Accessed: 2021.

[4] Falls. [Online]. Available: https://www.who.int/news-room/fact-sheets/detail/falls. Accessed: April 2021.

[5] How your home security camera detects motion. [Online]. Available: https://www.techhive.com/article/3263662/how-your-home-security-camera-detects-motion.html. Accessed: 2018.

[6] Qualcomm 802.11ad 60GHz WiFi. [Online]. Available: https://www.qualcomm.com/products/features/80211ad. Accessed: 2019.

[7] Soli Radar-Based Perception and Interaction in Pixel 4. [Online]. Available: https://ai.googleblog.com/2020/03/soli-radar-based-perception-and.html. Accessed: 2020.

[8] Vayyar Imaging Announces First 60GHz In-car Automotive Grade Radar-on-chip. [Online]. Available: https://www.prnewswire.com/news-releases/vayyar-imaging-announces-first-60ghz-in-car-automotive-grade-radar-on-chip-300983523.html. Accessed: 2020.

[9] Wi-Fi Alliance. [Online]. Available: https://www.wi-fi.org/. Accessed: 2021.

[10] Zigbee Alliance. [Online]. Available: http://www.zigbee.org/. Accessed: 2021.

[11] Md Abdullah Al Hafiz Khan, Ruthvik Kukkapalli, Piyush Waradpande, Sekar Kulandaivel, Nilanjan Banerjee, Nirmalya Roy, and Ryan Robucci. Ram: Radar-based activity monitor. In *IEEE INFOCOM 2016 - The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9, 2016.

[12] 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, 2015.

[13] Moeness G. Amin, Yimin D. Zhang, Fauzia Ahmad, and K.C. Dominic Ho. Radar signal processing for elderly fall detection: The future for in-home monitoring. *IEEE Signal Processing Magazine*, 33(2):71–80, 2016.

[14] Sheheryar Arshad, Chunhai Feng, Yonghe Liu, Yupeng Hu, Ruiyun Yu, Siwang Zhou, and Heng Li. Wi-chase: A wifi based human activity recognition system for sensorless environments. In *2017 IEEE 18th International Symposium on A World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, pages 1–6. IEEE, 2017.

[15] Giang Bui, Brittany Morago, Truc Le, Kevin Karsch, Zheyu Lu, and Ye Duan. Integrating videos with lidar scans for virtual reality. In *2016 IEEE Virtual Reality (VR)*, pages 161–162, 2016.

[16] Bo Chen, Vivek Yenamandra, and Kannan Srinivasan. Tracking keystrokes using wireless signals. In *Proceedings of the 13th Annual International Conference on Mobile Systems, Applications, and Services*, pages 31–44, 2015.

[17] 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, 2017.

[18] 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, 2017.

[19] Qingchao Chen, Bo Tan, Kevin Chetty, and Karl Woodbridge. Activity recognition based on micro-doppler signature with in-home wi-fi. In *2016 IEEE 18th International Conference on e-Health Networking, Applications and Services (Healthcom)*, pages 1–6, 2016.

[20] Wenqiang Chen, Yanming Lian, Lu Wang, Rukhsana Ruby, Wen Hu, and Kaishun Wu. Virtual keyboard for wearable wristbands. In *Proceedings of the 15th ACM Conference on Embedded Network Sensor Systems*, pages 1–2, 2017.

[21] Kevin Chetty, Graeme E. Smith, and Karl Woodbridge. Through-the-wall sensing of personnel using passive bistatic wifi radar at standoff distances. *IEEE Transactions on Geoscience and Remote Sensing*, 50(4):1218–1226, 2012.

[22] Laurie Davies and Ursula Gather. The identification of multiple outliers. *Journal of the American Statistical Association*, 88(423):782–792, 1993.

[23] Yu Gu, Fuji Ren, and Jie Li. Paws: Passive human activity recognition based on wifi ambient signals. *IEEE Internet of Things Journal*, 3(5):796–805, 2015.

