Human Activity Recognition Using Millimetre-Wave Radars With Machine Learning

By

HAN CUI

Department of Electrical and Electronic Engineering
UNIVERSITY OF BRISTOL

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of DOCTOR OF PHILOSOPHY in the Faculty of Engineering.

JULY 2022

Word count: 40834

Abstract

Human activity recognition (HAR) has been studied for decades in computer vision and has shown great success. However, as people are caring more about privacy issues, researchers are investigating non-intrusive HAR systems using radar-based techniques, among which millimetre-wave (mmWave) radars have received great popularity due to their capability of capturing high-resolution spatial information about the scene. This thesis presents a systematic study of HAR using mmWave radars. It explains the fundamentals of mmWave sensing techniques, discusses its use in HAR applications, and highlights the challenge of the sparse and noisy data through a purpose-built simulation system that can import arbitrary 3D models to form a scene and simulate the radar signal with configuration antenna settings. A software framework for managing multiple radars is presented that allows real-time data transmission, data processing, and result visualization.

Based on the software framework, three HAR systems are presented. First, a human detection and tracking system is presented as the fundamental of HAR. The system operates two radars simultaneously that verify each other's detection and significantly reduce the probability of false alarms. The system achieves 90.4% sensitivity and 98.6% precision when detecting up to four people in the room. Then, a human posture estimation system is presented that uses two radars as a vertical array and a neural network model to estimate the joint positions of the person. The system achieved over 71.3% accuracy when detecting postures that are commonly seen in an office environment with arbitrary limb motions. Finally, a human vital sign detection system is presented that uses one mmWave radar to detect a person's heart rate when exercising on a treadmill. It overcomes the challenge that the heartbeat signal can be difficult to extract when there is body movement, and achieved a low error rate of 5.4%.

Dedication and acknowledgements

I would like to thank my supervisor, Professor Naim Dahnoun, for all the academic and general support throughout my time at Bristol. You are always very attentive and put a lot of effort into providing timely feedback and guidance on my work that many supervisors would not do. I would like to thank my parents. You allow me to be able to finish my study and choose my career in the way I want, and always with the most assertive confidence. Special thanks to my girlfriend, Ling. Thank you for joining my life, adding an extra layer of brilliance to my study and making it some of my best memories.

Thanks to my colleagues who I have worked with. It has been a great pleasure to do the research work with you. Thanks to my friends for all the joyful time when away from work, and especially to mention the badminton games that helped me keep fit somehow. Thanks to my family, for no particular reasons, but just being the most kind and supportive family I could imagine. Finally, thanks to all who have provided feedback on my research in various ways, including academics at the University of Bristol, staff from Texas Instruments, reviewers and editors of my papers, and researchers who have cited my publications. Although not always positive, these experience have somehow increased my confidence in my work and, more importantly, let me feel the life as a real researcher.

Author's declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: ..... DATE: .....

Table of Contents

List of Tables

List of Figures

Figure **Page**

List of Abbreviations

AdaBoost adaptive boosting.

AdaGrad adaptive gradient algorithm.

Adam adaptive moment estimation.

ADC analogue to digital converter.

AoA angle-of-arrival.

AoV angle-of-view.

API application programming interface.

bpm beats per minute.

CFAR constant false alarm rate.

CNN convolutional neural network.

CPU central processing unit.

CUT cells-under-test.

CW continuous wave.

DBSCAN density-based spatial clustering of applications with noises.

DFT discrete Fourier transform.

DPC data processing chain.

DSP digital signal processor.

ECG electrocardiogram.

FAR false alarm rate.

FFT fast Fourier transfrom.

FIFO first in first out.

FIR finite impulse response.

FMCW frequency-modulated continuous-wave.

FMI Fowlkes–Mallows index.

FN false negative.

FP false postive.

FPGA field programmable gate arrays.

FPS frame per second.

GPS global positioning system.

GPU graphic processing unit.

HAR human activity recognition.

HoG histograms of oriented gradients.

HR heart rate.

IF intermediate frequency.

IMU inertial measurement unit.

IoU Intersection over Union.

ISM industrial, scientific and medical.

KNN k-nearest neighbour.

LIDAR light detection and ranging.

LSTM long short-term memory.

MDL minimum descriptive length.

MIMO multiple-in multiple-out.

mmWave millimetre wave.

MUSIC Multiple Signal Classifier.

MVDR Minimum Variance Distortionless Response.

NHS National Health Service.

PC personal computer.

R-CNN region-based convolutional neural network.

ReLU rectified linear unit.

RF radio frequency.

RGB red, green and blue.

RNN recurrent neural networks.

RNSProp root mean square propagation.

RSS received signal strength.

RX receiver.

SDK software development kit.

SGD stochastic gradient descent.

SIFT scale invariant feature transform.

SNR signal to noise ratio.

SURF speeded up robust features.

SVM support vector machine.

TI Texas Instruments.

TN true negative.

ToF time-of-flight.

TP true positive.

TX transmitter.

UWB ultra-wide band.

WHO World Health Organization.

WLAN wireless local area network.

List of Notations

$M^{(X \times Y)}$	A 2D matrix with X rows and Y columns.
$M_{x,\cdot}$	The x^{th} row of M as a vector.
$M_{\cdot,y}$	The y^{th} column of M as a vector.
$M_{x,y}$	The element at x^{th} row and y^{th} column of M..
$M_{x_1:x_2,y_1:y_2}$	A sub-matrix of M from row x_1 to x_2 and column y_1 to y_2.
$\|M\|$	The absolute values (or magnitude) of each element in M.
$\angle M$	The Euclidean plane angle (or phase) of each element in M..
M^T	The transpose of M.
M^H	The Hermitian transpose of M.
M^{-1}	The inverse matrix of M.
$\{x\}$	A set containing an element x..
$\{x...y\}$	A set containing integer elements from x to y..
$\|\{x\}\|$	The cardinality of the set $\{x\}$.
$[x, y]$	A continuous range from x to y.

CHAPTER 1

INTRODUCTION

With the development of micro-electronic technology, computers are playing an increasingly important role in many industries. Interaction and cooperation between humans and digital tools have become the key to improving working efficiency and saving human resources. Therefore, human activity recognition (HAR) has received a lot of interest in both the academic and industry fields. HAR can include recognizing people's identities, location, motion, activity, health status and other information that can be beneficial in human-computer interaction. Being able to gather this information allows tasks in many industries, such as health care and security, to be addressable by computers with reduced human intervention. HAR tasks have been challenging due to the high complexity and diversity of human activities in real-world scenarios. However, with the development of machine learning techniques, many solutions have been proposed for complex HAR tasks, such as human tracking, identification, and motion recognition.

Among all the industries that would benefit from HAR, health monitoring has received the greatest attention, especially during and post the Covid-19 pandemic. The pandemic has resulted in more people in the UK experiencing delayed hospital treatment. According to the National Health Service (NHS) statistics released in March 2022, people need to wait 12 weeks on average before getting treatment (in comparison to 7 weeks in 2019), and there are over 300,000 people who need to wait for over a year (in comparison to only around 2,000 people in 2019) [1]. Meanwhile, studies show that the national volume of surgical activity has dropped more than 33%, and this is expected to continue for years [2, 3]. The increase in waiting time and the reduced capacity for medical treatment make self-health monitoring increasingly important. Researchers have shown that a large variety of monitoring tasks can be performed by computers with different sensors, including monitoring people's positions, motions, vital signs, postures, and

actions.

One of the most important tasks in health monitoring is assessing a person's heart status and detecting abnormal heart activities. The World Health Organization estimates that there are 17.9 million deaths per year globally due to cardiovascular diseases, contributing to 32% of the total deaths [4]. In the UK, the number is estimated to be around 160 thousand, of which 73% are aged above 75 [5]. In addition, there are 7.6 million people living with cardiovascular diseases, who might require long-term monitoring of their health status [5]. Nonetheless, certain types of diseases are hard to diagnose, and the patients may not receive timely treatment. For example, it is estimated that 12.5% of people diagnosed with atrial fibrillation are not treated effectively, and 40% of people diagnosed with heart failure could benefit from earlier treatment [5]. Regular monitoring of the heart status and detecting any potential diseases at early stages is critical in reducing unexpected deaths, especially when medical resources are limited. Although professional checks like electrocardiograms might not always be possible at home or in the work environment, researchers have shown that simple monitoring of the heart rate can be an effective way of assessing a person's cardiac health and indicating the possibility of potential diseases [6]. A recent study has shown that the heart rate can also be used to analyse a person's emotion [7], which helps understand their mental health. However, analysing the health status through the heart rate often requires data over a long period of time and across different intervals of the day. Therefore, contactless and ubiquitous monitoring is required to allow people to monitor their heart rate at home and work, as well as to monitor patients at hospitals with a reduced cost of equipment and human resources.

Another important aspect to be monitored is the daily activity pattern. Nowadays, many people spend the majority of their time sitting, either for work (people in midlife) or for entertainment (elderly). An international survey shows that most people spend 180 to 480 minutes per day sitting [8], and there is also evidence that long-time sitting increases the possibility of a large range of health problems [9]. Even with enough daily activity, long-time sitting has deleterious cardiovascular and metabolic effects [10]. Therefore, it would be beneficial to have a posture monitoring system that could detect prolonged sitting time and suggest breaks automatically. Meanwhile, the sitting posture has also been shown to have various effects on the person's health [11] and reflects the person's emotion and stress level [12]. Similar effects are also found for walking postures [13] and sleeping postures [14]. These issues can be further addressed by having an effective posture monitoring system, to maintain a healthy posture in various daily activities and provide another dimension in understanding the person's health status.

Finally, detecting and understanding human posture allows advanced human-computer interaction systems to be developed and improve user experiences in many environments, including at home, at work and in public places. For example, smart homes have been proposed to help elderly and disabled people live more conveniently and comfortably at home [15]. These systems often use various types of sensors to understand the person's activity and provide appropriate

assistance, where a posture estimation system could provide more detailed information about the person and help recognize a wider range of activities, as well as detecting abnormal activities like a fall.

Effective HAR often relies on high-resolution data of the subject to be collected. Therefore, there is a large amount of literature in computer vision that uses optical cameras for HAR, including analysing a person's vital signs, activities and postures. However, cameras have the disadvantages of being intrusive, raising privacy concerns and relying on good lighting conditions. There are also many kinds of wearable devices that could measure a person's heart rate and breathing rate, but it is often practically impossible to ask the subject to wear it all the time, and there is a possibility that the device can be damaged or lost. Smartwatches can be used to monitor a person's heart rate and exercise level, but they are generally expensive and might not be affordable to people with low incomes. There are also non-contacting solutions using radio-frequency (RF) signals to monitor the heart rate and breathing rate [16, 17]. However, most of them require the subject to be stationary, which is not suitable for long-time monitoring or for monitoring people while exercising. Smart chairs, like [18], have been proposed to analyse a person's sitting posture by installing various sensors on an office chair. However, the information returned from the sensors is simple and does not include the full posture. Non-contacting solutions for posture monitoring often have a high cost, like the Microsoft Kinect. There is a lack of one system that is low-cost, contactless, non-intrusive, and capable of detecting the person's vital signs, activity level and postures at the same time.

To address the mentioned problems, this research proposes to use millimetre wave (mmWave) radars to build a complete and multi-functional HAR system, that is capable of long-term ubiquitous detection of a person's vital signs, activities and postures, hence providing an effective tool for monitoring, analysing and improving their health status. A mmWave radar sends a modulated electromagnetic signal, detects the signal reflection from any object, processes the signal and determines the range, velocity and angle of incidence of the object. It has the advantages of being non-intrusive, having a small antenna size, and being able to capture the scene at a higher resolution when compared with traditional radars. The high resolution allows detailed information about the subject to be analysed and advanced data processing algorithms to be developed, whereas smaller antenna sizes reduce the cost of the radar chip and provide high portability. For example, the cost of a mmWave radar chip from Texas Instruments (TI) can be around £10 [19], which can be affordable to people at all income levels. In addition, the non-intrusive nature means that only anonymous data will be collected, so that the privacy of the subject can be better protected. They can also operate in various conditions including darkness, smoke and fog, which are crucial in many applications. All these features increase the popularity of mmWave radars for human activity recognition and health monitoring. [1]

In this research, the mmWave radars from TI are used. The radars operate at 77 GHz to

[1] Some text in this paragraph have been published in [20, 21] ©2021-2022 IEEE.

81 GHz, with a bandwidth of 4 GHz and a wavelength of around 4 mm. The high bandwidth provides a distance resolution at 4 cm. In addition, tiny displacement at millimetre levels can be detected by analysing the phase change of the signal. The radar has three transmitters and four receivers that can operate concurrently, allowing the radar to separate signal sources in azimuth and elevation domains, and, therefore, to monitor multiple people in the range simultaneously. This research investigates the capability of mmWave radar in HAR, with a particular interest in three challenges: human detection and tracking, posture estimation, and heart rate detection.

1.1 Contribution

The contribution of this thesis can be summarized as follows:

- A human detection and tracking system using two radars is presented. A novel DPC is presented to filter out irrelevant information, such as clutter and noise, and locate people in the scene. It is shown that mmWave radars have good performance in indoor environments, with over 90% sensitivity. However, using a single radar can raise a large number of false alarms due to unstable data and noise, whereas the precision of the system can be improved significantly with two radars. The system achieves 90.4% sensitivity and 98.6% precision when detecting multiple people in an office environment. The system achieves a mean localization error of 5.6 cm on people tracking, which outperforms the state-of-the-art RF-based tracking systems. (Chapter 5.)

- A human posture estimation system using two radars is presented. The system detects people with arbitrary postures in indoor environments at close distances (within two metres) and estimates the posture by localizing the key joints. In contrast to much existing research that only focuses on standing postures, this work is the first mmWave radar-based system that can accurately estimate a rich set of postures that are commonly seen in an office environment, while having real-time processing speed and a low cost. Two mmWave radars are used to capture the scene and a neural network model is used to estimate the posture. The neural network model consists of a part detector that estimates the subject's joint positions, and a spatial model that learns the correlation between the joints. A temporal correlation step is introduced to further refine the estimate when in real-time operation. The system can provide an accurate posture estimate of the person in real-time at 20 fps, with a mean localization error of 12.2 cm and an average precision of 71.3%. (Chapter 6.)

- A vital sign detection system is presented that uses one mmWave radar to detect a person's heart rate when exercising on a treadmill. It overcomes the challenge that the heartbeat signal can be difficult to extract when there is body movement, by using a novel phase signal construction algorithm that measures the person's chest displacement at a high resolution, and a machine learning model that predicts the person's heart rate based on the

motion level. In contrast to many existing systems that require the person to stay static during the measurement, this is one of the first research that targets people in exercise. The system achieved a low error rate of 5.4%. (Chapter 7)

- A simulation system is designed for simulating the radar signal in a customized scene, as the first mmWave radar simulator for 3D imaging applications. It let researchers focus on the high-level DPC design and verification without worrying about hardware setup and data acquisition. It supports a rich set of parameters to be configured based on the desired use case, including the number of radars, antenna layout, chirp configuration, objects and/or people in the scene and their motions, and noise in the environment. 3D models from other datasets can be imported into the simulator as the ground truth, allowing real-world scenes to be studied. (Chapter 4)

- A super-resolution point cloud construction (SRPC) algorithm that aims to improve the distribution and resolution of the point cloud generated by the radar. Experiments showed that the quality of the radar data can be sparse and noisy when compared to cameras or light detection and ranging (LIDAR) systems. When evaluating the SRPC algorithm using the simulation system, the algorithm successfully improved the overall accuracy of the radar and provided a more natural point cloud with reduced outliers. (Chapter 4)

1.2 Thesis Outline

The outline of the thesis is described below.

Chapter 2 Background and Related Work

This chapter reviews related research in HAR and machine learning. It introduces the fundamental knowledge of machine learning and reviews HAR algorithms and systems based on the hardware used, including cameras, sensors, and wearable devices. It also discusses the background of mmWave sensors and their use in the HAR literature. Then, it discusses two fields that are particularly related to this research: human posture estimation and vital sign detection.

Chapter 3 mmWave Radars Fundamentals

This chapter explains the fundamentals of mmWave radars in terms of how they can capture information in the scene. It describes the underlying frequency-modulated continuous-wave (FMCW) model of the radars on the distance, velocity, and angle-of-arrival (AoA) detection of objects. It also describes the TI mmWave radar models that are used in this research, including the hardware aspect, the DPC implementation, and the communication between the radar and a host environment.

Chapter 4 Using mmWave Radar as 3D Sensor

This chapter presents a simulation system that implements the DPC described in Chapter 3. It allows 3D models from public datasets to be imported into the simulator to form a customized scene and serve as the ground truth, and can simulate the radar data as the radar is placed in the scene. Based on this system, a qualitative and quantitive study of how well a mmWave radar can capture information from a scene and serve as a 3D sensor is presented. The study evaluates several key factors that could affect radar detection, such as the DPC, environment and antenna layouts. It shows that there is still a big gap between the quality of mmWave radar detection and higher resolution systems like LIDARs, and emphases the importance of data post-processing in higher-level applications. Finally, a novel SRPC algorithm is proposed to improve the resolution and distribution of the radar detection and is verified using the simulation system. This work has contributed to a paper published in the MECO conference [22].

Chapter 5 Human Detection and Tracking [2]

In this chapter, a human detection and tracking system using mmWave radars is presented. The system uses two radars to verify each other's detection and a tracking module to continuously analyse the people's location over time, which significantly reduces the probability of false alarms. A purpose-built software framework for real-time radar management and data processing on a PC is presented. The content of this chapter has been published in the IEEE Aerospace and Electronic Systems Magazine [20] and in an international patent [23]. An extension of this work has been published in the MECO conference [24].

Chapter 6 Human Posture Estimation [3]

In this chapter, a novel human posture estimation system using mmWave radars is presented. The system achieved over 71.3% accuracy when detecting postures that are commonly seen in an office environment, like sitting, standing, and walking, with arbitrary limb motions. The system uses a two-phase neural network model to estimate the posture from the radar data and is one of the first research in the field that uses mmWave radars to detect a wide range of postures. The content of this chapter has been published in the IEEE Sensors Journal [21] and the MECO conference [25], and published in the patent with the detection and tracking system [23]. An extension of this work has been submitted to the Microprocessors and Microsystems [26].

Chapter 7 Human Vital Sign Detection

In this chapter, a vital sign detection system is presented that can detect a person's heart rate while exercising on a treadmill. One mmWave radar is used to detect the position of the person,

[2] This chapter contains reprinted content from [20] ©2021 IEEE.
[3] This chapter contains reprinted content from [21, 25] ©2020-2022 IEEE.

construct a phase signal representing the chest movement, and analyse the phase signal to identify the heart rate. While there has been much research on detecting the vital sign of a stationary person, detecting a moving person is significantly more difficult, as the body movement would be much stronger than the chest displacement due to heartbeats. To address the issue, a machine learning model is presented that predicts the trend of the heart rate change based on the motion level of the subject. Experiments showed that the proposed system can detect the heart rate of an exercising person with a low error rate of 5.4%. This work has contributed to a UK patent application [27], a paper published in the MECO conference [28], and a journal paper submitted to the Microprocessors and Microsystems [29].

1.3 Publications

This research has contributed to three journal papers (two published and one under review), four conference papers (all published) and two patents (one published and one filed).

Journal Papers

- **H. Cui** and N. Dahnoun, "Real-time short-range human posture estimation using mmWave radars and neural networks," *IEEE Sensors Journal*, vol. 22, no. 1, pp. 535-543, 2022.
- **H. Cui** and N. Dahnoun, "High precision human detection and tracking using millimeter-wave radars," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 1, pp. 22–32, 2021.
- J. Wu, **H. Cui**, and N. Dahnoun, "A voxelization algorithm for reconstructing mmwave radar point cloud and an application on posture classification," Submitted to *Microprocessors and Microsystems*, June 2022, under review.

Conference Papers

- J. Wu, **H. Cui**, and N. Dahnoun, "A novel heart rate detection algorithm with small observing window using millimeter-wave radar," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 2022.
- J. Wu, **H. Cui**, and N. Dahnoun, "An improved angle estimation algorithm for millimeter-wave radar," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 2022.
- J. Wu, **H. Cui** and N. Dahnoun, "A Novel High Performance Human detection, Tracking and Alarm System Based on millimeter-wave Radar," in *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, 2021.

- **H. Cui** and N. Dahnoun, "Human posture capturing with millimetre wave radars," in *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, 2020.

Patent Application

- N. Dahnoun and **H. Cui**, "Radar detection and tracking," International Patent Published WO 2022/130 350 A1, Dec. 18, 2020.
- N. Dahnoun, **H. Cui**, and J. Wu, "Determining vital signs," U.K. Patent Filed GB2 203 223.9, Mar. 08, 2022

CHAPTER 2

BACKGROUND AND RELATED WORK

HAR has been studied in depth in the literature and many systems have been proposed, especially during the past decade with the rapid development of microprocessors and machine learning techniques. This chapter gives literature reviews of the related work of this thesis. The chapter is divided into five sections. Section 2.1 provides the preliminary knowledge on machine learning for understanding this thesis. Section 2.2 reviews HAR techniques by the hardware used, including cameras, sensors and wearable devices. Section 2.3 and Section 2.4 discuss two subjects in HAR that are particularly relevant to this thesis: human posture estimation and vital sign detection. Section 2.5 concludes the chapter. Some content in this chapter has been published in [20, 21].

HAR tasks include but are not limited to:

- Detecting and recognizing the presence of people.
- Locating people and tracking their motion.
- Posture estimation, such as distinguishing between sitting and standing.
- Vital sign detection, such as detecting the heart rate and breathing rate.
- Activity classification, such as distinguishing between walking and running, distinguishing between working and entertaining, or detecting abnormal activities like fall detection.

A HAR system often focuses on one or a subset of these problems, as performing all the tasks using a single system is often impractical. Many HAR tasks are inherently machine learning problems. HAR systems often adopt machine learning techniques to process the data captured by the sensor and recognize the underlying human activity based on prior knowledge. These

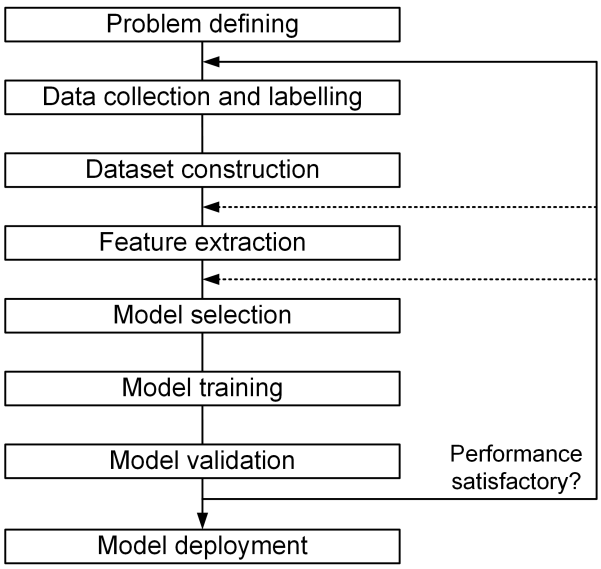


Figure 2.1: The process of training a supervised model.

systems often use a mathematical model to learn the relationship between the input data and the possible outcomes. This chapter discusses the fundamentals of machine learning techniques and their applications in HAR.

2.1 Machine Learning Fundamentals

There have been many reviews on the machine learning techniques in the literature, such as [30]. This section aims to give a brief introduction to the most important concepts of machine learning, in terms of supervised learning, unsupervised learning and neural networks, respectively.

2.1.1 Supervised Learning

Supervised learning models require a training stage, where a mathematical model between the data and the target classes has to be designed and trained using a set of known data. The model iteratively maps the input to a possible outcome, compares it with the ground truth and modifies its internal parameters to minimize the prediction error. Once the model has been trained, it can be applied to unseen data and predict the most possible outcome.

The process of training a supervised model is summarized in Figure 2.1. The first step is data collection and labelling. The data should be collected in the same or a similar environment as the deployment environment, and the amount of data should be large enough to cover common cases. Supervised learning requires data to be labelled with the ground truth, to guide the model during the training process. For classification problems, the output of the model is defined into classes and the model computes the most possible class that an input belongs to. Classification models can be used for simple decision-making problems when the possible outcomes have well-defined

boundaries, such as detecting if a person is present or not. In contrast, for regression problems, the output can span across a continuous space and the model predicts the most possible output in that range based on the input. Regression models can be used when the output is continuous and cannot be expressed as classes, such as detecting the position of a person in the room.

The data collection and ground truth generation process differs a lot depending on the hardware used and the application. For example, when using radar data for human detection, the scene should be set up based on the desired use case and the ground truth location of the person subjects should be captured using a separate reliable and high-resolution system, like a robust computer vision system. Once the data has been collected, it needs to be split into a training dataset and a test dataset. During the training process, only data from the training dataset should be used for updating the model parameters, to prevent the model from overfitting to the data collected. The test dataset simulates the real use case and should only be used for evaluating a trained model. Some applications also have an additional validation dataset to further prevent data leakage during the training process.

Machine learning models often rely on the features extracted from the input data. For image data, feature extraction algorithms on HAR include SIFT (scale invariant feature transform) [31], SURF (speeded up robust features) [32], HOG (histograms of oriented gradients) [33], Haar-like features for face detection [34], motion features extracted from temporal information [35], and many others as reviewed by Nixon and Aguado [36]. These algorithms convert the image input into various mathematical representations that can be handled by the model. For 1D signal, the features are often derived from the signal amplitude, phase, and frequency, and feature extraction algorithms can include time-domain analysis like peak detection and convolution, or frequency-domain analysis like fast Fourier transfrom (FFT) and wavelet transform. Then, depending on the number of features and the complexity of the task, there is a rich set of machine learning models that can be selected [30]. To name a few, support vector machine (SVM) is a powerful classification model when there are well-defined boundaries between the data from different classes, but does not scale to large datasets; Decision tree and random forest can handle a rich set of data types beyond numeric values, but at the expense of a higher complexity; Naive Bayes is a statistical model with low complexity and works well for large datasets, but with the condition that the input features should not be correlated [30]. A special type of model is the neural networks, which will be discussed separately in Section 2.1.3.

After determining which model and features to use, the datasets can be processed to a higher level abstraction and be fed into the model training stage. The training stage is an optimization process that can be applied to the entire dataset at once or applied iteratively to sub-datasets, whose objective is to find a set of parameters that minimize a loss function that defines the difference between the prediction and the ground truth. The model has a set of randomly initialized parameters, processes the input data to get the initial prediction, compares it with the ground truth, and updates the parameters to reduce the error. The training continues until

the model parameter converges and the accuracy no longer improves. One example optimization algorithm is the gradient descent algorithm. For each data instance, it computes the gradient of the loss with respect to the parameters and updates the parameters along the direction that would reduce the gradient, until the minimum gradient is found, which corresponds to a minimal loss between the prediction and the ground truth. The amount of updates depends on the user-defined learning strategy and learning rate, where a higher learning rate corresponds to a larger update rate and more aggressive learning. There have been many optimizations being proposed over the standard gradient descent to improve the learning efficiency and/or reduce the computational cost, such as the stochastic gradient descent (SGD) algorithm, the Adaptive Moment Estimation (Adam) algorithm [37] and the Adaptive Gradient (AdaGrad) algorithm [38]. A more detailed review of optimization algorithms can be found in [39, 40].

One particularly important problem in machine learning is preventing the model from overfitting to the training data. An overfitted model may perform extremely well on the training dataset, like when using a decision tree with an unlimited number of leaves, but may fail in real applications due to poor generalization. One way to reduce the chance of overfitting when training on a large-scale dataset is using regularization techniques, often by adding a penalty term in the loss function to constrain the importance of certain parameters, such as the shrinkage method in traditional machine learning and weight decaying in neural networks [41]. Having a separate test dataset (and a validation dataset when necessary) is also an important way to examine if a model is overfitting or not, given that the correlation between the datasets is kept minimal. Finally, if the performance is not satisfactory, there can be many reasons and one may need to re-evaluate each step of the workflow in Figure 2.1. For example, evaluating whether the quality of the data and the ground truth is good enough, whether the feature extracted can effectively represent the data, or whether the complexity of the model is capable of handling the task. If the model is showing a satisfactory performance without appearing to be overfitting, it can be deployed in the real application.

2.1.2 Unsupervised Learning

An unsupervised learning model does not require training data that have a ground truth label, and will not receive any feedback on its prediction. Instead, the model analyses and learns the most significant features of the input data and performs tasks like clustering or feature analysis. Unsupervised learning does not require the training stage, but it still plays an important role in the data processing.

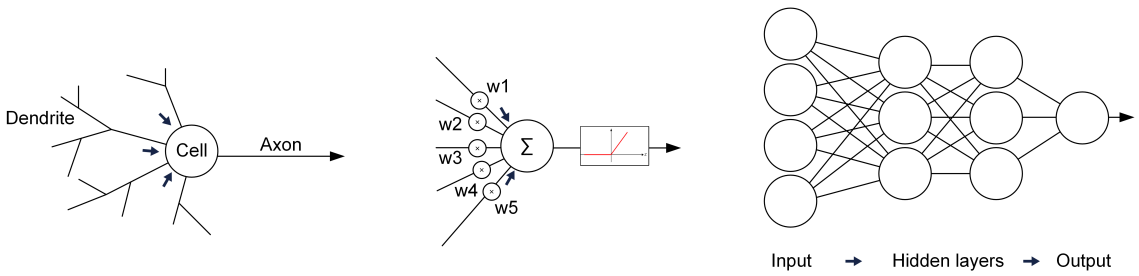
One well-known example of unsupervised clustering algorithms is the density-based spatial clustering of applications with noises (DBSCAN) algorithm [42], it examines all data points in the feature space, finds the points with a high density as the centroids of clusters, and allocates neighbouring points into the cluster. Points that are not close to any of the centroids will be regarded as noise. The clustering algorithm is useful in determining the number of

data sources and noise removal in radar signal processing. Another example is the mixture of Gaussian algorithm that fits the distribution of the data using a combination of Gaussian distribution models, and determine the possibility of incoming data belonging to the distribution. This algorithm has been used widely in removing noise and clutter from the data. A more detailed review of unsupervised learning models can be found in [43].

2.1.3 Neural Network

Neural networks, as an emerging supervised machine learning technique, have shown outstanding performance on various artificial intelligence tasks. They were first developed as a perceptron model for simulating the neurons in the human brain, and started to receive increasing popularity since the announcement of AlexNet [44], a convolutional neural network (CNN) for image classification that achieved significant improvement over traditional machine learning methods in object classification. Since then, many neural network architectures have been designed and implemented for various tasks.

Neurons, or artificial neurons, are the basic unit of a neural network. A neuron simulates the functionality of a biological neuron. It receives one or more inputs, modifies them with the neuron's internal states, applies certain non-linear transformations and sends out one output signal. Neural networks consist of a number of layers, each layer with a number of neurons connecting to each other, as shown in Figure 2.2.



(a) Biological neuron. (b) Artificial neuron. (c) Neural network.

Figure 2.2: Basic model of a neural network.

Neural networks are designed as universal function approximators to learn complex transformations from an input to a higher level abstraction. Mathematicians have proven that, with a combination of linear and non-linear functions, even single-layer neural networks can work as universal approximators [45, 46], though with a potentially very large number of neurons. With multi-layer neural networks, the number of neurons can be reduced to a feasible size for modern computers to process, while still being able to approximate complex functions for solving real-world problems.

Typical neural networks consist of an input layer, a number of hidden layers, and an output layer. The term deep learning is used to define neural networks with many hidden layers, though

the exact value to be qualified as deep is not strictly defined in the literature. The first deep neural network was designed by Hinton et al. [47] with three hidden layers. Therefore, many researchers refer to deep neural networks as networks with three or more hidden layers. However, the number of layers increased dramatically in the next few years due to the emergence of convolutional layers. For example, AlexNet [44] has eight layers, the VGG network [48] has more than 16 layers and the authors named it a very deep network, and the residual network [49] has up to 1200 hidden layers and the authors named it an extremely deep network. Although the author also claimed that the performance of the network hardly improves after 110 layers, an extension to their work by Huang et al. [50] showed that the 1200-layer network can still show meaningful improvement. Traditional machine learning methods often rely on effective feature extraction from the raw data, whereas a deep neural network can be applied directly to the raw data and extract features as part of its learning process.

2.1.3.1 Deep Neural Network Layers

This section provides the fundamental of deep neural networks required for this thesis, whereas more detailed information can be found in [51]. The dense layer, or fully connected layer, is one of the most important layers in deep neural networks. A dense layer contains a number of units, with each unit having independent weighted connections w to each of the inputs x and a bias b to be added to the output. The operation of a dense layer can be written as:

$$x_i = f(w \cdot x_{i-1} + b) \tag{2.1}$$

where x_i is the output of the layer, x_{i-1} is the output from the last layer and the input of the current layer, and f is the activation function. Dense layers usually have large numbers of trainable parameters. For example, to process an input with 100 features with a 10-unit dense layer, there will be 100 (features) $\times$ 10 (weights) + 10 (bias) trainable parameters. The output of a dense layer will have the number of features equal to the number of neurons in the layer. Dense layers consider all possible relationships between all input features and are strong at extracting hidden features.

The convolutional layer is another type of the most important layer in neural networks. Convolution is a well-known technique in signal processing, which defines the operation of applying a kernel to an input signal to extract a certain feature. Convolution can be performed on data with any dimension, though it is the most common on 1D (e.g. sound) and 2D signals (e.g. images). While dense layers extract all information from every possible combination of features, convolutional layers focus on features that are spatially close to each other. In image processing tasks, convolutional layers consider features in pixel blocks instead of arbitrary groups of pixels. This is especially helpful when extracting spatial features in images, such as edges, regular shapes and textures. Convolutional layers also reduce the computational cost significantly in comparison to a dense layer. For example, to process an $100 \times 100 \times 3$ RGB image with a $3 \times 3 \times 10$

convolution kernel, i.e. to learn 10 possible features by examining every 3×3 pixel block, there will be only $3 \times 3 \times 10 + 10$ (bias) trainable parameters and the output of the layer would be $98 \times 98 \times 10$ (without explicit padding and striding). The operation can be written in a similar way as the one for the dense layer, as:

$$x_i = f(w * x_{i-1} + b) \tag{2.2}$$

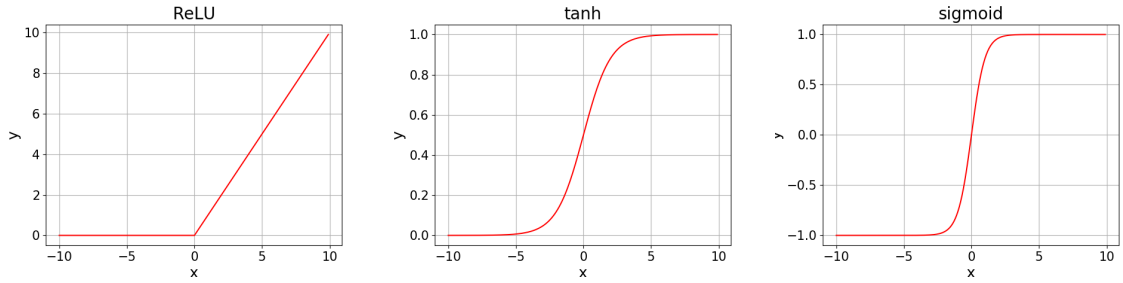
where $*$ denotes the convolution operation. The development of convolutional layers significantly improved the learning speed of neural networks and made them more suitable for graphic processing unit (GPU) acceleration, as the convolution operation can be processed in parallel.

There are also non-trainable layers that are designed for certain operations. For example, dropout layers randomly deactivate some neurons in a trainable layer to improve the generalizability of the network and avoid overfitting; Batch normalization layers normalize the data regarding a batch of data, to reduce the data variance caused by outliers from the previous layers or the input; Pooling layers compress the data from neighbouring locations and help the network to focus on certain spatial features. These functional layers help the network to extract features from input and train more efficiently.

Apart from the most common layers mentioned above, there are many other types of layers and architectures that are designed for different applications (as reviewed by Abiodun et al. [52]), such as the recurrent neural network which is designed to model temporal information within the data, and the graph neural network which is designed for graph-like or point cloud data. Some examples of their usage in HAR will be given in later sections.

2.1.3.2 Activation Function

While the dense layer and the convolution layer mentioned above perform linear operations on the input, activation functions are also an important element that performs non-linear operations on the data. These functions give networks the ability to perform non-linear transformation from input to output and are essential for the network to serve as a universal function approximator [45]. Activation functions are normally performed at the end of every layer and are applied to the intermediate result of the layer. They determine whether the output from the previous layer will be activated or discarded. There have been a number of activation functions proposed in the literature, a more detailed discussion of them can be found in Nwankpa et al. [53]. Three common activation layers, Rectified Linear Unit (ReLU), sigmoid and tanh are shown in Figure 2.3. The sigmoid function is one of the earliest activation functions that aims to map the output of a neuron to $[0, 1]$, but it has a non-zero mean and can become inefficient in training [53]. The tanh function helps address the problem as a zero mean function. However, they have been mostly replaced by the ReLU function and its variants, which have shown superior performance in many applications [53].



(a) ReLU. (b) sigmoid. (c) tanh.

Figure 2.3: Three typical activation functions.

2.1.3.3 Network Training

As a supervised machine learning model, the training step is very similar to that mentioned in Section 2.1.1. When a neural network is created, all trainable parameters in the layers are initialized randomly or following a user-defined pattern. The training is the process of modifying these parameters so that the network provides the best approximation from the input to the output. During each training step, the network takes a batch of data, applies the forward propagation, generates its prediction, compares it with the ground truth and calculates the cost accordingly using some cost functions.

The cost function plays the most important role in network training, as it defines how to evaluate the correctness of a model and how the network weights should be modified to approach the ground truth. The design of a cost function largely depends on the application. In a classification problem, the objective of the model is often to classify a data instance into a few possible categories, where the cross-entropy cost is one of the most popular choices that give a high cost when the model misclassifies the data [52]. Some example applications of this type of network include human identification, activity classification and fall detection.

Many applications do not require categorical output. For example, human tracking requires the coordinate of the person in a continuous space, where a mean-squared error or sum-squared error function is often used to measure the disparity between a prediction and the ground truth, which the network needs to minimize [52]. Another example would be face recognition, where a triplet loss function is used that assigns a low cost to faces that belong to the same person while assigning a high cost to faces that belong to different people [54]. Once the cost is calculated, it will be backpropagated through each layer of the network and the parameters will be updated using the gradient descent algorithm.

2.2 HAR with Machine Learning[1]

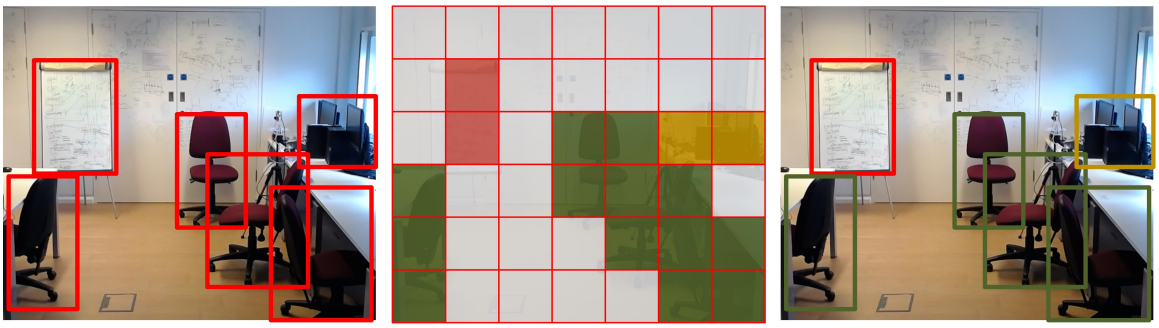
The main hardware required for data collection in a HAR system can be categorized into cameras, sensors, and wearable devices. While some researchers would refer to cameras as one kind of sensor, they are studied as a standalone subject in the context of this research, considering the significant difference in the data processing chain (DPC) between camera data and the others. Commonly used sensors include Doppler radars, radio frequency (RF) sensors and various environment sensors. They are often set at a fixed location and measure certain changes in the environment. Wearable devices are worn or carried by people and measure certain properties of the person or the environment. Some of them are single-purpose sensors, like GPS (Global Positioning System) or accelerometers, while others might have multiple functions. More complex wearable devices can contain one or more portable sensors, transceivers and processors integrated into a mobile platform, such as a mobile phone or smartwatch. This section reviews HAR systems based on the hardware used.

2.2.1 Cameras

Camera-based HAR has been studied in depth in computer vision, as reviewed by Poppe [55], Nguyen et al. [56] and Kong and Fu [57]. The data collection step is often simply collecting images or videos using the cameras. Pre-processing of the data, such as filtering and pixel normalization, is sometimes required to reduce the effect of inconsistent lighting conditions or camera artifacts, but varies a lot between different techniques.

Region selection is often an important aspect of a camera-based system. Since cameras record everything in the scene within the angle-of-view, it is necessary to reduce the input size and ideally only analyse areas around the object-of-interest. When there are multiple people in the scene doing different activities, it is also important to divide the image into subregions and analyse them independently. A sliding window algorithm divides the image into a number of rectangular subregions from different scales and analyses them in sequence. This is the simplest method and generates a regular and balanced workload, but the result often contains many irrelevant candidates and can result in a waste of resources. One advanced technique is foreground extraction, where the object-of-interest (the foreground) will be extracted from the scene using temporal information, such as the mixture of Gaussian model. A bounding box with a pre-defined size will be generated around the foreground for further processing. Another commonly used method is the segmentation-based method, where the image is segmented into irregular regions based on pixel features like edges or colours, which are then converted into regular shapes for the next steps. Regions generated with foreground extraction or segmentation often contain more precise features, but also require more processing power. An example of a region-based object classification algorithm using neural networks is shown in Figure 2.4.

[1] A condensed version of this section has been published in [20] ©2021 IEEE.



(a) Compute bounding boxes around potential objects.

(b) Divide the source image into a grid and compute the class probability distribution of each cell.

(c) Assign class labels to the bounding boxes.

Figure 2.4: Region-based object classification.

Once the candidate regions have been selected, the next step is to extract features within each region. Although with the emerging neural network techniques, feature extraction is often performed implicitly by the hidden layers in the network, many traditional machine learning techniques rely on features to be extracted explicitly before feeding into the classifier. Feature extraction algorithms on HAR include SIFT (scale invariant feature transform) [31], SURF (speeded up robust features) [32], HOG (histograms of oriented gradients) [33], Haar-like features for face detection [34], motion features extracted from temporal information [35], and many others as reviewed by Nixon and Aguado [36]. These algorithms convert the image input into various mathematical representations that can be handled easily by the classifier.

After the classification, some algorithms require an extra step to combine the results from different candidate regions. For example, if the algorithm uses a sliding window, results from overlapping regions will need to be combined to avoid repeat detection. Algorithms using different candidate regions for different purposes will also need to combine these results. For example, Chen and Yuille [58] performed upper-body and lower-body detection separately and combined the result for full posture recognition.

There has been much research focusing on the detection of a person in a scene. For example, Dalal and Triggs [33] applied SVM on HOG features for human detection. Viola and Jones [34] used AdaBoost (Adaptive Boosting) on Haar features for face detection. There is also research that assumes the presence of the person and focuses on the recognition of activities. For example, Dhulavvagol and Kundur [59] applied SVM with SIFT feature extractors for classifying a set of actions. Manosha Chathuramali and Rodrigo [60] applied SVM on spatial-temporal data from videos for action classification and claimed 100% recognition rate on public datasets. Nasution and Emmanuel [61] applied k-nearest neighbour (KNN) on image histograms for an elderly monitoring system and claimed over 90% accuracy on five postures.

Beyond human detection and activity classification, there are many other systems designed

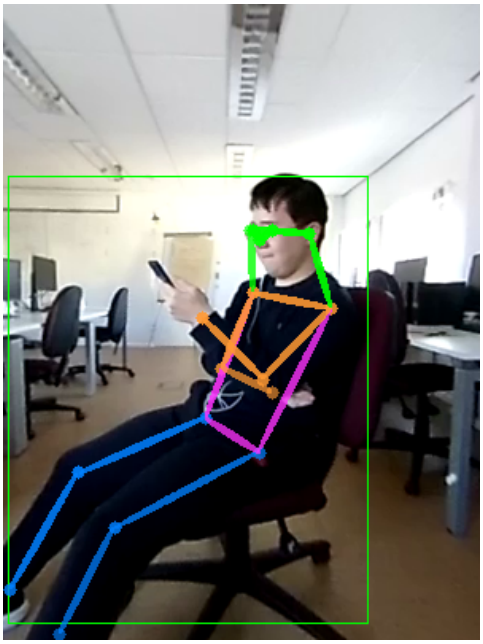


Figure 2.5: Posture estimation using HRNet.

for advanced tasks. Examples of such systems include face recognition using principal component analysis (PCA) [62] or neural networks [54], and posture recognition (or pose estimation) using a tree-based model [63] or neural networks [64–66]. One example of a posture recognition system is shown in Figure 2.5 (using the algorithm from Sun et al. [66]). These examples do not provide a direct classification of human activities, but they provide powerful tools for building an advanced HAR system.

Depth cameras, also known as 3D cameras, are specialized cameras that provide distance information from the camera to the object, in addition to the optical image. One type of depth camera is the stereo camera. A stereo camera has two camera lenses separated by a small distance. It captures two images at once, and the disparity between the left and right images can be used to estimate the depth information of the scene. Output from stereo cameras is often in the format of a dense depth map. Stereo cameras are low cost, but the disparity calculation algorithm often requires a significant amount of processing power [67]. The other type is a normal camera with depth sensors, where the depth information is captured either using time-of-flight or structured-light and can be represented as either depth maps or point clouds. One example of this kind of depth camera is the Kinect [68]. The 3D information captured by the depth cameras led to the study of 3D point-cloud-based HAR (reviewed by Aggarwal and Xia [69]) or skeleton-based HAR (reviewed by Han et al. [70]). These techniques share similar advantages and disadvantages with the normal camera-based methods. They provide dense information about the scene and have relatively low cost, but they are intrusive, and their performance is limited by the viewing condition.

2.2.2 Sensors

Sensors capture information from the environment other than the vision. Contact sensors require physical contact with people to measure certain biomedical properties, whereas non-contact sensors use certain signals, like RF signals, to sense the environment. One major difference between these data and the vision data is that, while the temporal dimension in vision data is supplementary, it is often more important in sensor data for a sensible interpretation. Some of the most common sensors for HAR are summarized as follows:

Doppler radars are designed for detecting the Doppler motion of the subject. They use various frequencies for different applications, from a few megahertz to a few hundred gigahertz. Researchers have used Doppler radars for classification of simple actions [71], motion detection [72], gait analysis [73], and vital signs detection [74].