[24] Daniel Halperin, Wenjun Hu, Anmol Sheth, and David Wetherall. Tool release: Gathering 802.11n traces with channel state information. *SIGCOMM Comput. Commun. Rev.*, 41(1):53, January 2011.

[25] David A. Hill. *Electromagnetic fields in cavities: deterministic and statistical theories*, volume 35. John Wiley & Sons, 2009.

[26] Yuqian Hu, M. Zahid Ozturk, Feng Zhang, Beibei Wang, and K. J. Ray Liu. Robust device-free proximity detection using wifi. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7918–7922, 2021.

[27] Yuqian Hu, Beibei Wang, Chenshu Wu, and K. J. Ray Liu. mmkey: Universal virtual keyboard using a single millimeter wave radio. *IEEE Internet of Things Journal*, pages 1–1, 2021.

[28] Yuqian Hu, Beibei Wang, Chenshu Wu, and K. J. Ray Liu. Universal virtual keyboard using 60 ghz mmwave radar. In *2021 IEEE 7th World Forum on Internet of Things (WF-IoT)*, pages 1–6. IEEE, 2021.

[29] Yuqian Hu, Feng Zhang, Chenshu Wu, Beibei Wang, and K. J. Ray Liu. A wifi-based passive fall detection system. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1723–1727. IEEE, 2020.

[30] Yuqian Hu, Feng Zhang, Chenshu Wu, Beibei Wang, and K. J. Ray Liu. Defall: Environment-independent passive fall detection using wifi. *IEEE Internet of Things Journal*, pages 1–1, 2021.

[31] Hui Ji, Jianxin Chen, Qingyu Lin, and Ang Li. A local fingertips movement and fingertips clustering based virtual keyboard adopting a camera. In *Proceedings of the 2018 International Conference on Computing and Pattern Recognition*, pages 61–67, 2018.

[32] Sanaz Kianoush, Stefano Savazzi, Federico Vicentini, Vittorio Rampa, and Matteo Giussani. Device-free rf human body fall detection and localization in industrial workplaces. *IEEE Internet of Things Journal*, 4(2):351–362, 2017.

[33] Rebecca Killick, Paul Fearnhead, and Idris A Eckley. Optimal detection of changepoints with a linear computational cost. *Journal of the American Statistical Association*, 107(500):1590–1598, 2012.

[34] Youngwook Kim and Brian Toomajian. Hand gesture recognition using micro-doppler signatures with convolutional neural network. *IEEE Access*, 4:7125–7130, 2016.

[35] Manikanta Kotaru, Kiran Joshi, Dinesh Bharadia, and Sachin Katti. Spotfi: Decimeter level localization using wifi. In *Proceedings of the 2015 ACM Conference on Special Interest Group on Data Communication*, SIGCOMM '15, page 269–282, New York, NY, USA, 2015. Association for Computing Machinery.

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

[37] Wenda Li, Robert J. Piechocki, Karl Woodbridge, Chong Tang, and Kevin Chetty. Passive wifi radar for human sensing using a stand-alone access point. *IEEE Transactions on Geoscience and Remote Sensing*, pages 1–13, 2020.

[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), September 2017.

[39] Chung-Chih Lin, Ping-Yeh Lin, Po-Kuan Lu, Guan-Yu Hsieh, Wei-Lun Lee, and Ren-Guey Lee. A healthcare integration system for disease assessment and safety monitoring of dementia patients. *IEEE Transactions on Information Technology in Biomedicine*, 12(5):579–586, 2008.

[40] Kang Ling, Yuntang Liu, Ke Sun, Wei Wang, Lei Xie, and Qing Gu. Spidermon: Towards using cell towers as illuminating sources for keystroke monitoring. In *IEEE INFOCOM 2020-IEEE Conference on Computer Communications*, pages 666–675. IEEE, 2020.