Radio frequency sensors sense the environment with RF signals at different frequencies. For example, WiFi systems use RF signals at 2.4 GHz and use multiplexing schemes to divide the available bandwidth into channels, where any change in the environment will be encoded in the channel state information as the signal propagates, such as the amplitude and the phase of each channel [75]. One common approach with RF sensors is to set up one or multiple signal transceivers at fixed locations in the environment, where people in the environment will reflect signals with different strengths and properties, based on their distance to the transceivers. Recent research has also shown the possibility of detecting the human pose [76] and generating a skeleton model [70] based on RF signals.

LIDAR sensors use the time-of-flight of laser pulses to measure the distance between an object and the sensor. LIDAR measurements can have extremely high accuracy and robustness and are often used in the fields of geology, astronomy and agriculture. As the cost of LIDARs decreases, it has also been used for automotive driving and robotics. There are also researchers studying its use in human detection and classification [77]. However, the cost of LIDARs is still much higher than many of the other sensors, preventing them from being used more widely.

Ultrasonic sensors detect human activities through ultrasound. Passive ultrasonic sensors listen to sound in the environment and extract useful information, whereas active ultrasonic sensors transmit modulated ultrasonic signals and analyse the received signals for action recognition. For example, Qi et al. [78] presented a gait analysis system that used a few passive ultrasonic sensors to determine the location of a moving person.

Biomedical sensors measure the biomedical activities of a person and provide an indication of their health status and activity status. They often require physical contact with the subject and are more commonly seen in clinical environments and hospitals. Some examples include the electrocardiogram (ECG) sensors for detecting the heart's electrical activity, and photoplethysmogram (PPG) sensors for detecting the blood volume change in the microvascular and, hence, the heart rate or blood pressure of the subject. These sensors can be used for daily health monitoring and the detection of diseases or abnormal activities. For example, Butt et al. [79] used ECG

sensors for fall detection.

Environment sensors include temperature sensors, thermal sensors, audio sensors, light sensors, humidity sensors, pressure sensors and others. They measure various information about the environment as their name states and are often used in conjunction with other sensors for HAR tasks (like [80]).

While each of these sensors can give detection on one aspect, a combination of them is often required to gather multidimensional data and hence obtain enough information for complex HAR tasks. Data can be combined either at the raw-data level when the same type of sensors are used, or at the decision level when each type of sensor requires a separate DPC [81].

When merging data from different sensors, it is often required to transfer all the data to a central processor for processing. The data transmission can be carried through a wire or wirelessly. Wireless transmission can be made through commercial communication protocols such as Bluetooth and WiFi, or customized short-range communication protocols [82]. For example, van Kasteren et al. [83] used a combination of infrared sensors, pressure mats and contact sensors for data collection and a wireless network for data communication. They collected data in three houses under daily conditions and evaluated a set of classification algorithms. A recurrent neural network-based approach using the same dataset was also published later [84].

2.2.3 Wearable Devices

While sensors have to be fixed at the point of interest, have a certain range of view and do not provide any information if the human is out of the region, wearable devices need to be worn or carried by people and contain sensors that provide continuous information about their activity. Apart from the data collected, the DPC with wearable devices is very similar to those of sensors. One difference is that wearable devices often require wireless transceivers to transfer the data to a central processor or have an embedded processor for processing the data and providing real-time feedback to the user. The processing power of embedded platforms is often constrained by power consumption and thermal dissipation. Common wearable devices for HAR can be summarized as follows:

Accelerometers and Gyroscopes measure the acceleration and the orientation of the device in the x-y-z dimension. A combination of the two is also referred to as an inertial measurement unit (IMU). These sensors have shown success on various tasks, such as movement detection [85], activities classification [86] and fall detection [87, 88].

GPS measures the geometrical position of a person. A commercial GPS can estimate the location of a person with a resolution of a few meters, and is often used in conjunction with other sensors. For example, the fall detection system designed by Wu et al. [88] used accelerometers for fall detection and a GPS for the location of the fall.

Portable biomedical sensors measure certain biomedical activities of a human, such as the heart rate, breathing rate and blood oxygen saturation levels. They are often in the form of a

smartwatch, chest band or wristband. Although they may not provide professional information as those for clinical uses, they have a much lower cost and higher convenience and are often used for personal health care.

Portable LIDAR systems are similar to fixed LIDAR systems but embedded into mobile platforms. They provide high-resolution information about the scene at a high cost. One of their most common usages in HAR is face recognition, as sometimes can be seen on high-end mobile phones.

Portable environment sensors provide information about the environment, as mentioned in the last section.

Again, each type of these sensors provides information from one aspect, and data fusion between multiple sensors is often required to provide multidimensional data for HAR tasks. For example, Kantoch [89] proposed a health monitoring system using a mixture of ECG sensors, temperature sensors and accelerometers, for monitoring physiological data during different activities.

In many cases, wearing multiple sensors can be inconvenient and impractical. This encourages the development of embedded wearable devices. Such devices are dedicated to having a few selected sensors, transceivers and processors, forming a portable HAR system with acceptable power consumption. For example, Wu et al. [88] designed a fall detection device with accelerometers and GPS modules. Maurer et al. [90] designed a smartwatch that contains multiple different sensors for activity monitoring.

With the development of embedding processing and integrated sensors, mobile phones have received lots of attention as wearable devices for HAR. Mobile phones have many integrated sensors, among which the IMU and the GPS are the most common, providing the motion information and location of the subject, respectively. In addition, modern mobile phones have strong processors, including CPUs (central processing units), GPUs and DSPs (digital signal processors), allowing the sensor data to be processed in real-time. The built-in network modules also allow data to be transferred for remote processing or cloud processing. Therefore, using mobile phones for HAR has become an emerging research topic, as reviewed by Su et al. [91]. For example, Khan et al. [92] designed a HAR system using accelerometers, pressure sensors and microphones on a mobile phone, demonstrated both an online processing approach and an offline approach, and showed that it can classify between 15 activities.

There are also public datasets available for wearable devices based HAR, where researchers ask a number of testers to bring the devices and perform activities in certain environments, collect the data from the sensors and label them manually [93–95]. These datasets often contain data measured from IMUs and a selection of other common sensors, and allow much theoretical work [96–98] to be developed without repeat data collection. Other reviews on wearable device based HAR can be found from Lara and Labrador [99], Mukhopadhyay [100] and Su et al. [91].

2.2.4 Multidimensional Data Fusion

While each type of the discussed hardware gives certain information from certain aspects, fusing them together enables the full potential of HAR systems to be explored. To give a few examples, Brdiczka et al. [101] used cameras, a set of audio sensors and the Hidden Markov Model for HAR at home. Gharghan et al. [102] presented a fall detection system using both wearable devices and environment sensors. Fotiadis et al. [103] used laser sensors and cameras for human detection. Huang et al. [104] fused mmWave radars and cameras for tracking moving subjects. Ulrich et al. [105] fused mmWave radars and cameras for estimating the size of subjects. Zouba et al. [80] used multiple environment sensors and cameras to monitor elderly activities at home. Chen et al. [106] reviewed HAR systems using depth sensors and inertial sensors.

The term "smart home" is used to refer to a home environment equipped with various sensors and interactive devices. They are designed to be able to monitor human activity at home and provide appropriate services at their best convenience. Common techniques in a smart home are reviewed by Demiris and Hensel [15]. Smart homes are originally designed for health and safety monitoring of elderly, disabled people or people under medical treatment, but they are tending to receive increased popularity among the general public due to the development of ubiquitous sensors and HAR systems.

2.3 Human Posture Estimation [2]

One of the main interests of this research is human posture estimation. The topic has been studied in depth in computer vision based on camera data. Postures are often represented by the position of a few key joints of the person. Many researchers use a two-phase approach: using local pixel features to estimate the positions of the joints independently, and then using spatial models to encode the correlation between the joints and refine the estimate. Traditional methods often rely on handcrafted models of these joints and require manual feature extraction, such as the pictorial structure [107, 108], hierarchical model [109] and deformable model [110]. While early work is restricted by the effectiveness of the tree-based models, researchers have proposed many improved methods, such as using strong part detectors [111] and using a mixture of models [63]. However, although these approaches improved the generalizability and robustness of the model, they have been outperformed by the neural network-based methods since their emergence [112].

In contrast to the traditional methods, neural network-based methods often have a strong ability to extract features from the input data, which relaxes the requirement of handcrafted features and models. Therefore, many researchers have adopted neural network models for posture estimation and have demonstrated significantly higher accuracy compared with the traditional methods [64, 66, 112, 113]. They are shown to be more effective in dealing with a complex environment and less common postures, and have achieved the top accuracy in many

[2] This section has been published in [21] ©2022 IEEE

datasets. The work from Toshev and Szegedy [112] is one of the first neural network-based posture estimation systems in the literature and has 7 network layers, whereas later work, like Sun et al. [66], have increased the number of layers to hundreds and achieved the state-of-the-art accuracy (78.2%) in public datasets.

These models often use fixed-size images as the input to the neural network and use the x-y coordinates of the joints as the ground truth to train the model. Tompson et al. [113] proposed a spatial model as a part of the neural network model to encode the relationship between the joints as a Markov Random Field (MRF) model and trained it using the same backpropagation algorithm as a typical CNN. In addition to estimating the posture of a single person, CNN models are also shown to be able to estimate postures for multiple people in one scene concurrently [65]. These techniques eliminate the requirement of locating the person in an image before estimating the posture.

Due to the various disadvantages of cameras, such as privacy concerns, researchers are also investigating alternative techniques for posture estimation using various kinds of sensors [114]. For example, Zhao et al. [115] used extensive radio-frequency antenna arrays to detect human posture. There are a few very recent studies on posture estimation using mmWave radars [116–119], but many only focused on a limited set of standing or walking postures. The performance of these models may degrade on unseen postures. For example, a sitting posture has a significantly different joint structure from a standing posture. Such situations require a robust and generalizable model. Therefore, this research aims to estimate arbitrary human postures commonly seen in an indoor environment (Chapter 6).

2.4 Human Vital Sign Detection

Vital sign detection can be an effective tool for monitoring people's health status and early detection of cardiovascular disease. Non-contact vital sign detection using sensors has been studied for decades and many systems have been proposed, as reviewed in [120–122]. While most early research was in the lower frequency band at sub-24 GHz, researchers are moving towards mmWave at higher than 60 GHz, which has several advantages, such as a higher resolution, a higher sensitivity to small movement and a smaller antenna size [120].

The existing work using mmWave radars to detect people's heart rates can be split into three categories: targeting stationary people, targeting people with restricted movement or slow and regular walking, and targeting people with free body movement. When a person is stationary, a phase signal of the chest movement can be easily constructed, and the frequency of the heart pulses can be computed through spectrum analysis. There are many works following this standard workflow presented for different scenarios, such as [16, 123–126], often with a low error rate below 5 beats per minute. However, the phase signal can be distorted and noisy when the person moves, as the phase change caused by movement is significantly higher than the displacement

due to heart pulses and makes phase extraction more difficult [122, 127]. There are studies that target restricted movement in an office environment that involves simple movement from the limbs and legs [128, 129], or walking in straight lines at a low speed of maximum 25 cm/s [130, 131]. These methods use certain signal processing techniques, like wavelet transform, to look for signal periods that are less noisy and then estimate the HR. However, in situations like exercising, the noise can be high for a long period of time and these methods may fail. Some researchers proposed to use cameras with radars to track the person [132–134], but are constrained by having privacy concerns and requiring good lighting conditions. Wang et al. [135] used two radars to cancel out the Doppler effect of the movement and obtain the heart rate of a person jogging. However, it requires explicit synchronization between the radars and can only measure the subject in the middle of the two radars. In addition, only one set of data was given in the original paper. Gong et al. [127] used a neural network to learn the heart rate of the person from the motion information captured by the radar, but requires a re-training for each unseen person using their static heart rate, as well as a long processing time (5 seconds) on a GPU. This research aims at estimating the heart rate of a person when exercising on a treadmill with real-time operation (Chapter 7).

2.5 Conclusion

This chapter discusses the fundamentals of machine learning that can be used for HAR. Machine learning tasks can be categorized as supervised learning and unsupervised learning, the former requires a training stage with ground truth data and can be used for various regression or classification tasks, whereas the latter is often used for data analysis and clustering. In this chapter, related work in HAR, categorized based on the use of cameras, sensors and wearable devices, have been reviewed. These systems have different applications and use cases. Camera-based systems, due to the high amount of information that can be captured by images, often achieve the top performance in HAR tasks. However, they have the disadvantages of being intrusive and relying on lighting conditions. Sensor-based systems use non-vision signals to capture information about the scene. They often have a fixed field of view and only capture anonymous data about the person, and are receiving increased popularity in the industry. Wearable devices can be worn or carried by the subject and are not limited by the field of view, but can be inconvenient and have a limited use case. Without the use of cameras, HAR requires a lot of data fusion between sensors for complex tasks, which increases the cost and the setup complexity of the system. There lacks one low-cost, multifunctional, real-time system that is capable of complex HAR tasks.

CHAPTER 3

mmWAVE RADARS FUNDAMENTALS

This chapter introduces preliminary knowledge of frequency-modulated continuous-wave (FMCW) mmWave radars, with a particular focus on the radars manufactured by Texas Instruments (TI). Section 3.1 reviews mmWave radar-based HAR systems in the literature. Section 3.2 discusses the typical FMCW mmWave radar model and explains how the distance, velocity, and angle of objects in the scene can be detected by a radar. Section 3.3 introduces the radar models made by TI, the data processing chain (DPC) implementation, host-device communication and data transfer.

3.1 Millimetre-wave Sensing[1]

mmWave is an electromagnetic wave with a frequency between 30 GHz and 300 GHz, and a wavelength of 1 mm to 10 mm. Signals at this high frequency were not explored until the last decades, when people started looking at 5G communication and mmWave radars. The largest benefit at this frequency is the available bandwidth. For radar and sensor usage, there is an industrial, scientific and medical (ISM) radio band at 24 GHz, but it only has a 250 MHz bandwidth. There is a UWB of 5 GHz from 21.65 GHz to 26.65 GHz GHz available for radars, but its usage will be phased out by 2022 in both Europe and the US [136, 137]. Currently, many industries are looking at mmWave sensors 76 GHz to 81 GHz [138] or around 60 GHz (e.g. 57 GHz to 64 GHz following European regulations [139]). The high bandwidth allows a potentially very high resolution for object detection, and the shorter wavelength allows a much smaller antenna size and allows the signal to penetrate through thin materials.

[1] Some content of this section has been published in [20] ©2021 IEEE.

mmWave radars used in different industries would have different types of antennas. Long-range radars often use large antennas, such as horn antennas and Yagi antennas, whereas short-range radars (within a few hundred metres) often use integrated patch antennas, due to the advantage of the much lower cost. Most mmWave radars are the frequency-modulated continuous-wave (FMCW) radars. These radars have at least one pair of a transmitter and receiver. The transmitter sends out a continuous radio frequency wave and the receiver receives the reflected signal. A DPC is performed on the two signals in order to extract information from the scene. Non-modulated continuous wave radars act as Doppler radars, which are only able to detect changes in the signal frequency caused by the Doppler shift, i.e. only being able to detect moving objects. FMCW radars modulate the frequency of the transmitted wave and are able to extract more information from the received signal [140]. There are a few different modulation schemes with different characterizations, where the most common modulation scheme is using the chirp signal, whose frequency is set to increase linearly during a certain period of time and the frequency and phase difference between the received signal and the transmitted signal will contain the information about the scene. A set of chirp signals forms a frame and allows continuous sensing of the environment. More details on the FMCW radar theorem will be given in Section 3.2.

Apart from FMCW radars, there are also mmWave sensors designed for active imaging. This type of radar often uses antenna arrays to scan objects in a certain area. In other words, the radar emits a signal and measures the received signal at every possible position, in a similar way to an optical camera. The resultant image can be a dense depiction of the scene and can be used for complex tasks like object classification [141, 142]. There are also sweeping radars, where the transmitters and the receivers move and/or rotate during the operation to obtain a larger field-of-view [143]. These radars achieve good performance in certain applications when the environment is known, like security checks at airports, but might not be suitable in general purpose applications where the environment can be arbitrary and the cost needs to be kept low.

The data processing of mmWave signal can be categorized as signal-level processing and data-level processing. Signal-level processing refers to the low-level signal processing techniques applied to the digital signals after the analogue to digital converter (ADC), such as signal filters and FFTs. These techniques are designed to extract the information-of-interest from the raw data and convert it to a higher-level representation that can be better understood by the user. For example, as will be shown in Chapter 3, FFTs can be used to process the raw data and construct a point cloud that represents the scene. Data-level processing refers to the high-level data processing techniques that are related to the desired applications, such as human detection and posture estimation based on the constructed point cloud.

A market research on commercial mmWave radars has been carried out and the result is shown in Table 3.1. The majority of the manufacturers focus on automotive applications and do not provide technical specifications for their products. Only a few of them are developing

Table 3.1: Main mmWave radar manufacturers and the frequency they use.

Company	**Frequency Range**	**Number of Antennas**
Ainstein	60-64 GHz and 76-81 GHz	3x4
Bosch	76-77 GHz	2
Continental Engineering	77 GHz	Not available
Denso	76-77 GHz	1x1
Infineon	60-66 GHz and 77 GHz	1x1 and 3x4
MediaTek	76-81 GHz	1x1
Metawave Corporation	77 GHz	Not available
NXP	76-81 GHz	3x4
Panasonic	79 GHz	Not available
Smartmicro	60-64 and 76-81 GHz	Not available
ST Microelectronics	76-81 GHz	3x4
Texas Instruments	76-81 GHz and 60-64 GHz	3x4
Uhnder	77 GHz	12x16
Vayyar (Minicircuits)	60 GHz and 79 GHz	20x20, up to 72
Veoneer	77 GHz	Not available
ZF Friedrichshafen AG	77 GHz	Not available

industrial and personal-use radars with complete datasheets and technical support, such as TI and Minicircuits, and are preferable to researchers.

mmWave radars have been used widely in autonomous driving due to their ability to detect the distance, velocity, and angle-of-arrival (AoA) of objects. Autonomous driving often uses short-range radars and signal processing techniques for understanding the road environment, as reviewed by Patole et al. [144] and Bengler et al. [145]. mmWave radars are shown to be able to perform many tasks in automotive driving, such as car tracking [146], traffic monitoring [147], navigation [148], and collision avoidance [149]. Data fusion between mmWave radars and other sensors has also been proposed. For example, Streubel and Yang [150] used mmWave radars and stereo cameras for pedestrian detection on the road. Kim et al. [151] used mmWave radars and infrared cameras for navigation in a strong smoking environment.

The use of mmWave is also receiving a lot of popularity in HAR. Yang et al. [124] used mmWave signals to detect the heart rate and breathing pattern of a person by analysing the variation of the signal strength reflected from the chest and achieved a low error of 0.43 breaths per minute and 2.15 beats per minute, respectively. Kianoush et al. [152] used separate mmWave transmitters and receivers, and presented a passive detection system of human movement between them at over 95% accuracy. When the person walks between the transmitters and receivers, the system records the signal signature and uses a neural network to determine the motion. Lien et al. [153] used mmWave radars for hand gesture recognition at a close distance. It uses mmWave signals to capture the motion of the hand and uses a neural network to classify between a set of pre-defined gestures, which achieved 92.1% accuracy. Björklund et al. [72] used mmWave radar as a Doppler radar to detect and classify human movement. However, most of

the research only used the Doppler signal or the received signal strength (RSS) in their systems. Only a few researchers have utilized the high bandwidth and, hence, the high range resolution of mmWave radars. There are only a few researchers who have used mmWave radar as a 3D sensor for HAR. For example, Singh et al. [154] used a neural network to process the point cloud captured by the radar and achieved 90.5% accuracy when classifying five actions. Similarly, Zhang and Cao [155] presented a system for classifying four actions and achieved 95.2% accuracy. These systems often use a classification model to classify between a set of pre-defined activities and are not generalizable to other unseen activities, which can be critical in real-world applications. Zhao et al. [156] used a neural network for people identification and tracking and achieved 96% accuracy when identifying 12 people. However, it had a localization error of 16 cm which, as will be shown in Chapter 5, can still be improved significantly. Meanwhile, although deep neural networks have shown outstanding performances in many tasks, their high computational cost makes real-time operation a concern.

The geometry information provided by the radar allows a point-cloud-based classification to be performed. Accurate 3D point-cloud data are often only obtainable from expensive sensors such as LIDAR systems. Although the data from mmWave sensors are much less accurate and sparse, their lower cost allows the use of several for reliable detection.

In this thesis, the use of mmWave radars for complex HAR in indoor environments is proposed. mmWave radars from TI were chosen, as they have a rather matured product line, detailed documentation and a good supporting community on their products.

3.2 FMCW mmWave Radar Preliminaries[2]

Radars can be categorized as pulse radars or continuous wave (CW) radars, depending on the length of each signal transmission [157]. Pulse radars can be used to detect the distance of the object based on the received signal strength and the time-of-flight, whereas CW radars can encode more information in the frequency and phase of the signal. One of the most common types of CW radars is the FMCW radars, where the frequency of the transmitted signal is modulated as a function of time. Commercial mmWave radars are generally FMCW radars. The radar sends a modulated signal, detects the signal reflection from any object, processes the signal and determines the range, velocity and angle-of-arrival (AoA) of the object. A typical frequency modulation scheme used in FMCW radars is to increase the frequency of the signal linearly with time. Since the reflected signal is a time-delayed version of the transmitted signal, the two signals will have a constant difference in the frequency domain that is determined by the time-of-flight and, hence, the distance between the object and the radar. The velocity and the AoA can be determined by the phase information in the reflected signal.

[2]A condensed version of this section has been published in [20] ©2021 IEEE.

3.2.1 Intermediate Frequency Signal

The transmitter sends a chirp signal S_{tx} (a signal with frequency increasing linearly with time) to detect any object in front of the radar. When S_{tx} is reflected by the object, the signal is received as S_{rx}. Assuming the signal has an initial frequency f_0 and a slope of S, then the frequency of S_{tx} is a function of t:

$$f_{tx}(t) = f_0 + S \cdot t \tag{3.1}$$

The instantaneous phase of the signal is a function of t and is the integral of f_{tx}:

$$\begin{aligned}
\phi_{tx}(t) &= \int_{\tau=0}^{t} 2\pi \cdot f_{tx}(\tau)\, d\tau \\
&= \int_{\tau=0}^{t} 2\pi \cdot (f_0 + S \cdot \tau)\, d\tau \\
&= 2\pi \cdot f_0 \cdot t + \int_{\tau=0}^{t} 2\pi \cdot S \cdot \tau\, d\tau \\
&= 2\pi \cdot f_0 \cdot t + 2\pi \cdot \frac{1}{2} S \cdot t^2 \\
&= 2\pi \cdot f_0 \cdot t + \pi \cdot S \cdot t^2
\end{aligned} \tag{3.2}$$

The transmitted signal S_{tx} can be written as a sinusoid signal:

$$S_{tx}(t) = A \cdot cos(2\pi f_0 t + \pi S t^2) \tag{3.3}$$

where A is the transmission power. The received signal is a delayed and downscaled version of S_{tx}:

$$S_{rx}(t) = \alpha A \cdot cos\big(2\pi f_0 (t-\tau) + \pi S (t-\tau)^2\big) \tag{3.4}$$

where τ is the time-of-flight of the signal and indicates the distance of the object, and α is the downscale factor that models the transmission loss. The two signals, S_{tx} and S_{rx}, are combined through a mixer (a multiplier) to generate one signal with both the sum frequency and the difference frequency. The trigonometric identities state that:

$$cos(X_1 \pm X_2) = cos(X_1)cos(X_2) \mp sin(X_1)sin(X_2) \tag{3.5}$$

which gives:

$$cos(X_1)cos(X_2) = \frac{1}{2}\big(cos(X_1 + X_2) + cos(X_1 - X_2)\big) \tag{3.6}$$

Applying Equation (3.6) to $S_{tx} \cdot S_{rx}$ gives:

$$\begin{aligned}
S_{tx}(t) \cdot S_{rx}(t) &= \frac{\alpha A^2}{2}\Big(cos\big((2\pi f_0 t + \pi S t^2) + (2\pi f_0 (t-\tau) + \pi S (t-\tau)^2)\big) + \\
&\qquad cos\big((2\pi f_0 t + \pi S t^2) - (2\pi f_0 (t-\tau) + \pi S (t-\tau)^2)\big)\Big) \\
&= \frac{\alpha A^2}{2}\Big(cos\big(2\pi(2f_0 - S\tau)t + 2\pi S t^2 + \pi S \tau^2 - 2\pi f_0 \tau\big) + \\
&\qquad cos\big(2\pi(S\tau)t + 2\pi f_0 \tau - \pi S \tau^2\big)\Big)
\end{aligned} \tag{3.7}$$

There are two cos terms in the result. The first one has a frequency of $2f_0$ and will be removed by a low pass filter. The second one is the called the intermediate frequency (IF) signal or the beat frequency. The IF signal has the equation:

$$IF(t) = B \cdot cos\big(2\pi(S\tau)t + 2\pi f_0 \tau - \pi S\tau^2\big) \tag{3.8}$$

where $B = \frac{\alpha A^2}{2}$. The signal has a frequency $S\tau$, i.e. the slope of the chirp multiplied by the time-of-flight. Therefore, the frequency of the IF signal is directly proportional to the time-of-flight. Given that the slope of the chirp S is known, the distance of the object can be calculated from the frequency of the IF signal. The phase of the IF signal, $(2\pi f_0 \tau - \pi S\tau^2)$, can be simplified to $(2\pi f_0 \tau)$, as the second term is negligible: S has an order of 10^{12}, τ has an order of 10^{-8}, so $(\pi S\tau^2)$ will have a negligible order of 10^{-4}. To interpret the phase term better, $(2\pi f_0 \tau)$ can be rewritten to $(4\pi d/\lambda_0)$ by substituting $\tau = 2d/c$ and $f_0 = c/\lambda_0$, where λ_0 is the wavelength of the mmWave signal at frequency f_0. In summary, the IF signal can be written as:

$$IF(t) = B \cdot cos(\omega_b t + \phi_b) \tag{3.9}$$

where the angular frequency ω_b and the phase of the signal ϕ_b are:

$$\omega_b = 2\pi \cdot S\tau, \quad \phi_b = 2\pi f_0 \tau = \frac{4\pi d}{\lambda_0} \tag{3.10}$$

The frequency in Hertz can be derived from the angular frequency as $f_b = \frac{\omega_b}{2\pi}$. The above equations assume that the object is stationary. If the object is moving, the time-of-flight τ will be varying with respect to t. However, considering that this variation is limited within a single chirp time, it is unlikely to produce a big change in the frequency. For example, a person moving at 1 m/s will give a frequency change of around 10 Hz, which is negligible as the frequency of the IF signal is often higher than 1 MHz. The change in phase can be more significant, but will only affect certain applications where the phase information is critical, such as vital sign monitoring. In such cases, the phase can be written as a function of t as $\phi_b(t) = \frac{4\pi d(t)}{\lambda_0}$, where $d(t)$ describes the displacement of the object during the chirp time.

Note that the TI mmWave radar uses a complex band architecture. It uses a complex mixer (an IQ mixer) to multiply the two signals S_{tx} and S_{rx}, which has several advantages like a lower noise figure, as discussed in [158]. When in complex form, the IF signal in Equation (3.9) can be written as:

$$IF(t) = B \cdot e^{j(\omega_b t + \phi_b)} \tag{3.11}$$

which has the same frequency and phase as in Equation (3.10). After obtaining the IF signal, a DPC will be applied to determine the presence of any object, as will be explained in the following sections.

3.2.2 Distance Calculation

For a single object, the frequency difference between the transmitted signal and received signal will be a constant value f_b. Figure 3.1 shows an example chirp signal and how an object can be represented as a constant frequency component in the IF signal. This frequency is equal to $S \times \tau$, where S is the slope of the chirp and τ is the time-of-flight. Let d denote the distance between the radar and the objects, then the time-of-flight can be expressed as $\tau = 2d/c$. Therefore, d can be estimated as shown below:

$$f_b = \frac{2d}{c} \cdot S \tag{3.12}$$

$$d = \frac{f_b c}{2S} \tag{3.13}$$

where f_b is the frequency of the IF signal in Equation (3.8). This frequency can be found by applying an FFT to the IF signal, known as the range-FFT.

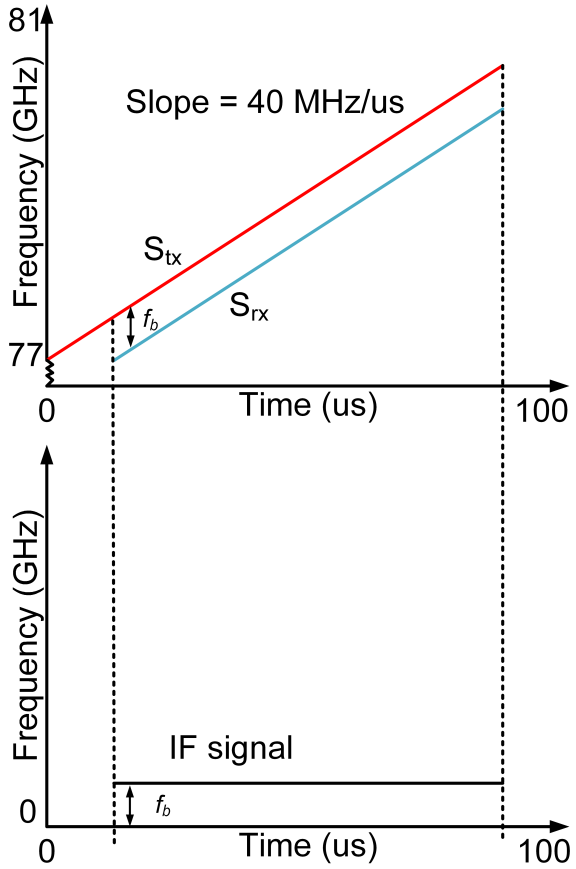


Figure 3.1: Example of the chirp signal and the IF signal of the radar.

3.2.2.1 Maximum Distance

The maximum distance that a radar can detect is limited by several factors. Some examples include:

- The power of the transmitters and the antenna gain of the receivers.
- The signal-to-noise ratio (SNR) in the environment.
- The reflectivity and cross-section area of the objects.
- The ADC sampling rate.
- The memory size for storing the ADC samples.
- The maximum allowed frequency of the IF signal.
- The slope of the chirp.

Some of these factors are limited by the environment and the hardware, as can be modelled by the well-known radar range equation [159]:

$$d_{max} = \sqrt[4]{\frac{\sigma P_{tx} G_{tx} G_{rx} \lambda^2}{(4\pi)^3 L P_{rx}}} \tag{3.14}$$

where d_{max} is the maximum detectable range, P_{tx} is the transmission power, G_{tx} and G_{rx} are the antenna gains of the transmitter and receiver, λ is the wavelength, L is the loss of the entire system, and P_{rx} is the minimum power required for the receiver to receive the signal.

On the other hand, the frequency of the IF signal and the slope of the chirp can be adjusted by the user to fit different applications. The maximum IF frequency depends on the signal processing chain. For example, for the IWR1443 mmWave radar from TI [160], the typical ADC sampling rate is 37.5 MHz, which limits the maximum IF frequency to 18.75 MHz according to the Nyquist sampling theorem. The digital filter uses an FIR filter with an 80% passband, and therefore the maximum IF frequency is 15 MHz. On the other hand, while the slope of the chirp can be configured within a certain range, its value affects many other factors. For example, a slower chirp could increase the distance of detection, but would also increase the memory required to store the processed signal and increase the time required to finish one chirp. For another example, when setting the IF frequency to 15 MHz and the slope of the chirp to S = 100 MHz/us, according to Equation (3.13), the maximum distance that can be measured will be 45 metres. However, considering the requirement of other measurements like the resolution, the maximum distance is often set much lower. As the scope of this research is in indoor environments, the range is typically configured to be within 10 metres.

3.2.2.2 Distance Resolution

In order to distinguish two objects that are close to each other, according to the Fourier Transform Theory, it is required that the frequencies of the two IF signals representing the two objects should be greater than the reciprocal of the signal period. Let T denote the length of the IF signal (roughly equal to the signal transmission time), the two objects can only be distinguished if the following equation is true:

$$\begin{aligned} \frac{2d}{c} \cdot S &> \frac{1}{T} \\ d &> \frac{c}{2ST} \\ d &> \frac{c}{2B} \end{aligned} \tag{3.15}$$

The minimal value of d is the distance resolution, i.e. the minimal distance required to distinguish two objects. It can be seen that the distance resolution solely depends on the available bandwidth of the radar. The larger the bandwidth, the smaller (better) distance resolution. In practice, TI mmWave radars use up to 4 GHz bandwidth and have a distance resolution of around 4 cm.

3.2.3 Velocity Calculation

In order to measure the velocity, the radar transmits a chirp every T_c seconds, and computes the phase differences between them. Assuming an object is moving at velocity v, then the displacement of the object between any two successive chirps would be $T_c \cdot v$. In other words, the second chirp signal would have travelled an additional distance of $\Delta d = 2 \cdot T_c \cdot v$ in comparison to the first chirp. Based on Equation (3.10), the phase of a sinusoid before and after travelling a distance Δd can be written as $2\pi f \tau$ and $2\pi f(\tau + \frac{\Delta d}{c})$, respectively, and the phase shift would be:

$$\Delta\phi = 2\pi f(\tau + \frac{\Delta d}{c}) - 2\pi f \tau = 2\pi f_0 \frac{\Delta d}{c} = 2\pi \frac{\Delta d}{\lambda} \tag{3.16}$$

Therefore, the phase shift between two chirps due to the velocity is:

$$\Delta\phi = 2\pi \frac{2 \cdot T_c \cdot v}{\lambda} = 4\pi \frac{T_c \cdot v}{\lambda} \tag{3.17}$$

Re-arranging this equation gives:

$$v = \frac{\lambda \Delta\phi}{4\pi T_c} \tag{3.18}$$

To get an accurate velocity estimation, the radar sends multiple chirps to form a frame and performs a Doppler-FFT over the phases received from these chirps to find $\Delta\phi$ and, hence, the velocity.

3.2.3.1 Maximum Velocity

Since the phase of a signal always has a range of $[-\pi, \pi]$, unambiguous measurement of the phase requires $|\Delta\phi| \leq 180°$ or $|\Delta\phi| \leq \pi$. This limits the maximum velocity of the subject that a radar can

measure. Considering the extreme situation where $\Delta\phi = \pi$, taking it into Equation (3.18) gives:

$$v_{max} = \frac{\lambda\pi}{4\pi T_c} = \frac{\lambda}{4T_c} \tag{3.19}$$

That is, within a certain period of time, measuring a higher velocity will require a shorter T_c, i.e. more frequent measurements. As an example, when the wavelength λ is 4 mm, a T_c of 100 us (10 kHz measuring rate) gives a maximum measurable velocity of $\pm$10 m/s.

3.2.3.2 Velocity Resolution

The velocity of the subject is determined through a Doppler-FFT, where the frequency component corresponds to the rate of the phase change due to the velocity. The theory of discrete Fourier transforms (DFT) states that two frequencies can be resolved if their difference $\Delta\phi$ is greater than $2\pi/N$, where N is the total number of samples, i.e. the number of chirps in a frame. Considering the extreme situation where $\Delta\phi = 2\pi/N$ and taking it into Equation (3.17), it becomes:

$$\frac{2\pi}{N} = 4\pi\frac{vT_c}{\lambda} \tag{3.20}$$

Defining $T_f = T_c \cdot N$ to be the duration of a frame,

$$\begin{aligned} \frac{1}{N} &= 2\frac{vT_f}{\lambda N} \\ v &= \frac{\lambda}{2T_f} \end{aligned} \tag{3.21}$$

That is, a longer frame time will give a smaller (better) velocity resolution. For example, with 50 chirps in a frame and each chirp being 100 us, then the velocity resolution will be 0.4 m/s.

3.2.4 Angle Calculation

The angular position of the object, or the AoA, can be estimated by having multiple antennas operating concurrently and by comparing the phase difference between neighbouring receivers. A number of receivers form an antenna array. Due to the spatial location difference between the receivers, the signal received at each receiver will have a slight phase difference depending on the relative position of the receivers and the AoA. The AoA can be computed in both azimuth and elevation directions, given that there exists more than one antenna in each direction. The azimuth angle θ_a is defined to be the angle between the object's projection on the horizontal plane and the front direction of the radar. As shown in Figure 3.2, the line of incidence of the object OA is projected onto the horizontal plane as OB, and the angle between OB and the y-axis is the azimuth angle θ_a. The elevation angle θ_e is defined to be the angle between the object and the horizontal plane (between line OA and the x-y plane). The azimuth and elevation angles together will be denoted as $\theta_{(a,e)}$.

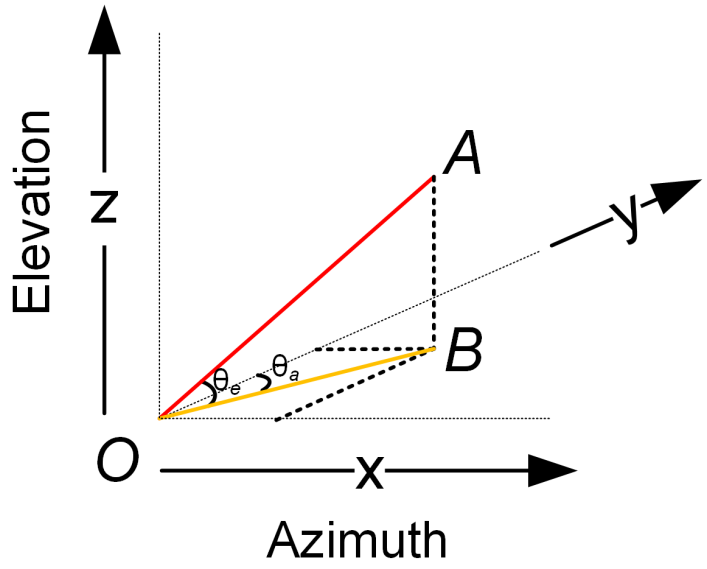


Figure 3.2: The azimuth and elevation angle of an object.

There are many algorithms designed for estimating the AoA based on a linear-spaced antenna array, such as the FFT-based method, beamforming method and subspace method. These algorithms provide a trade-off between the computational complexity and the angular resolution, as discussed in [161]. In the following sections, some of the most widely-used methods will be explained, including the FFT-based method, conventional beamforming (also known as the Bartlett beamforming or the delay-and-sum beamforming), the Minimum Variance Distortionless Response (MVDR) beamforming (also known as the Capon beamforming) [162], and the Multiple Signal Classifier (MUSIC) subspace method [163].

Assuming there are $N_a \times N_e$ receivers in the azimuth and elevation directions, and M objects in different directions $\theta_{(a,e)m}$, then each object can be viewed as a signal source and the antenna array will receive a signal (denoted as x) as a weighted sum of the M data source:

$$x^{(N_a \times N_e)} = \sum_{m=1}^{M} \alpha_m s(\theta_{(a,e)m}) + n \tag{3.22}$$

where $s(\theta_{(a,e)m})$ is the steering vector that represents the phase difference between receivers when a signal arrives with angle $\theta_{(a,e)m}$, α is an unknown parameter that models the signal transmission from the data source to the receivers, and n is the noise. The AoA estimation can be modelled as estimating the values of $\theta_{(a,e)}$ for each object m, given a set of receiver data (x).

3.2.4.1 Steering Vector

The steering vector is a function of the antenna layout and the incident angle, as will be shown in Equation (3.25) and Equation (3.29). To introduce the concept of steering vectors, it is easier to start with the one-dimensional situation. For linear-spaced arrays, the receivers are often

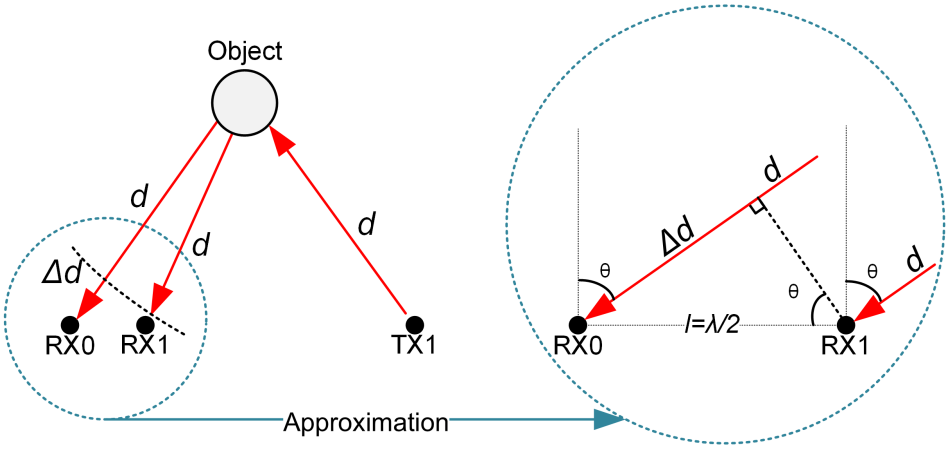


Figure 3.3: Phase difference between two receivers from one signal source.

separated by a small distance l that is equal to half of the signal wavelength, i.e. $l = \frac{\lambda}{2}$, to maximize the angle-of-view (AoV) (see Section 3.2.4.5). Assuming there are two receivers separated by l, a signal will travel an additional distance Δd to reach the second receiver, where the following approximation can be made (as shown in Figure 3.3):

$$\Delta d = l \cdot sin(\theta) \tag{3.23}$$

Given that the phase of a sinusoid signal travelled over any distance Δd will have a phase $\frac{2\pi\Delta d}{\lambda}$ (see Equation (3.16)), the phase difference between the two neighbouring receivers will be:

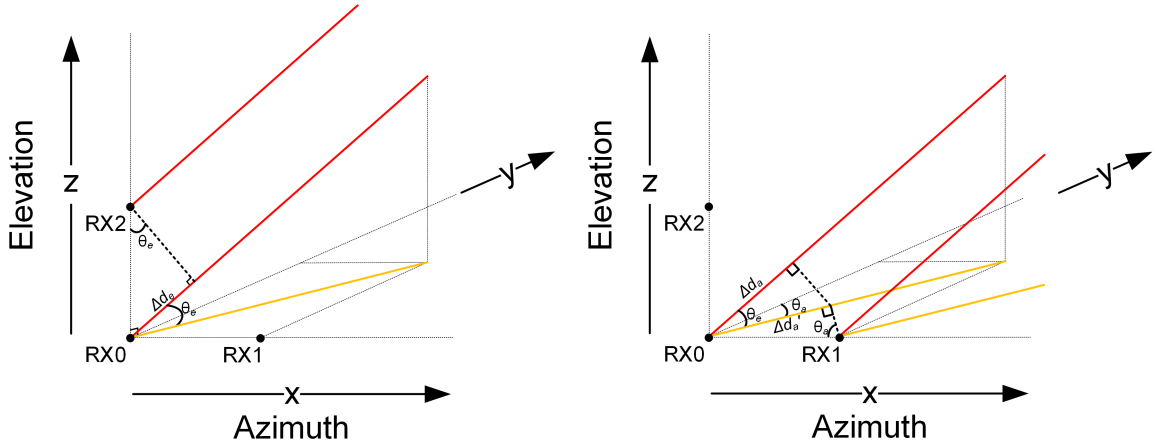
$$\begin{aligned} \Delta\phi &= 2\pi \cdot \frac{\Delta d}{\lambda} \\ &= 2\pi \cdot \frac{l \cdot sin(\theta)}{\lambda} \\ &= \pi \cdot sin(\theta) \quad \text{when } l = \frac{\lambda}{2} \end{aligned} \tag{3.24}$$

When using an antenna array with N azimuth receivers, each subsequent receiver beyond the first one will receive an additional phase change of $\Delta\phi$, which can be written as a steering vector:

$$s(\theta, N) = [1, e^{j\pi \cdot sin(\theta)}, e^{2j\pi \cdot sin(\theta)}, ..., e^{(N-1)j\pi \cdot sin(\theta)}] \tag{3.25}$$

When considering the AoA in both azimuth and elevation directions, the situation is shown in Figure 3.4. Distance Δd_a and Δd_e represent the extra distance travelled by the signal to reach receiver RX0 when compared with the azimuth receiver RX1 and the elevation receiver RX2 respectively. Similar to Equation (3.24), the estimation of the elevation angle θ_e is given by:

$$\begin{aligned} sin(\theta_e) &= \frac{\Delta d_e}{l} \\ &= \frac{\Delta\phi_e}{\pi} \end{aligned} \tag{3.26}$$



(a) Estimating the elevation angle.

(b) Estimating the azimuth angle.

Figure 3.4: The AoA can be estimated from the phase difference between adjacent antennas.

where $\Delta\phi_e$ is the phase difference between RX0 and RX2.

The azimuth angle requires a projection from the object's 3D location to the horizontal plane. As shown in Figure 3.4b, the projection from Δd_a to Δd_a' gives:

$$\Delta d_a = \Delta d_a' \cdot cos(\theta_e) \tag{3.27}$$

Then, the angle θ_a can be calculated as:

$$\begin{aligned} sin(\theta_a) &= \frac{\Delta d_a'}{l} \\ &= \frac{\Delta d_a}{l \cdot cos(\theta_e)} \\ &= \frac{\Delta\phi_a}{\pi \cdot cos(\theta_e)} \end{aligned} \tag{3.28}$$

where $\Delta\phi_a$ is the phase difference between RX0 and RX1.

When using an antenna array with N_a azimuth receivers and N_e elevation receivers, the steering vector can be written as:

$$s(\theta_{(a,e)}, N_a, N_e) = \begin{bmatrix} 1, & e^{j\Delta\phi_a}, & ..., & e^{j(N_a-1)\Delta\phi_a} \\ e^{j\Delta\phi_e}, & e^{j(\Delta\phi_e+\Delta\phi_a)}, & ..., & e^{j(\Delta\phi_e+(N_a-1)\Delta\phi_a)} \\ ..., & ..., & ..., & ... \\ e^{j(N_e-1)\Delta\phi_e}, & e^{j((N_e-1)\Delta\phi_e+\Delta\phi_a)}, & ..., & e^{j((N_e-1)\Delta\phi_e+(N_a-1)\Delta\phi_a)} \end{bmatrix} \tag{3.29}$$

where $\Delta\phi_e = \pi \cdot sin(\theta_e)$ and $\Delta\phi_a = \pi \cdot cos(\theta_e) \cdot sin(\theta_a)$, as shown Equation (3.26) and Equation (3.28).

Certain radar applications require the x-y-z coordinate of the object instead of the azimuth and elevation angles. Therefore, the calculation of the exact value of θ_a and θ_e is sometimes not required. Referring to Figure 3.2 and using Equation (3.26) and Equation (3.28), the coordinate

of the object can be calculated as:

$$\begin{aligned} x &= OB \cdot sin(\theta_a) = OA \cdot cos(\theta_e) sin(\theta_a) = OA \cdot \frac{\Delta\phi_a}{\pi} \\ z &= OA \cdot sin(\theta_e) = OA \cdot \frac{\Delta\phi_e}{\pi} \\ y &= \sqrt{OA^2 - x^2 - z^2} \end{aligned} \tag{3.30}$$

Given that OA can be obtained from the range-FFT as discussed in Section 3.2.2, the x-y-z coordinate can be obtained from the phase difference $\Delta\phi_a$ and $\Delta\phi_e$. Therefore, the AoA estimation of an object can be considered equivalently as searching for the best matching steering vector $s(\theta_{(a,e)m})$ of the object.