[41] Jian Liu, Yingying Chen, and Marco Gruteser. Vibkeyboard: virtual keyboard leveraging physical vibration. In *Proceedings of the 22nd Annual International Conference on Mobile Computing and Networking*, pages 507–508, 2016.

[42] K. J. Ray Liu and Beibei Wang. *Wireless AI: Wireless Sensing, Positioning, IoT, and Communications*. Cambridge University Press, 2019.

[43] Jiguang Lv, Dapeng Man, Wu Yang, Xiaojiang Du, and Miao Yu. Robust wlan-based indoor intrusion detection using phy layer information. *IEEE Access*, 6:30117–30127, 2018.

[44] Liang Ma, Meng Liu, Na Wang, Lu Wang, Yang Yang, and Hongjun Wang. Room-level fall detection based on ultra-wideband (uwb) monostatic radar and convolutional long short-term memory (lstm). *Sensors*, 20(4):1105, 2020.

[45] Xin Ma, Haibo Wang, Bingxia Xue, Mingang Zhou, Bing Ji, and Yibin Li. Depth-based human fall detection via shape features and improved extreme learning machine. *IEEE Journal of Biomedical and Health Informatics*, 18(6):1915–1922, 2014.

[46] Philip Marquardt, Arunabh Verma, Henry Carter, and Patrick Traynor. (sp) iphone: Decoding vibrations from nearby keyboards using mobile phone accelerometers. In *Proceedings of the 18th ACM conference on Computer and communications security*, pages 551–562, 2011.

[47] Paolo Melillo, Rossana Castaldo, Giovanna Sannino, Ada Orrico, Giuseppe de Pietro, and Leandro Pecchia. Wearable technology and ecg processing for fall risk assessment, prevention and detection. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7740–7743, 2015.

[48] Yan Meng, Jinlei Li, Haojin Zhu, Xiaohui Liang, Yao Liu, and Na Ruan. Revealing your mobile password via wifi signals: Attacks and countermeasures. *IEEE Transactions on Mobile Computing*, 19(2):432–449, 2019.

[49] Ghassem Mokhtari, Qing Zhang, and Amir Fazlollahi. Non-wearable uwb sensor to detect falls in smart home environment. In *2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, pages 274–278, 2017.

[50] Taichi Murase, Atsunori Moteki, Genta Suzuki, Takahiro Nakai, Nobuyuki Hara, and Takahiro Matsuda. Gesture keyboard with a machine learning requiring only one camera. In *Proceedings of the 3rd Augmented Human International Conference*, pages 1–2, 2012.

[51] Armando Muscariello, Guillaume Gravier, and Frédéric Bimbot. Variability tolerant audio motif discovery. In *International Conference on Multimedia Modeling*, pages 275–286. Springer, 2009.

[52] Sameera Palipana, Piyush Agrawal, and Dirk Pesch. Channel state information based human presence detection using non-linear techniques. In *Proceedings of the 3rd ACM International Conference on Systems for Energy-Efficient Built Environments*, pages 177–186, 2016.

[53] Sameera Palipana, David Rojas, Piyush Agrawal, and Dirk Pesch. Falldefi: Ubiquitous fall detection using commodity wi-fi devices. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 1(4):155, 2018.

[54] François Petitjean, Alain Ketterlin, and Pierre Gançarski. A global averaging method for dynamic time warping, with applications to clustering. *Pattern Recognition*, 44(3):678–693, 2011.

[55] François Petitjean, Germain Forestier, Geoffrey I. Webb, Ann E. Nicholson, Yanping Chen, and Eamonn Keogh. Dynamic time warping averaging of time series allows faster and more accurate classification. In *2014 IEEE International Conference on Data Mining*, pages 470–479, 2014.

[56] 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*, page 6. ACM, 2017.

[57] Kun Qian, Chenshu Wu, Zheng Yang, Yunhao Liu, and Zimu Zhou. Pads: Passive detection of moving targets with dynamic speed using phy layer information. In *2014 20th IEEE International Conference on Parallel and Distributed Systems (ICPADS)*, pages 1–8. IEEE, 2014.