3.2.4.2 Angle-FFT Method

The simplest way of estimating $s(\theta_{(a,e)m})$ of an object m in Equation (3.22) is by using correlation between the receiver data x and the steering vector from the candidate angles. A set of candidate steering vectors $s(\bar{\theta}_{(a,e)})$ is defined for $\theta_a \in [-\pi, \pi], \theta_e \in [-\pi, \pi]$, and the correlation is calculated as $s(\bar{\theta}_{(a,e)}) \cdot x$, which will yield a peak output when $\bar{\theta}_{(a,e)}$ equals to $\theta_{(a,e)m}$. This process is equivalent to applying an FFT over the receiver data x, since the steering vector can be considered the same as a set of FFT coefficients, which gives the frequency components in terms of $\Delta\phi_a$ and $\Delta\phi_e$. This FFT is also referred to as the angle-FFT.

As an example, Figure 3.5 shows the antenna layout of the TI IWR6843 radar. It has three transmitters and four receivers, which can form a 12-antenna array when using multiple-in multiple-out (MIMO) techniques [164]. The phase at each virtual antenna is also shown, where φ is the random initial phase of the first receiver. The azimuth antennas will form a signal $e^{j(\Delta\phi_a n + \varphi)}$ and the elevation antennas will form a signal $e^{j(\Delta\phi_a n + 2\Delta\phi_a + \varphi + \Delta\phi_e)}$, where n is the antenna index in each direction. The value of $\Delta\phi_a$ can be obtained by applying an azimuth-FFT over the azimuth antennas (RX1-RX4 and RX9-RX12), which will give the frequency $\Delta\phi_a$ and phase φ. The value of $\Delta\phi_a$ can be obtained by applying an FFT over the elevation antennas, which will give the frequency $\Delta\phi_a$ and phase $2\Delta\phi_a + \varphi + \Delta\phi_e$. Hence, the value of $\Delta\phi_e$ can also be calculated given $\Delta\phi_a$ and φ.

One limitation of the above approach is that, although the azimuth-FFT can distinguish objects from different azimuth angles, objects from the same azimuth but different elevation angles can hardly be distinguished. Separation of such objects would rely on the range- and Doppler-FFT, as the object only needs to be distinguished once during the entire DPC. An alternative approach to calculate $\Delta\phi_a$ is by applying an elevation-FFT over a set of antennas in the elevation direction. For example, Figure 3.6 shows the layout of the TI overhead detection sensor (ODS) model, where the antennas form a near-square shape and allows a 2D angle-FFT to be performed. The ODS models allow a higher elevation resolution at the cost of reduced azimuth resolution.

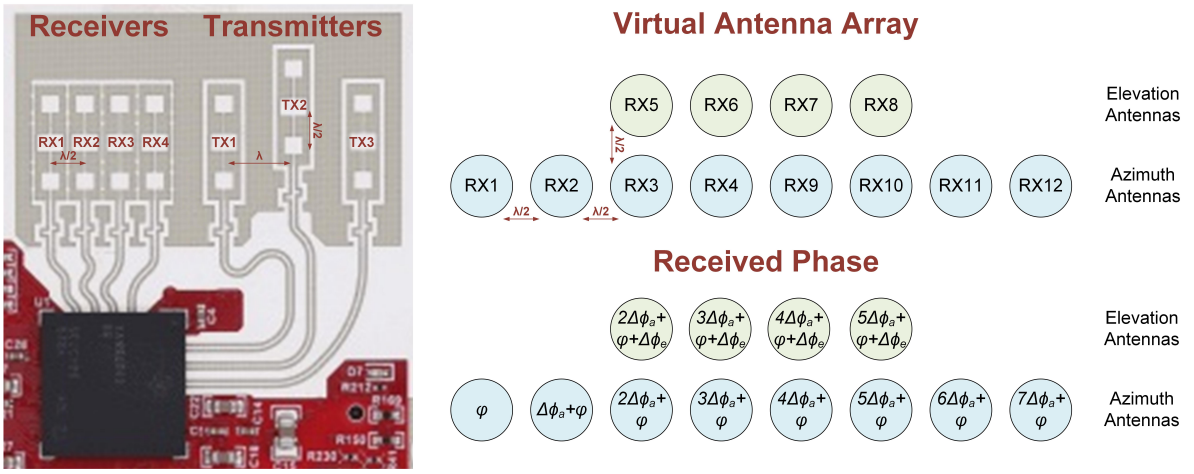


Figure 3.5: IWR6843/IWR1443/IWR1843 radar antenna layout, the virtual antenna array and the received phases.

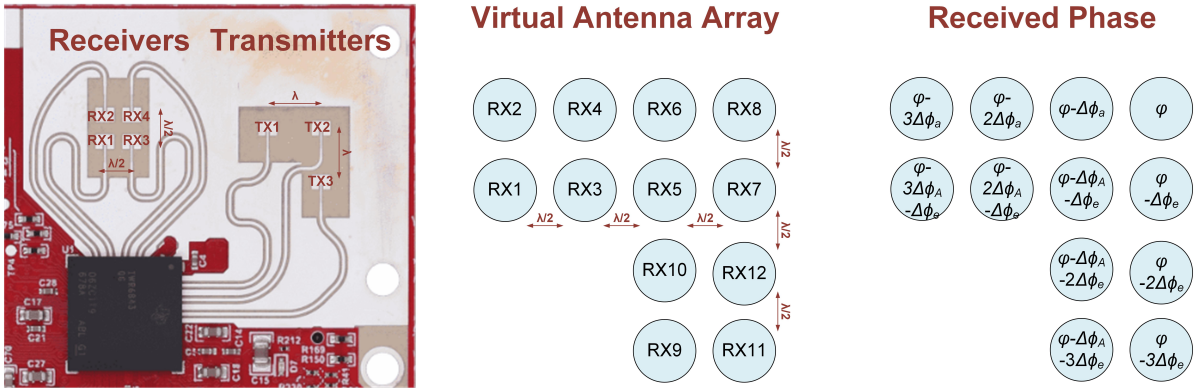


Figure 3.6: IWR6843ODS radar antenna layout, the virtual antenna array and the received phases.

The major problem of the angle-FFT method is the resolution. Considering there are two objects separated by an angle $\Delta\theta$, the phase difference between adjacent receivers for the two objects, according to Equation (3.24) and assuming $l = \frac{\lambda}{2}$, will be:

$$\begin{aligned} \Delta\phi_1 &= \pi \cdot sin(\theta) \\ \Delta\phi_2 &= \pi \cdot sin(\theta + \Delta\theta) \end{aligned} \tag{3.31}$$

and the difference between them will be:

$$\Delta\phi = \pi \cdot (sin(\theta + \Delta\theta) - sin(\theta)) \tag{3.32}$$

The small angle approximation states that $sin(\alpha+\beta) \approx sin(\alpha)+\beta \cdot cos(\alpha)$ for small angles. Applying the approximation to the equation above gives:

$$\Delta\phi = \pi \cdot cos(\theta)\Delta\theta \tag{3.33}$$

According to the DFT theorem, two frequencies can be resolved if $\Delta\phi > 2\pi/N$, where N is the number of samples. Therefore, in order to separate the two angles, the following equation needs to be satisfied:

$$\begin{aligned} \pi \cdot cos(\theta)\Delta\theta &> \frac{2\pi}{N} \\ \Delta\theta &> \frac{2}{N \cdot cos(\theta)} \end{aligned} \tag{3.34}$$

where N is how many receivers are available. With N_{tx} transmitters and N_{rx} receivers, a virtual antenna array of $N_{tx} \times N_{rx}$ can be generated with MIMO techniques. This is equivalent to 1 transmitter and $N_{tx} \times N_{rx}$ receivers and will result in an improvement in the angular resolution:

$$\theta_{res} = \frac{2}{N_{rx} \cdot N_{tx} \cdot cos(\theta)} \tag{3.35}$$

The resolution will be higher with smaller angles and more antennas. For example, when detecting objects at 30° with 8 virtual receivers, the angular resolution will be $\theta_{res} = 0.289 = 16.6°$.

3.2.4.3 Beamforming Method

Beamforming methods calculate a set of weights $w^{(N_{rx} \times \Theta)}$ for the N_{rx} virtual receivers in the array (both azimuth and elevation), and for all possible angles $\theta_{(a,e)} \in \Theta$ where $\theta_a \in [-\pi, \pi], \theta_e \in [-\pi, \pi]$. When applying a column of weights to the receiver data x, the signal from the direction θ will receive a constructive inference. By searching all possible angles $\theta_{(a,e)}$, a power spectrum p with size Θ can be obtained, where a high power in the spectrum indicates that there is a data source in that direction:

$$p = w^H x \tag{3.36}$$

where w^H is the Hermitian transposition of w. The angles of the M objects can be obtained by taking the M highest peaks in p and finding the corresponding entries in w.

In the data model shown in Equation (3.22), signals reflected from objects will be correlated when being received at each receiver, whereas the noise will be uncorrelated. Therefore, one way to extract signal information from x is by calculating a sensor covariance matrix R_x:

$$R_x = E\{x^H x\} \tag{3.37}$$

where E represents the statistical expectation. In practice, it is impossible to get an accurate covariance matrix, but only an estimation from a set of data snapshots over time:

$$R_x = E\{x^H x\} \approx \frac{1}{N} \sum_{t=1}^{N} x^H(t) x(t) \tag{3.38}$$

where $x(t)$ represents one snapshot (or one frame) of the receiver data x. When evaluating the beamforming power spectrum using multiple snapshots, the overall power spectrum becomes the

statistical expectation of p in Equation (3.36) over the snapshots, which gives:

$$\begin{aligned} P &= E\{|w^H x|^2\} \\ &= \frac{1}{N}\sum_{t=1}^{N}|w^H x(t)|^2 \\ &= \frac{1}{N}\sum_{t=1}^{N} w^H x(t) x^H(t) w \\ &= w^H R_x w \end{aligned} \tag{3.39}$$

There are many algorithms designed for calculating the weights w. The conventional beamforming uses the steering vector directly as the weights, which is conceptually equivalent to the angle-FFT method (or correlation-based method) in Section 3.2.4.2:

$$P_{conventional} = s^H R_x s \tag{3.40}$$

where s is the candidate steering vector in the format of Equation (3.29).

There are also adaptive beamforming algorithms that calculate the weights using the signal information embedded in the covariance matrix. For example, the MVDR algorithm aims at minimizing the variance from non-interested directions while keeping the signal from the candidate direction distortionless [162]. It defines the weight w as:

$$w_{mvdr} = \frac{R_x^{-1} s}{s^H R_x^{-1} s} \tag{3.41}$$

Putting Equation (3.41) together with Equation (3.39) simplifies to:

$$P_{mvdr} = \frac{1}{s^H R_x^{-1} s} \tag{3.42}$$

Once the beamforming power spectrum is computed, the peaks in the spectrum will correspond to the signal from the objects.

3.2.4.4 Subspace Method

The core of the subspace method is that, since the signal x should contain M correlated signals and uncorrelated noise, the covariance matrix R_x should have M non-zero eigenvalues and $N-M$ zero eigenvalues, where N is the rank of R_x that is equal to the number of receivers. The eigenvectors corresponding to the M eigenvalues form the signal subspace, and the eigenvectors corresponding to the zero eigenvalues form the noise subspace. The signal subspace and the noise subspace are orthogonal.

One of the most widely-used subspace-based algorithms is the MUSIC algorithm [163]. It searches for steering vectors that are orthogonal to the noise subspace. The power spectrum of the MUSIC algorithm can be written as:

$$P_{music} = \frac{1}{s^H U U^H s} \tag{3.43}$$

where U is the set of eigenvectors corresponding to the zero eigenvalues. A more detailed evaluation of the mentioned AoA estimation algorithms will be given in Chapter 4.

3.2.4.5 Maximum Angle-of-View

Unambiguous measurement of the angle requires $|\Delta\phi| \leq 180°$ or $|\Delta\phi| \leq \pi$, since phases of any signal have a range of $[-\pi, \pi]$. Considering the extreme situation where $\Delta\phi = \pi$, Equation (3.24) gives

$$\begin{aligned} \pi &= 2\pi \cdot \frac{l \cdot sin(\theta)}{\lambda} \\ sin(\theta) &= \frac{\lambda}{2l} \\ \theta &= sin^{-1}(\frac{\lambda}{2l}) \end{aligned} \tag{3.44}$$

which gives the maximal value of θ as

$$\theta_{max} = sin^{-1}(1) = 90° \quad \text{when } l = \frac{\lambda}{2} \tag{3.45}$$

That is, neighbouring receiving antennas should be separated by a distance of half the wavelength of the signal (approximately 2 mm), to achieve a ±90° AoV.

However, due to the antenna characterization and signal attenuation, the actual AoV of the radar is much smaller. Table 3.2 shows, for some TI radar models, at which angles the signal strength of the radar will drop to a certain level (data from TI [165–168]). As an example, all radars can hardly detect anything beyond ±50° horizontally, as the signal strength would reduce to 1/4 (−6 dB) of the original.

Table 3.2: TI mmWave radar AoV at given signal strength (H for Horizontal and V for vertical).

	Frequency	3 dB-H	3 dB-V	6 dB-H	6 dB-V
IWR1443	77 GHz	28°	14°	50°	20°
IWR1642	77 GHz	32°	14°	48°	18°
IWR1843	77 GHz	28°	14°	50°	20°
IWR6843	60 GHz	32°	11°	47°	17°
IWR6843ODS	60 GHz	28°	32°	40°	46°

3.3 TI mmWave Radars[3]

In this section, the details of the TI mmWave radars are discussed, including the DPC implementation on the radar, the radar configuration steps and the data transmission between the radar and a host environment.

[3]Some content of this section has been published in [20] ©2021 IEEE.

3.3.1 Hardware Models

The TI mmWave radars consist of two main series: the industrial IWR series and the automotive AWR series. There are also different models in each product series. The models differ mainly in the following aspects:

1. Number of antennas and the layout. Some models (like the IWR1443) have three transmitters that include both azimuth and elevation channels, whereas others (like the IWR1642) have two transmitters that only include the azimuth channel. All models have four receivers.

2. Processors and hardware accelerators. Most radars have an ARM processor for controlling the radio frequency subsystem and communication between the host environment. Some models (like the IWR1443) have hardware accelerators that are designed for the FFTs, and some (like the IWR1642) have programmable DSPs that perform the FFTs and other advanced post-processing.

3. Operational frequency, mostly either 77 GHz to 81 GHz or 60 GHz to 64 GHz.

4. Supported chirp configuration, which is mainly restricted by the maximum IF frequency, ADC sampling rate and memory size.

5. Peripherals for communications with external hardware and the host environment.

More detailed comparisons between the models are available from the TI radar datasheets [169]. For this research, the IWR1443 and IWR1843 radars have been used. They both have three transmitters and support 3D scanning of the scene. The IWR1443 models were used in the early stage of this research and were replaced by the IWR1843 models after their release, as the latter has additional DSPs on the chip. However, empirically no significant performance difference between the two models was observed, and the models did not affect the research outcome much as most of the data processing in this research was done in a host environment.

3.3.2 DPC

TI mmWave radars have on-chip processors that are able to implement the algorithms described in Section 3.2. Figure 3.7 shows the default DPC implemented on the radar, where more details on the other possible DPCs will be discussed in Chapter 4.

The radars use a complex mixer to mix the transmitted signal and the received signal, a lowpass filter to generate the IF signal, and an ADC to convert it into a digital signal. The IF signal is generated and processed independently regarding each receiver. A range-FFT is computed inline after the sampling of the IF signal, and the range-FFT result is saved to the on-chip memory. When a frame is finished and the range-FFT result of all chirps is saved, a Doppler-FFT is applied over each range bin across the chirps, which gives a 2D matrix containing

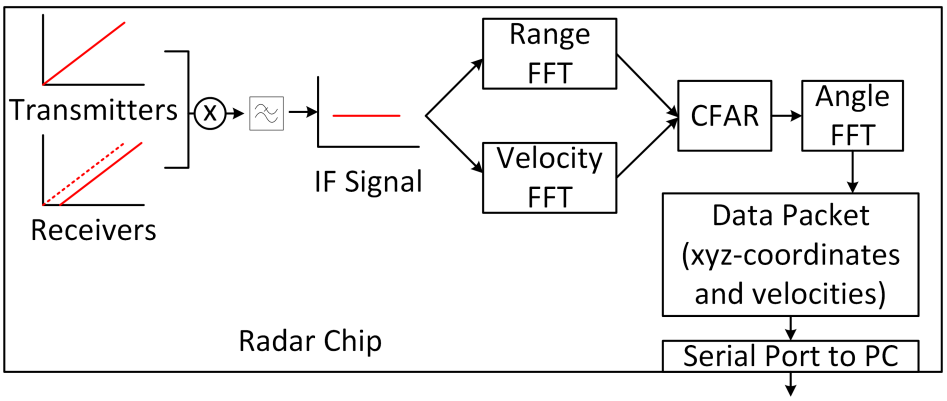


Figure 3.7: DPC of the TI mmWave radar.

the Doppler-FFT result at each range-Doppler bin. This process is also referred to as the range-Doppler-FFT. Once the range-Doppler-FFT is finished, a constant false alarm rate (CFAR) peak detection algorithm is applied to extract the peaks from the 2D matrix, where each peak is considered to represent one object. Finally, a virtual antenna array is constructed and an angle-FFT is applied over each peak across the virtual antennas to determine the AoA of the object. The final output will be in the format of a point cloud, where each point corresponds to an object and encodes the x-y-z coordinate and the velocity of the object. The point cloud could be post-processed using the on-chip DSP or be transmitted to a PC for further processing.

3.3.2.1 CFAR Peak Detection

After each stage of the FFTs, a peak detection algorithm is required to identify the presence of the subject. A common choice of the peak detection algorithm for radar applications is the CFAR algorithm [170]. It uses an adaptive threshold and aims to give a constant FAR (False Alarm Rate) at different signal strengths, which is particularly useful for radars as objects far away tend to give a low signal strength. The algorithm iterates through all data points (referred to as the cell-under-test (CUT)) and calculates the noise power from neighbouring cells in a window. The noise power is defined to be a scaled sum of all neighbouring cells, except those immediately next to the CUT (referred to as the guard cells), to avoid any power leakage from the CUT. The scale factor is determined based on the desired FAR. At any position, if the power of the CUT is higher than the noise power, then it is considered to be a peak and is likely to correspond to a real object in the scene. CFAR is performed on the chip using either the hardware accelerator or DSP. The window size and the desired FAR can be configured regarding the application.

3.3.3 Radar Configuration

Using the mmWave software development kit (SDK) provided by TI [171], the user can configure the chirp to fit their use case. The on-chip ARM processor can be used to read commands from

the host environment and configure the RF subsystem. The main properties that need to be configured include the following:

1. Transmitters and receivers to be used.
2. Structure of the chirp and frame.
3. Post-processing algorithms, including the FFTs and the CFAR algorithm.

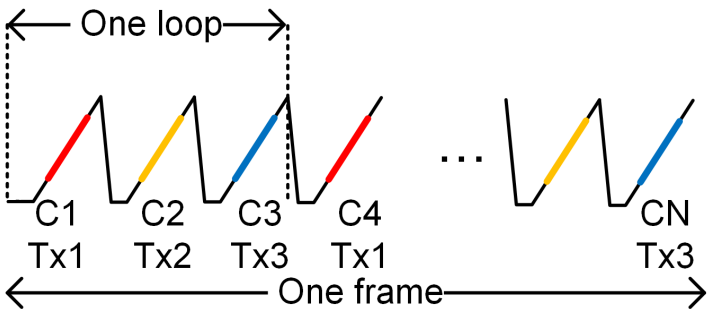


Figure 3.8: Configuration of a chirp frame.

The radars have two to three transmitters and four receivers that can run concurrently. However, when multiple transmitters are used at the same time, a modulation scheme is required to separate the signal from each transmitter, especially when estimating the AoA of the object. Two modulation schemes are supported, time-division multiplexing (TDM) and binary phase modulation (BPM) [164]. In the TDM mode, only one transmitter is enabled in each chirp and the chirps are interleaved in the time domain, which allows the signal from each transmitter to be separated easily (see Figure 3.8). However, the potential of the multiple transmitters is not fully used with TDM, as only one transmitter will be active at any timestamp. In contrast, in the BPM mode, all transmitters are enabled at the same time but send signals with a certain phase shift. When the signals are received, the signal from each transmitter can be restored by an appropriate decoding scheme. BPM has a higher complexity, but can provide a higher SNR as all the transmitters are enabled at the same time. However, TI radar models do not fully support the BPM mode, so this research focuses on the TDM mode and left BPM as future work.

The structure of a chirp is shown in Figure 3.9 [171]. A chirp mainly consists of three properties:

1. The **idle time** between any two successive chirps. A 2 us to 7 us idle time is generally required for the chirp to be reset after each transmission.
2. The **ramp time**, where the frequency of the chirp signal increases at a certain slope within this period. The transmitters are generally switched on at the start of the ramp time. However, they can be configured to be switched on before or after the ramp time,

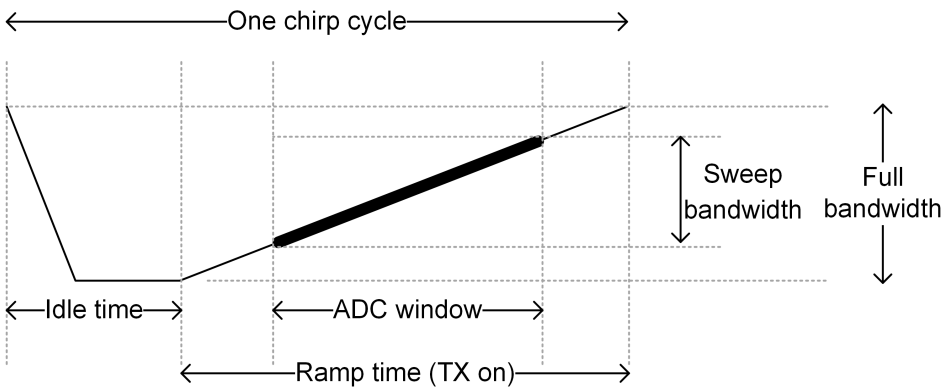


Figure 3.9: Structure of a chirp signal.

if required. The ramp time of a chirp is often around 50 us to 200 us and can vary a lot between applications.

3. The **ADC active time**, where the ADC actively samples the data from the receivers at its sampling rate. The ADC should only be active within the ramp time while setting aside a few microseconds of settling time before and after to ensure the quality of the chirp. The settling time is generally around 5 us to 10 us.

The sweeping bandwidth of the chirp is defined as **ramp time** × **chirp slope** and should be below the available bandwidth (4 GHz), whereas the effective bandwidth is **ADC active time** × **chirp slope**. An ideal chirp configuration should aim at utilizing the full 4 GHz bandwidth, but can also vary a lot depending on the desired detection resolution of the application. The detailed configuration used in this research will be given in each chapter later.

3.3.4 Data format

There are two ways to read data from a radar: reading the digitalized IF signal directly from the ADC or reading the processed point cloud from the serial port.

3.3.4.1 IF signal

Since the ADC sampling rate can reach up to 37.5 Msps, transferring the raw IF signal to a PC requires a very high bandwidth. Therefore, TI provides separate hardware platforms to capture the raw data: the TSW1400 [172] and the DCA1000 board [173]. Both of them are FPGA (Field Programmable Gate Arrays) based designs and allow raw IF signal capturing from the mmWave radar modules. The IF signal after the mixer can be transmitted to the data capture board through up to four low-voltage differential signalling (LVDS) lanes. The TSW1400 board has 1 GB memory that is able to hold the raw data, which can be dumped to a PC through a serial port once the capturing process is finished. In contrast, the DCA1000 board allows real-time data

Chirp	Sample	Real				Imag			
		RX0	RX1	RX2	RX3	RX0	RX1	RX2	RX3
A	0	A00	A01	A02	A03	A04	A05	A06	A07
	1	A08	A09	A10	A11	A12	A13	A14	A15
	2	A16	A17	A18	A19	A20	A21	A22	A23
	...	...							
B	0	B00	B01	B02	B03	B04	B05	B06	B07
	1	B08	B09	B10	B11	B12	B13	B14	B15
	2	B16	B17	B18	B19	B20	B21	B22	B23
	...	...							

Figure 3.10: Raw data format from an IWR1443 radar when using a DCA1000 board.

streaming from the radar to the PC through a gigabits Ethernet port. Since the DCA1000 allows real-time data capturing and does not have any memory constraints, it is the preferred approach for this research. This setup has been used in the vital sign detection system in Chapter 7.

The radar uses a complex band architecture that includes a quadrature mixer for generating the IF signal. It mixes the received signal with both an in-phase and a quadrature version of the transmitted signal, to generate a complex IF signal. This brings the advantage of an improved noise figure and detection performance [158]. The ADC can operate in three modes: real, complex-1x and complex-2x. The ADC only takes the in-phase signal when in the real mode, and takes both the in-phase and the quadrature signal when in the complex mode. When in the complex mode, the signal consists of an in-band signal and an image band signal. The former contains signals reflected from the object and is used for object detection, and the latter contains noise and inference information that can be used for signal quality evaluation and inference cancellation. When in the complex-1x mode, the complex signal will be filtered and shifted so that only the in-band signal is kept. Complex-2x, on the other hand, contains both the in-band and the image band, and can be used for custom applications when information in the image band is necessary.

Each ADC sample is a 2-byte word. The captured data will be arranged in different layouts depending on the capture board, the radar model and the antenna configuration (more details in [174]), and the corresponding decoding scheme is required when reading the data for post-processing. For example, Figure 3.10 shows the data format when using four receivers on an IWR1443 radar with the DCA1000 board.

3.3.4.2 Point Cloud

The on-chip hardware processors on the radar implement a complete DPC for processing the ADC data. Therefore, it is much easier for the user to configure the processors and only capture

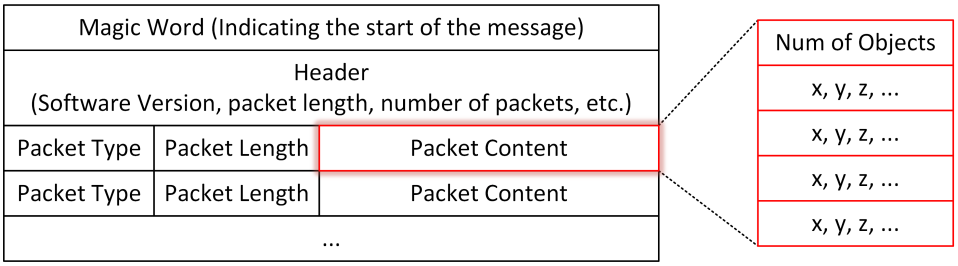


Figure 3.11: Example message structure from IWR1443.

the processed data. Communication with the radar is made through the use of two serial ports: one configuration port and one data port. The configuration port allows the PC to interact with the radar and send commands, such as configuring the antennas and switching on/off the radar. The data port is read-only from the PC side, where the radar will start dumping the processed data to this port once it starts operating.

The on-chip DPC is user-programmable. By using the out-of-box system firmware provided by TI, the processed data can be captured in the form of data messages. As an example, the structure of a data packet from the IWR1443 radar is shown in Figure 3.11. Each message will have a header to indicate the start of a packet, its content type and its length. Decoding the data packet includes detecting and parsing the header, fetching the content whose length is specified by the header, and processing the content. The most important data packets are those stating the presence of any object in front of the radar, which will be reported with its x-y-z coordinates, velocity and signal strength.

3.4 Conclusion

This chapter introduces the use of mmWave radars in HAR and describes the fundamentals of mmWave sensing techniques required for developing the rest of this research. It explains the underlying FMCW theorem of the radar that is used for detecting the distance, velocity and AoA of an object in front of the radar. The radar sends a chirp signal, receives its reflection and processes the signal to get the frequency and phase variation due to the signal passing through the scene. The distance of the object can be determined by the signal frequency, the velocity can be determined by sending successive chirps and analysing the phase variation between the chirps, and the AoA can be determined by using multiple transmitters and receivers and analysing the phase variation between the receivers. There are many AoA estimation algorithms in the literature, including angle-FFT methods, beamforming methods and subspace methods, among which the angle-FFT is the least computationally expensive and can be used on embedded processors, whereas the other algorithms provide a higher resolution. A combination of the distance and AoA allows a point cloud to be generated that encodes the spatial shape of the object, which forms the foundation for HAR systems in the next chapters.

CHAPTER 4

USING MMWAVE RADAR AS 3D SENSOR

As discussed in Chapter 3, mmWave radars are able to generate point clouds to represent objects in the scene. However, the accuracy and density of the generated point cloud are still much lower than LIDARs. Although researchers have used mmWave radars for various applications, there are few quantitative evaluations on the quality of the point cloud generated by the radar and there is a lack of a standard on how this quality can be assessed. This work aims to fill the gap in the literature. A radar simulator is built to verify the DPC described in Chapter 3 and to examine the capability of the mmWave radar as a high resolution 3D sensor. It will be shown that the point cloud generated from the radars can be noisy and have an imbalance distribution. To address the problem, a novel super-resolution point cloud construction (SRPC) algorithm is proposed to improve the spatial resolution of the point cloud and is shown to be able to produce a more natural point cloud and reduce outliers.

The rest of the chapter is organized as follows. Section 4.1 presents the details of the simulator for simulating the radar raw data towards a scene. Section 4.2 describes the DPCs that are commonly seen on a mmWave radar. Section 4.3 describes the FAUST dataset used with the simulator for evaluating the radar. Section 4.4 presents and discusses the evaluation results. Section 4.5 presents the novel SRPC algorithm to improve the resolution of the radar point cloud and shows its effectiveness. Section 4.6 concludes the chapter.

4.1 mmWave Radar Simulator

Radar data simulation allows researchers to focus on algorithm design and verification, instead of investing too much time in the hardware and real-world data collection. Existing radar simulators are often not designed for 3D imaging and have certain constraints. For example, the system in [175] generates range and Doppler information of the radar rather than the raw data, the system

in [176] only supports single antenna data generation and cannot be used to estimate the AoA, and the system in [177] only supports up to four receivers in one direction and cannot be used for 3D imaging. In this research, a lightweight mmWave radar simulator is designed that supports raw data generation of a multi-antenna mmWave radar, configurable antenna parameters and layout, and customized scene construction using 3D human models with programmable motions.

The radar is simulated to have one transmitter and one receiving antenna array, which is practically equivalent to a multi-transmitter multi-receiver radar using an appropriate modulation scheme [164]. Any two neighbouring receivers in the array are separated by $\lambda_0/2$, as explained in Section 3.2.4.5, where λ_0 (approximately 3.9 mm) is the wavelength of the mmWave signal at its chirp starting frequency (77 GHz). The simulator simulates the IF signal at each receiver of a mmWave radar when pointing toward a scene. The scene consists of M points, where each point has a unique x-y-z coordinate and represents the spatial location of the object in the scene. Each point is modelled as a corner reflector and reflects the mmWave signal sent out by the radar with a certain reflectivity. The IF signal at a receiver during one chirp is modelled using Equation (3.10) and Equation (3.11) in Chapter 3. Given a certain chirp configuration, the frequency and phase of the IF signal from one point are determined by the distance d between the point and the receiver. The amplitude of the IF signal is set to be inversely proportional to d^4, to simulate the power loss due to distances according to the radar range equation (Equation (3.14)). The final IF signal at a receiver is the accumulated IF signals from all M points in the scene, with an additional white Gaussian noise n, as shown in Equation (4.1).

$$IF(t) = \sum_{i=1}^{M} \frac{1}{d_i^4} e^{j(2\pi \cdot S\tau_i \cdot t + \frac{4\pi d_i}{\lambda_0})} + n \tag{4.1}$$

where d_i and τ_i are the distance and time-of-flight (ToF) from the transmitter to the point i and then to the receiver, and S is the slope of the chirp. The amplitude of the noise n is controlled by the desired SNR during the experiment. The signal $IF(t)$ is sampled into a digital signal of length N_s, where N_s = (duration of the chirp) × (ADC sampling rate). During one chirp, the radar receives a signal that can be represented as a 2D matrix of size $N_{rx} \times N_s$, where N_{rx} is the number of receivers in the array. One frame includes N_c chirps that form a 3D matrix of size $N_{rx} \times N_c \times N_s$, which becomes the input matrix of the point cloud construction algorithm, as shown in Figure 4.1.

4.2 Point Cloud Construction Algorithm

The construction of a point cloud takes an input matrix of size $N_{rx} \times N_c \times N_s$ and outputs a 2D matrix PC_K of size $K \times 3$ (referred to as the output point cloud), where K is the number of

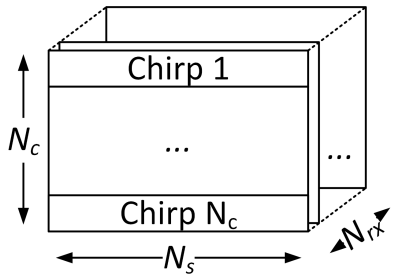


Figure 4.1: One frame of radar data represented as a 3D matrix.

detected points and 3 is the x-y-z coordinates.

$$PC_K = \begin{bmatrix} x_1, & y_1, & z_1 \\ x_2, & y_2, & z_2 \\ ..., & ..., & ... \\ x_k, & y_k, & z_k \end{bmatrix} \tag{4.2}$$

The distance and AoA of the object are required to estimate its x-y-z coordinate. In an FMCW radar model, the distance of an object can be computed through a range-FFT over the IF signal, whereas the AoA estimation can be performed by analysing the signal difference between multiple receivers. As discussed in Section 3.2.4, there are a variety of algorithms for computing the AoA based on a uniform antenna array, including the angle-FFT, beamforming and subspace methods. The angle-FFT method is a single-snapshot method that can make an estimate based on a single chirp, whereas the other methods are multi-snapshot methods that require a few chirps to make one estimate. The performance of the algorithms depends on several factors, including the antenna layout, number of antennas, chirp configuration, number of snapshots, SNR, environment, etc. This section studies one of the most common DPCs used on mmWave radars and its variant, which have shown success in many HAR systems, like in [116, 154, 156].

4.2.1 Data Processing Chains

There are two possible DPCs depending on the use of a Doppler-FFT or not, as shown in Figure 4.2. Both DPCs require a range-FFT over the raw data. The range-FFT identifies the frequency components in the IF signal that corresponds to the distance of an object. It transforms the input matrix X of size $N_{rx} \times N_c \times N_s$ into a range matrix R of size $N_{rx} \times N_c \times N_s^*$, where N_s^* is the length of the range-FFT. The first DPC is used by TI in the radar firmware. It applies a Doppler-FFT on the data from all the chirps and generates a Range-Doppler heatmap of size $N_{rx} \times N_c^* \times N_s^*$, where N_c^* is the length of the Doppler-FFT. Then, it searches for peaks in the Range-Doppler heatmap (using the average of all receivers), extracts the receivers' data for each peak and generates a 2D matrix of size $K \times N_{rx}$, where K is the number of detected peaks and, equivalently, the number of detected points. The CFAR algorithm described in Section 3.3.2.1 is

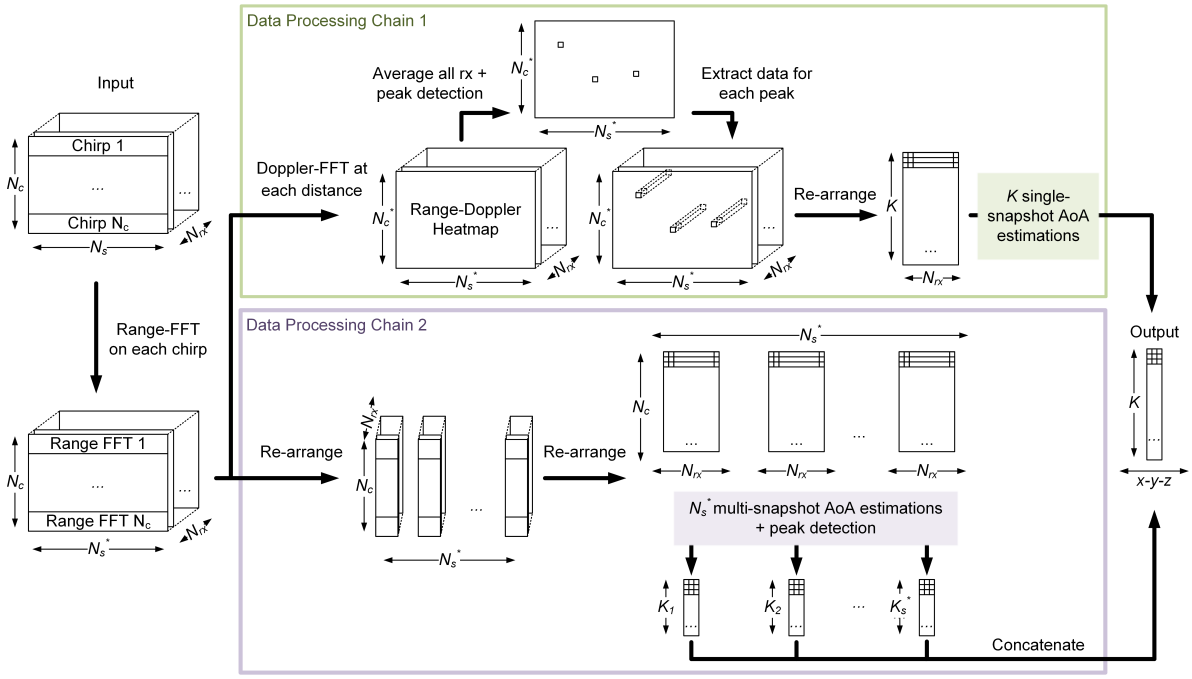


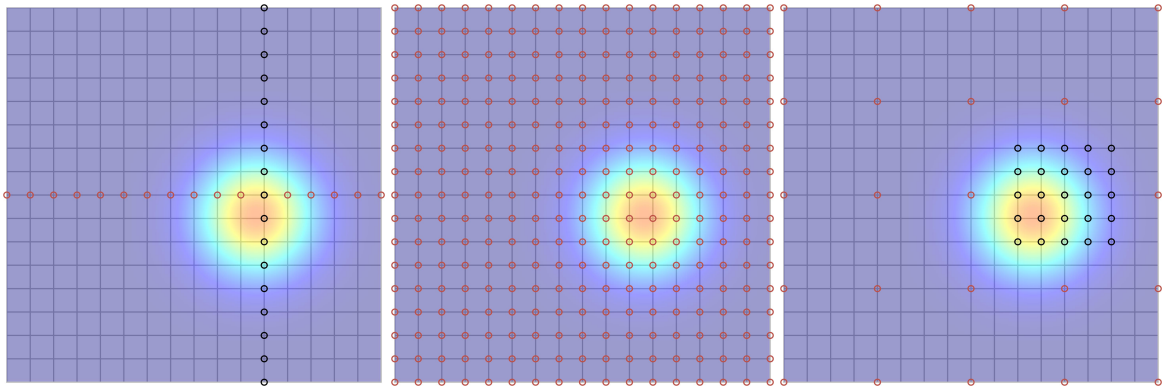
Figure 4.2: Two possible DPCs for mmWave radar point cloud construction.

used for detecting peaks from the Range-Doppler heatmap. The parameters of CFAR control the sensitivity of the peak detection and are considered the hyperparameters of the system. Finally, a single-snapshot AoA estimation is applied to each point in the matrix for a total of K times, to obtain the x-y-z coordinates of all detected points. The AoA estimation algorithm can be any of the angle-FFT, beamforming or subspace methods. Although the beamforming and subspace methods are multi-snapshot algorithms, the Doppler-FFT implicitly uses the information from all chirps and allows a good estimate of the covariance matrix at the AoA estimation stage.

The second DPC does not include a Doppler-FFT. Instead, it considers the chirps as different snapshots and performs one multi-snapshot estimation for each range bin for a total of N_s^* times. More specifically, the input range matrix of size $N_{rx} \times N_c \times N_s^*$ is re-arranged into N_s^* instances of $N_{rx} \times N_c$ matrix, and the AoA estimation is applied on each $N_{rx} \times N_c$ matrix using N_c snapshots. The AoA estimation algorithm can be any of the beamforming or subspace methods. Finally, the points detected at each range bin are concatenated into one point cloud. In this research, the angle-FFT, conventional beamforming, MVDR beamforming and MUSIC subspace methods described in Section 3.2.4 are being studied.

4.2.2 Model Order Estimation

As described in Section 3.2.4.3 and Section 3.2.4.4, the beamforming and subspace methods include an angle power spectrum computation step, where each peak in the spectrum corresponds to an incoming signal from a point. However, in both DPCs, the expected number of incoming



(a) An azimuth search (red) followed by an elevation search (black).

(b) A full 2D azimuth-elevation search.

(c) A 2D azimuth-elevation search using sub-grids.

Figure 4.3: Three approaches when searching for the steering vectors.

signals will be unknown in practice. Therefore, this number needs to be estimated from the signal data. This step is referred to as model order estimation. For this purpose, the covariance matrix of the signal data and its eigenvalues are computed. As described in Section 3.2.4.4, the covariance matrix should have a size of $N_{rx} \times N_{rx}$ and has a full rank equal to N_{rx}, and there should be M large eigenvalues that correspond to the number of incoming signals and $N_{rx} - M$ zeros corresponding to noise. In practice, due to the presence of noise, the difference between these eigenvalues may not be significant. Therefore, the minimum descriptive length (MDL) algorithm [178] is used for estimating the value of M. It fits a statistical model using the eigenvalues and searches for the optimal value of M that minimizes a cost function. The MDL algorithm is used in the AoA estimation step in both DPCs to estimate the number of incoming signals. Once the angle power spectrum has been calculated, all the local maxima will be found and the largest M_{mdl} peaks will be taken as the output, where M_{mdl} is the value found from the MDL algorithm.

4.2.3 Steering Vector Searching

The beamforming and subspace methods search for the steering vectors that maximize a power function. This process can be carried out using three approaches: an azimuth search followed by an elevation search, a 2D azimuth-elevation search or a 2D search using sub-grids. An example of the three approaches is shown in Figure 4.3. In the example, the power spectrum shows the incoming direction of the signal. The space of the spectrum is sampled into a 17×17 grid and each vertex on the grid represents a candidate AoA to be tested. In the first approach, an azimuth AoA search is performed using the data from azimuth receivers and steering vectors that only consider the azimuth angle. Then, based on the azimuth AoA output, a secondary search is performed in the elevation direction using the data from all receivers. This approach has the

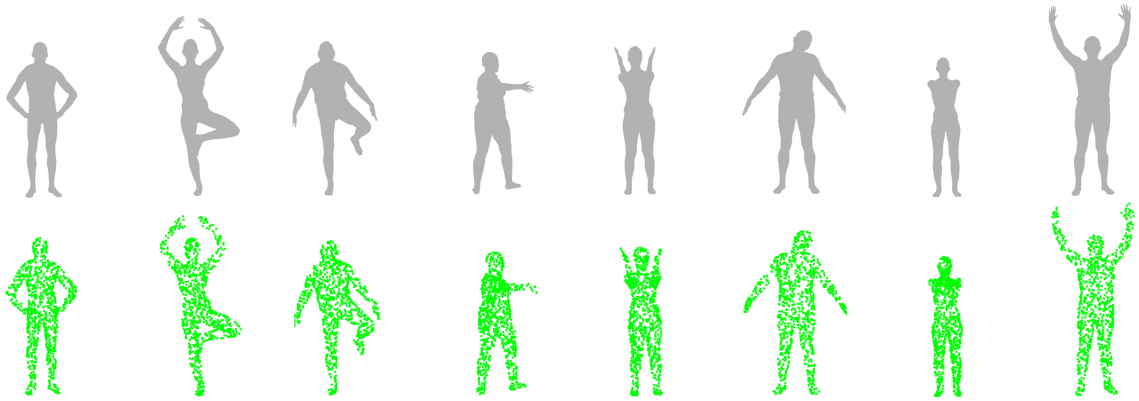


Figure 4.4: Some examples of the mesh models and point clouds from the FAUST dataset.

least computational cost (34 searches in the example), but the performance can be suboptimal as the azimuth search may not cover the actual AoA. The second approach performs a 2D search that considers all possible combinations of the azimuth and elevation directions and uses data from all receivers. It is computationally expensive (289 searches in the example) but provides the most accurate estimate. The third approach defines several levels of grids and performs the AoA search at different granularities. It starts the searching with a sparse grid, finds the peaks, defines a denser grid around each peak and performs the next search. The process can be performed iteratively until the desired resolution is achieved. It reduces the computational cost of the second approach significantly as it skips certain regions in the spectrum (50 searches in the example), at the cost of the potential possibility of missing some peaks.

4.3 Dataset

The FAUST dataset [179] is used to serve as the ground truth for the simulator, to evaluate the point cloud construction algorithms described. The datasets contain human models in the form of watertight triangulated meshes. The meshes are generated from a high-resolution camera system containing stereo cameras, RGB cameras and speckle projectors. The FAUST dataset contains 10 subjects and 30 static postures per subject, of which 10 postures are provided with aligned watertight models, giving 100 models in total.

In the simulation, the models are placed at 2 m from the radar and facing towards the radar. The height of the radar is set to be in the middle of each model. A ground truth point cloud is constructed from each model by randomly sampling M points from the surface of the mesh model, where each point is assumed to be a corner reflector. Some examples of the mesh models and point clouds are shown in Figure 4.4. The simulator will compute a signal matrix for each point cloud to simulate the IF signal that would be received by the radar when placed towards a subject, as described by Equation (4.1). The entire dataset containing the 100 models is split into 80 training data and 20 test data, where the training data is used for hyperparameters searching in the point

cloud construction algorithms, and the test data is used for evaluating the algorithms.

When generating the IF signal matrix, there are two sources of randomness: the noise term n introduced in Equation (4.1) and the random sampling of the ground truth point cloud from the mesh model. Therefore, all the evaluation processes were repeated 10 times for each mesh model and the average metrics are reported, to minimize any potential effect of the randomness.