[58] 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*, MobiSys '18, page 350–361, New York, NY, USA, 2018. Association for Computing Machinery.

[59] Kun Qian, Chenshu Wu, Zimu Zhou, Yue Zheng, Zheng Yang, and Yunhao Liu. Inferring motion direction using commodity wi-fi for interactive exergames. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*, pages 1961–1972, 2017.

[60] Ramin Ramezani, Yubin Xiao, and Arash Naeim. Sensing-fi: Wi-fi csi and accelerometer fusion system for fall detection. In *2018 IEEE EMBS International Conference on Biomedical Health Informatics (BHI)*, pages 402–405, 2018.

[61] Sai Deepika Regani, Chenshu Wu, Beibei Wang, Min Wu, and K. J. Ray Liu. mmwrite: Passive handwriting tracking using a single millimeter-wave radio. *IEEE Internet of Things Journal*, 8(17):13291–13305, 2021.

[62] Wenjie Ruan, Quan Z Sheng, Lei Yang, Tao Gu, Peipei Xu, and Longfei Shangguan. Audiogest: enabling fine-grained hand gesture detection by decoding echo signal. In *Proceedings of the 2016 ACM international joint conference on pervasive and ubiquitous computing*, pages 474–485, 2016.

[63] Hiroaki Sakoe and Seibi Chiba. Dynamic programming algorithm optimization for spoken word recognition. *IEEE transactions on acoustics, speech, and signal processing*, 26(1):43–49, 1978.

[64] Reinhold Scherer, GR Muller, Christa Neuper, Bernhard Graimann, and Gert Pfurtscheller. An asynchronously controlled eeg-based virtual keyboard: improvement of the spelling rate. *IEEE Transactions on Biomedical Engineering*, 51(6):979–984, 2004.

[65] Ralph Schmidt. Multiple emitter location and signal parameter estimation. *IEEE transactions on antennas and propagation*, 34(3):276–280, 1986.

[66] Souvik Sen, Božidar Radunovic, Romit Roy Choudhury, and Tom Minka. You are facing the mona lisa: Spot localization using phy layer information. In *Proceedings of the 10th International Conference on Mobile Systems, Applications, and Services*, MobiSys '12, page 183–196, New York, NY, USA, 2012. ACM.

[67] Mehmet Saygın Seyfioğlu, Ahmet Murat Özbayoğlu, and Sevgi Zubeyde Gürbüz. Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities. *IEEE Transactions on Aerospace and Electronic Systems*, 54(4):1709–1723, 2018.

[68] Stephan Sigg, Ulf Blanke, and Gerhard Tröster. The telepathic phone: Frictionless activity recognition from wifi-rssi. In *2014 IEEE International Conference on Pervasive Computing and Communications (PerCom)*, pages 148–155, 2014.

[69] Vincent S. Staggs, Lorraine C. Mion, and Ronald I. Shorr. Assisted and unassisted falls: Different events, different outcomes, different implications for quality of hospital care. *The Joint Commission Journal on Quality and Patient Safety*, 40(8):358–364, 2014.

[70] Bo Yu Su, K. C. Ho, Marilyn J. Rantz, and Marjorie Skubic. Doppler radar fall activity detection using the wavelet transform. *IEEE Transactions on Biomedical Engineering*, 62(3):865–875, 2015.

[71] Xiaolin Su, Yunzhou Zhang, Qingyang Zhao, and Liang Gao. Virtual keyboard: A human-computer interaction device based on laser and image processing. In *2015 IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER)*, pages 321–325. IEEE, 2015.

[72] Vincent Tam and Ling-Shan Li. Integrating the kinect camera, gesture recognition and mobile devices for interactive discussion. In *Proceedings of IEEE International Conference on Teaching, Assessment, and Learning for Engineering (TALE) 2012*, pages H4C–11–H4C–13, 2012.