4.4 Evaluation

4.4.1 Evaluation Metrics

To evaluate the quality of the point cloud constructed by an algorithm, it is necessary to define the evaluation metrics for comparing the output point cloud against the ground truth point cloud. Let PC_M denote the ground truth point cloud and PC_K denote the point cloud generated by the radar, which are a $M \times 3$ matrix and a $K \times 3$ matrix, respectively. It is important to note that, the point cloud construction algorithm can provide an uncertain number of points that might be different to the ground truth ($M \neq K$), and PC_K can have a non-uniform distribution while PC_M is distributed uniformly on the mesh model. The evaluation metrics should take the two point clouds PC_M and PC_K as input and measure the similarity between them. First, two points are defined to be close to each other if their Euclidean distance is less than a certain distance D. In this research, D is set to 10 cm as an empirical estimation of the error tolerance of a HAR system. Then, the following terms and metrics are defined:

- Precision: Number of points in PC_K that has at least one close point from PC_M, divided by K. It evaluates how many points in PC_K are considered to be accurate.
- Sensitivity/Recall: Number of points in PC_M that has at least one close point from PC_K, divided by M. It evaluates how well PC_K can cover the space of PC_M.
- Fowlkes–Mallows index (FMI): the geometric mean of precision and sensitivity, that is $\sqrt{\text{precision} \times \text{sensitivity}}$.
- Intersection over Union (IoU): Establish two regular 3D voxel grids for PC_K and PC_M with the voxel size set to 10 cm × 10 cm × 10 cm, consider a voxel to be occupied if there is at least one point present in the voxel, then the IoU is calculated as the number of overlapping voxels of the two voxel grid, divided by the union. The IoU evaluates the similarity of the two point clouds at the granularity of the voxel size.

An ideal system should have both high precision and high sensitivity, whereas the relative importance of the two depends on the application. In this section, the FMI, i.e. the geometric mean of precision and sensitivity, is used to indicate the performance of the system. The IoU also provides a good indication of how the generated point cloud can represent the scene. However,

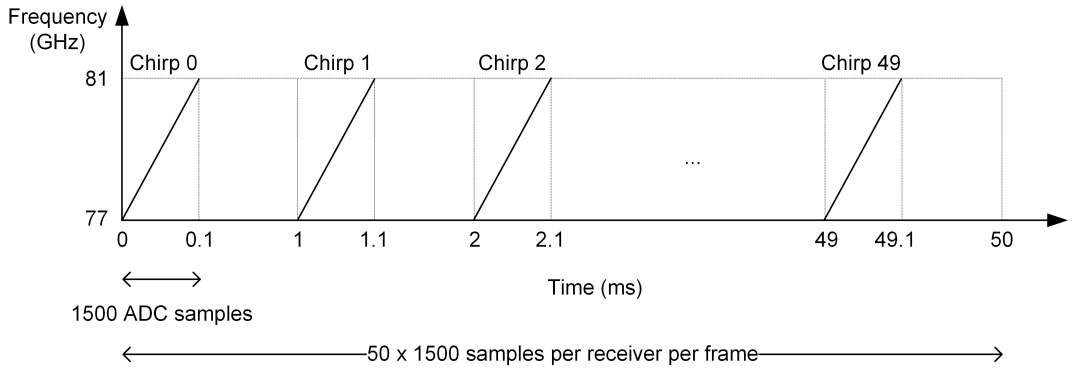


Figure 4.5: Chirp configuration of one frame in the baseline setup.

as the calculation of the IoU is highly sensitive to the voxel size and outliers, it is used as a secondary metric.

4.4.2 Data Processing Chain and Algorithms

In the first experiment, the two DPCs combined with different AoA algorithms were evaluated and compared, in terms of the quality of the estimated point cloud and the computational cost. A baseline radar and scene configuration were designed to approximate a typical setup in a common indoor environment as follows:

- The radar has one transmitter and a 4×4 uniform receiver array.
- The chirp frequency is 77 GHz to 81 GHz, the slope is 40 MHz/us, the chirp duration is 100 us, the ADC sampling rate is 15 MHz, each frame is 50 ms with 50 chirps, and each chirp has 1500 samples (as shown in Figure 4.5).
- Each human mesh model is sampled into 512 points and placed at 2 m away from the radar.
- SNR is 30 dB.
- The subject has a velocity of 0.05 m/s moving away from the radar.
- The AoA algorithm uses 512 bins to cover the $\pm 90°$ AoV, i.e. the angular resolution is 0.35°.

The velocity of the subject is introduced following the assumption that a real person cannot stay absolutely stationary during the measurement. At a velocity of 0.05 m/s and a frame time of 50 ms, the total displacement will be 2.5 mm and is considered negligible. The velocity provides a variation on the signal received at different chirps, as otherwise the multi-snapshot AoA estimation algorithms would receive an identical signal at all chirps and would yield a poor performance.

Combining the two DPCs with different AoA estimation algorithms, there are 14 methods in total to be evaluated. For each method, both the 1D search approach and the 2D sub-grid

Table 4.1: FMI (standard deviation in parentheses) comparison between the algorithms when using a 4×4 antenna array and a subject velocity of 0.05 m/s.

FMI in %	Angle-FFT		Conventional Beamforming		MVDR Beamforming		MUSIC	
	1D	2D	1D	2D	1D	2D	1D	2D
DPC1	68.3 (7.5)	68.2 (7.9)	60.6 (8.7)	67.2 (7.6)	67.7 (7.6)	74.5 (6.7)	69.7 (6.9)	77.0 (6.2)
DPC2	NA		43.7 (7.8)	46.5 (7.1)	50.2 (7.6)	53.1 (7.4)	52.7 (7.4)	53.2 (7.0)

Table 4.2: IoU (standard deviation in parentheses) comparison between the algorithms when using a 4×4 antenna array and a subject velocity of 0.05 m/s.

IoU in %	Angle-FFT		Conventional Beamforming		MVDR Beamforming		MUSIC	
	1D	2D	1D	2D	1D	2D	1D	2D
DPC1	21.2 (4.3)	22.5 (4.6)	14.6 (3.9)	20.6 (4.1)	18.0 (4.4)	23.4 (4.1)	19.0 (4.1)	22.7 (3.5)
DPC2	NA		11.2 (3.2)	12.2 (3.0)	13.2 (3.3)	14.7 (3.4)	14.6 (3.2)	14.6 (3.3)

approach described in Section 4.2.3 are included. For the 2D angle-FFT method, the full-grid approach is used instead of the sub-grid approach, since the benefit of the lower computational cost is less significant for FFTs. The algorithms will be referred to using the format "DPC-Method-1D/2D" throughout the chapter. For example, DPC1-Conventional-2D refers to the conventional beamforming method in DPC1 that uses a 2D steering vector search. The angle-FFT method is not applicable in DPC2 as it is not a multi-snapshot algorithm. Algorithms in DPC1 include a CFAR peak detection step on the Range-Doppler heatmap, where the optimal parameters for the CFAR were searched on the training dataset. Then, the performance of the algorithms on the test dataset were evaluated and compared. The result is shown in Table 4.1 and Table 4.2 as FMI and IoU (in % and with the standard deviation in parentheses), respectively.

There are a few important observations from the experiment. Even though the subject had a low velocity, the DPC1 with a Doppler-FFT outperformed the other significantly. One main reason is that, as the number of antennas is much lower than the number of signals, the AoA estimation algorithm can fail to distinguish points with a close angle. Instead, these points will be identified as one strong signal source. On the contrary, the CFAR peak detection step in DPC1 picks a set of points around the peak that are above the CFAR threshold. As these points also contribute to the point cloud, the output becomes denser and the sensitivity is improved. This effect can be observed from the example detection shown in Figure 4.6.

In terms of the different algorithms, the MVDR and MUSIC methods outperformed the angle-FFT and conventional methods, at the expense of higher complexity. Meanwhile, all the 2D methods outperformed the 1D methods due to a more fine-grained resolution (as shown earlier in Figure 4.3). The best performance was achieved with the DPC1-MVDR-2D and DPC1-MUSIC-2D

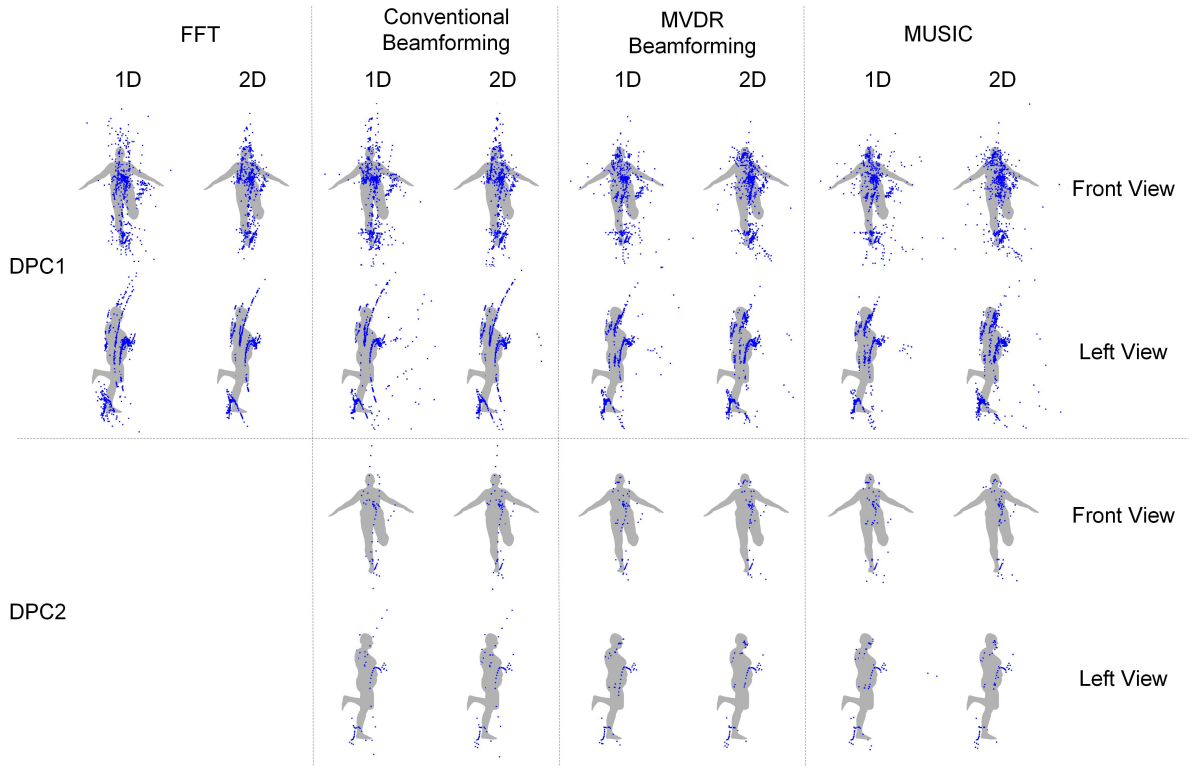


Figure 4.6: Examples of the radar detection using the different algorithms, when using a 4×4 antenna array and a subject velocity of 0.05 m/s.

methods, with an FMI of 74.5% and 77.0%, respectively. However, the IoU metrics show that the point clouds were still far from the objective of high-accuracy scene reconstruction, as the highest IoU was only 23.4%. It can be seen from Figure 4.6 that, while the distribution of the point cloud mostly fitted the subject, the distribution was not even and there were body parts (like the hands) that received fewer points. Therefore, there is still a big gap before the radar output can be directly used by applications that require high quality data.

Table 4.3 compares the algorithms in terms of computational complexity. The algorithms were run using the same dataset and parameters multiple times. The algorithms were written in Python without any processor-specific optimization and were run on one Intel i7-9700K CPU core. The result is shown as the relative execution time of each algorithm when compared with the DPC1-FFT-1D method (the most lightweight method) and normalized with the number of detected points, to give an indication of their relative complexity. All the 2D methods have a higher complexity than the 1D methods. For algorithms in DPC1, the 1D angle-FFT method has the lowest computational cost. With the sub-grid optimization, the complexity of the 2D beamforming and MUSIC methods can be kept at around twice the 1D methods. The complexity without the sub-grid optimization is expected to be much higher, as can be estimated from the difference between the 2D and 1D angle-FFT methods. When considering both the complexity and the performance, the DPC1-FFT-1D method provides a good trade-off between them. The

Table 4.3: Normalized execution time comparison between the algorithms using the baseline setup.

Normalized Complexity	Angle-FFT		Conventional Beamforming		MVDR Beamforming		MUSIC	
	1D	2D	1D	2D	1D	2D	1D	2D
DPC1	1.00	13.42	4.38	9.32	3.51	8.99	4.02	8.85
DPC2	NA		5.69	12.38	5.31	10.89	5.67	10.61

Table 4.4: Relative FMI difference of the algorithms when using a 4×4 antenna array and a subject velocity of 0.5 m/s in comparison to 0.05 m/s.

FMI in %	Angle-FFT		Conventional Beamforming		MVDR Beamforming		MUSIC	
	1D	2D	1D	2D	1D	2D	1D	2D
DPC1	+8.3	+11.4	+12.1	+10.7	+10.4	+8.8	+10.1	+7.7
DPC2	NA		+4.6	+2.6	+5.3	+4.6	+8.4	+5.6

MVDR methods and MUSIC methods in DPC1 give the best performance at the cost of 9x higher complexity and require additional efforts on the hardware and implementation.

4.4.3 Subject Velocity

The motion of the subject being sensed has a significant impact on the detection output. In DPC1, a higher velocity makes a subject easier to be identified in the Range-Doppler heatmap. Due to the relative position difference between the body parts of the subject, they will have a different radial velocity with respect to the radar, making them distinguishable in the Range-Doppler heatmap. In DPC2, a higher velocity increases the variance of the signal between chirps and allows a better estimate of the data covariance matrix. To verify the theorem, an experiment was carried out using the same configuration as the baseline setup, except that the velocity of the subject was set to different values from 0.1 m/s to 1 m/s. The ground truth point cloud was taken as the average position of the subject during the motion.

Table 4.4 and Table 4.5 show two examples of the experiment where the subject velocity was set to 0.5 m/s and 1 m/s, respectively. When compared with Table 4.1, all algorithms achieved a 2.6% to 14.5% improvement in terms of the FMI when the subject had an increased velocity. Figure 4.7 shows the FMI and IoU of the DPC1-MUSIC-2D method with different subject velocities from 0.1 m/s to 1 m/s. An overall positive correlation can be observed between the subject velocity and the detection performance, and the impact is the most obvious at lower velocities (<0.5 m/s). Some examples of the detection at 1 m/s are shown in Figure 4.8.

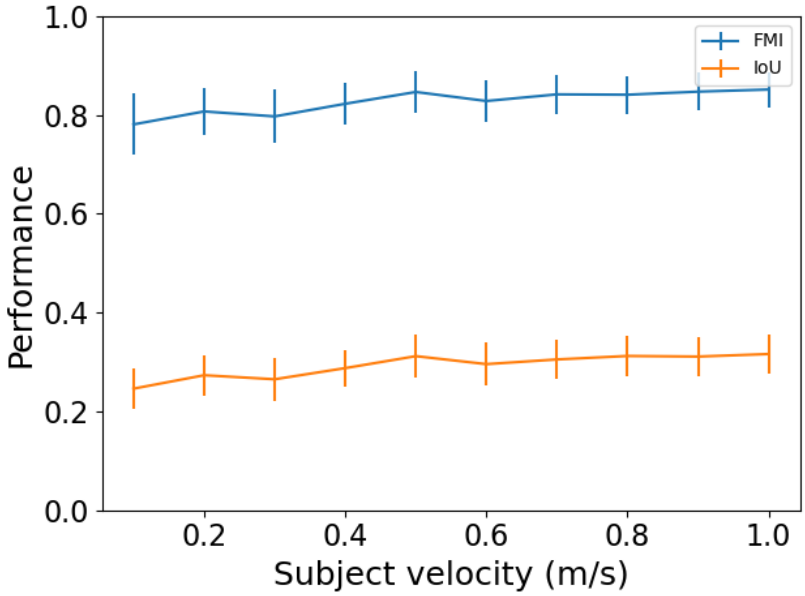


Figure 4.7: FMI of the DPC1 2D MUSIC algorithm with different subject velocities.

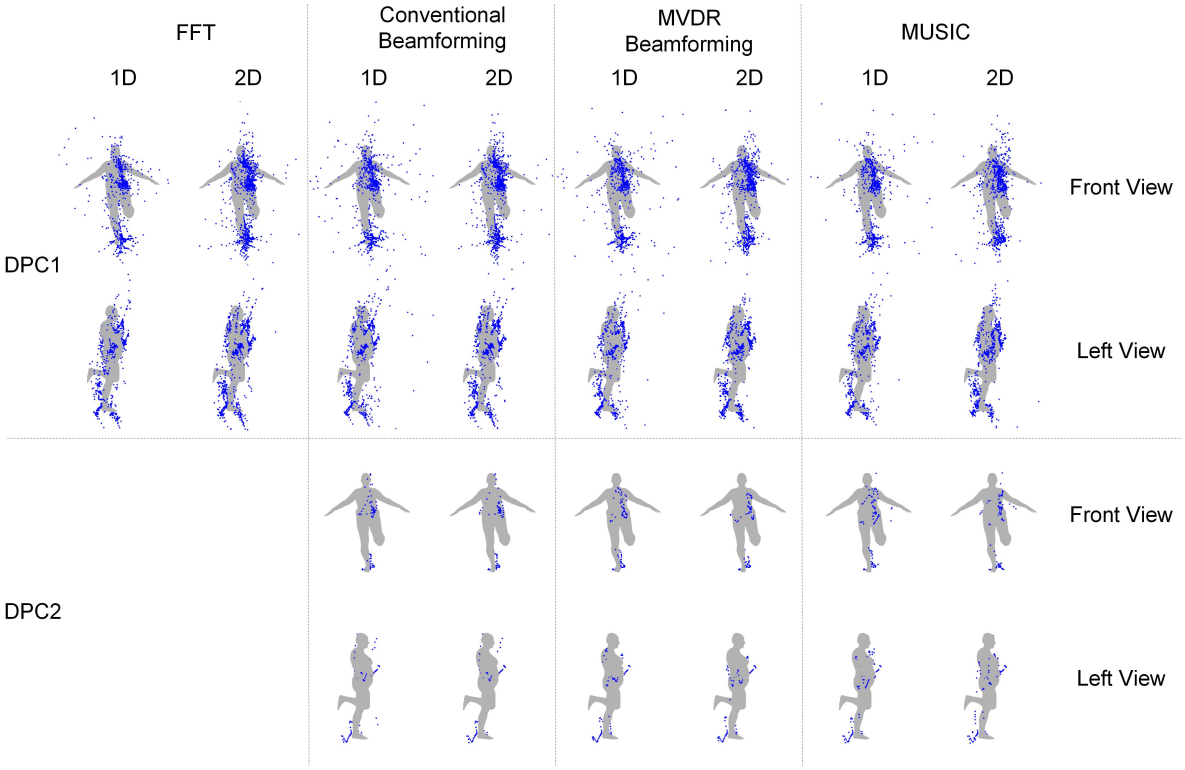


Figure 4.8: Examples of the radar detection using the different algorithms, when using a 4×4 antenna array and a subject velocity of 1 m/s.

Table 4.5: Relative FMI difference of the algorithms when using a 4×4 antenna array and a subject velocity of 1 m/s in comparison to 0.05 m/s.

FMI in %	Angle-FFT		Conventional Beamforming		MVDR Beamforming		MUSIC	
	1D	2D	1D	2D	1D	2D	1D	2D
DPC1	+9.7	+13.0	+14.5	+12.8	+12.0	+9.8	+11.6	+8.2
DPC2	NA		+5.3	+2.6	+4.3	+3.4	+9.3	+5.8

Table 4.6: Performance difference when using a 4×4 antenna array and a subject velocity of 0.05 m/s in a low SNR environment (5 dB in comparison to 30 dB).

FMI in %	Angle-FFT		Conventional Beamforming		MVDR Beamforming		MUSIC	
	1D	2D	1D	2D	1D	2D	1D	2D
DPC1	-8.1	-5.8	-5.5	-5.6	-6.4	-6.3	-5.7	-6.2
DPC2	NA		+2.2	+2.8	+1.7	+2.7	+1.6	+2.4

Table 4.7: Performance difference when using a 4×4 antenna array and a subject velocity of 0.5 m/s in a low SNR environment (5 dB in comparison to 30 dB).

FMI in %	Angle-FFT		Conventional Beamforming		MVDR Beamforming		MUSIC	
	1D	2D	1D	2D	1D	2D	1D	2D
DPC1	-7.8	-6.2	-7.2	-6.1	-8.3	-7.5	-8.9	-7.4
DPC2	NA		+2.5	+2.8	+1.0	+0.8	-0.7	+0.9

4.4.4 SNR

In a practical environment, a radar system can experience noise from different sources, such as the thermal noise of the radar chip. The SNR also depends on the distance between the radar and the subject, as the signal power drops quickly along with the distance. In the simulator, the SNR can be controlled by the power of the noise term n in Equation (4.1). In this section, the performance of the algorithms between a high SNR environment (40 dB) and a lower SNR environment (5 dB) is compared. Two experiments were carried with the subject velocity set to 0.05 m/s and 0.5 m/s, respectively. The results are shown in Table 4.6 and Table 4.7.

In the low SNR environment, all the algorithms in DPC1 experienced a similar drop in performance, as expected. However, the algorithms in the DPC2 showed a higher performance. The reason is that the higher noise affected the model order estimation step and the system tends to report a higher number of points. Taking the DPC2-Conventional-2D method as an example, the average size of the detected point cloud was found to be 20.3% higher in a low SNR environment than in a higher SNR environment. However, this was still insufficient to reach a similar performance as DPC1.

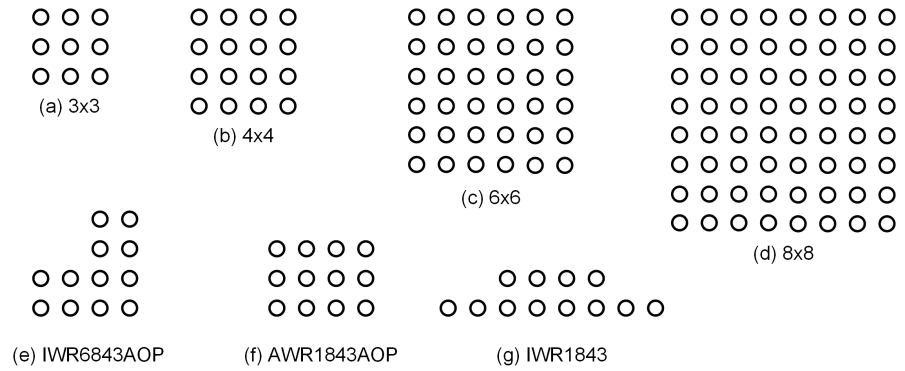


Figure 4.9: The list of receiver layouts being evaluated. (a)-(d) are square antenna arrays. (e)-(f) are non-regular antenna arrays implemented on TI radars.

Table 4.8: Performance comparison between different antenna layouts using the baseline configuration and the DPC1-MUSIC-2D algorithm (standard deviation in parentheses).

Antenna Layouts	a	b	c	d	e	f	g
FMI in %	76.7 (6.4)	77.0 (6.2)	76.8 (4.8)	77.9 (4.5)	72.4 (6.3)	77.8 (5.9)	65.0 (6.0)
IoU in %	23.4 (4.1)	22.7 (3.5)	20.5 (2.8)	18.8 (2.7)	20.8 (4.0)	23.9 (4.2)	17.0 (3.5)

4.4.5 Antenna Layout

Theoretically, the antenna layout determines the angular resolution that an AoA estimation algorithm can achieve. The more receivers in one direction, the higher resolution the radar can measure. However, this is questionable when the signal sources are spatially close and continuous. Meanwhile, having more antennas also increases the cost of the hardware, as more circuit components, processing units and memory would be required. Therefore, it is beneficial to study the relationship between the antenna layout and the output quality and find the optimal trade-off for an application.

Common commercial mmWave radars use up to three transmitters and up to four receivers, giving up to twelve virtual receivers as a receiving array. Some radar models are designed for automotive applications and prioritize the azimuth direction, while others are designed for general purpose applications and have a similar resolution in both the azimuth and elevation directions. In this section, common antenna layouts implemented on the TI radars are evaluated and compared, as well as a few square-shape antenna layouts that are more common in research projects, as listed in Figure 4.9. The same radar configuration and scene setup in Section 4.4.2 were used. The experiment compares the antenna layouts using the DPC1-MUSIC-2D algorithm (the best performing algorithm) to fully exploit the potential of the antennas. The result is shown in Table 4.8 and Figure 4.10.

It can be seen that most antenna layouts had similar performance, except the layout (g)

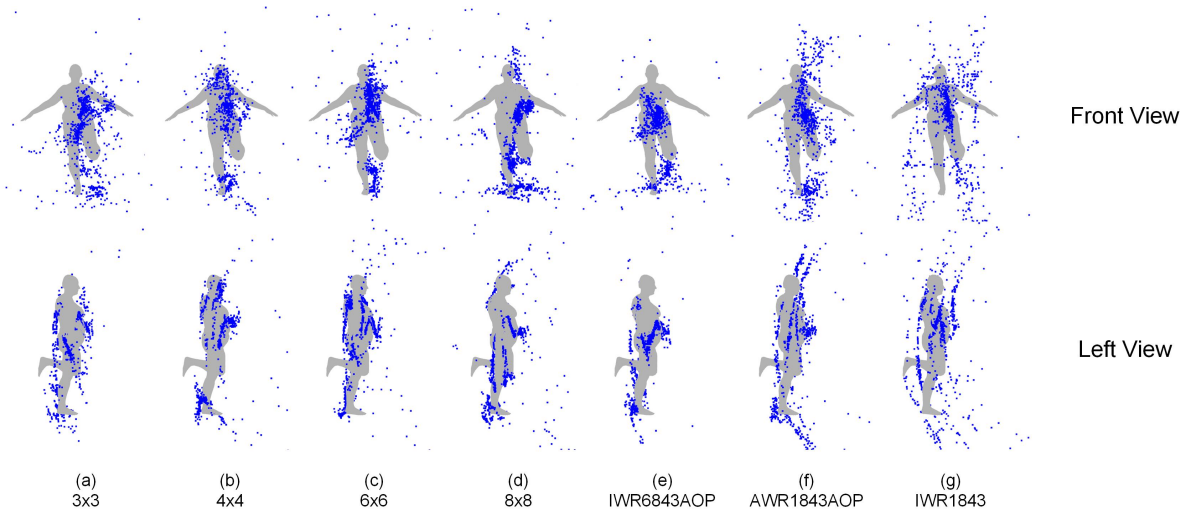


Figure 4.10: Examples of the radar detection using the different antenna layouts, when using a 4×4 antenna array and a subject velocity of 0.05 m/s.

which had a worse performance as it is designed for automotive applications. The layout (e) has a non-uniform antenna distribution that slightly affected its performance. All other layouts showed a similar performance regardless of the antenna size. Therefore, considering the increased hardware cost and computational cost of increasing the number of antennas, a small antenna size can be preferable for 3D sensing applications.

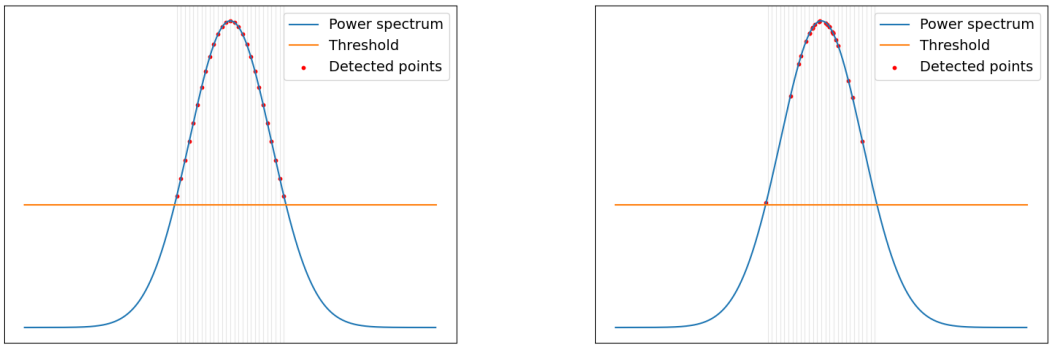
4.4.6 Chirp Configuration

The chirp configuration can have various effects on the distance detection and velocity detection, as discussed in Chapter 3. These factors can indirectly affect the quality of the final point cloud. In this section, three different chirp configurations are tested and compared against the baseline configuration in Section 4.4.2. The details of the three configurations (named A, B and C) and the performance are shown in Table 4.9. Each configuration has certain parameter cut to 80% to evaluate the effect on the output. Configuration A had an 80% reduced chirp slope and, hence, a reduced effective bandwidth from 4 GHz to 3.2 GHz. Configuration B had an 80% reduced ADC sampling rate that reduced the samples per chirp from 1500 to 1200. Configuration C had an 80% reduced number of chirps per frame, from 50 to 40. All other parameters were kept the same as the baseline with the DPC1-MUSIC-2D algorithm.

The result shows that the performance can be strongly affected by the effective bandwidth and the number of chirps. The former affects the distance resolution of the detection, and the latter affects the Doppler resolution. Reducing either of these parameters reduces the accuracy of the range-Doppler heatmap and the estimation of the covariance matrix. On the other hand, the effect of reducing the ADC sampling rate and the number of samples per chirp is much less significant.

Table 4.9: FMI (standard deviation in parentheses) comparison between four chirp configurations using the DPC1-MUSIC-2D algorithm.

Chirp Configuration	Baseline	A	B	C
Slope of the chirp	40 MHz/us	32 MHz/us	40 MHz/us	40 MHz/us
ADC sampling rate	15 MHz	15 MHz	12 MHz	15 MHz
Chirps per frame	50	50	50	40
FMI in %	77.0 (6.2)	71.1 (6.8)	76.5 (6.0)	70.2 (6.0)



(a) Points detected without SRPC.

(b) Points detected with SRPC.

Figure 4.11: Using SRPC algorithm to improve the resolution and distribution of the data.

4.5 Super-resolution Point Cloud Construction Algorithm

It can be seen from Figure 4.6 and Figure 4.8 that the constructed point clouds can be noisy and the distribution of the points can be imbalance. One major reason is that the point cloud construction relies on the peak detection result over the range-Doppler-FFT spectrum, so the distribution of the points will be limited by the resolution of the FFT, and the points will have a discrete distribution in the range domain (as the curve-like data from the left view). Although it is possible to improve this resolution, such as zero padding the data before applying the FFT, it would also increase the computational cost and memory consumption. Meanwhile, there are false detected points due to the outliers from the peak detection stage. To address the mentioned issue and improve the quality of the constructed point cloud, a novel super-resolution point cloud construction (SRPC) algorithm is proposed.

The SRPC algorithm aims to improve the distribution of the point cloud and make it span more naturally in the spatial space. The rationale of the algorithm is shown in Figure 4.11. When detecting peaks in a range-Doppler spectrum or an angle spectrum, a common approach is taking all points above a static or dynamic threshold, where the distribution of the points is limited by the resolution of the original data. An example of this effect is shown in Figure 4.11a, where the grid represents the resolution of the data and all the detected points must fall on the grid. The

SRPC algorithm aims to return a set of points that have a higher resolution than the original data and fall more naturally on the distribution curve, as shown in Figure 4.11b.

The algorithm can be broken down into the following steps. First, the power spectrum is upsampled into the desired resolution using linear interpolation. Then, for each of the originally detected points i, the algorithm randomly samples n_i points around it with a probability distribution being the amplitude of the upsampled power spectrum. The value of n_i is calculated as:

$$n_i = \frac{p_i \cdot \alpha_{SRPC}}{th} \tag{4.3}$$

where p_i is the power of the point, th is the threshold of the peak detection algorithm, and α_{SRPC} is a global hyperparameter that controls the aggressiveness of the algorithm. The term p_i ensures that a point with higher power will be sampled into more points, as the power indicates the confidence that a point can represent a real signal source. The parameter α_{SRPC} amplifies the importance of p_i, where a higher α_{SRPC} pushes the distribution of the points towards the peak of the spectrum and gives a more dense distribution. The sampling process is repeated for each point i to form a new point list. Finally, a number of points of the length of the original detection are randomly selected from the new point list, so that the total number of detected points is kept the same and the computational cost of the rest of the system is not affected. Since the algorithm tends to sample more points at higher power, the distribution of the final points will also tend to be around higher powers, and, hence, gives a more natural distribution regarding the power spectrum and overcomes the limitation of the original data resolution.

When constructing the point cloud, the SRPC is applied when detecting peaks from the range-FFT spectrum and detecting peaks from the angle spectrum in the AoA estimation step. The former improves the data distribution in the range domain and eliminates the curve-like effect when looking at the point cloud from the left view. The latter improves the data distribution in the angle domain so that the points tend to span into the space rather than appearing as a dense cluster. Meanwhile, since the points will be distributed around higher powers, the probability of outliers will be reduced.

To evaluate the proposed SRPC algorithm, it has been inserted into the DPC1-FFT-1D and DPC1-MUSIC-2D methods mentioned in Section 4.4.2 when using the baseline setup. The two methods are chosen as they represent the most lightweight algorithm and the most accurate algorithm, respectively. Since the SRPC is likely to produce point clouds with different sizes and to ensure a fair comparison, a fixed number of 512 points will be randomly taken from the point cloud generated by each algorithm for the evaluation. The result is shown in Figure 4.12. After applying the SRPC algorithm, the distribution of the point cloud appears to be more natural and better distributed around the ground truth, and the outliers in the original detection have been reduced. The result of a quantitive evaluation is shown in Table 4.10. The performance without SRPC drops slightly when compared with Table 4.1 because the output size has been forced to be 512, but both metrics have improved after applying SRPC. Therefore, it is shown that the SRPC

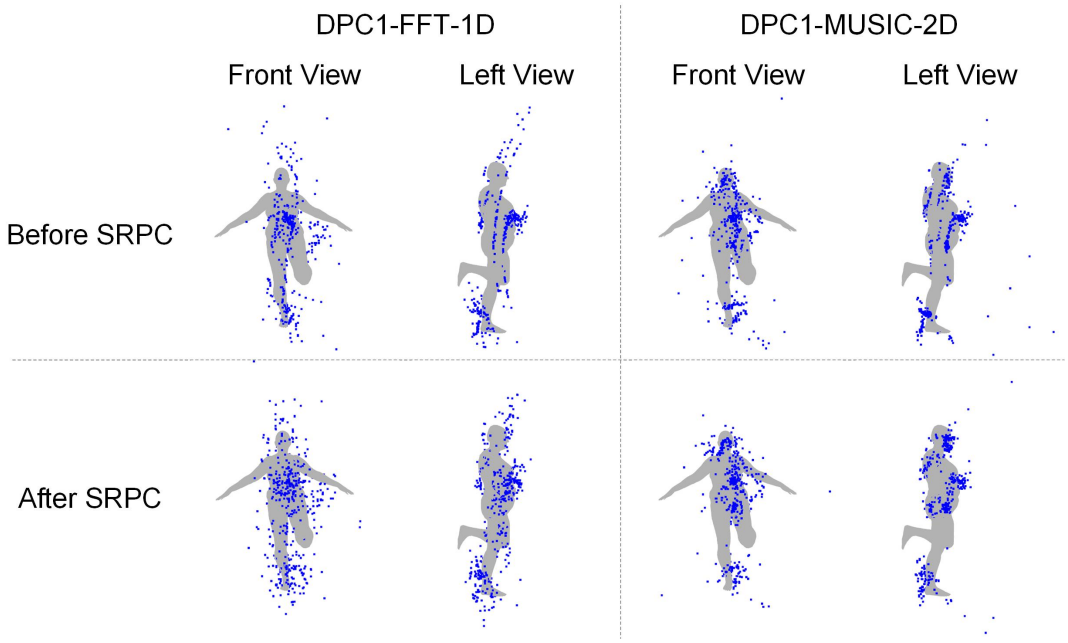


Figure 4.12: Examples of point clouds constructed with and without the SRPC algorithm.

Table 4.10: Performance comparison of two algorithms with and without SRPC.

	DPC1-FFT-1D		DPC1-MUSIC-2D	
	FMI	IoU	FMI	IoU
Without SRPC	64.9	20.2	72.1	22.9
With SRPC	69.5	23.6	72.9	25.9

algorithm successfully improves the data point distribution, reduces the outliers and produces a more natural point cloud that can be potentially preferable for higher-level applications. Future work of this research includes an efficient hardware implementation of this algorithm using the radar on-chip processors so that it can be further verified in real-world scenarios, as well as an evaluation on its effectiveness in higher-level applications like posture estimation.

4.6 Conclusion

In this chapter, a mmWave radar simulator is presented. The system is used to evaluate the ability of the mmWave radar as a 3D imaging sensor. A mmWave radar dataset is constructed using the FAUST dataset as the ground truth to provide 3D mesh models of human subjects, from which mmWave radar IF signals are simulated and used to evaluate different point cloud construction algorithms. The FMI and IoU metrics are defined to evaluate the quality of the generated point cloud. The evaluation is performed regarding a set of different factors, including the DPCs, AoA estimation algorithms, subject velocity, SNR, antenna layout and chirp configuration. It was found that the DPC combining a range-Doppler-FFT and a single-snapshot AoA estimation

algorithm gives better performance. Among all the AoA estimation algorithms, the angle-FFT method gives a good trade-off between high performance and low computational cost, whereas the more advanced AoA estimation algorithms, like MVDR and MUSIC, give the best performance at up to 9x higher computational time. The velocity of the subject helps significantly in the detection, as the algorithms are better at detecting a moving subject than a stationary object. When comparing common antenna layouts, large square antenna arrays give the best performance, but the advantage is not significant in a 3D sensing application when the data sources are spatially close and continuous. It is shown that the performance of the point cloud detection benefits from higher effective bandwidth and a higher number of chirps per frame. Finally, a novel SRPC algorithm has been proposed for improving the resolution and distribution of the point cloud and reducing the probability of outliers. The algorithm applies to the range-Doppler-FFT peak detection stage and the AoA estimation stage and detects points at a higher resolution that fits the power spectrum better. When evaluating the algorithm using the simulation system, it has been shown that the algorithm can successfully improve the data distribution and produces a more natural point cloud.

CHAPTER 5

HUMAN DETECTION AND TRACKING

In this chapter, a novel human detection and tracking system using mmWave radars is presented. It uses two radars from different perspectives to detect the presence of people in an office environment and track their locations. The system achieves 90.4% sensitivity and 98.6% precision when detecting up to four people in the room. The content of this chapter has been published in the IEEE Aerospace and Electronic Systems Magazine [20] [1] and in a patent application [23].

The rest of the chapter is organized as follows. Section 5.1 gives an overview of this work. Section 5.2 presents the experimental setup of the system. Section 5.3 presents a real-time software framework for operating the radars and implementing the data processing algorithms. Section 5.4 presents a study on the possible signal interference when operating multiple radars simultaneously. Section 5.5 presents the human detection and tracking algorithm. Section 5.6 shows the evaluation result of the system and a comparison to the state-of-the-art systems. Section 5.7 concludes the chapter.

5.1 Overview

In HAR, human detection and tracking is often an implicit requirement of many applications. For example, advanced tasks like posture estimation or identification are often triggered only when a person has been detected and the data corresponding to the subject has been captured, and a continuous operation requires the position of the person to be tracked over time. The performance of these tasks can be largely affected by the accuracy of the detection and tracking system.

[1] 2021 IEEE. Reprinted, with permission, from H. Cui and N. Dahnoun, "High precision human detection and tracking using millimeter-wave radars," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 1, pp. 22–32, 2021.

As discussed in Chapter 4, mmWave radars can capture high-resolution information about the subject in the scene, but the quality of the detection can depend on a lot of factors, such as the velocity of the person and the noise in the environment. Even in an optimal situation, the accuracy of the detected point cloud is still far from camera systems or LIDAR systems, and there can be a significant amount of false detection that falls around the real location of the subject. Therefore, it is essential to quantitively evaluate the capability of mmWave radars in detecting the presence of people in a scene and continuously tracking their location.

This chapter presents a novel human detection and tracking system using mmWave radars. A novel data processing chain is proposed to filter out the noise and determine the presence of any people in the scene, and a tracking module is designed to track the status of the people over time. It will show that mmWave radars can have a high detection sensitivity when people enter the scene. However, using a single radar can raise a high number of false alarms, but the precision could be improved significantly with the use of two radars. It will show that the system outperforms RFID and WiFi-based systems and achieves similar performance as other UWB or mmWave systems.

5.2 Experimental Setup

Throughout the experiments, a radar configuration tuned for indoor environments was used, with a maximum range of 8 m, a range resolution of 4 cm, a maximum velocity of 1 m/s and a velocity resolution of 0.1 m/s. The time of each chirp is 125 us, with 10 us idle time (for resetting the chirp) and 115 us chirp ramp time. With a slope rate of 35 MHz/us, the full 4 GHz bandwidth available for the radar was utilized. As the target use case is human activity recognition, the CFAR threshold was set to a relatively low value so that enough data can be received for post-processing.

The hardware setup of the system is shown in Figure 5.1. Two radars were placed at different perspectives and the camera was placed on the top of one radar. The camera is used only to provide the ground truth for evaluating the system and is not involved in the detection process. Both of the radars are the IWR1443 model with the same antenna configuration (as shown in Figure 3.5), and the detection area is defined as the intersection area in the sight of both radars. The radars are calibrated offline, where a rotation matrix and a translation matrix are generated for each radar based on their orientations and locations. The metrics are recorded in a configuration file. They will be loaded into the Frame Manager module at runtime and be used to translate the detection results into one coordinate system.

5.3 Real-Time Software Framework

For real-time data streaming and processing, a multithreaded software framework for managing the radar and processing the radar data has been designed and implemented. It can process

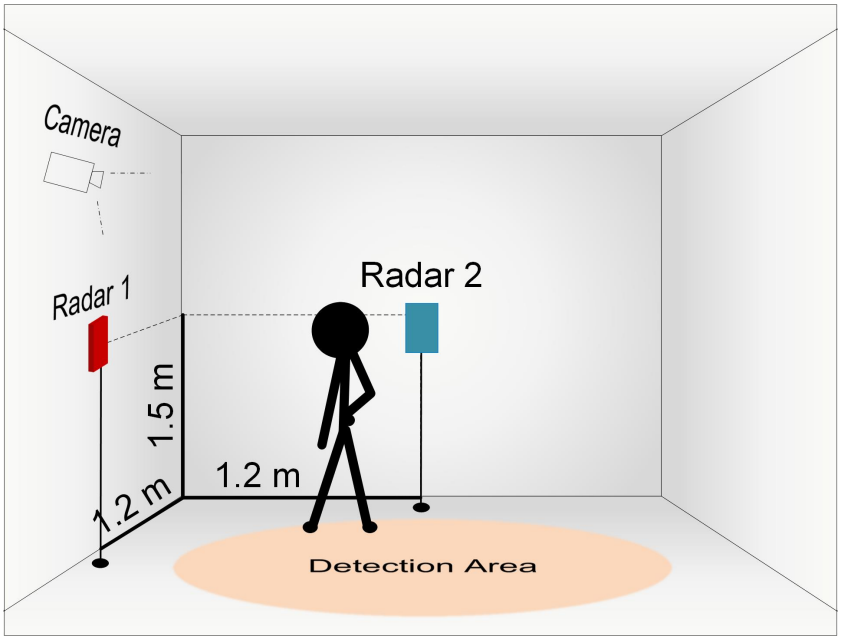


Figure 5.1: Hardware setup of the two radars for human detection.

either the raw IF signal or the point cloud data. In this section, the point cloud data processing is discussed.

The framework is capable of managing multiple radars simultaneously. An overview of the software framework is shown in Figure 5.2. It uses one thread per radar to manage the communication to the radar and the data transmission, and one thread to fuse and process the data. First, The data processing consists of two stages: the first stage applies to the point cloud received by each radar independently, and the second stage applies to the fused data from all the radars. The system is written in Python and has the following main modules:

Radar Handler (RH): connects to the radar through the serial ports, loads the configuration files, sends them to the radars, receives detection results, packs them into data matrices and applies the first stage of the data processing. The data matrices containing the point cloud are referred to as one frame.

Frame Processor (FP): takes one frame of data as input, performs customized data processing tasks and outputs data in the same format. A customized DPC can be formed by using a sequence of FPs.

Visualizer: manages the DPC composed of FPs and the visualization of the data in 2D or 3D. It also allows cameras or other peripherals to connect to the system and interact with the FPs.

Through the use of a configuration file, the user can specify the number of radars and each radar's model, serial port number, the antenna and chirp configuration, and the relative position in the form of a rotation and translation matrix. When the system starts, one independent thread will be spawned for each radar, referred to as the radar threads. These threads will each execute an RH module, connect to the serial ports and handle the communication between

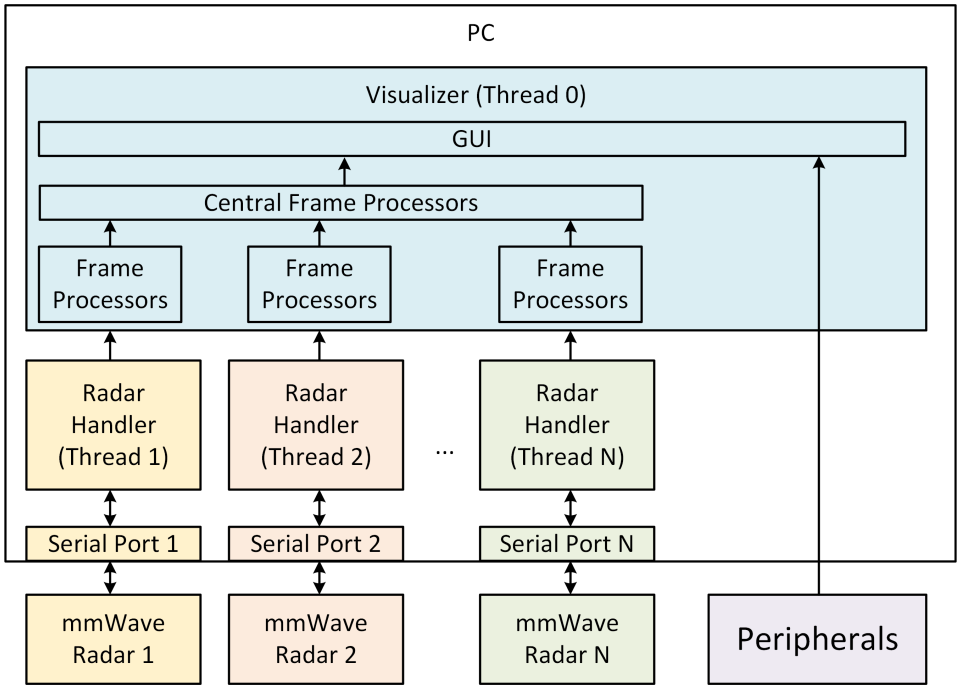


Figure 5.2: Software framework for managing multiple radars and applying customized processing chain.

the host and the radar. In addition, one visualizer thread will be spawned with a Visualizer module. The visualizer thread manages a number of FPs to achieve a customized post-processing chain on the received data. These FPs include individual FPs that are applied independently to each radar's data, and central FPs that are applied to the fused data from all radars. One data queue (first-in-first-out) will be created for each radar thread and shared with the visualization thread. The radar threads will read data from the serial ports continuously, parse them into an appropriate matrix format, and push the result into the shared queue. Any old data in the queue that has not been read will be replaced by the new data, so that the visualizer thread will always have the latest frame from the radar. This design minimizes the effect of out-of-synchronization caused by inconsistent processing speed between the radar threads. The visualizer thread will fetch the data from each queue, perform the user-defined post-processing chain, display the result and fetch the next batch of data. The system operates in real-time from data collection to result visualization. The performance bottleneck of the system will be either the transmitting speed of radars or the processing speed of all FPs, whichever is slower. The system works best on multicore CPUs when each thread can utilize one physical CPU core, but it can also work on single-core machines with reduced performance. The following sections provide a more detailed discussion of each module.

5.3.1 Radar Handler

As discussed in Section 3.3.4.2, a radar has two serial ports that can be accessed by the PC, one for configuring the radar and the other for transmitting the data. The RH performs the following tasks: opens up the two serial ports as specified by the system configuration file, loads the commands from the antenna configuration file, writes the commands to the configuration port, checks the response of each command and starts listening to the data port upon success. When decoding data from the data port, it searches the packet header containing the point cloud (as shown in Figure 3.11), parses the packet and extracts the point cloud. The data will be re-arranged into an $N \times 3$ matrix, where N is the number of detected points and 3 is the x-y-z coordinates. The thread will then push the matrix into the queue and continue listening to the port for the next data packet.

5.3.2 Frame Processor

FPs define the data processing algorithms to be performed on each frame of data. The three types of FPs used throughout this research are listed below:

Temporal Stacking: This module stores the frames at each timestamp using a fixed-length first-in-first-out queue. The output of this module is the sum of all the point clouds in the queue. During experiments, it was found that stacking data in the temporal domain can help to stabilize detection, as data points from real objects will be emphasized, but the noise will not.

Clustering: This module groups data points in one frame into clusters according to their Euclidean distances between each other, and filters out small clusters with low numbers of points. The DBSCAN (density-based spatial clustering of applications with noise) algorithm is used for clustering, which does not require prior knowledge of the scene and can extract all qualified clusters. This module shows a significant effect in reducing noise.

Background Subtraction: This module attempts to learn the environment during the first few frames (e.g. one minute). It collects the data in these frames, performs the DBSCAN clustering algorithm, and records the detected objects in a local database as clutter. Then, for new frames, the module will compare each new cluster with the clutter in the database and filter out those with a similar size and location. This module can be useful when irrelevant static objects are presented in the area and should be removed.

The Temporal Stacking module is applied individually to each radar frame, whereas the other two modules are applied to the fused frame. The FP module provides a standard interface for any other customized operations, such as the neural network module that will be discussed in Chapter 6. All modules work independently and can be loaded as per user requirement, and additional functionality can be easily integrated into the system with new modules, which allows the system to be adapted and deployed for different use cases.

5.3.3 Visualizer

The Visualizer is responsible for receiving the data matrices from the shared queues, applying the individual FPs, combining the result into a single matrix, applying the central FPs and visualizing the final result in a graphical user interface (GUI). The Visualizer shares a data queue with each of the RH modules. Since the RH modules always push the latest frame into the queue, the Visualizer always has access to the latest frame from each radar and can query the data as soon as it finishes the last frame. In other words, data from different devices are implicitly synchronized based on their arrival time at the Visualizer. It is also responsible for saving the data to the local drive if an offline dataset needs to be established. While combining the data from different radars, it applies a pre-defined rotation and translation to the point cloud according to their relative positions and perspectives, so that a consistent coordinate system can be established between all devices.

5.3.4 Peripherals

The framework allows peripheral devices to be incorporated into the Visualizer. For example, an optical camera can be connected to the PC and loaded into the system as a peripheral device, to provide visual information or serve as the ground truth during data collection. Another important peripheral device that will be used is the heart rate monitor, as will be discussed in Chapter 7. A separate thread will be spawned for each peripheral device to manage the device operation and data transmission, similar to an RH module.

5.4 Signal Interference between Multiple Radars

When using multiple radars, it is important to ensure that they do not interfere with each other. In this section, it will be shown that the probability of interference between multiple radars is very low even without any explicit synchronization.

Assuming that a maximum distance of 6 m is being measured, then the time-of-flight of a round trip would be 0.04 us. With a 35 MHz/us slope (as used in this work), this time period gives a frequency change of around 1.4 MHz, as shown in Figure 5.3. Assuming that two radars are working simultaneously, based on Equation (3.2) and Equation (3.3), the transmitter signal and the receiver signal of the two radars can be represented as Equation (5.1) to Equation (5.4) (the amplitude term has been omitted as they are not important for this section):

$$S_{tx1}(t) = cos(2\pi f_0 t + \pi S t^2) \tag{5.1}$$

$$S_{rx1}(t) = cos(2\pi f_0 (t-\tau) + \pi S (t-\tau)^2) \tag{5.2}$$

$$S_{tx2}(t) = cos(2\pi f_0 |t-t_d| + \pi S (t-t_d)^2) \tag{5.3}$$

$$S_{rx2}(t) = cos(2\pi f_0 |t-t_d-\tau| + \pi S (t-t_d-\tau)^2)) \tag{5.4}$$

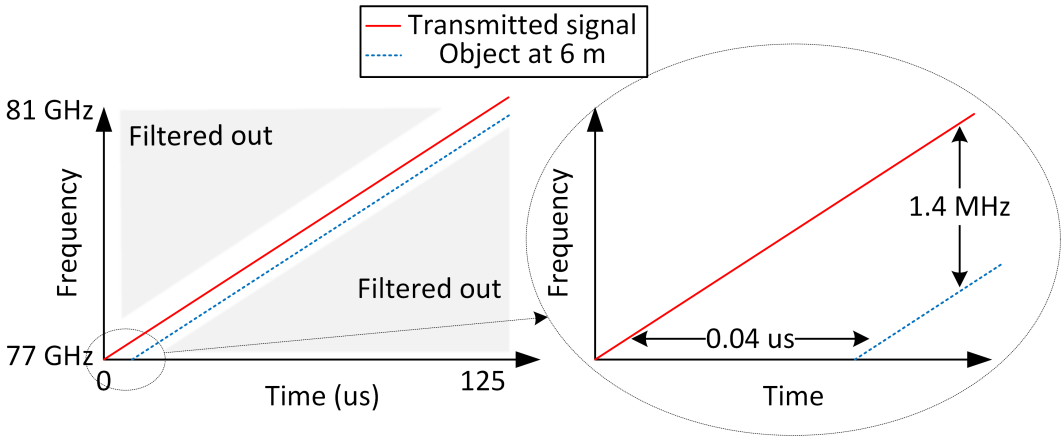


Figure 5.3: Transmitted and received signals when detecting an object at 6 m.

where f_0 is the starting frequency of the chirp 77 GHz, τ is the time-of-flight 4 us, and t_d is the difference of the starting time of the two transmitters and can be either positive or negative. Assuming the signals $S_{tx2}(t)$ and $S_{rx2}(t)$ are also detected by the first radar, then the mixer will multiply the signals as $S_{tx1}(t)\cdot(S_{rx1}(t)+S_{tx2}(t)+S_{rx2}(t))$, which, similar to Equation (3.8), can be derived to:

$$\begin{aligned} S_{tx1}(t)\cdot(S_{rx1}(t)+S_{tx2}(t)+S_{rx2}(t)) = & cos(2\pi(S\tau)t)+2\pi f_0\tau) \\ & + cos(2\pi S|t_d|t)+2\pi f_0|t_d|) \\ & + cos(2\pi S|\tau+t_d|t)+2\pi f_0|\tau+t_d|) \end{aligned} \tag{5.5}$$

The mixed signal can be viewed as the sum of three sinusoids with three different frequencies, $S\tau$, $|St_d|$, and $|S(\tau+t_d)|$. The first term is the desired signal, whereas the other two are the possible interference signals. By configuring the ADC sampling rate and with the help of the built-in digital filter, frequencies beyond 1.4 MHz could be filtered out. In other words, the radar will only keep the detection within the 0.04 us period (the 6 m range). Assuming the cut-off frequency of the radar is set to 1.4 MHz, then the two extra frequency terms will only stay if $|St_d| < 1.4$ MHz or $|S(\tau+t_d)| < 1.4$ MHz, which evaluates to:

$$-2.8 \text{ MHz} < St_d < 1.4 \text{ MHz} \tag{5.6}$$

This means that the two radars will only interfere with each other if their frequency difference falls into the 4.2 MHz range, i.e. if the two radars are switched on within 0.12 us. With a 4 GHz bandwidth, this is a probability of around 0.1%, assuming that the radars are switched on at a random time.

As an experiment, two radars were placed at a close distance and pointed towards the same scene from different perspectives, one of them was kept switched on (referred to as the main radar) and the other one was switched on and off periodically and randomly (referred to as the interference radar). The scene is set up with static objects placed between 0.5 m to 5 m and kept

Table 5.1: Average variances of the main radar's detection on static objects.

	Interference radar active	Interference radar inactive
All detection	0.23	0.20
Detection within 3-metre	0.08	0.07
Detection with signal strength >-3 dB	0.06	0.07

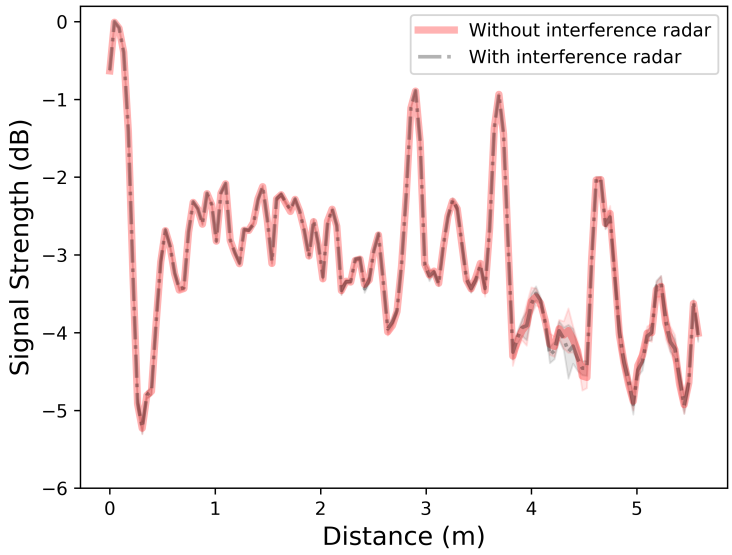


Figure 5.4: Received signal strength (and the standard deviation represented by the coloured area) at zero-Doppler domain from the main radar, when the interference radar is placed at a close distance.

unchanged at all times. The FFT results in the range domain from the main radar were recorded and analysed.

The experiment was carried out multiple times with different radar locations and lengths of recording. The average variances of the main radar's detection results were recorded and are shown in Table 5.1. It can be shown that, in all cases, the variances are very similar for the entire scene within the 6 m range, regardless of the status of the interference radar. When paying particular attention to the detection within 3 m (the main region-of-interest of this research), or the detection with signal strength greater than −3 dB (when the signals are strong enough to be identified), the variances are even lower. Therefore, it was concluded that the probability of interference is very low when using two radars concurrently.

One example of the experiment results is shown in Figure 5.4. The red solid line shows the detection result of the main radar when the interference radar was switched off, and the blue dashed line shows the result when the interference radar was switched on/off every three seconds.

The results shown were recorded and averaged over a 5-minute period (3000 frames). It can be seen that, as the two plots are overlapping, they do not have any significant differences and the variances are low most of the time.

The chance of interference can increase if more than two radars are used. When having N radars picking random 4.2 MHz frequency bands in the 4 GHz band, the probability of interference is the probability that any two of the radars pick the same frequency, which is

$$P(N) = 1 - \prod_{i=1}^{N} \frac{4000 - 4.2 \cdot (i-1)}{4000} \tag{5.7}$$

The probability of interference is generally low (less than 1% with four radars and less than 5% with ten radars). This figure will be higher with more than ten radars, which will then require explicit synchronization between radars or an interference detection algorithm.

5.5 Detection and Tracking Algorithm

The detection and tracking procedure can be divided into three stages: the two radars sense the scene independently and pass the data to a central processor on a computer; the central processor fuses the data from the two radars and detects the presence of people; the processor invokes the tracking module to verify the detection and refine the results. The full algorithm is shown in Algorithm 1. In the next sections, each part of the algorithm will be discussed respectively.

Algorithm 1 Human detection algorithm using two radars.

Input: Two radar modules r_1 and r_2.
Output: A set of detection D representing the detected people.

$D \leftarrow \varnothing$
while True **do**
  for $r_i \in \{r_1, r_2\}$ **do**
    $P_i \leftarrow r_i$.detect() ▷ Get individual detection from each radar
    for $f_i \in$ {FrameProcessors} **do**
      $P_i \leftarrow f_i(P_i)$ ▷ Apply individual Frame Processors
  $O \leftarrow P_0 \cap P_1$ ▷ Combine the two radars' detection
  for $o_i \in O$ **do** ▷ Iterate through each detection
    if $o_i \in D$ **then** ▷ Check if it matches a previous detection
      $D \leftarrow \text{update}(D, o_i)$ ▷ Update the database
    else
      $D \leftarrow D \cup o_i$ ▷ Add it to the database
  for $d_i \in D$ **do** ▷ Remove inactive detections in the database
    if $\neg\text{live}(d_i)$ **then**
      $D \leftarrow D \setminus d_i$

5.5.1 Individual Detection

As discussed in Chapter 3, the radar has a complete on-chip data processing chain to process the analogue mmWave signal and output objects in the form of a data cloud with x-y-z coordinates. This data will be transmitted to the central processor and be processed by the Frame Processor module independently. The frames will be stacked along the temporal domain using the first in first out (FIFO) queue module. A window of 10 frames is used to generate one stacked frame, which gives a few hundred points for each subject, at a cost of around 0.4 s processing delay. The data will then be clustered using the DBSCAN algorithm, which examines all the detected points and groups them based on their Euclidean distances between each other, where points within 15 cm will be classified into one cluster. The DBSCAN algorithm is selected as it has a low computational cost and does not rely on prior knowledge of the scene. Clusters with a low population will be treated as noise and discarded. The foreground extraction module can be loaded here to remove static objects in the area. It is considered as an optional module depending on the amount of clutter in the environment. The resulting clusters from each radar will then be passed to the Central Frame Processor for data fusion.

5.5.2 Data Fusion

The Central Frame Processor will be triggered once both radar results are ready. It will first transform all the data into one coordinate system by using the calibration parameters. Then, based on the size and the location of the clusters, it will calculate the eigenvectors of each cluster, estimate the distance and the overlapping region between every pair of the clusters and only keep them if their centroids are close and the majority of the areas overlap, where the threshold can be adjusted to provide a trade-off between high sensitivity and high precision. An illustration of the procedure is shown in Figure 5.5. The raw data from the two radars can be clustered into six candidate subjects (①-⑥), among which only ② and ⑥ are overlapping and are considered as one candidate. The rest of the clusters are treated as noise that can come from various sources, such as the error in the DPC and the signal multi-path effect.

A candidate human model will be constructed based on each verified cluster pair and the underlying point cloud data, which contains the estimation of the person's position, height, and volume. While these properties are not expected to be an accurate representation of the real subject, they provide essential information for these candidates to be compared and distinguished. These candidates will be passed to the tracking module to be correlated with previous frames.

5.5.3 Tracking

The tracking module records all the candidates at each timestamp and exploits the temporal relationship between them. The concept is similar to a Kalman filter, where prior information about the object is used to estimate the probability distribution of its new position. The system

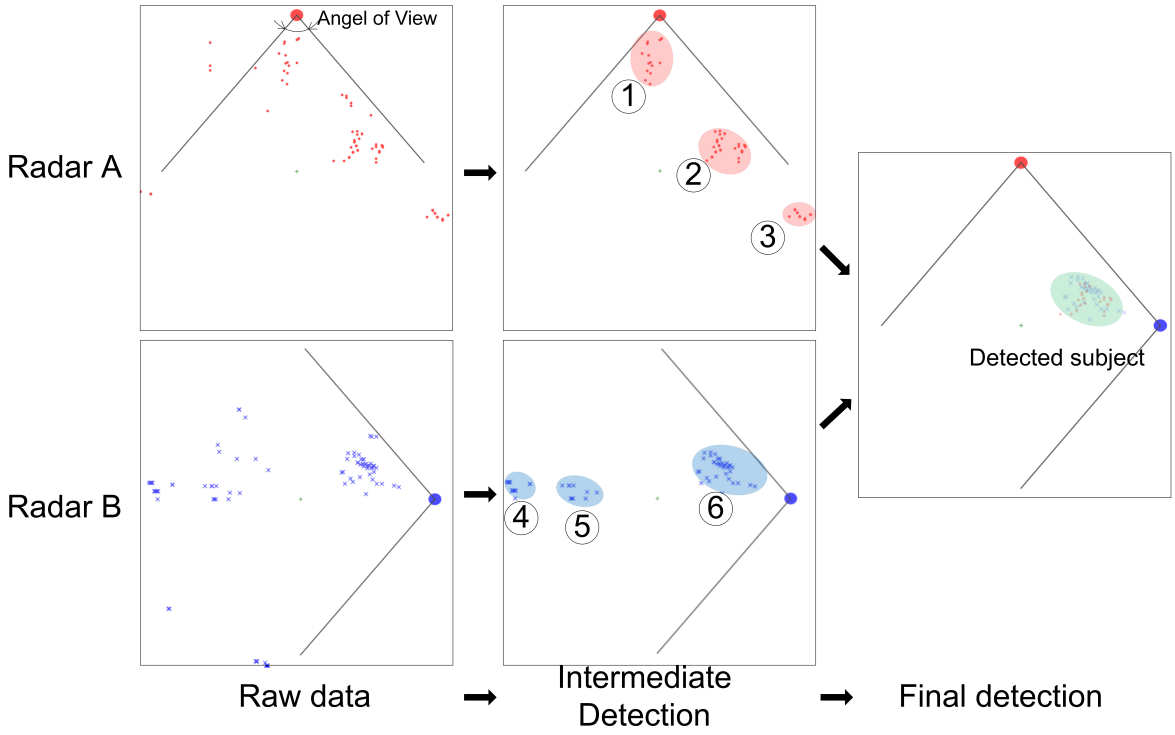
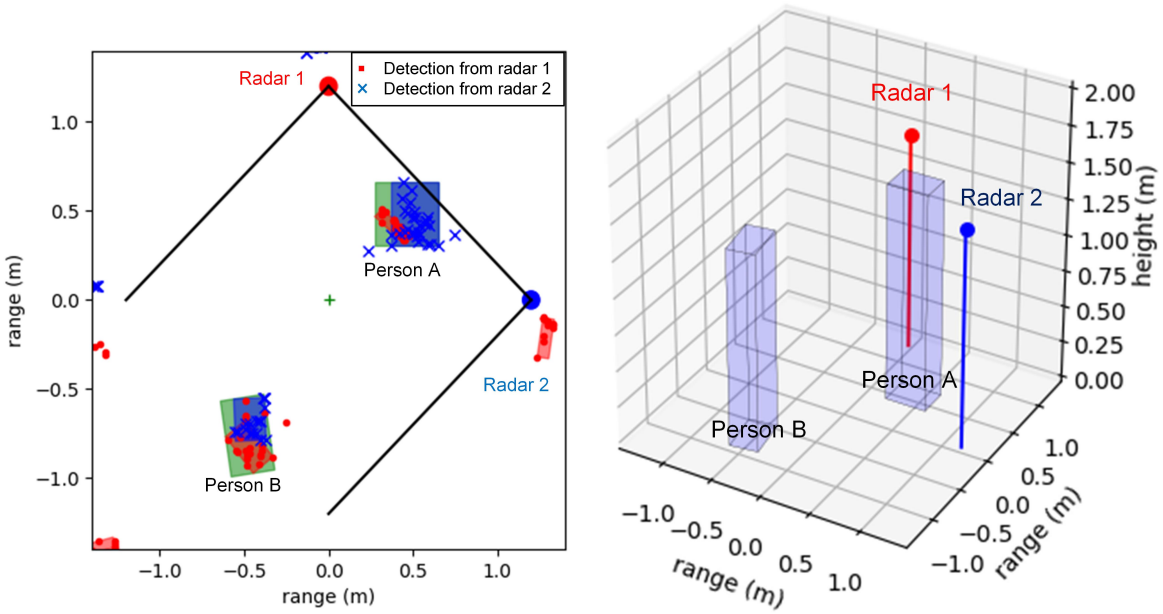


Figure 5.5: Workflow of the human detection system, with one person present in the area (top-down view).

will take a 25-frame temporal window, compare the new candidate with each detected object from the previous frames, and look for the best match using the candidate properties. If a match is found, i.e. the new candidate is close to a detected object and has a similar size, then it is considered to be the same object being detected again. The decision thresholds are learned during a training stage with a person moving at different speeds and along different paths, to model the possible variation of the parameters. If a match is not found, then the candidate is recorded as a potential new subject and the module waits for further frames to verify it.

The module keeps records of the live time of each detected subject and will only report the presence of a subject if the presence has lasted for more than a second, to avoid any phantom effect caused by signal noises. Meanwhile, the position of the subject will be smoothed over the past seconds to provide a more accurate estimation and reduce outlier effects, taking the assumption that the person will not move at a high speed in an indoor environment and the position should not vary too much within a second. The system is able to resolve multiple people in the area, as the detection process for each subject is independent. An example detection is shown in Figure 5.6 where two people are presented in the scene and have been detected successfully. An example of human tracking is shown in Figure 5.7. The current system uses the estimated properties of the human subject (the position, height, and volume) only to correlate them in the temporal domain. However, it is possible that this information can be further exploited for other



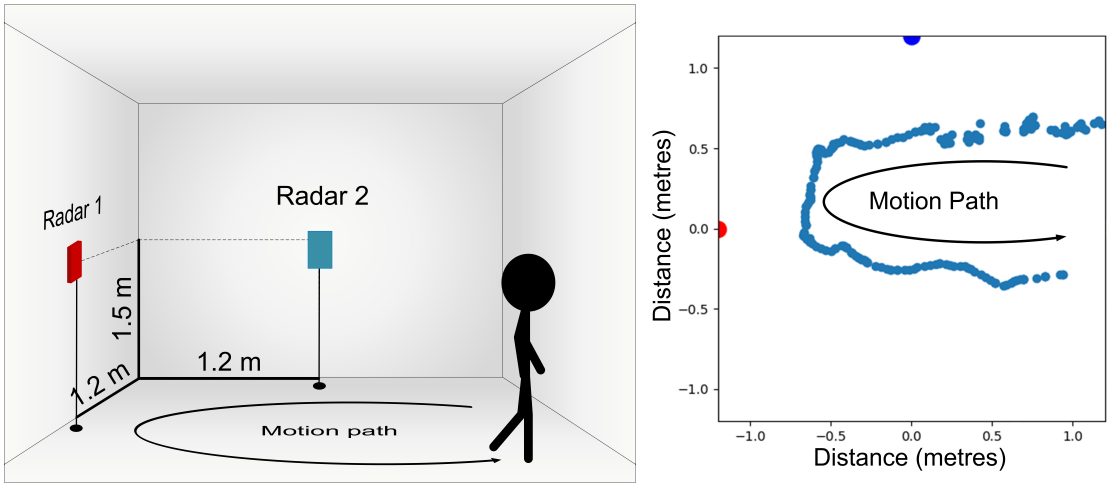
(a) The top view of the radar detection.

(b) The 3D view of the radar detection.

Figure 5.6: Example detection when two people are present in the area, from a top-down view (left) and a 3D view (right).

tasks, like human identification, which is left for future work.

The system requires a low memory usage and has a low computational cost, allowing the entire process to be performed in real-time. When running on an Intel i7-6700 CPU, the system can achieve 25 fps with only 10% average CPU utilization. The processing speed is only limited by the data processing and transmission speed of the radar. Computationally expensive algorithms, like neural networks on vision-based methods, were avoided, as they would require additional GPUs and a much higher cost and power consumption. Therefore, it is possible to port the proposed system onto low power consumption platforms and embedded processors. The system also benefits from its high configurability due to the Frame Processor module, which allows customized functions to be incorporated into the system based on the use case. For example, the foreground extraction module would be useful when the monitored area has clutter that needs to be removed prior to performing human detection. When using multiple radars, the independent detection stage and the calibration stage mean that the system does not have any restriction on the position or the orientation of the radars, nor the number of radars being used. While in this research two radars were used in a short area, it would be possible to extend the range of view by using more radars without modifying the framework.



(a) Room setup and the motion path of the person.

(b) Radar detected path.

Figure 5.7: Example of human tracking.

5.6 System Evaluation

5.6.1 Ground Truth from Cameras

In order to evaluate the performance of the system, an accurate ground truth on the presence of people is necessary. Since camera-based human recognition has been studied in depth and a lot of successful systems have been developed, They can be used for calculating the ground truth and providing a baseline for evaluation. The camera is calibrated using markers on the floor so that a 2D coordinate in the camera image can be projected into the 3D coordinate system used by the radar. The Yolo-v3 neural network model [180] for human detection was used in this research.

When the system starts, the Visualizer thread reads in the camera data, applies the neural network to the image, obtains the coordinates of the bounding boxes around the people and approximates the 3D areas accordingly. Meanwhile, the radar frame is clustered by the Frame Processors and each cluster is verified with the 3D areas. The 3D areas are estimated using trigonometry and have sector-shapes, and the system will validate a radar-detected object only if it fits closely in the sector. More specifically, the centroid of the radar detection and the camera detection need to be within 25 cm and have at least 70% overlapping area. For evaluating the tracking performance, a person walked into the region following several pre-defined paths, and a mean localization error is calculated based on the Euclidean distance between the radar detected location and the ground truth path. This provides a low-cost and real-time approach for verifying radar detection, and has the potential to allow more complex data labelling for other applications.

5.6.2 Evaluation Result

The following metrics were used for evaluating the human detection system.

- **Positives** (P): Number of people presented in the detection area.
- **True Positives** (TP): Number of people in the detection area that are successfully detected by the radar, with the position verified by the camera detection.
- **False Positives** (FP): Noise or other objects in the detection area that are falsely detected as a person, or if the detection is too far from the camera detection.
- **Sensitivity** (TP/P): The ability to detect a person when they are presented in the detection area.
- **Precision** (TP/(TP+FP)): The ability to distinguish a person from a false detection.

An ideal system should have both high sensitivity and a high precision. All the experiments were carried out in a 2.4 m by 2.4 m region in a laboratory under daily conditions. The system was run for two days and data was collected when at least one human was present in the area. During 56.8% of the time, there was only one person in the area, 12.1% with two people, 19.6% with three people and the rest with more than three people. The results are shown in Table 5.2.

Table 5.2: Performance evaluation of the system

	Sensitivity	Precision
One Radar	96.4%	46.9%
Two Radars	90.4%	98.6%

The high sensitivity in both cases indicates that, whenever a person is present in the area, the system has a very high probability of detecting it. However, with one radar, the 46.9% precision indicates that more than half of the detections would be false detections. With two radars, the system sensitivity was reduced slightly, but the precision improved significantly to 98.6%. In other words, when the one-radar setup detects an object, there is over half the chance that it is a false detection, whereas with two radars the system can be very confident in its detection.

When detecting with one radar, the system reports a large number of false alarms due to noise and flickering of the results. The flickering is observed because of the FFT process and the peak detection algorithm, where a small change in the signal, once it comes through the FFT, can result in a change in the FFT bins and hence a few centimetres' displacement on the object coordinates. This effect will be enlarged when carried over to the angle-FFT, where a displacement in the angle will result in a much larger displacement in the 3D space. On the other hand, when using two radars, the system has access to two independent detections and can verify the results from each other. As a result, the false alarm rate was reduced significantly (represented by the rise in precision) with only a slight reduction in the sensitivity.

Table 5.3 shows a comparison of the tracking performance between proposed system and several state-of-the-art RF-based tracking systems. The proposed system achieves a low mean localization error of (5.6 ± 4.2) cm, outperforming all other systems. The table shows a general

Table 5.3: Tracking performance comparison between the proposed system and the literature.

	Method	Frequency (GHz)	Bandwidth (MHz)	Error (cm)
Ours	mmWave	77	4000	**6**
Ruan et al. [181]	RFID	0.86	90	64
Qian et al. [182]	WiFi	5	20	38
Li et al. [183]	WiFi	5	20	35
Nguyen and Pyun [184]	UWB	3.8	2500	22
Will et al. [185]	UWB	24	200	20
Zhao et al. [156]	mmWave	77	4000	16
Wu et al. [186]	mmWave	60	3520	10

trend that systems with higher frequency and bandwidth tend to have a better performance due to the reduced signal interference and larger amount of information, especially with mmWave radars when the bandwidth can reach up to 4 GHz. when compared with the other mmWave systems, the localization error of the proposed system has been reduced further due to the use of two radars. Meanwhile, the proposed system has other advantages, including a real-time processing time and the ability to detect multiple people at the same time, which were not presented in the mentioned work.

One limitation of the system is the ability to distinguish multiple people at short distances. The DBSCAN clustering algorithm used in this work cluster points within 15 cm as one subject. Given that the radar detection can be noisy (as shown in Chapter 4), experiments found that two people within 0.5 m have a high probability to be recognized as one subject. Therefore, the performance of the system will drop in certain situations, such as counting people in a queue. When there are three or more people and people are occluded by others, the system can only confidently report people in the front, which results in a loss of sensitivity. The occlusion can potentially be solved by using more radars to cover the scene from more angles. As discussed in Section 5.4 and Section 5.3, it is possible to adapt more radars into the system without modifying it much. Therefore, the system can be easily adapted to fit different use cases if necessary. Similarly, although all experiments were carried out in a 2.4 m by 2.4 m region, as the radar's sensitivity to stationary targets would drop significantly beyond 2.5 m, the range of detection can also be extended by incorporating more radars into the system.

5.7 Conclusion

In this chapter, a real-time human detection and tracking system using two mmWave radars has been presented. Lightweight algorithms were used that can achieve real-time processing at 25 fps with a low CPU utilization, making it possible to port the system onto low power-consumption platforms. It is shown that the system is able to detect people in indoor environments with over 90% sensitivity. The problem of high false alarm rates with a single radar has been discussed,

and it has been shown that the precision can be improved from 46.9% to 98.6% with a two-radar setup. The system is able to track the path of people walking in the region with a low mean localization error of 5.6 cm.

CHAPTER 6

HUMAN POSTURE ESTIMATION

In this chapter, a novel human posture estimation system using mmWave radars is presented. The system uses two radars to capture the posture information of the subject in the form of point clouds, and uses a neural network model to estimate the position of the key joints. The system achieved an average precision of 71.3% when detecting common human postures in an indoor environment. The content of this chapter has been published in the MECO conference [25] [1] and the IEEE Sensors Journal [21] [2], and the patent with the tracking system [187].

The rest of the chapter is organized as follows. Section 6.1 gives the background of this work. Section 6.2 presents the experimental setup of the system. Section 6.3 presents the architecture of the proposed neural network model. Section 6.4 presents a novel temporal correlation algorithm for improving the smoothness of the prediction during real-time operation. Section 6.5 shows the evaluation result of the system and a comparison to the state-of-the-art systems. Section 6.6 shows how the posture estimation system can be integrated into the framework presented in the last chapter to achieve real-time operation. Section 6.7 discusses the feasibility of porting the system to an embedded system as future work. Section 6.8 concludes the chapter.

6.1 Overview

Human posture analysis has become a popular topic in computer vision. Being able to obtain an accurate estimate of a person's posture enables computers to understand human behaviours and provide appropriate assistance or interaction, which can be beneficial in many applications, such

[1] ©2020 IEEE. Reprinted, with permission, from H. Cui and N. Dahnoun, "Human posture capturing with millimetre wave radars," in *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, 2020.

[2] ©2022 IEEE. Reprinted, with permission, from H. Cui and N. Dahnoun, "Real-time short-range human posture estimation using mmWave radars and neural networks," *IEEE Sensors Journal*, vol. 22, no. 1, pp. 535-543, 2022.

as health care, security, and gaming. While camera-based methods have shown an impressive accuracy on optical images [66, 112], their intrusive nature makes them unsuitable for certain applications. Posture analysis using RF signals and radars has been an emerging area, among which mmWave sensing has received a great popularity due to its ability of capturing high-resolution information of the subject. Although there are some researchers using mmWave radars for posture estimation, such as [116, 119], few have achieved a satisfactory performance in terms of the accuracy, processing speed and complexity of the postures. Accurate posture estimation using non-intrusive devices is still a challenge.

In this chapter, a real-time posture estimation system is presented. The system detects people with arbitrary postures in indoor environments at close distances (within two metres), and estimates the posture by localizing the key joints. Two mmWave radars are used to capture the scene and a neural network model is used to estimate the posture. The model consists of two parts: a part detector model that provides an initial estimate of the person's key joint positions, and a spatial model that learns the position relationship between these joints and refines the estimate. The position of the joints forms a concrete representation of the entire body posture. During real-time operation, the temporal correlation of the joints between timestamps has been exploited to improve the smoothness of the estimate. The system can provide an accurate posture estimate of the person in real-time at 20 fps, with a mean localization error of 12.2 cm and an average precision of 71.3%.

In contrast to much existing research that only focuses on standing postures, this work is the first mmWave radar-based system that can accurately estimate a rich set of postures that are commonly seen in an office environment, while having real-time processing speed and a low cost. The posture provides a fine-grained description of the person's activity and allows higher-level applications to be developed, such as security and health monitoring.

6.2 Experimental Setup

The same radar configuration and the software framework mentioned in Chapter 5 are used, but with a different radar layout. It was found that the angle-of-view (AoV) of one radar is not sufficient to cover the body area of an adult, so two radars are used as a vertical radar array to increase the vertical AoV.

6.2.1 Single Radar Angle-of-view

Human posture analysis requires high-resolution detection across the vertical plane around the person, where the quality of the detection is determined by the antenna radiation pattern of the radar. The receivers on the IWR1443 radar module can receive signals from both the horizontal plane and the vertical plane. However, the horizontal AoV is designed to be much larger than the

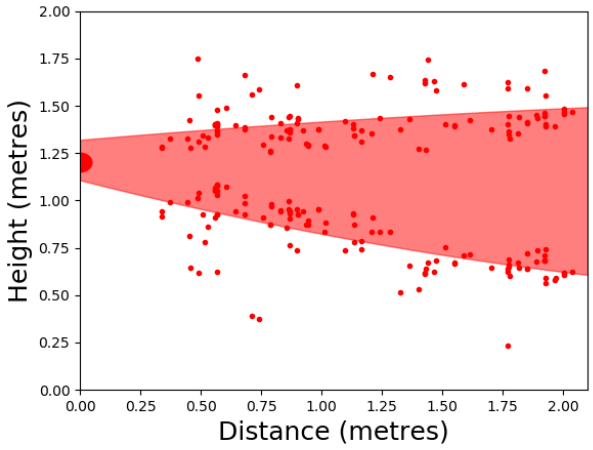


Figure 6.1: Radar vertical AoV at various distance when pointing to a flat wall.

vertical one. The radar will receive an attenuation of -6 dB (1/4) on the signal when detecting objects at ±50° horizontally or ±20° vertically, as discussed earlier in Section 3.2.4.5.

While the antenna radiation pattern could work well in autonomous driving, the situation in human posture analysis is different, as the vertical AoV becomes equally important. For indoor environments, when the person is close to the radar, the limited vertical beam-width will make the radar only be able to sense a part of the person, and might not have enough features for advanced analysis.

In order to investigate this effect in real-world applications, a set of experiments was performed to test the radiation pattern with a focus on the vertical AoV. The radar was placed at 1.2 m high and was pointed at a flat wall at various distances between 0.3 m to 2 m, and the detected frames were recorded. At each distance, the amount of and the distribution of the detected points were analysed and the effective AoV was calculated to be the 95% confidence interval range. In other words, the radar is considered to have a V_{low} to V_{high} vertical AoV if 95% of the detected points are above V_{low} and below V_{high}, to exclude any outliers. The results are shown in Figure 6.1. The red dots in the figure are the V_{low} to V_{high} values at each distance, and the highlighted area is the estimated AoV region, fitted as a quadratic polynomial function. It can be seen that the AoV increases near-linearly with the distance. The detectable range was around 30 cm to 45 cm at close distances (0.3 m to 0.5 m) and increased to 87 cm at 2 m. The angular AoV was approximately 15° to 20°, conforming to the −3 dB to −6 dB range in Table 3.2.

The same experiment was also carried out with a person being the subject. The radar was set at 1 m high. The detectable range was calculated using the same method, and the results are shown in Figure 6.2. The AoV region follows a lobe shape, similar to the standard radiation pattern of a general antenna. The radar has a limited AoV of around 50 cm at close distances, where the antennas can hardly receive signals beyond 35°. The AoV increases with the distance and reaches a peak of 98 cm at 1.4 m. When the person moves further away, the signal attenuates.

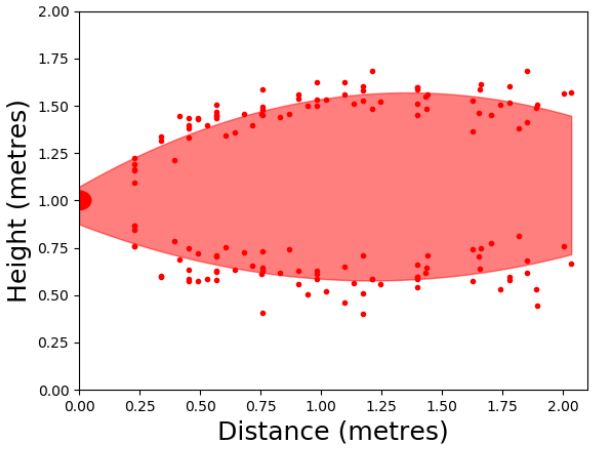


Figure 6.2: Radar vertical AoV at various distance when pointing to a person.

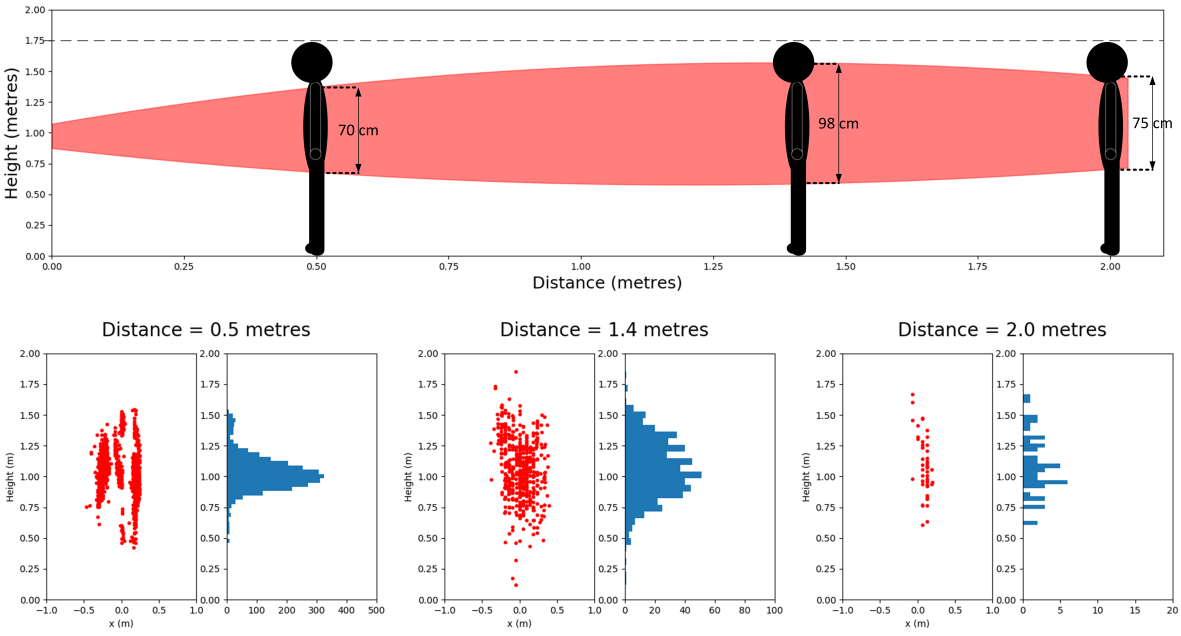


Figure 6.3: Radar vertical AoV when detecting human-size subjects. Top: Only a limited area of the person can be detected with one radar. Bottom: The detection results and their distribution at various distances.

Although the radar can still detect the presence of the person, the number of detected points drops dramatically and the result does not carry sufficient information for determining the posture.

Figure 6.3 shows an illustration of this problem when detecting human postures at an adult-scale. It shows some detection results and a histogram of their distribution at 0.5 m, 1.4 m, and 2 m. Based on the experiments, the best AoV can be obtained at around 1.4 m with a 98 cm coverage. Therefore, this distance is considered to be the best distance for posture capturing due

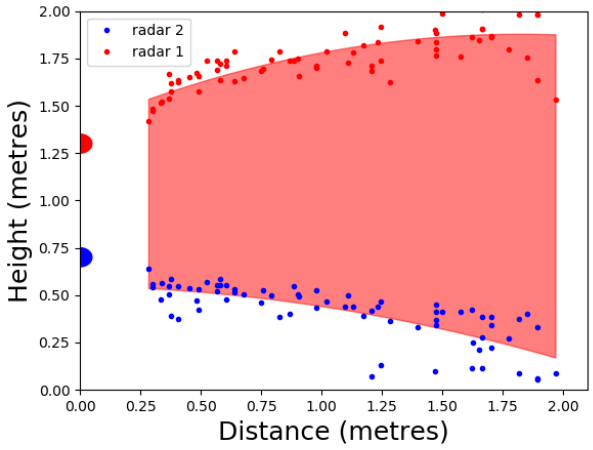


Figure 6.4: Two radars vertical AoV at various distances when pointing to a person.

to the wide AoV and a sufficient amount of data.

6.2.2 Radar Array Angle-of-view

In order to address the problem, at least two radars are required as a vertical radar array to cover the entire height of an adult. A radar array was constructed by placing two identical radars at the same vertical axis and pointing them forward at the same angle, one at 0.7 m high and the other at 1.3 m high. The setup is found to provide a consistent vertical coverage across different distances. As shown in Figure 6.4, using two radars can significantly extend the vertical AoV at all distances. In comparison to using only one radar, the average detectable range at close distances (around 0.5 m) has increased from 70 cm to 110 cm, and the peak range (at around 1.4 m) has increased from 98 cm to over 140 cm. The average improvement from all distances is around 55%. The variance in the coverage is much smaller than one radar. Figure 6.5 shows the detection results and their distribution at 0.5 m and 1.4 m. The two peaks from the two radars are still observable at close distances, but the overlapping region is receiving more data and, hence, could contain more useful data. The distribution is much smoother at larger distances, and the two detection results are no longer separable. In both cases, the radar array can detect over 100 points at each height bin and can capture useful shape information from the scene and the human body-parts.

6.2.3 Radar Array Posture Capturing

As shown in Figure 6.6, the two radars are placed at 0.7 m and 1.3 m height respectively above the ground level and in the same orientation, with an optical camera in the middle to provide the ground truth of the person's posture. It was found that the resolution of the detection will decrease dramatically along the distance, and the radar is only able to obtain enough information

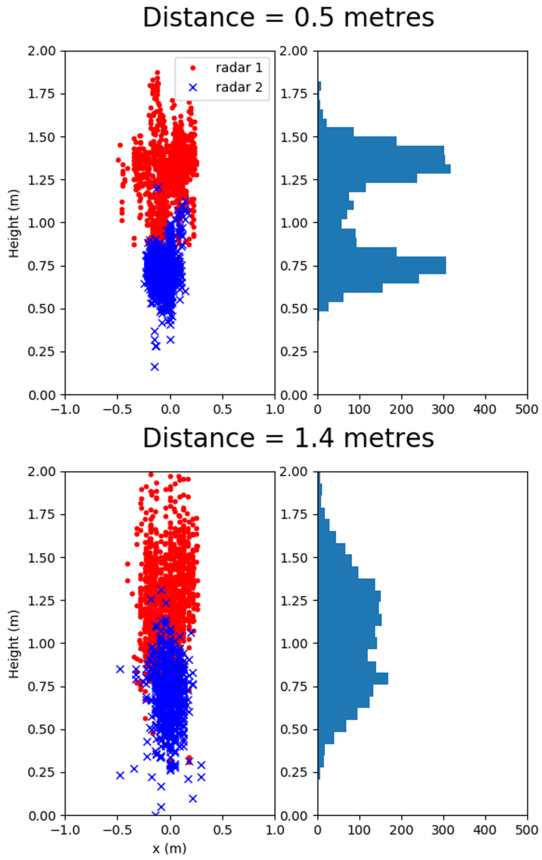


Figure 6.5: Detection results and their distribution using two radars.

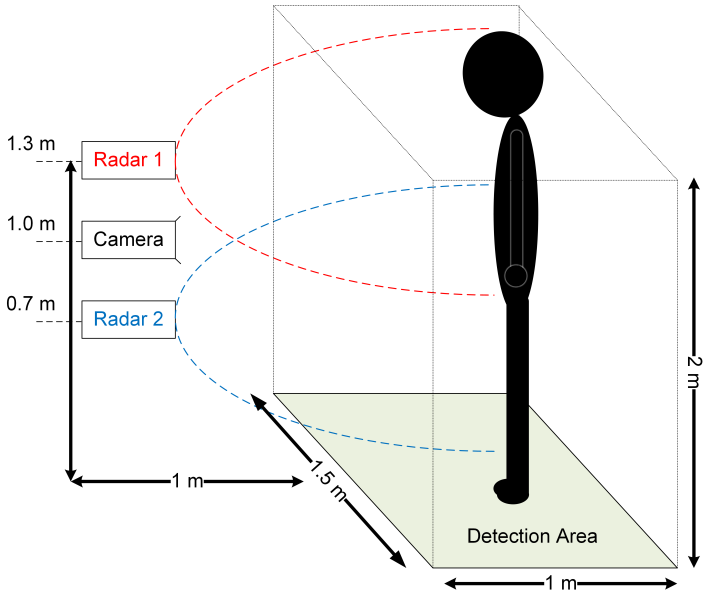


Figure 6.6: Experimental setup of the system.