[73] David Tse and Pramod Viswanath. *Fundamentals of wireless communication.* Cambridge university press, 2005.

[74] M. S. Varela and M. G. Sanchez. Rms delay and coherence bandwidth measurements in indoor radio channels in the uhf band. *IEEE Transactions on Vehicular Technology*, 50(2):515–525, 2001.

[75] Lochan Verma, Mohammad Fakharzadeh, and Sunghyun Choi. Wifi on steroids: 802.11ac and 802.11ad. *IEEE Wireless Communications*, 20(6):30–35, 2013.

[76] 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.

[77] Fengyu Wang, Feng Zhang, Chenshu Wu, Beibei Wang, and K. J. Ray Liu. Vimo: Multiperson vital sign monitoring using commodity millimeter-wave radio. *IEEE Internet of Things Journal*, 8(3):1294–1307, 2021.

[78] Hao Wang, Daqing Zhang, Yasha Wang, Junyi Ma, Yuxiang Wang, and Shengjie Li. Rt-fall: A real-time and contactless fall detection system with commodity wifi devices. *IEEE Transactions on Mobile Computing*, 16(2):511–526, 2017.

[79] Junjue Wang, Kaichen Zhao, Xinyu Zhang, and Chunyi Peng. Ubiquitous keyboard for small mobile devices: harnessing multipath fading for fine-grained keystroke localization. In *Proceedings of the 12th annual international conference on Mobile systems, applications, and services*, pages 14–27, 2014.

[80] Tao Wang, Dandan Yang, Shunqing Zhang, Yating Wu, and Shugong Xu. Wi-alarm: Low-cost passive intrusion detection using wifi. *Sensors*, 19(10):2335, 2019.

[81] Wei Wang, Alex X. Liu, Muhammad Shahzad, Kang Ling, and Sanglu Lu. Device-free human activity recognition using commercial wifi devices. *IEEE Journal on Selected Areas in Communications*, 35(5):1118–1131, 2017.

[82] Xueyi Wang, Joshua Ellul, and George Azzopardi. Elderly fall detection systems: A literature survey. *Frontiers in Robotics and AI*, 7:71, 2020.

[83] Yuxi Wang, Kaishun Wu, and Lionel M. Ni. Wifall: Device-free fall detection by wireless networks. *IEEE Transactions on Mobile Computing*, 16(2):581–594, 2016.

[84] Chenshu Wu, Zheng Yang, Zimu Zhou, Xuefeng Liu, Yunhao Liu, and Jiannong Cao. Non-invasive detection of moving and stationary human with wifi. *IEEE Journal on Selected Areas in Communications*, 33(11):2329–2342, 2015.

[85] 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.

[86] Chenshu Wu, Feng Zhang, Beibei Wang, and K. J. Ray Liu. Easitrack: Decimeter-level indoor tracking with graph-based particle filtering. *IEEE Internet of Things Journal*, 7(3):2397–2411, 2020.

[87] Chenshu Wu, Feng Zhang, Beibei Wang, and K. J. Ray Liu. mmtrack: Passive multi-person localization using commodity millimeter wave radio. In *IEEE International Conference on Computer Communications*, April 27-30 2020.

[88] 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.

[89] Chien-Min Wu, Chih-Wen Hsu, Tzu-Kuei Lee, and Shana Smith. A virtual reality keyboard with realistic haptic feedback in a fully immersive virtual environment. *Virtual Reality*, 21(1):19–29, 2017.

[90] Jiang Xiao, Kaishun Wu, Youwen Yi, Lu Wang, and Lionel M. Ni. Pilot: Passive device-free indoor localization using channel state information. In *2013 IEEE 33rd International Conference on Distributed Computing Systems*, pages 236–245, 2013.

[91] Yaxiong Xie, Zhenjiang Li, and Mo Li. Precise power delay profiling with commodity wi-fi. *IEEE Transactions on Mobile Computing*, 18(6):1342–1355, 2019.

[92] Qinyi Xu, Yan Chen, Beibei Wang, and K. J. Ray Liu. Trieds: Wireless events detection through the wall. *IEEE Internet of Things Journal*, 4(3):723–735, 2017.

[93] Zheng Yang, Zimu Zhou, and Yunhao Liu. From rssi to csi: Indoor localization via channel response. *ACM Computing Surveys (CSUR)*, 46(2):1–32, 2013.

[94] Yafeng Yin, Qun Li, Lei Xie, Shanhe Yi, Edmund Novak, and Sanglu Lu. Camk: A camera-based keyboard for small mobile devices. In *IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on Computer Communications*, pages 1–9. IEEE, 2016.

[95] 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.

[96] 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.

[97] 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. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 3(3):1–24, 2019.

[98] Feng Zhang, Chenshu Wu, Beibei Wang, and K. J. Ray Liu. mmeye: Super-resolution millimeter wave imaging. *IEEE Internet of Things Journal*, 8(8):6995–7008, 2021.

[99] Feng Zhang, Chenshu Wu, Beibei Wang, Min Wu, Daniel Bugos, Hangfang Zhang, and KJ Ray Liu. Smars: sleep monitoring via ambient radio signals. *IEEE Transactions on Mobile Computing*, 2019.

[100] Jie Zhang, Xiaolong Zheng, Zhanyong Tang, Tianzhang Xing, Xiaojiang Chen, Dingyi Fang, Rong Li, Xiaoqing Gong, and Feng Chen. Privacy leakage in mobile sensing: Your unlock passwords can be leaked through wireless hotspot functionality. *Mobile Information Systems*, 2016, 2016.

[101] Lei Zhang, Zhirui Wang, and Liu Yang. Commercial wi-fi based fall detection with environment influence mitigation. In *2019 16th Annual IEEE International Conference on Sensing, Communication, and Networking (SECON)*, pages 1–9, 2019.

[102] Lingyan Zhang, Qinghua Gao, Xiaorui Ma, Jie Wang, Tingting Yang, and Hongyu Wang. Defi: Robust training-free device-free wireless localization with wifi. *IEEE Transactions on Vehicular Technology*, 67(9):8822–8831, 2018.

[103] Yuliang Zhao, Chao Lian, Xueliang Zhang, Xiaopeng Sha, Guangyi Shi, and Wen J Li. Wireless iot motion-recognition rings and a paper keyboard. *IEEE Access*, 7:44514–44524, 2019.

[104] Rui Zhou, Xiang Lu, Pengbiao Zhao, and Jiesong Chen. Device-free presence detection and localization with svm and csi fingerprinting. *IEEE Sensors Journal*, 17(23):7990–7999, 2017.

[105] Zimu Zhou, Chenshu Wu, Zheng Yang, and Yunhao Liu. Sensorless sensing with wifi. *Tsinghua Science and Technology*, 20(1):1–6, 2015.

[106] Hai Zhu, Fu Xiao, Lijuan Sun, Ruchuan Wang, and Panlong Yang. R-ttwd: Robust device-free through-the-wall detection of moving human with wifi. *IEEE Journal on Selected Areas in Communications*, 35(5):1090–1103, 2017.

[107] Hai Zhu, Fu Xiao, Lijuan Sun, Xiaohui Xie, Panlong Yang, and Ruchuan Wang. R-pmd: Robust passive motion detection using phy information with mimo. In *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)*, pages 1–8. IEEE, 2015.

[108] Tong Zhu, Qiang Ma, Shanfeng Zhang, and Yunhao Liu. Context-free attacks using keyboard acoustic emanations. In *Proceedings of the 2014 ACM SIGSAC conference on computer and communications security*, pages 453–464, 2014.

[109] Li Zhuang, Feng Zhou, and J Doug Tygar. Keyboard acoustic emanations revisited. *ACM Transactions on Information and System Security (TISSEC)*, 13(1):1–26, 2009.