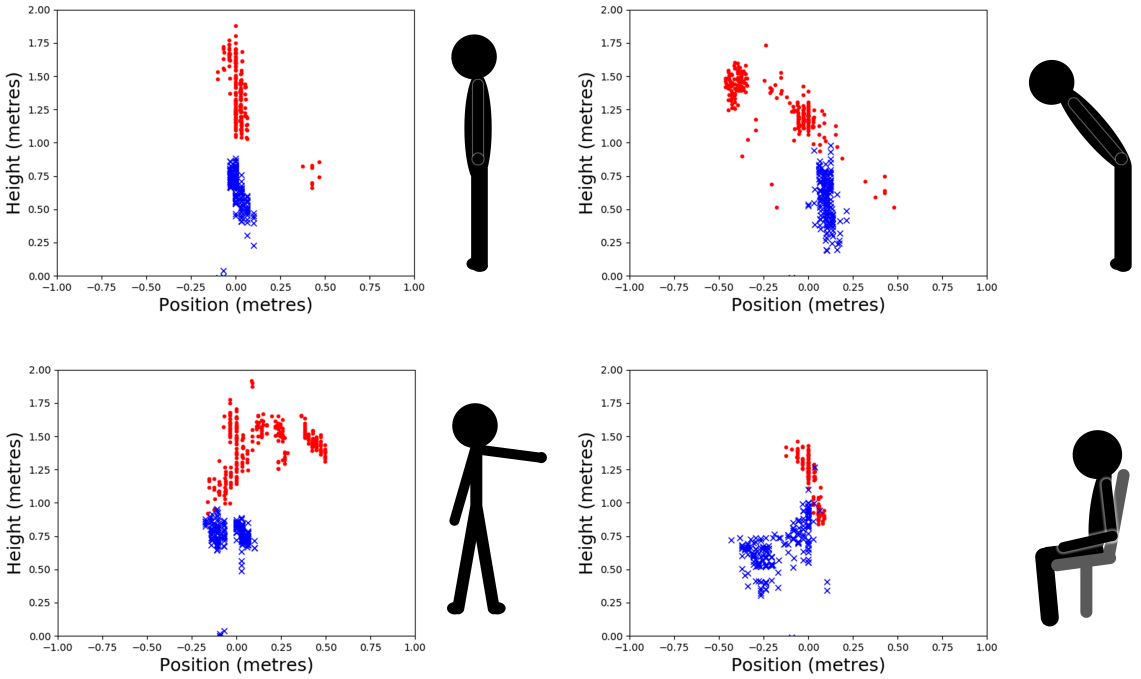


Figure 6.7: Radar array vertical AoV at various distances when pointing to a person. Top-left: Standing still. Top-right: Bowing. Bottom-left: Standing and holding one arm. Bottom-right: Sitting.

for posture estimation at a short distance within two metres. Therefore, the detection area of this research was restricted to a (2 m × 1.5 m × 1 m) region to ensure the quantity and accuracy of the received data. As the angle estimation algorithm used by the radar can have a decreased accuracy at a larger angle of incidence, the two radars are placed to have an overlapped detection region in the middle to complement each other. Figure 6.7 shows some example data captured when the person was holding different postures. It can be seen that the radar array successfully captured a complete view of the subject, and the data contained distinguishable features corresponding to the postures.

It is worth noting that the captured image did not have any data close to the floor, i.e. from 0 cm to 25 cm. The hypothesis is that the lower body parts of a person have a close distance to the floor and have a small area, which makes them hard to identify using the CFAR peak detection algorithm. Therefore, they are not reported by the radar as real objects. This issue should not have a big effect on posture analysis, as the lower body parts, like feet, are often less important than the other body parts like limbs, and this is left as a future research topic.

The radars are configured to use a 4 GHz bandwidth, have a 4 cm range resolution and operate at 25 fps, which is the same configuration as for the human tracking system. The two radars operate concurrently and independently using the software framework mentioned in Section 5.3, where a computer is used to read the data from the two radars simultaneously, align the data

packets into frames based on their arrival time and concatenate the data at each frame. The posture estimation system can also work as a standalone module and can be incorporated into any other mmWave radar system.

6.2.4 Data Collection and Pre-processing

The radar outputs a 3D point cloud in arbitrary sizes, representing the geometric information about any subject in front of it. As the radar obtains the point cloud in one scan and relies on the signal reflected from the surface of the subject, the x-y position of the obtained point cloud is found to be more accurate than the depth. Therefore, to transform a point cloud into a fixed-size data format as required by a typical CNN model, the point cloud is projected to a 2D image from the front view through trigonometry, aiming to retain the most essential spatial information for determining the posture of a person. The projection is calculated from a virtual viewing point that is defined to be the middle of the two radars, which is also the position of the camera shown in Figure 6.6, so that the projected radar image will share the same coordinate system as the vision images captured by the camera.

Once the point cloud data from the two radars has been transmitted to the computer, the data processing chain described in Section 5.5 is used as a pre-processing technique, to locate the point cloud representing the person and filter out clutter. Other human detection system can also be used as a pre-processing stage to prepare the data for posture estimation. The filtered point cloud is projected to a 200×150 grayscale image I, where the intensity of the pixels represents the projected position of the subject (see the leftmost image in Figure 6.8). The cropped space is found to be sufficient to cover most types of postures, and the projected image has a resolution of 1 cm/pixel that retains enough detail in the 3D space.

The goal of the neural network is, based on the input image I, to estimate 9 heatmaps $P_{v\in\{1\ldots9\}}$ of size 45×32 for 9 joints of a person: the head, left and right shoulders, hips, elbows, and knees. The wrists and ankles were ignored, as the mmWave signal reflection from these joints is relatively weaker and less important in estimating the overall body posture. Each heatmap P_v is a 2D image where each pixel value has an intensity between 0 and 1 that denotes the probability of joint v being at that location. The position of the maximum intensity in a heatmap P_v represents the most likely position of joint v. Since the number of parameters and operations of the network will increase proportionally with the size of the heatmap, and since it is targeted for real-time operation, a relatively low output resolution was used and the result was interpolated to a higher resolution for visualization only. Meanwhile, considering that the range resolution of the radar is around 4 cm, a higher output resolution will not provide a significant improvement to the system. Therefore, the size of the heatmap is chosen to be 45×32, where each pixel represents approximately 4 cm $\times$ 4 cm.

The ground truth of the posture is generated using a standard optical camera and the HRNet algorithm [66], a state-of-the-art algorithm for human posture estimation on optical images.

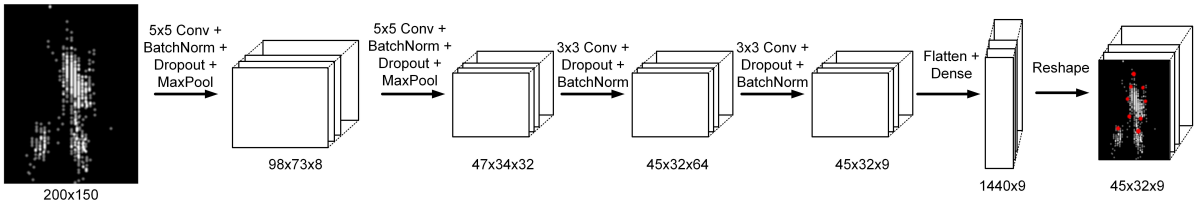


Figure 6.8: The architecture of the part detector model.

The camera is placed in the middle of the two radars pointing towards the scene and is set to operate at the same speed as the radars. Images captured by the camera are cropped according to the detection area, timestamped and associated with each frame of the radar data. The HRNet algorithm takes the camera image as the input and outputs the position of the 9 selected joints of the person. A heatmap is generated for each joint by placing a 9×9 Gaussian kernel on the joint's true location, which determines the error-tolerance of the neural network in the training stage and improves the training speed and robustness of the neural network. The generated heatmaps are cut according to the detection area and are resized to 45×32 to serve as the ground truth for the radar data.

During the data collection, the person stayed in the specified detection area at around 1 m to 2 m from the radars and the camera, and performed arbitrary postures that are commonly seen in an office environment. More specifically, the postures include standing, walking, and sitting in different directions and angles with arbitrary hand and arm positions. The training data and the test data are collected separately to reduce any potential correlation between the two datasets and improve the generalizability of the model. Before feeding into the neural network, the training data is shuffled randomly and augmented with random rotations and translations, to increase the robustness of the model and avoid overfitting. The final dataset has 24k training data instances and 2.6k test data instances.

6.3 Neural Network

The proposed neural network consists of two parts: a part detector and a spatial model. The part detector consists of a few standard convolutional layers and dense layers. It is applied directly to the input point cloud and produces heatmaps as an initial estimate of the position of the 9 joints. The spatial model is a customized layer that models the correlation between the joints and refines the heatmaps.

6.3.1 Part Detector

The structure of the part detector model is shown in Figure 6.8. The model has four convolutional layers to extract the initial features from the input, where the numbers of channels are set to 8, 32, 64, and 9, respectively. The nine channels of the final convolutional layer are designed to

correspond to the nine joints considered in this work, which are followed by nine independent dense layers to estimate the location of each joint. Two max-pooling layers are used with the first two convolutional layers to emphasize the features. Batch normalization layers and dropout layers are used between each pair of the convolutional layers to reduce the variance in the training data and avoid overfitting. All the intermediate layers use the rectified linear unit (ReLU) activation function, and the last dense layer uses the Softmax function to generate the heatmap. The model takes a 200×150 image I as the input containing the projected point clouds and outputs 9 heatmaps $P_{v\in\{1...9\}}$ of size 45×32 as the position estimate of all the nine joints. The model has around 107M multiply-accumulate (MAC) operations and 19M parameters.

6.3.2 Spatial Model

The part detector can provide a rough estimate of the joint positions. However, since the regression process of each joint is independent, the model does not consider the relative position between the joints, which sometimes leads to anatomically incorrect postures. To address this issue, and inspired by [113], a spatial model was added into the system. In [113], the authors proposed an MRF model to formulate the spatial relationship between the joints. This model was adapted into the system by designing a new dependency graph, which refines the joint positions using Equation (6.1):

$$\hat{P}_{v\in\{1...9\}} = exp(\frac{1}{|C_v|}\sum_{c\in C_v} log(W_{c\rightarrow v} * P_c + b_{c\rightarrow v})) \tag{6.1}$$

where C_v is the set of joints that will contribute to the position estimation of joint v, including v itself; $|C_v|$ is the cardinality of C_v and $\frac{1}{|C_v|}$ is the normalization term used to scale the calculation with respect to the number of joints involved; $P_{v\in\{1...9\}}$ and $P_{c\in C_v}$ are the heatmap output from the part detector for joint v and c respectively; $\hat{P}_v$ is the refined heatmap output of the spatial model; $W_{c->v}$ and $b_{c->v}$ are the weight and bias terms that model the spatial relationship between joints c and v.

The architecture of the spatial model is shown in Figure 6.9. The MRF operation is implemented as a convolution operation (the MRF Conv block in Figure 6.9), where $W_{c\rightarrow v}$ is defined as the convolution kernel and $b_{c\rightarrow v}$ is the matrix containing the bias term. The convolution operation models how the estimate of joint c contributes to the estimate of joint v. The ReLU function is applied to the heatmaps (P) and the convolution kernels (W) before performing the convolution to ensure non-negative values and improve the stability of the network.

Five of the nine joints, the head, the left and right shoulders and hips, were defined to be the primary joints. These joints are chosen because they have a relatively larger size and produce a stronger reflection of the mmWave signal, when compared with the elbows and the knees. Meanwhile, the position of these joints are more important in understanding the overall posture of the person, and their relative positions regarding each other have a more regular pattern. Therefore, the primary joints are set to have a bidirectional pair-wise dependency among them,

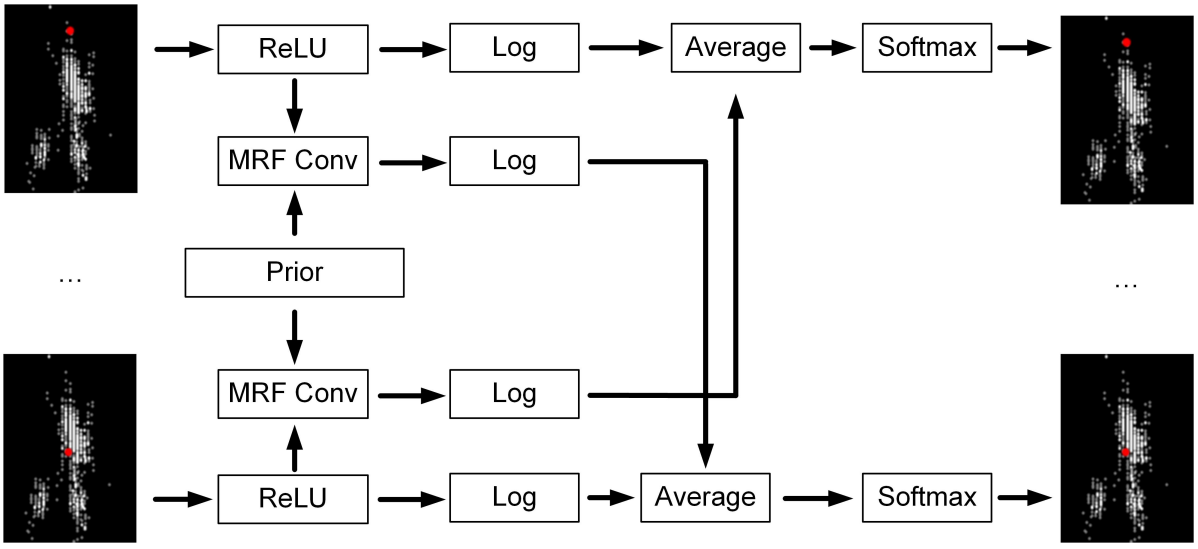


Figure 6.9: The architecture of the spatial model, showing the head and the hip as an example.

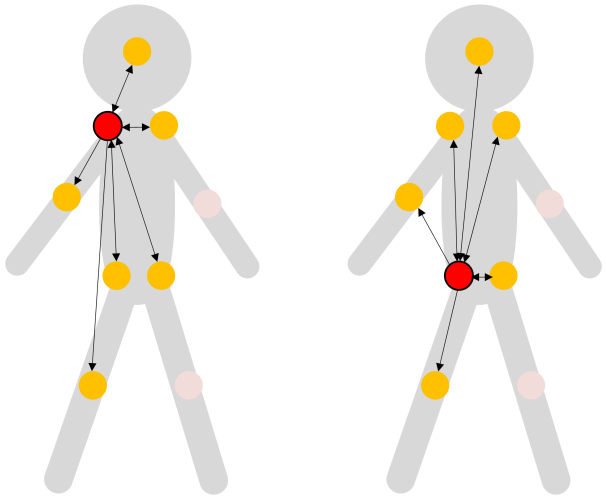


Figure 6.10: Dependency graph of the left shoulder and left hip.

i.e. the position of any primary joint will contribute to the prediction of other primary joints. The other joints, the left and right elbows and knees, can have more random motions and reflect less signal. The prediction of the secondary joints is set to be dependent on the neighbouring primary joints and the head. For example, when predicting the position of the left elbow, the network will refer to the position of the head, the left shoulder and the left waist. As an example, Figure 6.10 shows the dependency graph of the left shoulder and left hip. A double arrow indicates that the two joints are dependent on each other, and a single arrow indicates a one-way dependency.

The convolution kernels W are set to be twice the size of a heatmap along each dimension, i.e. 90×64 pixels, because the position dependency can be from any direction. Since the kernels are supposed to encode the prior knowledge of the joints' spatial relationship, the weight of the kernels was initialized by collecting the pairwise position dependency between each pair of

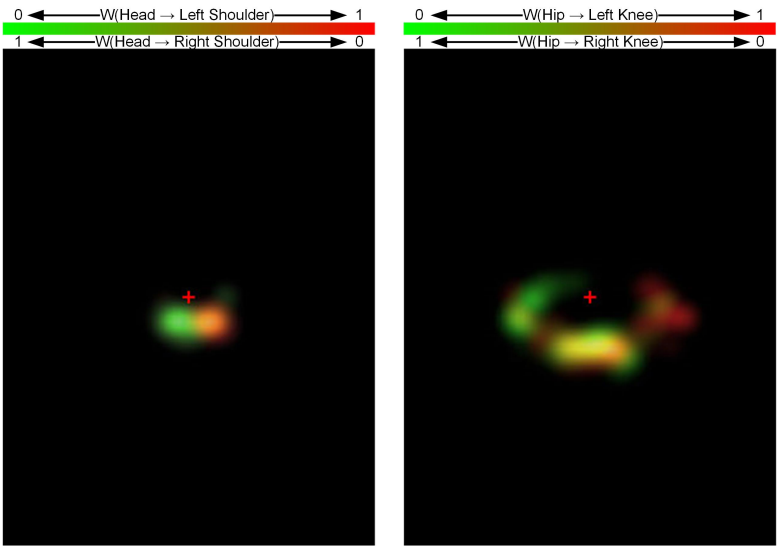


Figure 6.11: Example of how prior knowledge helps predict a joint's position. Left: the likely position of the left and right shoulders given the position of the head. Right: the likely position of the knees given the position of the hips.

the joints from the training dataset. Figure 6.11 gives an example of how the prior knowledge encodes the relationship between the joints. The left figure shows, given the position of the head (represented by the red cross at the centre), the likely position of the left and right shoulders, in red and green respectively. In other words, assuming that the position of the head is known at the centre, the red region is the likely position of the left shoulder where the intensity shows the corresponding probability, and the green region is the likely position of the right shoulder. The yellow region indicates the intersection between the two probability distributions. Similarly, the right figure shows the likely position of the left and right knees given the position of the hips.

The spatial model has 32 dependencies between all the joints, which requires 32 convolutions, 265M MAC operations and 0.3M trainable parameters. The spatial model is appended at the end of the part detector model. It takes the heatmap output from the part detector and generates a refined heatmap of the same size, as shown in Figure 6.9. The spatial model can improve the accuracy and the robustness of the network, as will be shown in Section 6.5.

6.3.3 Model Training

A block diagram of the training procedure is shown in Figure 6.12. The two parts of the model were trained separately in two phases. In the first phase, the part detector was trained on its own. In the second phase, the parameters in the part detector were frozen, and the spatial model was trained. The Adam optimizer was used with a dynamic learning rate between 10^{-2} and 10^{-5}. The cross-entropy loss was used to compute the difference between the estimated heatmap and the ground truth. The full model has 372M MAC operations and 19M parameters.

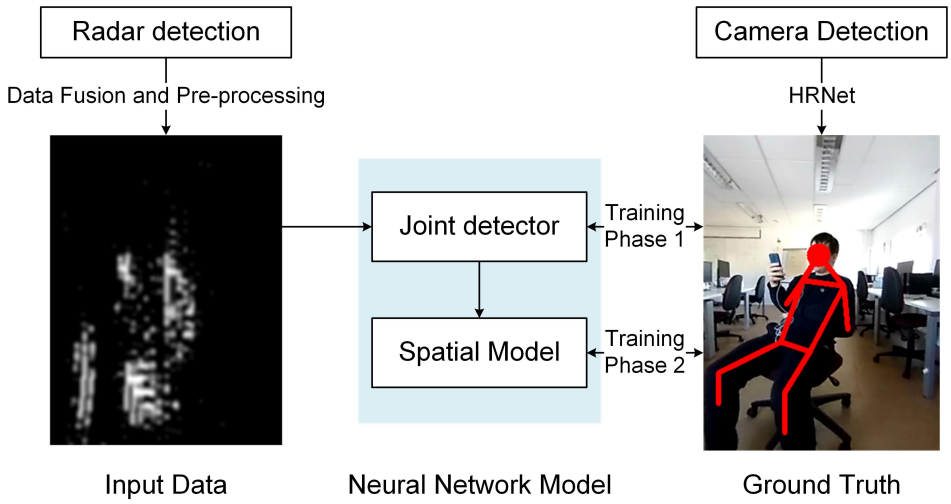


Figure 6.12: The training procedure of the proposed neural network model.

6.4 Temporal Correlation

The neural network model estimates the position of the joints independently at each timestamp. However, as the radar is prone to noise and the point cloud can sometimes be inaccurate, the estimate can be further refined by exploiting the temporal correlation between frames, following the assumption that the joints will not move much over one timestamp.

The estimate at each timestamp is recorded and analysed. Two parameters were used to evaluate the smoothness of the estimate in the temporal domain: the confidence of the neural network's estimate (C) and the speed of the joints' motion (M). The confidence of an estimate is inherited in the heatmap, represented by the intensity distribution from the Softmax layer. A sharp and dense distribution indicates that the network is confident in the joint position, as opposed to a sparse and flat distribution. The value of C is calculated by taking the peak values of the heatmaps and averaging across all the 9 joints, as shown in Equation (6.2).

$$C = \frac{\sum_{v \in \{1...9\}} max(\hat{P}_v)}{9} \tag{6.2}$$

The second parameter, M, examines the speed of the joint's motion, i.e. the rate of change of the joint position across timestamps. Once the neural network has predicted the heatmaps for the joints, the position of the maximum intensity within each heatmap is taken as the estimated position of that joint. If a joint v was at (x_{v,t_0}, y_{v,t_0}) at timestamp t_0 and (x_{v,t_1}, y_{v,t_1}) at timestamp t_1, then the speed of motion could be calculated from the Euclidean distance between them as $\sqrt{((x_{v,t_1} - x_{v,t_0})^2 + (y_{v,t_1} - y_{v,t_0})^2)}$. The value of M (at timestamp t_1) is calculated by taking the average of this distance across all the 9 joints, as shown in Equation (6.3).

$$M = \frac{\sum_{v \in \{1...9\}} \sqrt{(x_{v,t_1} - x_{v,t_0})^2 + (y_{v,t_1} - y_{v,t_0})^2}}{9} \tag{6.3}$$

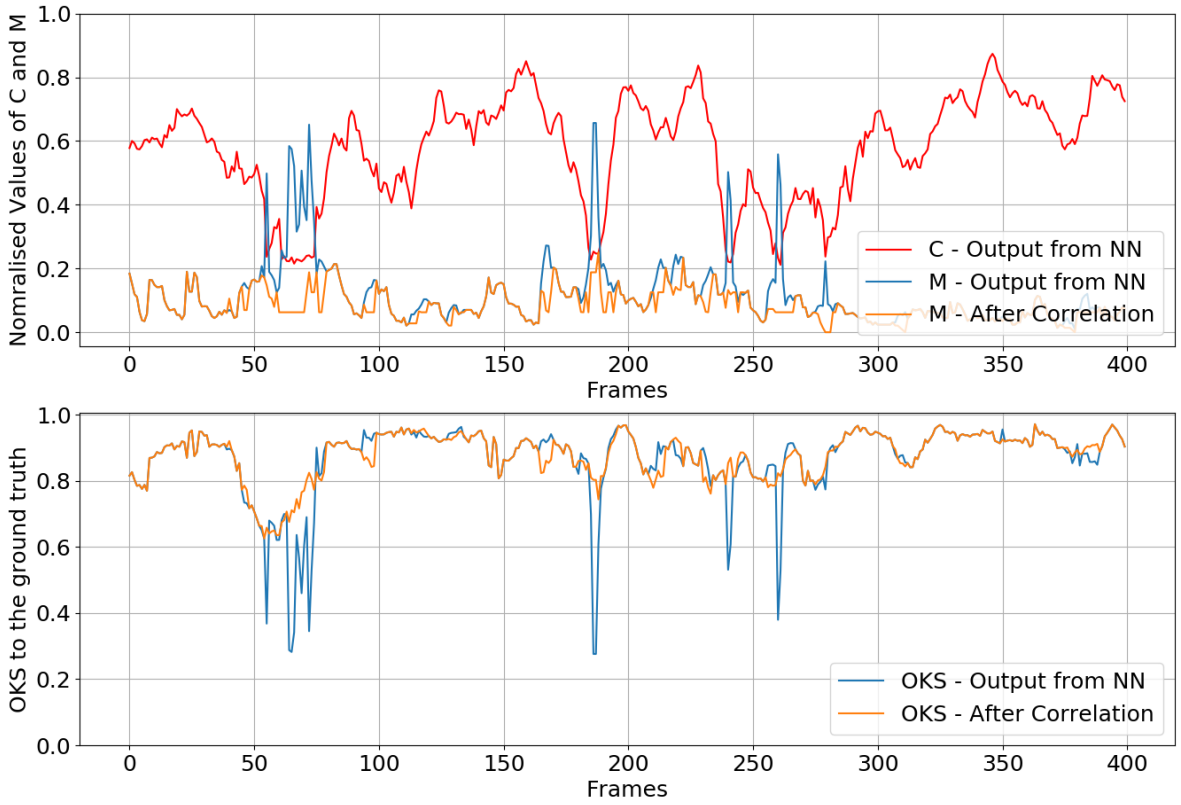


Figure 6.13: An example of 400 continuous frames, showing that the stability of the estimation can be improved by assessing and correlating C and M in the temporal domain.

The two variables, C and M, are recorded as the system operates. The mean values and the variances of C and M are calculated across a 5-frame time window (approximately 0.25 s), to assess the stability of the estimate. An estimate will be determined as unstable if either C or M has fluctuated over 5% regarding their mean; for example, if the confidence has dropped significantly, or if a new estimate greatly differs from the last one. It should be noted that the comparison is made against the mean value of C and M in the 5-frame time window. Therefore, a constant low C or a constant high M will not be treated as unstable. An unstable estimate will not be accepted as an output. Instead, only the horizontal position of the estimate will be recorded, and the estimate from the last frame will be used and shifted as the output. The estimate will be treated as stable again if the variances of both C and M fall within 15% of their mean values, i.e. when the last 5 estimates have a similar C and M. The thresholds used here provide a trade-off between a higher error tolerance to noisy data and a higher sensitivity to posture changes, which can be configured for the application. Figure 6.13 shows an example consisting of 400 continuous frames. The OKS (object keypoint similarity) metric in the figure measures the similarity between an estimate and the ground truth, which will be explained in Section 6.5. Occasionally, due to the noisy radar data, the confidence C of the neural network will drop dramatically and the motion term M will increase, indicating that the system might be making a

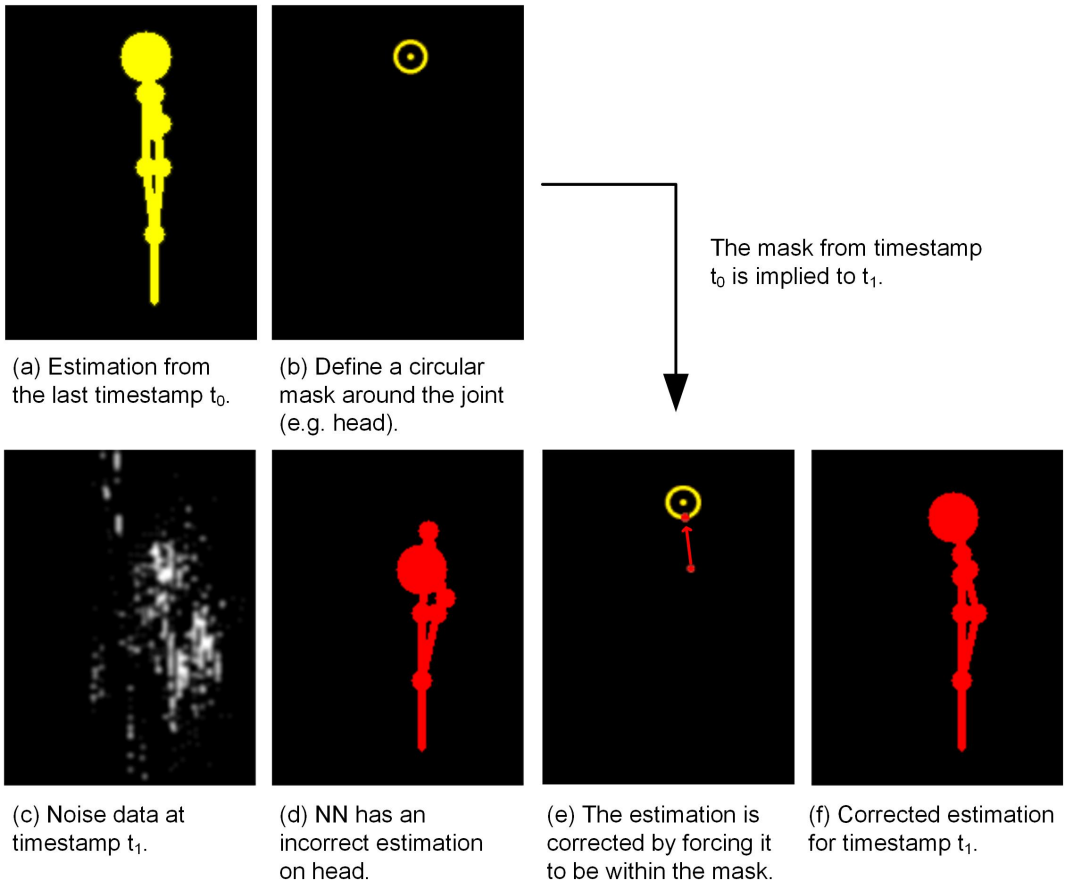
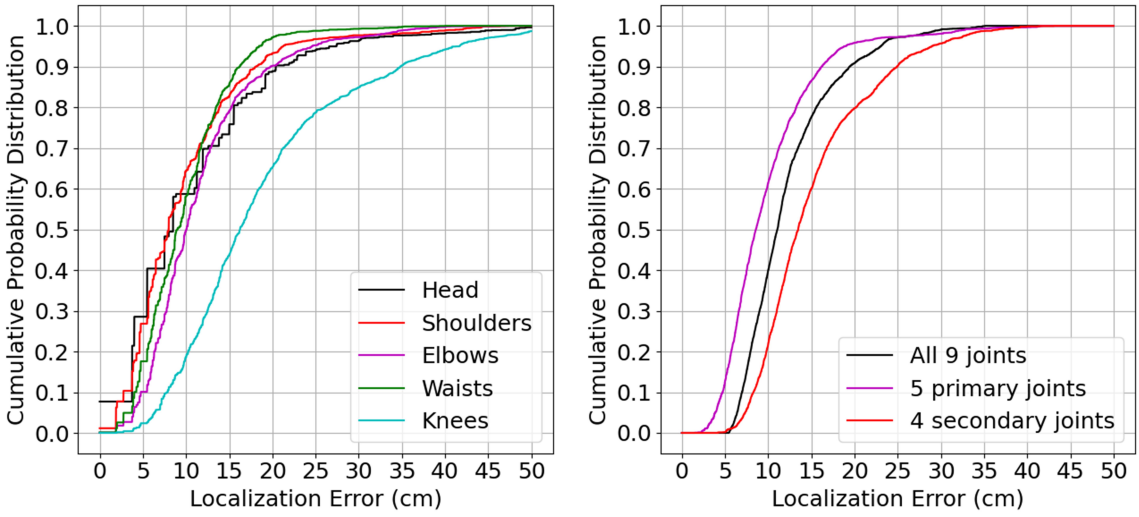


Figure 6.14: An example of how an estimate can be improved by restricting the maximum displacement of the joints.

mistakenly aggressive estimate. Such estimates are detected and corrected automatically through the temporal correlation step.

In addition, it is assumed that the joints will not move much between frames and a restriction on the maximum distance that each joint could move was set. Given the x-y coordinates (x_{v,t_0}, y_{v,t_0}) of a joint v at timestamp t_0, then for the next timestamp t_1, the estimate (x_{v,t_1}, y_{v,t_1}) will be forced to be within a certain distance of (x_{v,t_0}, y_{v,t_0}), as shown in Figure 6.14. The threshold is set to be 2 pixels in this system (approximately 8 cm), which is a rather loose criterion considering that the time between two frames is only 5 ms.

The temporal correlation step aims to reject any outlier estimate caused by any instability within the data, and hence improves the smoothness in the time domain. It is found to improve the stability of the estimation and reduce the variance, as will be shown in Section 6.5.



(a) Comparison between each type of joint.

(b) Comparison between all joints, the primary joints and secondary joints.

Figure 6.15: The cumulative distribution function of the localization error.

6.5 System Evaluation

The output from the proposed system was evaluated against the ground truth generated by the camera system. The first metric is the localization error of each joint. The localization error is calculated from the Euclidean distance between the estimated position and the ground truth position of the joints. Given that the 45×32 heatmap is supposed to project a $2\,\text{m} \times 1.5\,\text{m}$ area at $1.5\,\text{m}$ to $2\,\text{m}$, each pixel will represent approximately $4\,\text{cm}$. The mean localization error of all the 9 joints is $(12.2 \pm 5.2)\,\text{cm}$. The cumulative distribution function of the localization error of the full system is given in Figure 6.15. Figure 6.15a shows the localization error of each joint (or the mean localization error for paired joints). Figure 6.15b compares the mean localization error between all joints, primary joints and secondary joints. Both figures indicate that the system achieves a higher accuracy on the primary joints (the head, shoulders, and waist) over the secondary joints. During the experiment, it was found that the mmWave signal reflected from the legs and the feet is noisy due to a large amount of background reflection from the floor, which affects the system's ability to determine the position of the knees.

The average precision (AP) of the system using the OKS metric is also reported, as commonly used in computer vision [188]. The OKS measures how close an estimate is in comparison to the ground truth. The calculation of the OKS considers the relative size of different joints concerning the subject scale, as shown in Equation (6.4):

$$OKS = \sum_{v \in \{1 \ldots 9\}} (e^{-\frac{d_v^2}{2s^2k_v^2}})/9 \tag{6.4}$$

where d_v is the Euclidean distance between the joint's estimated position and the ground truth;

Table 6.1: AP (using OKS) and mean localization error (MLE) of the system, and a comparison to the literature.

	Method	AP (OKS=0.5)	AP (OKS= 0.5:0.95:0.05)	MLE (cm)
This research	**mmWave**	**0.959**	**0.713**	**12.2**
HRNet [66]	Camera	0.928	0.782	NA
UDP-Pose [189]	Camera	0.949	0.808	NA
RF-Pose2D [76]	RF	0.933	0.624	NA
RF-Pose3D [115]	RF	NA	NA	4.0-4.9
mm-Pose [116]	mmWave	NA	NA	2.7-7.5
CLGNet [119]	mmWave	NA	NA	27.9

s is the size of the subject to detect; k_v is a constant that controls the weight of joint v, which is pre-calculated based on the average size of the joints as in the optical images. The rationale behind the OKS metric is that a larger joint (e.g. a head) should have a higher error tolerance than a smaller joint (e.g. an elbow). A more accurate estimate will give a higher OKS, and an exact match will give an OKS of 1. The AP over the dataset is then calculated as the percentage of correct estimates from all frames, where an estimate is considered correct if its OKS value is higher than a certain threshold. For example, AP at OKS=0.5 is a loose metric that accepts an estimate if its OKS is greater than 0.5, whereas OKS=0.5:0.95:0.05 is a stricter metric that calculates the AP over 10 OKS thresholds from 0.5 to 0.95. A more detailed explanation of the AP+OKS metric can be found in [188]. The evaluation result of the system and a quantitative comparison between this research and the literature are shown in Table 6.1.

It should be noted that the performance figures of the research being compared are taken from the referenced papers rather than reproduced on the dataset developed for this research, since they are based on different methods and are designed with different setups, including the equipment used, types of postures involved, data format, training procedures, etc. Therefore, due to the lack of a common benchmark, it is difficult to carry out a strict comparison between these systems. Computer vision datasets are generally collected by an optical camera at different distances and scales, where the 3D geometrical information is not available. Therefore, the AP+OKS indicator is often used. In contrast, 3D datasets often use dedicated systems to collect the geometrical information (e.g. the 12-camera system in RF-Pose3D), which can be better evaluated through the localization error rather than the AP.

The result indicates that the system can effectively extract spatial features from the radar data and determines a person's posture, at a competitive performance to the state-of-the-art systems in both the computer vision field and the sensor field. The work by Sun et al. [66] and Huang et al. [189] are two of the state-of-the-art computer vision systems for posture estimation. They both used multiscale convolutional blocks to capture information from the images at different resolutions, and have achieved the top performance in 2D posture estimation due to the high amount of details in images. However, they are constrained by intrusiveness and lighting

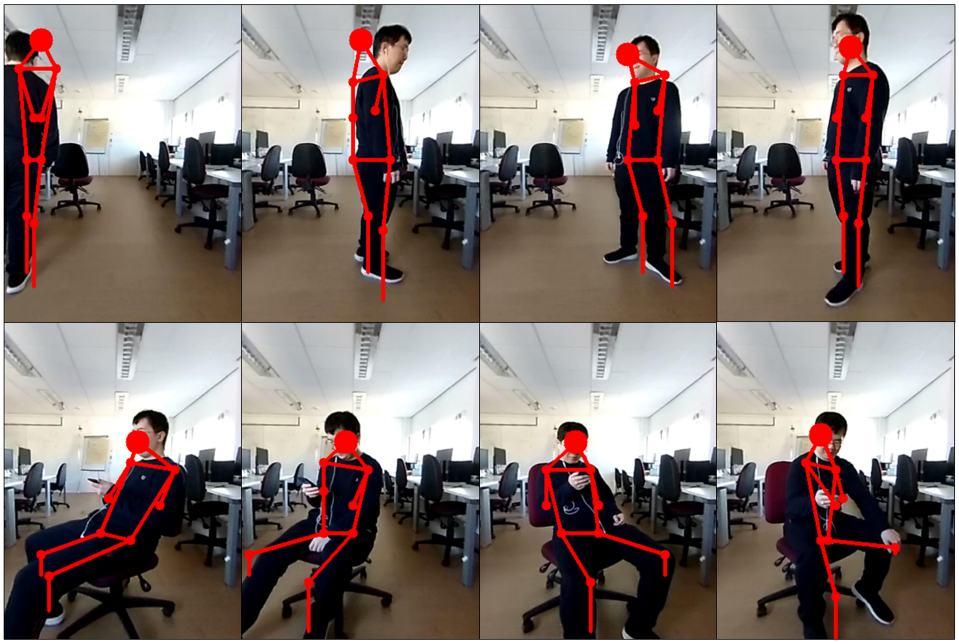


Figure 6.16: Example posture estimation results from the full system.

Table 6.2: Mean localization error at different stages.

	PD	PD+SM	PD+SM+TC
Mean (cm)	14.2	12.7	12.2
STD (cm)	6.8	6.5	5.2

conditions. Zhao et al. [76] and Zhao et al. [115] used a complex RF antenna array to scan the scene and obtain a signal heatmap, from which a neural network is applied to predict the joint position. The work by Sengupta et al. [116] and Wang et al. [119] are the most similar to this research. They both used mmWave radars to capture a point cloud that represents the spatial shape of the subject, and trained different CNN models for posture estimation. Although the localization errors in [115] and [116] are lower than found in this research, it still has its unique advantage. The research in [115] used a time window of 3 s for one input and used a large neural network (0.4 s processing time on a high-end GPU), whereas this system can process real-time radar data at 20 fps (more details in Section 6.6). The research in [116] contained only a few motions (walking and standing with arms swing) and the variation of certain joints is not significant, whereas this system is more generalizable and is capable of more complex and arbitrary postures. Therefore, this system provides a competitive solution for potential real-world applications. A visualization of some results from the system is given in Figure 6.16.

In order to evaluate the effectiveness of the spatial model and the temporal correlation step of the system, the localization error of the system was compared at different stages in Figure 6.17 and Table 6.2 (PD = part detector, SM = spatial model, TC = temporal correlation). While both the spatial model and temporal correlation have improved the performance of the system, the

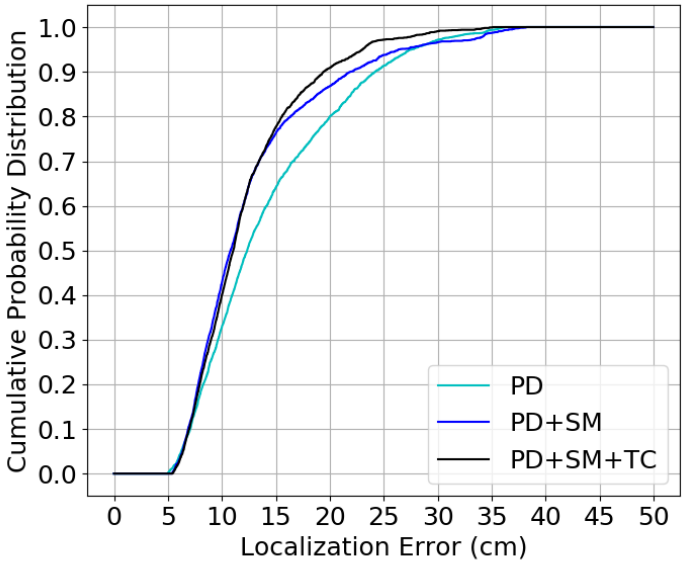


Figure 6.17: The cumulative distribution function of the localization error at different stages.

former improves more on the overall localization accuracy, whereas the latter is more effective at rejecting outliers and reducing the variance.

Figure 6.18 gives some examples that specifically compare the output between the part detector and the spatial model. When the point cloud data is noisy or ambiguous, the prior knowledge encoded in the spatial model can significantly improve the robustness of the model and avoid anatomically incorrect postures.

6.6 Real-time System Integration

The neural network model proposed in this chapter has been integrated into the framework mentioned in Section 5.3 to build a real-time posture estimation system. The system framework contains two Radar Handler modules for managing the communication between the computer and the two radars, two Frame Processor modules to process the data from the radar, and one Central Frame Processor to fuse the data and invoke the neural network model. The neural network model is initialized on the GPU when the system starts, including allocating memory, constructing the computational graph and loading the pre-trained weights into the model. The block diagram of the system framework is shown in Figure 6.19.

In the Frame Processor modules, a FIFO module was applied to stack data in the temporal dimension and a DBSCAN clustering module was used to filter out noise, as discussed in Section 5.5. The Central Frame Processor module synchronizes the output from individual frame processors and fuses the data into one frame. It transforms the fused frame into 2D images,

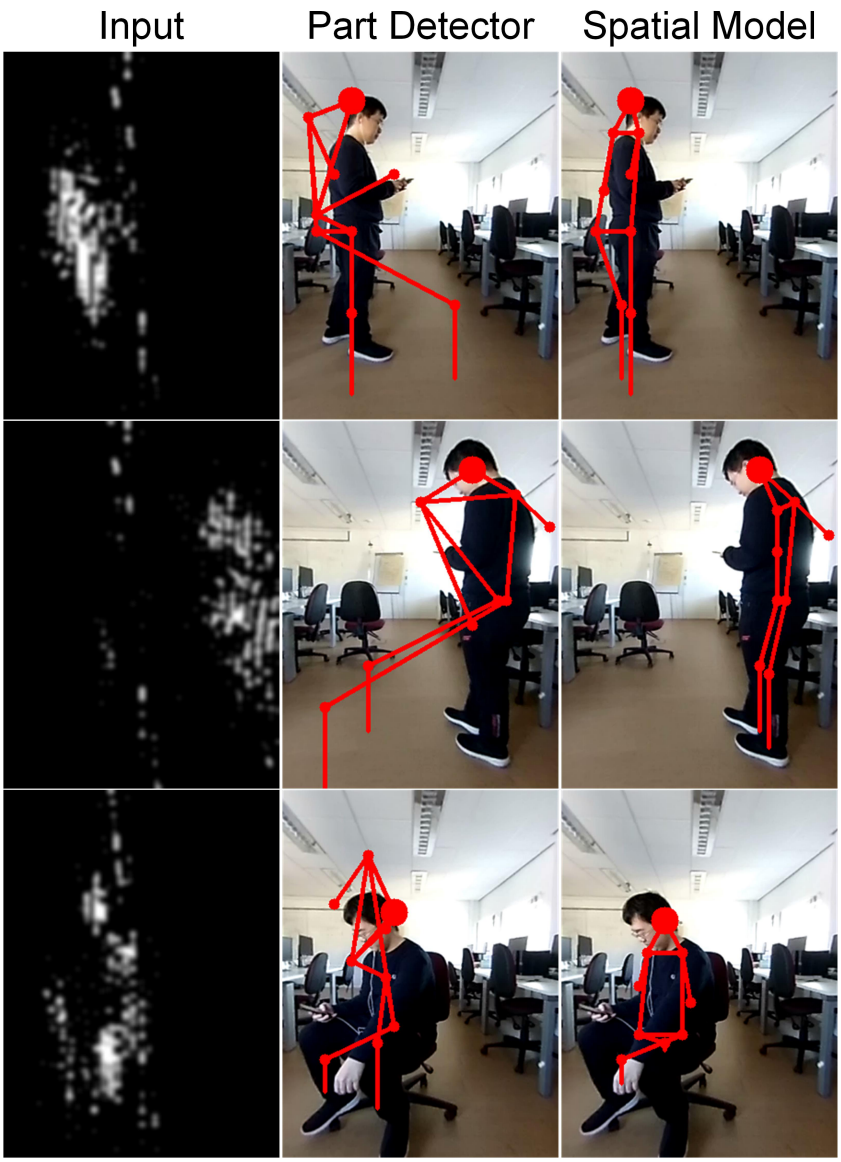


Figure 6.18: Some examples of the comparison between the part detector and the spatial model. Left: data input from the radar. Middle: output from the part detector. Right: output from the spatial model.

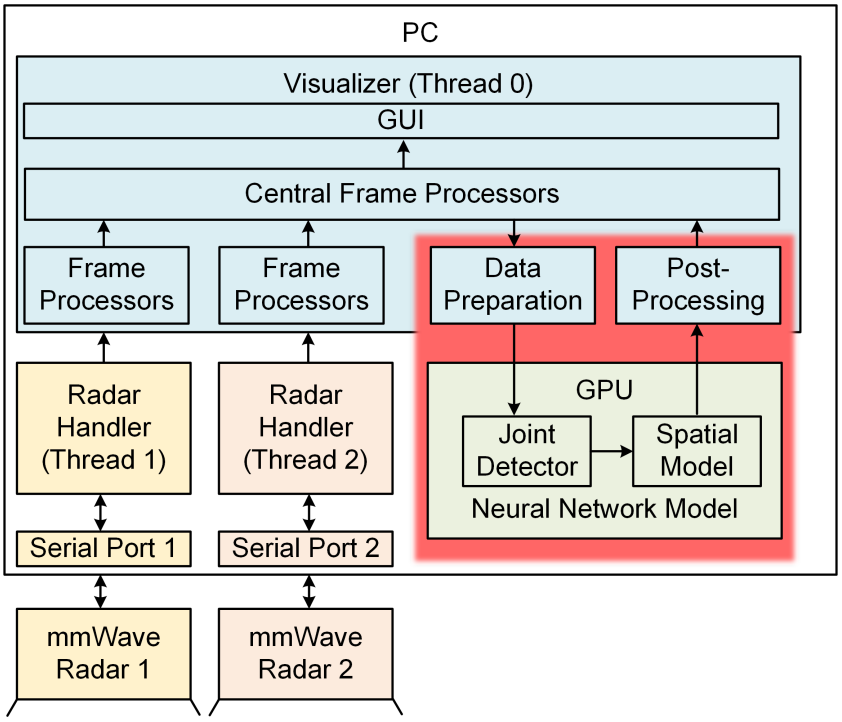


Figure 6.19: The complete system framework. The posture estimation part is highlighted in red.

feeds it into the neural network model, receives the output and displays the interpolated result. Since the size of a point cloud is always only a few hundred points, the time consumption of the pre-processing is negligible in comparison to the neural network, which is around 49 ms on an Nvidia RTX3070 GPU. Therefore, given that the bottleneck of the full system is at the neural network, the system can operate in real-time at 20 fps.

6.7 Operating on Embedded Platforms

As the computational resource and processing power can be limited in certain applications, the possibility of running the system on embedded platforms was evaluated. As neural networks on mobile and embedded platforms are becoming more common, many manufacturers are making dedicated systems for lightweight machine learning tasks. For example, TI provides the AM57x SoC at around £30, with the C66x digital signal processors [190, 191] and the Embedded Vision Engine (EVE) subsystems that are dedicated for accelerating neural network operations, with a low power-consumption of around 5 W. Although it only supports 8-bit integer operation rather than floating-point numbers like a GPU, it is possible to compress a network through quantization, at the expense of lowering the precision. According to the TI deep learning framework [192], the EVE unit can perform 16 8-bit MAC operations per clock cycle and it is typically clocked at 650 MHz, which is 10.4G MAC per second. TI has verified that the EVE unit can execute a few small neural networks in real-time, such as the InceptionNetV1 network (1.5G MAC operations

and 6.8M parameters [193]) in 785 ms [192].

Alternatively, the Nvidia Jetson GPU is a more powerful product series of embedded platforms. For example, the Jetson Nano has a power consumption of 10 W and a performance of 472G floating point operations per second, which can execute large state-of-the-art networks like the InceptionNetV4 network (12G MAC operations and 43M parameters [194]) in 13 fps. The GPU architecture, although more expensive, does not require further optimization like quantization on the network, and therefore can retain a higher performance of the network.

The proposed neural network in this research has only around 373M MAC operations, significantly less than the InceptionNetV1 (26%) and InceptionNetV4 (3%). The total number of parameters is around 19M and can be easily fitted into the memory of an embedded platform. Therefore, it is possible to port the network to either the AM57x SoC or the Jetson platform while still operating in real-time.

6.8 Conclusion

In this chapter, a real-time human posture estimation system using commercial mmWave radars has been shown. The system consists of a data pre-processing module to convert the radar point cloud into 2D images, a neural network model to process the images and generate heatmaps of joint positions, and a post-processing module to exploit the temporal correlation between time frames and refine the estimate. The experimental setup focused particularly on short-range detections within two metres. In contrast to many existing mmWave-based systems which mainly focus on standing postures, this system can estimate arbitrary standing and sitting postures of a person, with a mean localization error of 12.2 cm and an average precision of 71.3%, while still maintaining a high processing speed and low cost. It was shown that the system can operate in real-time at 20 fps, from data collection, data processing to result visualization. The system provides a low-cost and non-intrusive monitoring solution as it only collects anonymous data rather than sensitive visual information. Therefore, it can be of great interest in many real-world applications where privacy is important, such as health monitoring and elderly care.

CHAPTER 7

HUMAN VITAL SIGN DETECTION

In this chapter, a novel human heart rate (HR) detection system using one mmWave radar is presented. In contrast to many existing systems that focus on detecting the vital signs of a stationary person, this system aims to detect the heart rate of a person while exercising on a treadmill. A novel phase signal construction algorithm is presented for detecting the chest displacement of the person, and a machine learning model is trained for predicting the trend of HR change during exercise. It will be shown that the system can detect the person's HR in a complete exercise cycle with a low error rate of 5.4%. The phase signal construction algorithm described in this chapter has been submitted in a patent application [27].

The rest of the chapter is organized as follows. Section 7.1 gives the background of this work. Section 7.2 discusses the problem of phase ambiguity when measuring the heart rate using radar and how it can be addressed. Section 7.3 presents the heart signal detection algorithm, including the phase construction algorithm and the machine learning model. Section 7.4 shows the evaluation result of the system when measuring an exercising person's heart rate. Section 7.5 concludes the chapter.

7.1 Overview

Cardiovascular disease is one particularly important health problem around the world and is one of the major threats for people in midlife and older ages. Regular monitoring of people's HR has been shown to be an effective way of assessing their cardiac health and early detection of potential diseases [6]. Although some commercial products, like chest bands and smartwatches, are available for long-term HR monitoring, it is often impractical to ask the subject to wear it all the time, and there is a possibility that the device can be damaged, need calibration or be

lost. These devices can also be expensive to people with a low income. Therefore, contactless HR monitoring has been studied for decades as an alternative to contact sensors, among which radars have received the greatest interest. However, the majority of the research assumes the person being monitored is stationary or has restricted movement, which is often a too optimistic assumption and is not suitable for daily use. Accurate HR monitoring under free body movement remains a challenge [120, 121].

Generally, a person's chest movement due to heartbeat is around 0.2 mm to 0.5 mm at 1 Hz to 1.34 Hz when at rest [195]. The phase information of radio-frequency signals emitted from the radar and reflected from the person can be analysed to capture the chest displacement. There is research on using mmWave radars to measure the HR [16, 123, 126, 196]. However, most of them require the subject to sit or lie at a known distance, as any movement of the person would be significantly higher than the chest displacement and make the detection much harder. Some work measure the HR when the person is in limited motion, such as when walking in a straight line at a low speed [131], but cannot deal with more complex motion.

In this paper, a mmWave radar-based system is proposed to detect a person's HR while exercising on a treadmill, as an attempt to solve the problem of HR estimation with free body movement and provide insight for other applications in future work. The radar operates at an extremely high frequency to capture the information of the scene at the highest possible resolution. Then, a novel phase signal construction algorithm is presented that can track the position of the person among all the noise signals, where the phase signal would encode the displacement of the person's chest. Finally, an HR tracking algorithm is proposed based on a support vector machine (SVM), to continuously estimate the person's HR during the exercise.

The principle of the mmWave radar has been introduced in Chapter 3. The radar sends a chirp signal, receives the reflection, and generates an IF signal that encodes the distance between the radar and the subject. The IF signal at any chirp can be expressed as Equation (7.1):

$$IF(t) = e^{j(\omega t+\phi)} \quad \text{where } \omega = 2\pi \cdot S\tau, \quad \phi = \frac{4\pi d}{\lambda_0} \tag{7.1}$$

where S is the chirp slope, τ and d are the ToF and distance between the subject and the radar, and λ_0 is the signal wavelength at the chirp starting frequency. The radar sends chirp signals at a high rate (5000 Hz in this research), where the chest displacement of the subject in front of the radar will be embedded in the phase variation of the IF signal. Let $\phi_0 = \frac{4\pi d_0}{\lambda_0}$ denote the initial detected phase of the subject at chirp 0, then $\phi_c = \frac{4\pi(d+\Delta d_c)}{\lambda_0}$ denotes the phase at chirp c with a displacement Δd_c. Collecting the phase ϕ for a number of chirps N_c gives a phase signal PS that can be expressed as:

$$\begin{aligned} PS &= \left\{ \frac{4\pi(d_0 + \Delta d_c)}{\lambda_0} | c \in \{1...N_c\} \right\} \\ &= \frac{4\pi d_0}{\lambda_0} + \left\{ \frac{4\pi(\Delta d_c)}{\lambda_0} | c \in \{1...N_c\} \right\} \end{aligned} \tag{7.2}$$

where the constant term $\frac{4\pi d_0}{\lambda_0}$ represents the initial position of the subject, and the second term represents the displacement of the subject at each chirp.

HR detection can be achieved by analysing the phase signal PS and extracting the periodical change of d_c that corresponds to the HR. However, this is more challenging when the person is not stationary. Equation (7.1) implicitly assumes that the distance between the person and the radar does not change within the duration of a chirp. This assumption is mostly true when the person is stationary or moving at a low speed. However, the accuracy of the estimation can drop when the person is exercising. Assuming the person has an instantaneous speed of 1 m/s and the chirp duration is 100 us, then the displacement within the chirp will be 0.1 mm and will give a phase noise of around 0.1π. In addition, the movement of other body parts, like the limbs, can make the IF signal much more noisy and make it harder to extract the correct frequency and phase term corresponding to the chest. This research aims to solve the mentioned problem by using a combination of the traditional frequency-based approach and a machine learning-based HR tracking algorithm to estimate a person's HR in exercise.

7.1.1 Raw Data Capturing

HR estimation requires raw IF signal of the radar to be transmitted to the PC for post-processing. This can be performed by using a modified version of the framework discussed in Section 5.3, with a DCA1000 data capturing card. The overview of the data flow is shown in Figure 7.1. The framework uses a modified version of TI's DCA1000EVM CLI software, a dedicated driver designed for controlling the DCA1000 board and receiving the data [197]. The original version of the software only supports dumping data to files, whereas the modified version support real-time data streaming to the presented software framework through socket programming.

In addition to the modules discussed in the previous sections, the framework uses a DCA1000 Handler module to control the DCA1000 board. When the system starts, the DCA1000 board will be configured and started through the DCA1000 Handler module and the DCA1000EVM CLI software. A configuration file is used to specify the radar model, the data capturing mode and the corresponding parameters. Then, the radar will be configured and started with LVDS streaming enabled. Once the radar starts operation, it will dump raw IF signals through the LVDS lanes to the DCA1000 board, which will then be transmitted to the PC through a gigabit Ethernet port. The data will be received by the modified DCA1000EVM CLI software and be streamed to the DCA1000 handler module. The DCA1000 handler module will decode the data, arrange it into an appropriate matrix format and send it to the Visualizer for further processing.

While the point cloud data is sparse and only takes up to a few Kbps (kilobits per second) bandwidth, the raw data is dense and requires significantly higher bandwidth. For example, the proposed system in this chapter captures 5000 chirps per second and 1500 samples per chirp, which corresponds to 240 Mbps (million bits per second) bandwidth per receiver. The DCA1000 board supports up to four LVDS lanes, and each lane supports up to 600 Mbps. However, the

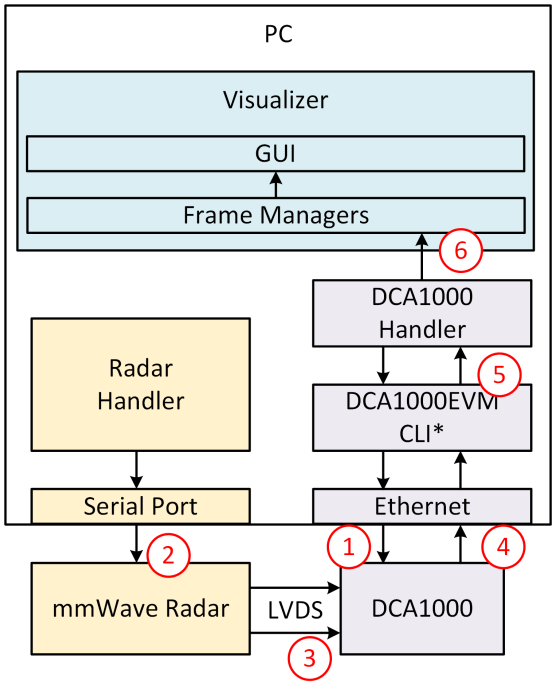


Figure 7.1: Software framework when capturing the raw data from a radar. 1) Configure the DCA1000 board. 2) Configure the radar. 3) The radar starts dumping data to the DCA1000 board. 4) The data is received by the DCA1000EVM CLI software. 5) The data is transmitted to the DCA1000 handler. 6) Process the data.

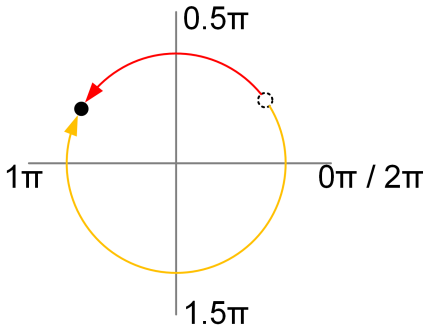


Figure 7.2: When a change in the phase is observed, the red and the yellow path show two possible interpretations of the object's motion.

Ethernet port supports up to only 706 Mbps [173] and becomes the bottleneck of the system. Therefore, the bandwidth can become an important constraint when designing systems that require raw data capturing from multiple receivers at a high data rate.

7.2 Phase Ambiguity and Unwrapping

Since the phase of a signal has a range of $[0, 2\pi]$, it can become ambiguous when the subject is moving fast. For example, as shown in Figure 7.2, when a change in the phase is observed, there exists more than one possible interpretation of the object's motion. Therefore, to avoid such ambiguity, the sampling rate of the phase has to be greater than the motion of the object, so that the maximum phase change will be within π:

$$\Delta\phi < \pi \tag{7.3}$$

where $\Delta\phi = \frac{4\pi\Delta d}{\lambda_0}$ (derived from Equation (7.2)) is the phase change between two successive chirps due to the displacement Δd of the object. Substituting this into Equation (7.3) gives:

$$\begin{aligned} \frac{4\pi\Delta d}{\lambda_0} &< \pi \\ \Delta d &< \frac{\lambda_0}{4} \end{aligned} \tag{7.4}$$

where λ_0 is the wavelength of the signal at the chirp starting frequency (77 GHz) and corresponds to a wavelength of 3.9 mm. Therefore, to avoid phase ambiguity, the displacement of the object between each measurement has to be within 1 mm. In the experiment, the radar is configured to operate at 5000 Hz (5000 chirps per second), so that it can measure a person moving at up to 5 m/s without introducing any phase ambiguity. Taking this assumption to the example in Figure 7.2, it can then be inferred that the red path is the correct one and the yellow path should not happen.

By assuming that the phase changes between any two successive measurements would be within π, the phase can be unwrapped to restore the original motion of the object. Taking a set of phase measurements Φ, the phase difference between every measurement can be calculated and any difference greater than π can be unwrapped, as shown in Algorithm 2. An example of how phase unwrapping helps restore the object's motion can be seen in Figure 7.3, where the yellow points are the measured phase in the range $[0, 2\pi]$ and the blue points are the unwrapped phases representing the actual motion (a sinusoidal oscillation). The phase unwrapping concept will be applied during the processing of the phase signal in later stages.

Algorithm 2 Phase unwrapping algorithm.

Input: Raw phase signal Φ.
Output: Unwrapped phase signal Φ.

```
n = length(Φ)                          ▷ Get the signal length
for i in {1, n} do                     ▷ Iterate through each sample
    if (Φ_i − Φ_{i−1} > π) then        ▷ If phase increases by over π
        Φ[i : n] = Φ[i : n] − 2π       ▷ Shift signal downward
    else if (Φ_{i−1} − Φ_i > π) then   ▷ If phase decreases by over π
        Φ[i : n] = Φ[i : n] + 2π       ▷ Shift signal upward
```

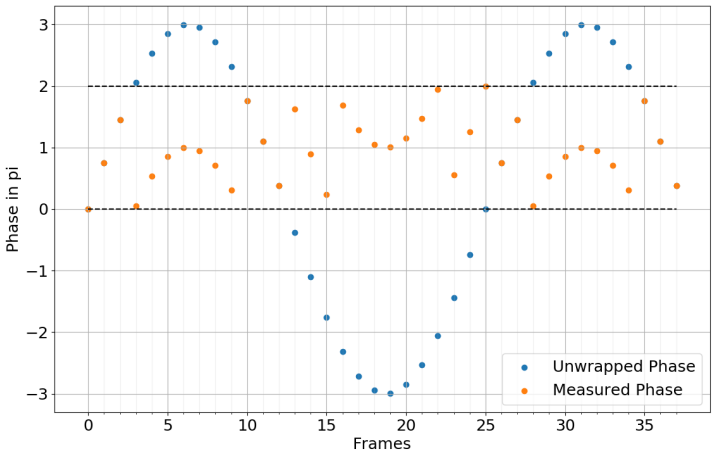


Figure 7.3: The motion of an object can be restored by unwrapping the phase signal.

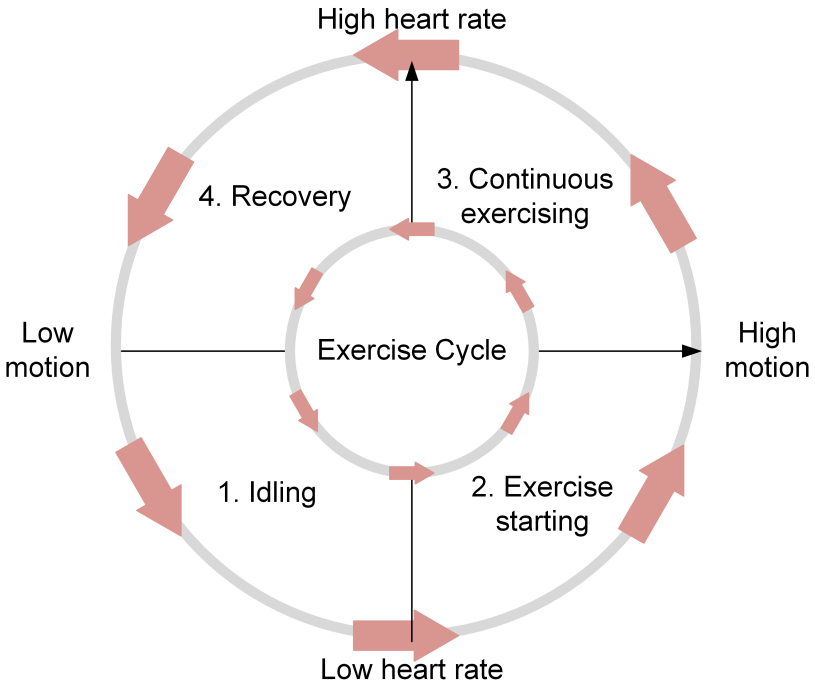


Figure 7.4: The four stages of a common exercise cycle.

7.3 Heart Signal Detection

When exercising on a treadmill, a full exercise cycle can be defined with four stages: idling, exercise starting, continuous exercising, and recovery, as shown in Figure 7.4. In stages one and three, there is often a positive correlation between the motion level and the HR, and researchers have designed systems to detect the person's HR during these periods, such as [35]. However, there are few researches considering stages two and four, where the motion level and the HR might not be correlated. The system proposed in this research aims to track the HR of the person

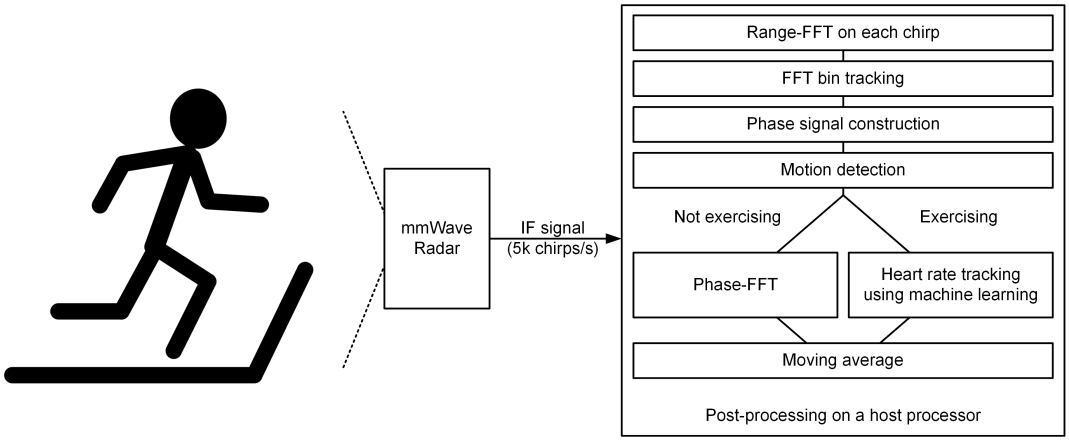


Figure 7.5: flowchart of the proposed algorithm.

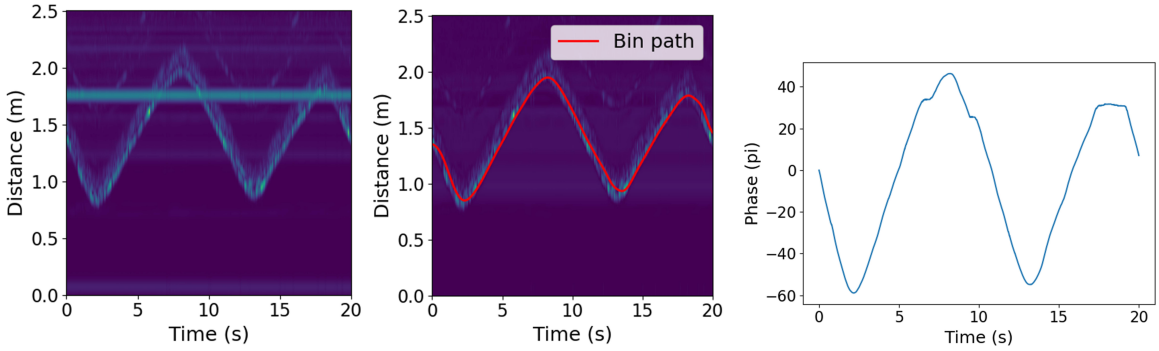
during the complete exercise cycle.

The procedure of using mmWave radars for HR detection is shown in Figure 7.5. The system operates following a sliding-window approach. It uses a time window of 5 s (referred to as one frame) and a refresh rate of 1 Hz. Each second of data contains 5000 chirps. First, a range-FFT is applied to the IF signal at each chirp to obtain its frequency spectrum, and a bin tracking algorithm is used to find the bins corresponding to the person's position across all the chirps. Each range bin will have a corresponding phase, so a continuous path of range bins in the time domain can form a phase signal of length 5000 per second. Then, based on the phase signal, the system judges if the person is exercising or not, and applies the corresponding algorithm to determine the HR. The system predicts the HR once per second, where the predictions are smoothed with the previous predictions using a moving average approach, to ensure a stable reading. The following sections explain each stage of the system.

7.3.1 Phase Signal Construction

The radar generates an IF signal at each chirp, as shown in Equation (7.1). A range-FFT is applied to each IF signal to compute a frequency spectrum of length N_s^*, where N_s^* is the length of the FFT and the peak in the spectrum corresponds to the frequency ω_b. For all chirps in one frame, a 2D range-time spectrum as a matrix of size $TN_c \times N_s^*$ can be constructed that encodes the position of the person over time, where $N_c = 5000$ represents 5000 chirps per second, $T = 5$ represents the 5 s time window and TN_c represents the total number of chirps in the time window. Figure 7.6a shows an example of the 2D matrix where a person is walking back and forth for a short distance.

To construct a phase signal of the person, the range bin at each chirp needs to be identified. Ideally, the chosen range bin should reflect the motion of the subject's chest. However, with the assumption of free body movement, the range-FFT can become noisy and the location of the bin can have a large variance. Therefore, a novel bin tracking and phase construction algorithm is



(a) The 2D range-time spectrum obtained from the IF signal that encodes the motion of the person over time.

(b) The range-time spectrum after clutter removal and the detected bin path.

(c) The extracted phase signal of the person.

Figure 7.6: An example of the phase construction step.

proposed.

Let $R^{(TN_c \times N_s^*)}$ denote the range-time spectrum from the range-FFT. First, a clutter removal step is applied, where the averaged power from all chirps is subtracted from the spectrum and the absolute value of the resulting spectrum is calculated. This step filters out the signal from static objects that have a near-zero variance in the spectrum, as a person can hardly be absolutely stationary. The clutter removed range-time spectrum is denoted as $\bar{R}^{(TN_c \times N_s^*)}$ and an example is shown in Figure 7.6b. When the system starts, a range spectrum is generated by averaging the magnitude of the spectrum ($|\bar{R}|$) in the first 50 chirps, and the subject's initial position s_1 is found by applying a CFAR peak detection algorithm on the range spectrum.

Given an initial range bin s_1, the tracking algorithm attempts to determine the most possible range bin in the subsequent chirps s_c where $c \in [2, TN_c]$. For each chirp c, it is assumed that the new bin s_c would fall within a certain range around the last bin s_{c-1}, where the probability distribution follows a Gaussian distribution. Therefore, a Gaussian distribution centred at s_{c-1} is established as $GD(s_{c-1})$ and is multiplied by the range-FFT magnitude at chirp c, to produce a spectrum R'_c of length N_s^*:

$$R'_c = GD(s_{c-1}) \cdot |\bar{R}_{c,\cdot}| \tag{7.5}$$

The position of the peak of the output spectrum R'_c is taken as the new bin s_c and the algorithm is applied iteratively to all chirps. The width of the Gaussian distribution controls the trade-off between the sensitivity to motions and the stability of the tracked phase. A minimal score is set empirically for accepting the bin index, so that the tracking process will terminate when no person is detected or when the person has left the scene. An example of the tracking processing is shown in Figure 7.7, where the red line is $GD(s_{c-1})$, the green line is $|\bar{R}_{c,\cdot}|$ and the blue line is R'_c.

Once the bins have been determined for all chirps, a bin path in the range-time spectrum

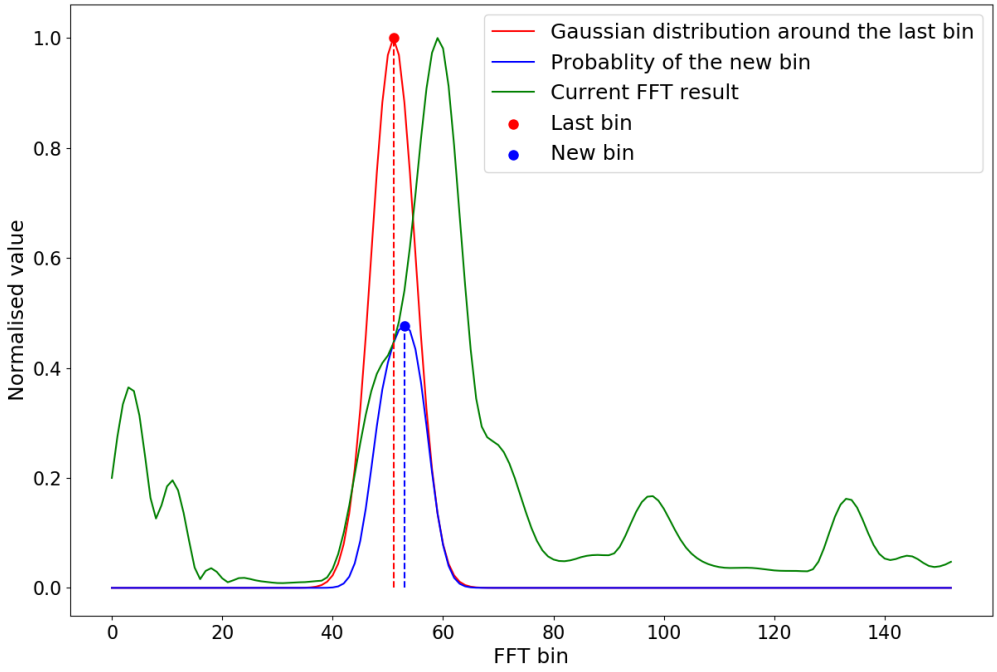


Figure 7.7: Tracking the FFT bin index using a Gaussian distribution.

can be constructed as $BP = \{s_c | c \in \{1...TN_c\}\}$ (as shown as the red line in Figure 7.6b). The bin path represents correspondence of the person's motion in the frequency domain. Then, the phase signal of the person can be extracted from the range-time spectrum following the bin path:

$$PS = \{\angle \bar{R}_{c,s_c} | c \in \{1...TN_c\}\} \tag{7.6}$$

The phase signal PS represents the motion of the person with respect to the signal wavelength and can achieve millimetre-level resolution, as described in Equation (7.2).

However, a phase signal extracted directly from the path bins will contain discontinuities whenever there is a bin change. Therefore, to maintain the smoothness of the phase signal, one additional processing step is introduced. First, a phase spectrum $\Phi^{(TN_c \times N_s^*)} = \angle \bar{R}^{(TN_c \times N_s^*)}$ is calculated for all chirps and range bins. For each range bin s, the phases of all chirps are collected as a signal $\Phi_{.,s}$ and are unwrapped (denoted as $U\Phi_{.,s}$) as described in Section 7.2. The first-order derivative of $U\Phi_{.,s}$ is then calculated as $U\Phi'_{.,s}$. Then, the bin path BP is smoothed using a Gaussian kernel. The smoothed path would have non-integer bins that need to be interpolated from neighbouring bins using the corresponding value in $U\Phi'_{.,s}$. For example, the phase value of a non-integer bin 10.1 at a chirp c will be linearly interpolated from bin 10 and 11 as $U\Phi'_{c,10.1} = (0.1U\Phi'_{c,10} + 0.9U\Phi'_{c,11})$. Applying this procedure to all bins in BP gives a phase signal PS' as a smoothed first-order derivative of PS and allows a smooth phase signal to be

re-constructed. An example of the tracked phase signal is shown in Figure 7.6c.

While the absolute phase values encode the position of the person, the changing of the phase is more important in identifying the chest motion due to the heartbeat. Chest movement due to heartbeats happens within a very short time (around 0.1 s), which will result in a rapid change in the phase. Researchers (as in [16]) have shown the effectiveness of using the phase difference, or more specifically the first-order derivative of the phase signal, in detecting the HR. Due to the movement of the person, the unwrapped phase signal can vary from a few π to thousands of π, whereas the phase derivative is limited to the range of $[-\pi, \pi]$ and is easier to compute, given the assumption that the measuring rate is high enough so that the displacement of the subject between two chirps is small. Therefore, instead of reconstructing the raw phase signal PS, its derivative PS' can be used directly for the next stage.

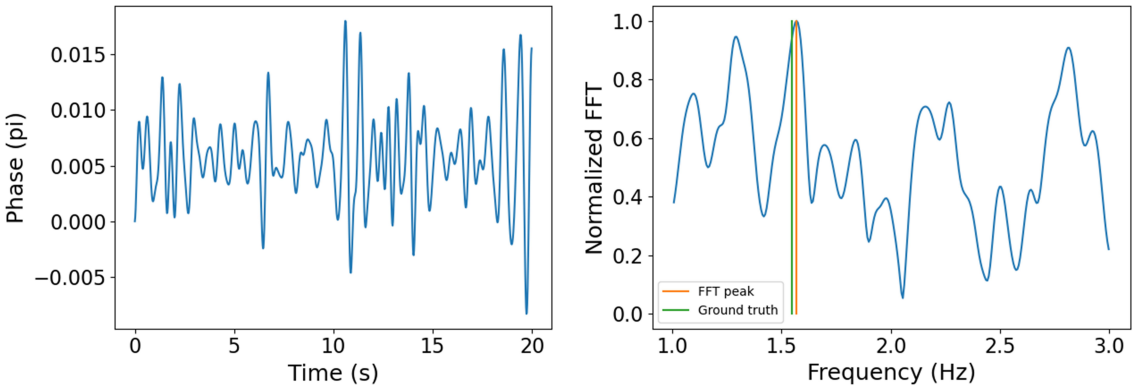
Algorithm 3 Phase signal construction algorithm.

Input: A range-time spectrum R of size $(TN_c \times N_s^*)$.
Output: The first order derivate of the phase signal, PS', with length TN_c.

for s in $\{1...N_s^*\}$ **do** ▷ Clutter removal
  $\bar{R}_{\cdot,s} = |R_{\cdot,s} - \text{mean}(R_{\cdot,s})|$
$s_1 = \text{peak}(\text{mean}(\bar{R}_{1:50,\cdot})$ ▷ Predict the initial bin
$BP = \{s_1\}$
for c in $\{2...TN_c\}$ **do** ▷ Bin tracking
  $R'_c = GD(s_{c-1}) \cdot |\bar{R}_{c,\cdot}|$
  $BP_c = \text{peak}(R'_c)$
$BP = \text{Gaussian\_filter}(BP)$ ▷ Smooth the bin path
$\Phi = \angle \bar{R}$ ▷ Calcualte the phase
for s in $\{1...N_s^*\}$ **do** ▷ Unwrap and calcualte the derivative
  $U\Phi_{\cdot,s} = \text{unwrap}(\Phi_{\cdot,s})$
  $U\Phi'_{\cdot,s} = \text{derivative}(U\Phi_{\cdot,s})$
$PS' = 0$
for c in $\{1...TN_c\}$ **do** ▷ Generate the phase signal
  $b_{float} = BP_c$
  $b_{int} = \text{round}(b_{float})$
  $PS'(b) = \text{interpolate}(b_{float}, U\Phi'_{c,b_{int}}, U\Phi'_{c,b_{int}+1})$

7.3.2 Phase-FFT

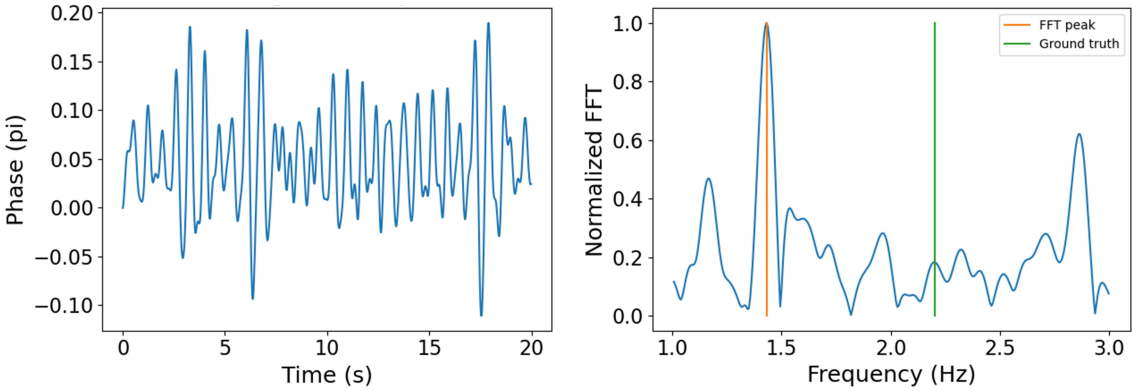
When the subject is stationary, the displacement of the chest due to heartbeats can be considered as a periodical signal with a certain frequency between 1 Hz to 3 Hz. Therefore, it is possible to extract the signal by applying a bandpass filter over the desired frequency range and an FFT on the filtered phase signal. For a stationary subject, the FFT spectrum will have minimal interference from other sources of movement, making the HR distinguishable. However, as the phase signal is highly sensitive to movement and the subject cannot stay absolutely stationary in



(a) Phase signal of the person when stationary.

(b) Phase FFT result.

Figure 7.8: Example of the phase signal and phase-FFT when a person is stationary.



(a) Phase signal of the person when exercising.

(b) Phase FFT result.

Figure 7.9: Example of the phase signal and phase-FFT when a person is exercising.

practice, the FFT output will be sensitive to noise and have outliers. Two examples of the phase signal and the phase-FFT spectrum are shown in Figure 7.8 and Figure 7.9, when the person is stationary and exercising, respectively.

When the person is stationary, the amplitude of the phase signal is low and the FFT peak from the HR is easier to detect. When the person is moving or exercising, the amplitude becomes much more significant due to the body movement, and there could be a strong frequency component from the movement that dominates the FFT spectrum. Although the HR is still visible from the spectrum, it is much smaller and impractical to distinguish from the noise. Therefore, further processing is required to identify the correct frequency component for the HR.

7.3.3 Heart Rate Tracking

When the subject is exercising, the signal from the heartbeats will be small when compared with the signal from the body movement, as the body movement will also be periodical but with a much larger amplitude. To address the mentioned issue, a machine learning model that attempts to model the relationship between the motion level and the HR is trained. Given a phase signal PS', the motion level of the subject is defined as the trimmed mean of $|PS'|$ with a 20% cut-off and denoted as M. When M is above a certain threshold, the HR tracking model will be triggered, which takes several properties of the phase signal as the input and predicts the most possible HR change in the frame.

First, an FFT is applied to the phase signal PS' as described in Section 7.3.2. All local peaks in the FFT spectrum are extracted as the candidate frequencies of the heartbeat. The candidates are considered to have an equal probability regardless of their power, since the HR signal can have a weak amplitude. Then, the problem of HR detection is transformed into a classification problem: given certain properties of the phase signal and the last known heartbeat frequency, the model needs to determine if the frequency will increase, decrease or stay at the same level. Based on the model prediction, the system can then pick the corresponding frequency among all the candidates.

Let H_{last} denote the heartbeat frequency in the previous frame, and the two closest frequencies in the candidate set of the current frame will be taken as H_{hi} and H_{lo} ($H_{hi} > H_{lo}$). Let H_{max} and H_{min} denote the maximum and minimum HR to detect (3 Hz and 1 Hz in this research). A normalization term HN is defined as $\frac{1}{(H_{max}-H_{min})^2}$. Then, the following variables are calculated as the input to the machine learning classifier:

1. $HN(H_{last} - H_{max})^2$.
2. $HN(H_{last} - H_{min})^2$.
3. $HN(H_{last} - H_{hi})^2$.
4. $HN(H_{last} - H_{lo})^2$.
5. M_t: The current motion level.
6. $M_t - \text{mean}(M_{t-15:t-10})$: The difference between the current motion level and the average motion level in the past 15 s to 10 s.

A linear SVM is used as a binary classifier, where a positive prediction corresponds to H_{hi} and a negative prediction corresponds to H_{lo}. An example of this process is shown in Figure 7.10. Given the last prediction is around 125, the two most possible HRs based on the phase-FFT result would be the left and right peaks (120 and 127, respectively), where the SVM will be used to select the result based on the described variables.

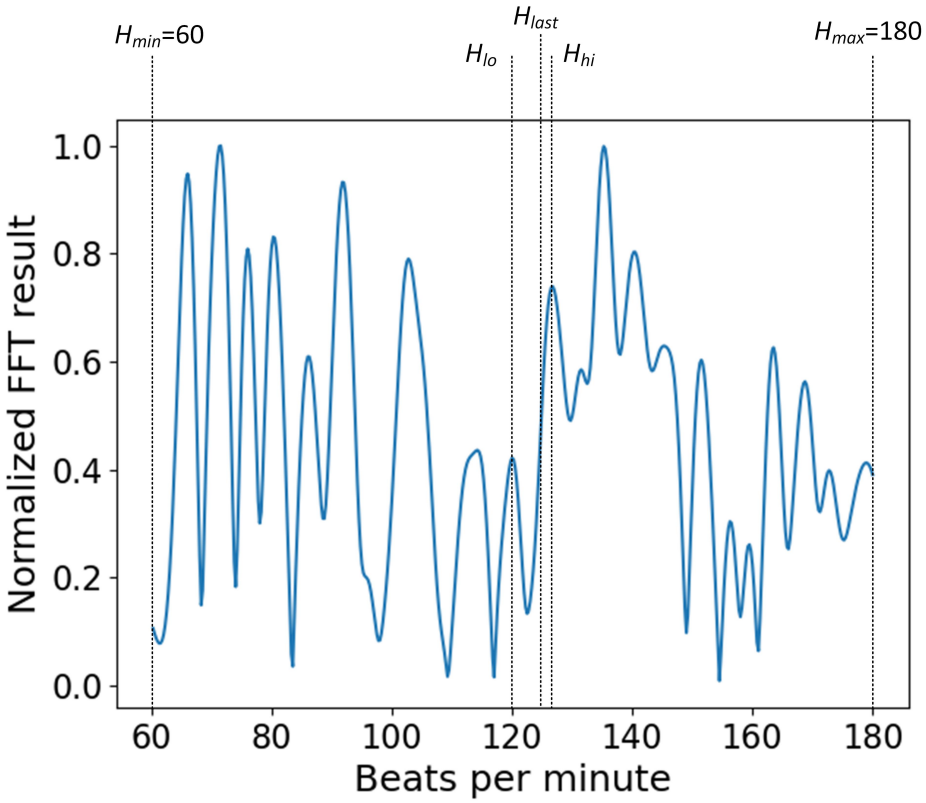


Figure 7.10: Example of using an SVM to predict the HR based on the phase-FFT result.

When considering a complete exercise cycle, it is always assumed that the person starts from the idle state, where the stationary HR estimation procedure is applied. Then, upon the detection of a higher motion level, the machine learning model is applied to estimate the HR for the rest of the time. In order to provide a smooth real-time reading, the estimation is performed once per second and a moving average of the HR within a 30 s window is reported. Reporting the moving average can strongly improve the system's tolerance to outliers and relax the requirement on the phase-FFT accuracy, which allows a small time window to be used when constructing the phase signal.

7.4 Evaluation

7.4.1 Experimental Setup and Dataset

The experimental setup is shown in Figure 7.11. In the experiment, one IWR1843 radar is used and configured as follows: The chirp frequency is 77 GHz to 81 GHz, the slope is 21 MHz/us, the chirp duration is 180 us, the ADC sampling rate is 9 MHz, each frame is 50 ms with 250 chirps (i.e. 5000 chirps per second), and each chirp has 1500 samples. The radar is placed at approximately the same height as the person's chest and 1 m to 2 m away from the person. Only one pair of

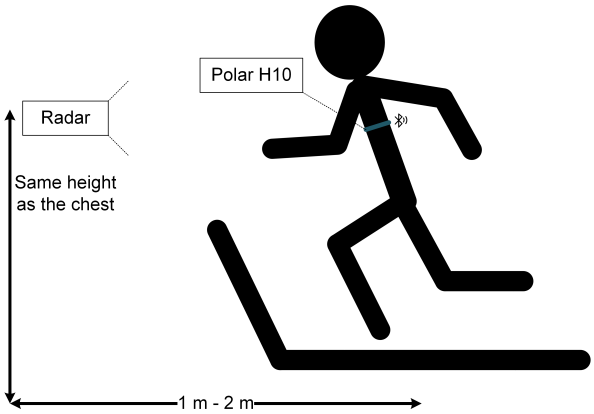
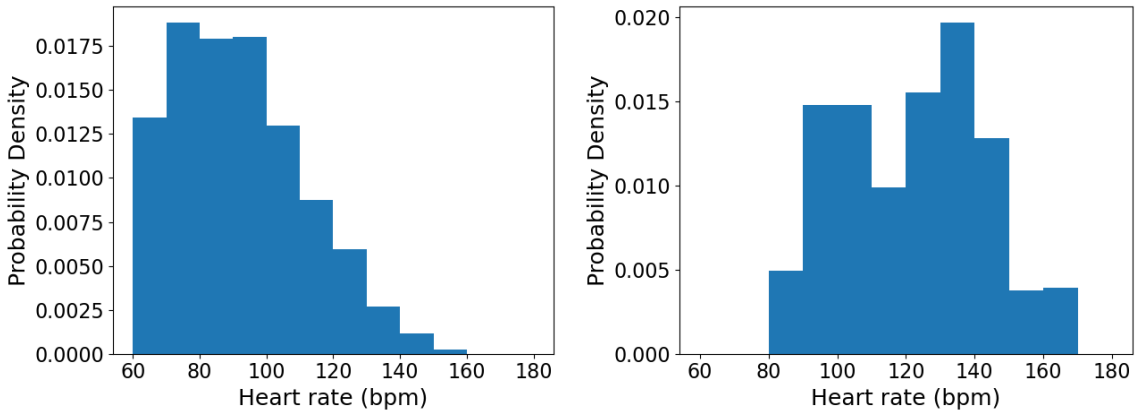


Figure 7.11: Experimental setup.

transmitter and receiver is used, as this research focuses on the HR estimation of only one person. The software framework described in Section 7.1.1 is used for data collection and processing. The ground truth HR is collected using a Polar H10 chest band, a portable ECG-based sensor that have demonstrated reliable performance in the industry [198]. The H10 device is connected to the PC using Bluetooth and is integrated into the framework using the standard Bluetooth Low Energy protocol. It detects and reports the HR of the person approximately once per second. The data from the H10 device and the radar data will be timestamped and synchronized based on their arrival time to the PC. Since the update rate of the HR estimation system is set to one update per second, explicit data synchronization at a higher rate is not required.

Two datasets were collected for evaluating the system. The first dataset contains the radar signal when a person is either sitting, standing, walking back and forth, or exercising in front of the radar, for a total of 72 minutes. The data were collected without a particular order, and the HR ranged from 60 to 160 beats per minute (bpm). This dataset is used to evaluate the phase signal construction algorithm and aims to show that the constructed signal can effectively encode the heartbeat signal. The second dataset contains three data segments, where each of them corresponds to one complete exercise cycle (as in Figure 7.4) and lasts 5 to 10 minutes. In each run, the person stood in front of the radar for around one minute, exercised for several minutes, and then stopped and rested for another one minute. One segment is used for training the SVM model described in Section 7.3.3, and the other two are used for testing it. The HR distribution of the two datasets is shown in Figure 7.12, both of them have a wide HR variability that ensures the generalizability of the system.

The evaluation includes two parts. The first part aims to show that, although the heartbeat frequency in the phase signal FFT spectrum can be hard to identify, the frequency always exists in the spectrum. It examines the effectiveness of the phase construction stage so that the heartbeat signal is embedded in the phase signal as expected. The second part evaluates the overall accuracy of the system when predicting the HR of a person during a complete exercise



(a) Heart rate distribution in dataset 1.

(b) Heart rate distribution in dataset 2.

Figure 7.12: HR distribution of the two datasets.

cycle.

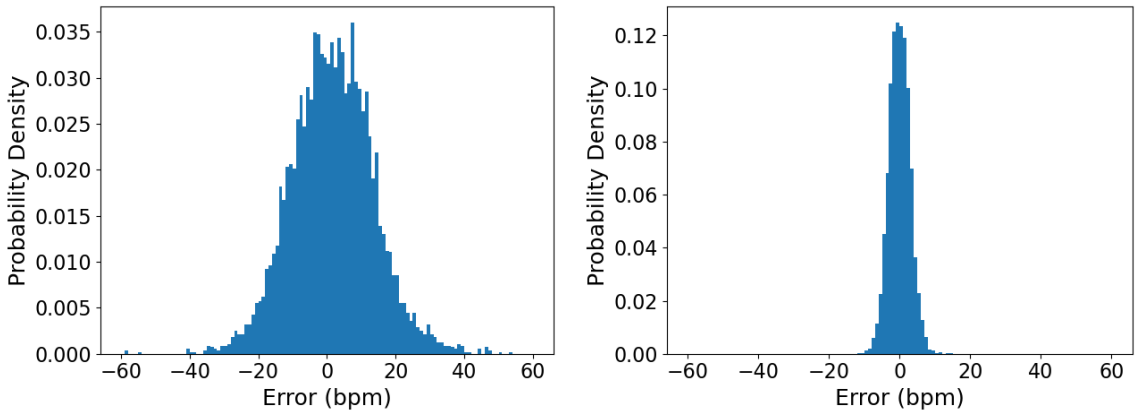
7.4.2 Phase Signal

The phase signal should ideally represent the chest displacement of the person. However, the chest displacement can become less significant and hard to identify when the person is moving. Therefore, it is important to make sure that the heartbeat signal can be captured by the phase signal constructed using the algorithm proposed in Section 7.3.1. Ideally, the phase signal should contain a frequency component that corresponds to the HR. This component should be present when an FFT is applied to the phase signal, although it may not be the only peak in the FFT spectrum due to the noise.

To verify this, the phase construction algorithm was applied to the first dataset to generate a phase signal every second, followed by a phase-FFT. The closest peak to the ground truth was found and the distance to the ground truth was recorded. The process was repeated using a 5 s and 20 s time window, respectively. The result is shown in Figure 7.13. It shows that there was always a peak around the ground truth regardless of the person's status, where the error can be up to ± 20 bpm when using a 5 s time window, and up to ± 5 bpm when using a 20 s time window. It can be seen that a larger time window helps the FFT to identify the HR signal, and the frequency corresponding to the HR is very likely to be present in the phase-FFT spectrum. Therefore, the only question left is how well the system can identify this frequency among the noise.

7.4.3 Heart Rate Estimation

The system is evaluated using the dataset containing complete exercise cycles. The result, as well as a comparison to the literature, is shown in Table 7.1 and Figure 7.14. The error is calculated as the mean absolute error between the estimation and the ground truth (measured once per second)



(a) Error with a 5 s time window.

(b) Error with a 20 s time window.

Figure 7.13: The distribution of the error between the ground truth and the nearest peak in the phase-FFT spectrum.

throughout the exercise cycles, which was (6.73 ± 5.10) bpm (5.9% in relative to the ground truth) when using a 5 s time window. The system is compared against research that focuses on the HR detection involving some degrees of body movement. The state-of-the-art systems can achieve a low error of below 5 bpm with various motions. However, one potential limitation of the systems is the variation range of the HR, which was not given in the mentioned work. When the motion is simple, like sitting or walking, the variation of the HR can be low, and the system may overfit to a certain range of HR.

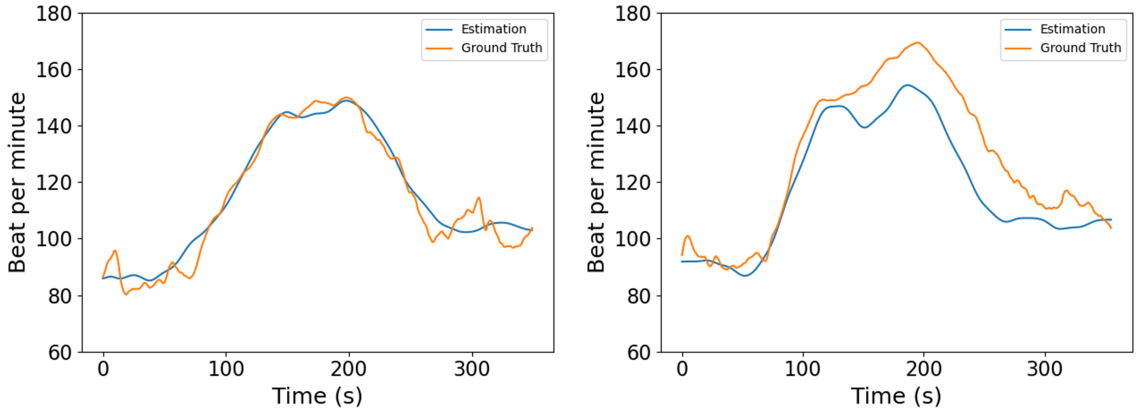
Although the proposed system does not outperform the state-of-the-art in the literature, it has several advantages. It is one of the few research studies that targets the HR detection of an exercising person and covers a wide range of possible HRs between 60 bpm to 180 bpm. The research in [127] is the most similar to this research. It used a neural network to estimate the HR of the person based on the motion level directly. However, the neural network requires much longer data acquisition and processing time (30 s and 5 s) and requires a high-end GPU to execute the network, so it has limitations in real-time applications. In addition, the research in [127] did not discuss the second and fourth stages of the exercise cycle, where the relationship between the HR and the motion level may fail. In contrast, the presented system uses a machine learning model to learn the trend of the HR changes and combines it with the traditional FFT-based method, to successfully track the HR during the entire exercise cycle. The presented system works in real-time and uses a 5 s time window to achieve a fast response.

7.5 Conclusion

In this chapter, a real-time HR detection system using a mmWave radar has been shown. The system is designed to monitor the HR during a complete exercise cycle, including the idling,

Table 7.1: Result of the proposed system and a comparison to the literature.

Method	Motion involved	Time window (s)	Average error rate
This research	One complete exercise cycle	5	6.73 bpm / 5.4%
Gong et al. [127]	Standing and exercising	30	5.57 bpm
Hu and Toda [131]	Walking at 0.25 m/s	5	3.66 bpm
Mercuri et al. [128]	Limb movements and desk work	20	3 bpm
Yang et al. [199]	Siting and moving at 47.6 mm/s	5	0.87%
Jang et al. [200]	Siting and moving at 53.4 mm/s	5	2.20%
Chen et al. [201]	Eight motions	20	3%



(a) First run.

(b) Second run.

Figure 7.14: HR estimation result using the proposed system.

exercise starting, continuous exercising and recovery phase. The system uses one mmWave radar to detect the chest displacement of the person and extract the HR signal. A phase construction algorithm is proposed to provide an accurate representation of the chest displacement under free body movement, and an HR tracking algorithm using an SVM is proposed to estimate the trend of the HR changes based on the person's motion level. The evaluation shows that, although the HR signal can be weak when compared with the motion signal, it can still be captured in the phase signal. When combined with the HR tracking algorithm, the system can provide an accurate estimation of the HR. The system can achieve a low error rate at 6.73 bpm when monitoring a person exercising on a treadmill. This research provides a non-intrusive, low-cost and real-time solution that helps people assess their health status without any professional medical equipment.

CHAPTER 8

CONCLUSION

In the past, the HAR field has been highly related to the computer vision field and has relied on vision data. But as HAR systems become more common and practical, the problem of privacy has become one of the main concerns of many people, which promotes the research of radar-based HAR that only collects anonymous data. mmWave radars, due to their much higher frequency and bandwidth than traditional radars, have demonstrated strong potential in HAR. In addition to the non-intrusive nature, the high bandwidth allows mmWave radars to capture the spatial features of the scene at a much higher resolution than traditional radar systems, and the short wavelength enables them to have a small antenna size and makes them easy to deploy. Along with the success in autonomous driving, researchers are starting to investigate the use of mmWave radars in HAR, which has also motivated this research.

This research presents a comprehensive study of mmWave radars in HAR. First, it gives a literature review of HAR and mmWave sensing techniques. HAR is inherently a machine learning problem that uses experiences from observed data to predict human activity in future scenarios. Therefore, most HAR systems adopt one or more machine learning algorithms to deal with various tasks. For example, supervised learning trains a mathematical model that maps the input to output using labelled data and can be used for classification or prediction, whereas unsupervised learning models the features within the data and can be used for data clustering. Machine learning forms an essential part of designing a HAR system and is used throughout this research.

This research uses the mmWave radar models from TI. The radars use the FMCW radar model that sends a chirp signal, receives its reflection and computes the IF signal, whose amplitude and phase can be used to determine the distance, velocity, and AoA of the objects in the scene. The performance of the range and velocity estimation relies on the chirp configuration, whereas

the AoA estimation relies on the number of antennas and the AoA estimation algorithm. The radars have on-chip processors that can be used for processing the radar signals, including the range-FFT, Doppler-FFT, angle-FFT and CFAR peak detection. A combination of these techniques forms a complete DPC that can construct a point cloud to represent objects in the scene. The radars also have high-speed interfaces that allow the raw IF signal to be captured through a data capturing card and transmitted to a PC for processing.

A mmWave radar simulation system is presented for efficient algorithm design and verification. The system allows a customized scene to be set up using 3D models from public datasets and simulates the radar signal as if the radar is placed in front of the scene. In this research, the FAUST dataset is used to simulate the radar signal when pointing toward a person in different postures, and a few DPCs and AoA estimation algorithms are evaluated. The algorithms provide a trade-off between a low computational cost and a higher resolution. For example, the basic DPC using a range-FFT, a Doppler-FFT and an angle-FFT is shown to be the most computationally efficient, whereas a better result can be achieved by using an advanced AoA estimation algorithm like MUSIC, but at an expense of a much higher processing time. The experiments show that the quality of the radar detection depends on several factors and tends to be more accurate when using a higher bandwidth and a higher number of chirps, as well as when detecting moving subjects. However, it is shown that even in an optimal setup, the quality of the radar detection can still be noisy and incomplete when compared with cameras or LIDARs, which can be the major challenge when using mmWave radars for higher-level applications.

The problem of the data quality is further verified when using the radars for human detection and tracking in a real-world scene. Locating the human is often considered the foundation for more complex HAR tasks, which would otherwise be impossible. When using a single mmWave radar to detect the presence of people in a room, the radar shows high sensitivity but also a high false alarm rate. To address the problem, a novel system using two radars from two perspectives is proposed, which uses information from both radars to verify each other's detection and produces a more robust solution. To implement the system, a software framework has been designed that can connect to and synchronize multiple radars at the same time. The software framework utilizes a multithreaded design to manage the data from the radars, ensures real-time performance and allows customized DPC to be implemented based on the use case. The resulting system improved the precision from 46.9% to 98.6% using two radars instead of one, while keeping the sensitivity at over 90%. The system achieved a low localization error of 0.56 cm when tracking a person's motion and outperformed many state-of-the-art RF-based systems.

Then, a novel human posture estimation system is presented using the two radars as a vertical array. The system identifies the key joint position of the person using a two-phase neural network model. Experiments show that the system is able to estimate an arbitrary posture of a person in an office environment at around two metres distance, with high accuracy at 71.3%. In contrast to much research that targets either standing or sitting postures, this research is the

first mmWave radar system that can detect a rich set of postures with a real-time processing time.

Finally, a human vitals sign detection system is presented that is able to detect a person's heart rate while exercising on a treadmill. In contrast to many existing systems that often require the subject to sit or lie at a known distance, the proposed system uses a novel phase signal construction algorithm that can accurately measure the chest displacement of the subject, and a machine learning model to predict the trends of the heart rate change based on the motion level of the subject. Experiments show that the system is able to estimate a subject's heart rate with a low error rate of 5.4%. Although the system does not outperform the state-of-the-art systems in HR detection, it is one of the few research studies that targets people with large body movement and can estimate HR in a wide range.

This research provides the fundamentals for developing HAR systems using mmWave radars. From the theoretical perspective, the thesis explains the principle of the mmWave radars, identifies their advantages and disadvantages and discusses the design choices that can affect the performance in real-world applications. From the practical perspective, a simulation system is presented for fast application design and verification before investigating the hardware, and a software framework is presented for managing the data communication of the radars and efficient system implementation.

This research shows that, although the data from mmWave radars is not as accurate as cameras and LIDARs, they can be used in several important fields in HAR with appropriate DPC and machine learning algorithms. The thesis has presented three HAR systems that have demonstrated competitive performances to traditional systems while having a real-time processing time. The human detection and tracking system provides an effective solution to locate a person in the field of view of the radar and can be used as the foundation for any other HAR applications. Once the subject has been located, the human posture estimation system helps the computer understand the underlying activity of the subject and potential provide interaction or assistance. Although the accuracy of the system is less than the state-of-the-art camera-based systems, it has the advantage of collecting only anonymous data and working in any lighting conditions. The vital sign detection system can be used to determine the HR of the subject in front of the radar, either standing, sitting, or exercising. The system is one of few research studies that addresses the challenge of HR detection under large body movement and targets a wide range of possible HRs. It helps the user to monitor their cardiovascular health conditions at home or work environment, without the requirement of professional medical equipment. Meanwhile, the proposed systems benefit from a low cost as the price of a radar chip can be below £10, which can be affordable to people from all income levels. It is believed that mmWave radars will have unique advantages and strong competitiveness in many industries that require HAR, including health care, elderly care, security, home monitoring and gaming.

8.1 Future Work

This research opens many possible directions for future work. The SRPC algorithm presented in Chapter 4 has demonstrated its effectiveness in improving the quality of the radar detection using simulation data, and it would be necessary to verify its effectiveness in real applications. For example, evaluating whether the posture estimation can perform better on the point cloud generated with SRPC. The evaluation would require a hardware-efficient implementation so that the real-time processing speed of the system can be retained.

Another research direction is porting the HAR systems to a single embedded platform. In this research, most of the data processing is performed on a PC using a modern CPU or/and GPU, which may not be available when targeting a low-cost HAR system. The radar models have on-chip DSP processors that can be programmed for higher-level applications. Since the effectiveness of the proposed HAR systems has been verified, the next step would be optimizing the algorithms and porting them to the DSP processors, so that the cost and power consumption of the final system would be kept low.

The framework and methodology presented in this research can also be used for similar practical problems in HAR. One such problem is human identification, which is important when there will be multiple people presenting in the environment and when the HAR information needs to be linked to the person's identity, such as in security and health care. The problem is more challenging as identification often requires high accuracy and reliable information about the person to be captured. Another possibility is gesture recognition. As mmWave radars provide high-resolution motion information of the person, it might be possible to extract a region of interest around the person's hand and recognize the underlying gesture, allowing the user to provide non-contact feedback to a computer system in certain applications. These research directions on mmWave radars can greatly encourage the development of human-computer interaction systems and contribute to a low-cost, non-intrusive and multifunctional HAR system.

BIBLIOGRAPHY

[1] National Health Service. (2021) Consultant-led referral to treatment waiting times data 2021-22. [Online]. Available: https://www.england.nhs.uk/statistics/statistical-work-areas/rtt-waiting-times/rtt-data-2021-22/

[2] T. D. Dobbs, J. A. Gibson, A. J. Fowler, T. E. Abbott, T. Shahid, F. Torabi, R. Griffiths, R. A. Lyons, R. M. Pearse, and I. S. Whitaker, "Surgical activity in england and wales during the covid-19 pandemic: a nationwide observational cohort study," *British Journal of Anaesthesia*, 2021.

[3] British Heart Foundation. (2021) Heart surgery and other heart procedures fall by 39%. [Online]. Available: https://www.bhf.org.uk/what-we-do/news-from-the-bhf/news-archive/2021/march/heart-surgery-procedures-fall-39-per-cent

[4] World Health Organization. (2021) Cardiovascular diseases (cvds). [Online]. Available: https://www.who.int/en/news-room/fact-sheets/detail/cardiovascular-diseases-(cvds)

[5] British Heart Foundation. (2021) Facts and figures - information for journalists | bhf. [Online]. Available: https://www.bhf.org.uk/what-we-do/news-from-the-bhf/contact-the-press-office/facts-and-figures

[6] U. R. Acharya, K. P. Joseph, N. Kannathal, C. M. Lim, and J. S. Suri, "Heart rate variability: a review," *Medical and biological engineering and computing*, vol. 44, no. 12, pp. 1031–1051, 2006.

[7] H. Ferdinando, L. Ye, T. Seppänen, and E. Alasaarela, "Emotion recognition by heart rate variability," *Australian Journal of Basic and Applied Science*, vol. 8, no. 14, pp. 50–55, 2014.

[8] A. Bauman, B. E. Ainsworth, J. F. Sallis, M. Hagströmer, C. L. Craig, F. C. Bull, M. Pratt, K. Venugopal, J. Chau, M. Sjöström *et al.*, "The descriptive epidemiology of sitting: a 20-country comparison using the international physical activity questionnaire (ipaq)," *American journal of preventive medicine*, vol. 41, no. 2, pp. 228–235, 2011.

[9] P. T. Katzmarzyk, T. S. Church, C. L. Craig, and C. Bouchard, "Sitting time and mortality from all causes, cardiovascular disease, and cancer," *Medicine & science in sports & exercise*, vol. 41, no. 5, pp. 998–1005, 2009.

[10] M. T. Hamilton, G. N. Healy, D. W. Dunstan, T. W. Zderic, and N. Owen, "Too little exercise and too much sitting: inactivity physiology and the need for new recommendations on sedentary behavior," *Current cardiovascular risk reports*, vol. 2, no. 4, pp. 292–298, 2008.

[11] J. P. Caneiro, P. O'Sullivan, A. Burnett, A. Barach, D. O'Neil, O. Tveit, and K. Olafsdottir, "The influence of different sitting postures on head/neck posture and muscle activity," *Manual Therapy*, vol. 15, no. 1, pp. 54–60, 2010.

[12] J. Michalak, J. Mischnat, and T. Teismann, "Sitting posture makes a difference—embodiment effects on depressive memory bias," *Clinical Psychology & Psychotherapy*, vol. 21, no. 6, pp. 519–524, 2014.

[13] J. Hackford, A. Mackey, and E. Broadbent, "The effects of walking posture on affective and physiological states during stress," *Journal of Behavior Therapy and Experimental Psychiatry*, vol. 62, pp. 80–87, 2019.

[14] A. Menon and M. Kumar, "Influence of body position on severity of obstructive sleep apnea: a systematic review," *International Scholarly Research Notices*, vol. 2013, 2013.

[15] G. Demiris and B. K. Hensel, "Technologies for an aging society: a systematic review of "smart home" applications," *Yearbook of medical informatics*, vol. 17, no. 01, pp. 33–40, 2008.

[16] A. Ahmad, J. C. Roh, D. Wang, and A. Dubey, "Vital signs monitoring of multiple people using a fmcw millimeter-wave sensor," in *2018 IEEE Radar Conference (RadarConf18)*. IEEE, 2018, pp. 1450–1455.

[17] F. Adib, H. Mao, Z. Kabelac, D. Katabi, and R. C. Miller, "Smart homes that monitor breathing and heart rate," in *Proceedings of the 33rd annual ACM conference on human factors in computing systems*, 2015, pp. 837–846.

[18] D. Bibbo, M. Carli, S. Conforto, and F. Battisti, "A sitting posture monitoring instrument to assess different levels of cognitive engagement," *Sensors*, vol. 19, no. 3, p. 455, 2019.

[19] T. Instrument. (2021) IWR1443 data sheet, product information and support. [Online]. Available: https://www.ti.com/product/IWR1443

[20] H. Cui and N. Dahnoun, "High precision human detection and tracking using millimeter-wave radars," *IEEE Aerospace and Electronic Systems Magazine*, vol. 36, no. 1, pp. 22–32, 2021.

[21] H. Cui and N. Dahnoun, "Real-time short-range human posture estimation using mmwave radars and neural networks," *IEEE Sensors Journal*, vol. 22, no. 1, pp. 535–543, 2022.

[22] J. Wu, H. Cui, and N. Dahnoun, "An improved angle estimation algorithm for millimeter-wave radar," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 2022, pp. 1–4.

[23] N. Dahnoun and H. Cui, "Radar detection and tracking," International Patent Published WO 2022/130 350 A1, Dec. 18, 2020. [Online]. Available: https://patentscope.wipo.int/search/en/detail.jsf?docId=WO2022130350

[24] J. Wu, H. Cui, and N. Dahnoun, "A novel high performance human detection, tracking and alarm system based on millimeter-wave radar," in *2021 10th Mediterranean Conference on Embedded Computing (MECO)*, 2021, pp. 1–4.

[25] H. Cui and N. Dahnoun, "Human posture capturing with millimetre wave radars," in *2020 9th Mediterranean Conference on Embedded Computing (MECO)*, 2020, pp. 1–4.

[26] J. Wu, H. Cui, and N. Dahnoun, "A voxelization algorithm for reconstructing mmwave radar point cloud and an application on posture classification," *Microprocessors and Microsystems*, 2022, under review.

[27] N. Dahnoun, H. Cui, and J. Wu, "Determining vital signs," U.K. Patent Filed GB2 203 223.9, Mar. 08, 2022. [Online]. Available: https://www.ipo.gov.uk/p-ipsum/Case/ApplicationNumber/GB2203223.9

[28] J. Wu, H. Cui, and N. Dahnoun, "A novel heart rate detection algorithm with small observing window using millimeter-wave radar," in *2022 11th Mediterranean Conference on Embedded Computing (MECO)*, 2022, pp. 1–4.

[29] J. Wu, H. Cui, and N. Dahnoun, "A health monitoring system with posture estimation and heart rate detection based on millimeter-wave radar," *Microprocessors and Microsystems*, 2022, under review.

[30] G. Bonaccorso, *Machine learning algorithms*. Packt Publishing Ltd, 2017.

[31] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *International journal of computer vision*, vol. 60, no. 2, pp. 91–110, 2004.

[32] H. Bay, T. Tuytelaars, and L. Van Gool, "Surf: Speeded up robust features," in *European conference on computer vision*. Springer, 2006, pp. 404–417.

[33] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern*

Recognition (CVPR'05) - Volume 1 - Volume 01, ser. CVPR '05. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893.

[34] P. Viola and M. Jones, "Robust real-time object detection," *International journal of computer vision*, vol. 4, no. 34-47, p. 4, 2001.

[35] D. Tran and A. Sorokin, "Human activity recognition with metric learning," in *European conference on computer vision*. Springer, 2008, pp. 548–561.

[36] M. Nixon and A. Aguado, *Feature extraction and image processing for computer vision*. Academic press, 2019.

[37] D. Kingma and J. Ba, "Adam: A method for stochastic optimization," *International Conference on Learning Representations*, Dec. 2014.

[38] J. Duchi, E. Hazan, and Y. Singer, "Adaptive subgradient methods for online learning and stochastic optimization," *Journal of Machine Learning Research*, vol. 12, no. Jul, pp. 2121–2159, 2011.

[39] C. Gambella, B. Ghaddar, and J. Naoum-Sawaya, "Optimization problems for machine learning: A survey," *European Journal of Operational Research*, vol. 290, no. 3, pp. 807–828, 2021.

[40] S. Ruder, "An overview of gradient descent optimization algorithms," *arXiv preprint arXiv:1609.04747*, 2016.

[41] Y. Tian and Y. Zhang, "A comprehensive survey on regularization strategies in machine learning," *Information Fusion*, vol. 80, pp. 146–166, 2022.

[42] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "Dbscan revisited, revisited: why and how you should (still) use dbscan," *ACM Transactions on Database Systems (TODS)*, vol. 42, no. 3, pp. 1–21, 2017.

[43] M. E. Celebi and K. Aydin, *Unsupervised learning algorithms*. Springer, 2016.

[44] A. Krizhevsky, I. Sutskever, and G. E. Hinton, "Imagenet classification with deep convolutional neural networks," in *Advances in neural information processing systems*, 2012, pp. 1097–1105.

[45] G. Cybenko, "Approximations by superpositions of a sigmoidal function," *Mathematics of Control, Signals and Systems*, vol. 2, pp. 183–192, 1989.

[46] K. Hornik, "Approximation capabilities of multilayer feedforward networks," *Neural networks*, vol. 4, no. 2, pp. 251–257, 1991.

[47] G. E. Hinton, S. Osindero, and Y.-W. Teh, "A fast learning algorithm for deep belief nets," *Neural computation*, vol. 18, no. 7, pp. 1527–1554, 2006.

[48] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," in *International Conference on Learning Representations*, 2015.

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

[50] G. Huang, Y. Sun, Z. Liu, D. Sedra, and K. Q. Weinberger, "Deep networks with stochastic depth," in *European conference on computer vision*. Springer, 2016, pp. 646–661.

[51] A. Bhardwaj, W. Di, and J. Wei, *Deep Learning Essentials: Your hands-on guide to the fundamentals of deep learning and neural network modeling*. Packt Publishing Ltd, 2018.

[52] O. I. Abiodun, A. Jantan, A. E. Omolara, K. V. Dada, N. A. Mohamed, and H. Arshad, "State-of-the-art in artificial neural network applications: A survey," *Heliyon*, vol. 4, no. 11, p. e00938, 2018.

[53] C. Nwankpa, W. Ijomah, A. Gachagan, and S. Marshall, "Activation functions: Comparison of trends in practice and research for deep learning," *arXiv preprint arXiv:1811.03378*, 2018.

[54] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 815–823.

[55] R. Poppe, "A survey on vision-based human action recognition," *Image and vision computing*, vol. 28, no. 6, pp. 976–990, 2010.

[56] D. T. Nguyen, W. Li, and P. O. Ogunbona, "Human detection from images and videos: A survey," *Pattern Recognition*, vol. 51, pp. 148 – 175, 2016.

[57] Y. Kong and Y. Fu, "Human action recognition and prediction: A survey," *arXiv preprint arXiv:1806.11230*, 2018.

[58] X. Chen and A. L. Yuille, "Parsing occluded people by flexible compositions," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2015, pp. 3945–3954.

[59] P. M. Dhulavvagol and N. C. Kundur, "Human action detection and recognition using sift and svm," in *International Conference on Cognitive Computing and Information Processing*. Springer, 2017, pp. 475–491.

[60] K. G. Manosha Chathuramali and R. Rodrigo, "Faster human activity recognition with svm," in *International Conference on Advances in ICT for Emerging Regions (ICTer2012)*, Dec. 2012, pp. 197–203.

[61] A. H. Nasution and S. Emmanuel, "Intelligent video surveillance for monitoring elderly in home environments," in *2007 IEEE 9th Workshop on Multimedia Signal Processing*. IEEE, 2007, pp. 203–206.

[62] E. Gumus, N. Kilic, A. Sertbas, and O. N. Ucan, "Evaluation of face recognition techniques using pca, wavelets and svm," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6404 – 6408, 2010.

[63] Y. Yang and D. Ramanan, "Articulated pose estimation with flexible mixtures-of-parts," in *CVPR 2011*. IEEE, 2011, pp. 1385–1392.

[64] A. Newell, K. Yang, and J. Deng, "Stacked hourglass networks for human pose estimation," in *European conference on computer vision*. Springer, 2016, pp. 483–499.

[65] H.-S. Fang, S. Xie, Y.-W. Tai, and C. Lu, "RMPE: Regional multi-person pose estimation," in *ICCV*, 2017.

[66] K. Sun, B. Xiao, D. Liu, and J. Wang, "Deep high-resolution representation learning for human pose estimation," in *CVPR*, 2019.

[67] H. Cui and N. Dahnoun, "Real-time stereo vision implementation on the texas instruments keystone ii soc," in *2016 IEEE International Conference on Imaging Systems and Techniques (IST)*, 2016, pp. 195–200.

[68] L. Xia, C.-C. Chen, and J. K. Aggarwal, "Human detection using depth information by kinect," in *CVPR 2011 workshops*. IEEE, 2011, pp. 15–22.

[69] J. K. Aggarwal and L. Xia, "Human activity recognition from 3d data: A review," *Pattern Recognition Letters*, vol. 48, pp. 70–80, 2014.

[70] F. Han, B. Reily, W. Hoff, and H. Zhang, "Space-time representation of people based on 3d skeletal data: A review," *Computer Vision and Image Understanding*, vol. 158, pp. 85–105, 2017.

[71] M. S. Seyfioğlu, A. M. Özbayoğlu, and S. Z. Gürbüz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 54, no. 4, pp. 1709–1723, 2018.

[72] S. Björklund, H. Petersson, A. Nezirovic, M. B. Guldogan, and F. Gustafsson, "Millimeter-wave radar micro-doppler signatures of human motion," in *2011 12th International Radar Symposium (IRS)*. IEEE, 2011, pp. 167–174.

[73] A.-K. Seifert, A. M. Zoubir, and M. G. Amin, "Radar classification of human gait abnormality based on sum-of-harmonics analysis," in *2018 IEEE Radar Conference (RadarConf18)*. IEEE, 2018, pp. 0940–0945.

[74] C. Li, V. M. Lubecke, O. Boric-Lubecke, and J. Lin, "A review on recent advances in doppler radar sensors for noncontact healthcare monitoring," *IEEE Transactions on microwave theory and techniques*, vol. 61, no. 5, pp. 2046–2060, 2013.

[75] W. Wang, A. X. Liu, M. Shahzad, K. Ling, and S. Lu, "Understanding and modeling of wifi signal based human activity recognition," in *Proceedings of the 21st annual international conference on mobile computing and networking*. ACM, 2015, pp. 65–76.

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

[77] Z. Yan, T. Duckett, and N. Bellotto, "Online learning for human classification in 3d lidar-based tracking," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017, pp. 864–871.

[78] Y. Qi, C. B. Soh, E. Gunawan, K.-S. Low, and R. Thomas, "Assessment of foot trajectory for human gait phase detection using wireless ultrasonic sensor network," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 24, no. 1, pp. 88–97, 2015.

[79] F. S. Butt, L. La Blunda, M. F. Wagner, J. Schäfer, I. Medina-Bulo, and D. Gómez-Ullate, "Fall detection from electrocardiogram (ecg) signals and classification by deep transfer learning," *Information*, vol. 12, no. 2, p. 63, 2021.

[80] N. Zouba, F. Bremond, and M. Thonnat, "Multisensor fusion for monitoring elderly activities at home," in *2009 Sixth IEEE International Conference on Advanced Video and Signal Based Surveillance*. IEEE, 2009, pp. 98–103.

[81] R. Gravina, P. Alinia, H. Ghasemzadeh, and G. Fortino, "Multi-sensor fusion in body sensor networks: State-of-the-art and research challenges," *Information Fusion*, vol. 35, pp. 68–80, 2017.

[82] B. L. R. Stojkoska and K. V. Trivodaliev, "A review of internet of things for smart home: Challenges and solutions," *Journal of Cleaner Production*, vol. 140, pp. 1454 – 1464, 2017.

[83] T. L. van Kasteren, G. Englebienne, and B. J. Kröse, "Human activity recognition from wireless sensor network data: Benchmark and software," in *Activity recognition in pervasive intelligent environments*. Springer, 2011, pp. 165–186.

[84] D. Singh, E. Merdivan, I. Psychoula, J. Kropf, S. Hanke, M. Geist, and A. Holzinger, "Human activity recognition using recurrent neural networks," in *International Cross-Domain Conference for Machine Learning and Knowledge Extraction*. Springer, 2017, pp. 267–274.

[85] A. M. Sabatini, C. Martelloni, S. Scapellato, and F. Cavallo, "Assessment of walking features from foot inertial sensing," *IEEE Transactions on biomedical engineering*, vol. 52, no. 3, pp. 486–494, 2005.

[86] K. Altun and B. Barshan, "Human activity recognition using inertial/magnetic sensor units," in *International workshop on human behavior understanding*. Springer, 2010, pp. 38–51.

[87] G. Wu and S. Xue, "Portable preimpact fall detector with inertial sensors," *IEEE Transactions on neural systems and Rehabilitation Engineering*, vol. 16, no. 2, pp. 178–183, 2008.

[88] F. Wu, H. Zhao, Y. Zhao, and H. Zhong, "Development of a wearable-sensor-based fall detection system," *International journal of telemedicine and applications*, vol. 2015, p. 2, 2015.

[89] E. Kantoch, "Technical verification of applying wearable physiological sensors in ubiquitous health monitoring," in *Computing in Cardiology 2013*. IEEE, 2013, pp. 269–272.

[90] U. Maurer, A. Smailagic, D. P. Siewiorek, and M. Deisher, "Activity recognition and monitoring using multiple sensors on different body positions," Carnegie-Mellon University, School of Computer Science, Tech. Rep., 2006.

[91] X. Su, H. Tong, and P. Ji, "Activity recognition with smartphone sensors," *Tsinghua science and technology*, vol. 19, no. 3, pp. 235–249, 2014.

[92] A. M. Khan, A. Tufail, A. M. Khattak, and T. H. Laine, "Activity recognition on smartphones via sensor-fusion and kda-based svms," *International Journal of Distributed Sensor Networks*, vol. 10, no. 5, p. 503291, 2014.

[93] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "A public domain dataset for human activity recognition using smartphones." in *ESANN*, 2013.

[94] D. Anguita, A. Ghio, L. Oneto, X. Parra, and J. L. Reyes-Ortiz, "Human activity recognition on smartphones using a multiclass hardware-friendly support vector machine," in *International workshop on ambient assisted living*. Springer, 2012, pp. 216–223.

[95] A. Reiss and D. Stricker, "Introducing a new benchmarked dataset for activity monitoring," in *2012 16th International Symposium on Wearable Computers*. IEEE, 2012, pp. 108–109.

[96] C. A. Ronao and S.-B. Cho, "Human activity recognition with smartphone sensors using deep learning neural networks," *Expert systems with applications*, vol. 59, pp. 235–244, 2016.

[97] J.-L. Reyes-Ortiz, L. Oneto, A. Samà, X. Parra, and D. Anguita, "Transition-aware human activity recognition using smartphones," *Neurocomputing*, vol. 171, pp. 754–767, 2016.

[98] N. Y. Hammerla, S. Halloran, and T. Plötz, "Deep, convolutional, and recurrent models for human activity recognition using wearables," in *Proceedings of the Twenty-Fifth International Joint Conference on Artificial Intelligence*, ser. IJCAI'16. AAAI Press, 2016, pp. 1533–1540.

[99] O. D. Lara and M. A. Labrador, "A survey on human activity recognition using wearable sensors," *IEEE communications surveys & tutorials*, vol. 15, no. 3, pp. 1192–1209, 2012.

[100] S. C. Mukhopadhyay, "Wearable sensors for human activity monitoring: A review," *IEEE Sensors Journal*, vol. 15, no. 3, pp. 1321–1330, Mar. 2015.

[101] O. Brdiczka, M. Langet, J. Maisonnasse, and J. L. Crowley, "Detecting human behavior models from multimodal observation in a smart home," *IEEE Transactions on Automation Science and Engineering*, vol. 6, no. 4, pp. 588–597, Oct. 2009.

[102] S. Gharghan, S. Mohammed, A. Al-Naji, M. Abu-AlShaeer, H. Jawad, A. Jawad, and J. Chahl, "Accurate fall detection and localization for elderly people based on neural network and energy-efficient wireless sensor network," *Energies*, vol. 11, no. 11, p. 2866, 2018.

[103] E. Fotiadis, M. Garzón, and A. Barrientos, "Human detection from a mobile robot using fusion of laser and vision information," *Sensors*, vol. 13, no. 9, pp. 11 603–11 635, 2013.

[104] W. Huang, Z. Zhang, W. Li, and J. Tian, "Moving object tracking based on millimeter-wave radar and vision sensor," *Journal of Applied Science and Engineering*, vol. 21, no. 4, pp. 609–614, 2018.

[105] M. Ulrich, F. Maile, A. Löcklin, B. Yang, B. Kleiner, and N. Ziegenspeck, "A model for improved association of radar and camera objects in an indoor environment," in *2017 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*. IEEE, 2017, pp. 1–6.

[106] C. Chen, R. Jafari, and N. Kehtarnavaz, "A survey of depth and inertial sensor fusion for human action recognition," *Multimedia Tools and Applications*, vol. 76, no. 3, pp. 4405–4425, Feb. 2017.

[107] M. A. Fischler and R. A. Elschlager, "The representation and matching of pictorial structures," *IEEE Transactions on computers*, vol. 100, no. 1, pp. 67–92, 1973.

[108] P. F. Felzenszwalb and D. P. Huttenlocher, "Pictorial structures for object recognition," *International journal of computer vision*, vol. 61, no. 1, pp. 55–79, 2005.

[109] Y. Tian, C. L. Zitnick, and S. G. Narasimhan, "Exploring the spatial hierarchy of mixture models for human pose estimation," in *European Conference on Computer Vision*. Springer, 2012, pp. 256–269.

[110] B. Sapp and B. Taskar, "Modec: Multimodal decomposable models for human pose estimation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 3674–3681.

[111] M. Andriluka, S. Roth, and B. Schiele, "Pictorial structures revisited: People detection and articulated pose estimation," in *2009 IEEE conference on computer vision and pattern recognition*. IEEE, 2009, pp. 1014–1021.

[112] A. Toshev and C. Szegedy, "Deeppose: Human pose estimation via deep neural networks," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2014, pp. 1653–1660.

[113] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler, "Joint training of a convolutional network and a graphical model for human pose estimation," in *Advances in neural information processing systems*, 2014, pp. 1799–1807.

[114] T. Xu, D. An, Y. Jia, and Y. Yue, "A review: Point cloud-based 3d human joints estimation," *Sensors*, vol. 21, no. 5, p. 1684, 2021.

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

[116] A. Sengupta, F. Jin, R. Zhang, and S. Cao, "mm-pose: Real-time human skeletal posture estimation using mmwave radars and cnns," *IEEE Sensors Journal*, 2020.

[117] G. Li, Z. Zhang, H. Yang, J. Pan, D. Chen, and J. Zhang, "Capturing human pose using mmwave radar," in *2020 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops)*, 2020, pp. 1–6.

[118] A. Sengupta, F. Jin, and S. Cao, "Nlp based skeletal pose estimation using mmwave radar point-cloud: A simulation approach," in *2020 IEEE Radar Conference (RadarConf20)*, 2020, pp. 1–6.

[119] Q. Wang, K. Wang, and W. Chen, "Clgnet: A new network for human pose estimation using commodity millimeter wave radar," in *2020 3rd International Conference on Algorithms,*

Computing and Artificial Intelligence, ser. ACAI 2020. New York, NY, USA: Association for Computing Machinery, 2020.

[120] A. Singh, S. U. Rehman, S. Yongchareon, and P. H. J. Chong, "Multi-resident non-contact vital sign monitoring using radar: A review," *IEEE Sensors Journal*, vol. 21, no. 4, pp. 4061–4084, 2021.

[121] E. Cardillo and A. Caddemi, "A review on biomedical mimo radars for vital sign detection and human localization," *Electronics*, vol. 9, no. 9, 2020.

[122] T. Hall, D. Y. Lie, T. Q. Nguyen, J. C. Mayeda, P. E. Lie, J. Lopez, and R. E. Banister, "Non-contact sensor for long-term continuous vital signs monitoring: A review on intelligent phased-array doppler sensor design," *Sensors*, vol. 17, no. 11, p. 2632, 2017.

[123] M. Alizadeh, G. Shaker, J. C. M. De Almeida, P. P. Morita, and S. Safavi-Naeini, "Remote monitoring of human vital signs using mm-wave fmcw radar," *IEEE Access*, vol. 7, pp. 54 958–54 968, 2019.

[124] Z. Yang, P. H. Pathak, Y. Zeng, X. Liran, and P. Mohapatra, "Vital sign and sleep monitoring using millimeter wave," *ACM Transactions on Sensor Networks (TOSN)*, vol. 13, no. 2, pp. 1–32, 2017.

[125] I. V. Mikhelson, S. Bakhtiari, T. W. Elmer, A. V. Sahakian *et al.*, "Remote sensing of heart rate and patterns of respiration on a stationary subject using 94-ghz millimeter-wave interferometry," *IEEE Transactions on Biomedical Engineering*, vol. 58, no. 6, pp. 1671–1677, 2011.

[126] S. Wang, A. Pohl, T. Jaeschke, M. Czaplik, M. Köny, S. Leonhardt, and N. Pohl, "A novel ultra-wideband 80 ghz fmcw radar system for contactless monitoring of vital signs," in *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*. IEEE, 2015, pp. 4978–4981.

[127] J. Gong, X. Zhang, K. Lin, J. Ren, Y. Zhang, and W. Qiu, "Rf vital sign sensing under free body movement," *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.*, vol. 5, no. 3, Sep. 2021.

[128] M. Mercuri, I. R. Lorato, Y.-H. Liu, F. Wieringa, C. Van Hoof, and T. Torfs, "Vital-sign monitoring and spatial tracking of multiple people using a contactless radar-based sensor," *Nature Electronics*, vol. 2, no. 6, pp. 252–262, 2019.

[129] L. Liu, S. Zhang, and W. Xiao, "Non-contact vital signs detection using mm-wave radar during random body movements," in *2021 IEEE 16th Conference on Industrial Electronics and Applications (ICIEA)*, 2021, pp. 1244–1249.

[130] I. V. Mikhelson, S. Bakhtiari, T. W. Elmer, and A. V. Sahakian, "Remote sensing of patterns of cardiac activity on an ambulatory subject using millimeter-wave interferometry and statistical methods," *Medical & biological engineering & computing*, vol. 51, no. 1, pp. 135–142, 2013.

[131] Y. Hu and T. Toda, "A novel adaptive range-bin selection method for remote heart-rate measurement of an indoor moving person using mm-wave fmcw radar," *IEICE Communications Express*, vol. 10, no. 5, pp. 277–282, 2021.

[132] C. Gu, G. Wang, Y. Li, T. Inoue, and C. Li, "A hybrid radar-camera sensing system with phase compensation for random body movement cancellation in doppler vital sign detection," *IEEE Transactions on Microwave Theory and Techniques*, vol. 61, no. 12, pp. 4678–4688, 2013.

[133] K. Konishi and T. Sakamoto, "Automatic tracking of human body using millimeter-wave adaptive array radar for noncontact heart rate measurement," in *2018 Asia-Pacific Microwave Conference (APMC)*, 2018, pp. 836–838.

[134] I. V. Mikhelson, P. Lee, S. Bakhtiari, T. W. Elmer, A. K. Katsaggelos, and A. V. Sahakian, "Noncontact millimeter-wave real-time detection and tracking of heart rate on an ambulatory subject," *IEEE Transactions on Information Technology in Biomedicine*, vol. 16, no. 5, pp. 927–934, 2012.

[135] F.-K. Wang, T.-S. Horng, K.-C. Peng, J.-K. Jau, J.-Y. Li, and C.-C. Chen, "Single-antenna doppler radars using self and mutual injection locking for vital sign detection with random body movement cancellation," *IEEE Transactions on Microwave Theory and Techniques*, vol. 59, no. 12, pp. 3577–3587, 2011.

[136] K. Ramasubramanian and K. Ramaiah, "Moving from legacy 24 ghz to state-of-the-art 77-ghz radar," *ATZelektronik worldwide*, vol. 13, no. 3, pp. 46–49, 2018.

[137] Federal Communications Commission, "Permitting radar services in the 76-81 ghz band," 2017. [Online]. Available: https://www.federalregister.gov/documents/2017/09/20/2017-18463/permitting-radar-services-in-the-76-81-ghz-band

[138] Federal Communications Commission, "Equipment authorization guidance for 76-81 GHz radar devices," 2019.

[139] European Telecommunications Standards Institute, "Short Range Devices (SRD); Radio equipment to be used in the 40 GHz to 246 GHz frequency range; Harmonised Standard for access to radio spectrum." 2017.

[140] A. G. Stove, "Linear fmcw radar techniques," *IEE Proceedings F (Radar and Signal Processing)*, vol. 139, no. 5, pp. 343–350, 1992.

[141] T. Liu, Y. Zhao, Y. Wei, Y. Zhao, and S. Wei, "Concealed object detection for activate millimeter wave image," *IEEE Transactions on Industrial Electronics*, vol. 66, no. 12, pp. 9909–9917, 2019.

[142] L. Wang and K. You, "3d holographic millimeter-wave imaging for concealed metallic forging objects detection," in *Emerging Microwave Technologies in Industrial, Agricultural, Medical and Food Processing*. Univ. of Technology Malaysia, 2018, p. 125.

[143] T. Gu, Z. Fang, Z. Yang, P. Hu, and P. Mohapatra, "Mmsense: Multi-person detection and identification via mmwave sensing," in *Proceedings of the 3rd ACM Workshop on Millimeter-Wave Networks and Sensing Systems*, ser. mmNets'19. New York, NY, USA: Association for Computing Machinery, 2019, p. 45–50.

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

[145] K. Bengler, K. Dietmayer, B. Farber, M. Maurer, C. Stiller, and H. Winner, "Three decades of driver assistance systems: Review and future perspectives," *IEEE Intelligent Transportation Systems Magazine*, vol. 6, no. 4, pp. 6–22, 2014.

[146] M. Z. Ikram and M. Ali, "3-d object tracking in millimeter-wave radar for advanced driver assistance systems," in *2013 IEEE Global Conference on Signal and Information Processing*. IEEE, 2013, pp. 723–726.

[147] Texas Instruments, "Traffic monitoring object detection and tracking reference design using single-chip mmwave radar sensor," 2017.

[148] S. Clark and H. Durrant-Whyte, "Autonomous land vehicle navigation using millimeter wave radar," in *Proceedings. 1998 IEEE International Conference on Robotics and Automation (Cat. No.98CH36146)*, vol. 4, May 1998, pp. 3697–3702 vol.4.

[149] K. Dutta and S. Chakraborty, "Obstacle avoidance robot using mmwave," *International Journal of Scientific and Engineering Research*, vol. 07, pp. 0196–0198, Feb. 2018.

[150] R. Streubel and B. Yang, "Fusion of stereo camera and mimo-fmcw radar for pedestrian tracking in indoor environments," in *2016 19th International Conference on Information Fusion (FUSION)*. IEEE, 2016, pp. 565–572.

[151] J.-H. Kim, J. W. Starr, and B. Y. Lattimer, "Firefighting robot stereo infrared vision and radar sensor fusion for imaging through smoke," *Fire Technology*, vol. 51, no. 4, pp. 823–845, 2015.

[152] S. Kianoush, S. Savazzi, and V. Rampa, "Passive detection and discrimination of body movements in the sub-thz band: A case study," in *2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 1597–1601.

[153] J. Lien, N. Gillian, M. E. Karagozler, P. Amihood, C. Schwesig, E. Olson, H. Raja, and I. Poupyrev, "Soli: Ubiquitous gesture sensing with millimeter wave radar," *ACM Transactions on Graphics (TOG)*, vol. 35, no. 4, p. 142, 2016.

[154] A. D. Singh, S. S. Sandha, L. Garcia, and M. Srivastava, "Radhar: Human activity recognition from point clouds generated through a millimeter-wave radar," in *Proceedings of the 3rd ACM Workshop on Millimeter-wave Networks and Sensing Systems*. ACM, 2019, pp. 51–56.

[155] R. Zhang and S. Cao, "Real-time human motion behavior detection via cnn using mmwave radar," *IEEE Sensors Letters*, vol. 3, no. 2, pp. 1–4, 2018.

[156] P. Zhao, C. X. Lu, J. Wang, C. Chen, W. Wang, N. Trigoni, and A. Markham, "mid: Tracking and identifying people with millimeter wave radar," in *2019 15th International Conference on Distributed Computing in Sensor Systems (DCOSS)*. IEEE, 2019, pp. 33–40.

[157] M. A. Richards, J. Scheer, W. A. Holm, and W. L. Melvin, *Principles of modern radar*. Citeseer, 2010, vol. 1.

[158] K. Ramasubramanian, "Using a complex-baseband architecture in fmcw radar systems," Texas Instruments, Tech. Rep., 2017.

[159] D. K. Barton, "Modern radar system analysis," *Norwood*, 1988.

[160] Texas Instrument, "Iwr1443 single-chip 76- to 81-ghz mmwave sensor," 2017. [Online]. Available: http://www.ti.com/lit/ds/symlink/iwr1443.pdf

[161] Z. Chen, G. Gokeda, and Y. Yu, *Introduction to Direction-of-arrival Estimation*. Artech House, 2010.

[162] J. Capon, "High-resolution frequency-wavenumber spectrum analysis," *Proceedings of the IEEE*, vol. 57, no. 8, pp. 1408–1418, 1969.

[163] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Transactions on Antennas and Propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[164] S. Rao, "MIMO radar," Texas Instruments, Tech. Rep., 2017.

[165] Texas Instrument, "IWR1443BOOST evaluation module mmWave sensing solution," 2018. [Online]. Available: https://www.ti.com/lit/ug/swru518d/swru518d.pdf

[166] Texas Instrument, "IWR1642 Evaluation Module (IWR1642BOOST) Single-Chip mmWave Sensing Solution," 2017. [Online]. Available: https://www.ti.com/lit/ug/swru521c/swru521c.pdf

[167] Texas Instrument, “xWR1843 Evaluation Module (xWR1843BOOST) Single-Chip mmWave Sensing Solution,” 2018. [Online]. Available: https://www.ti.com/lit/ug/spruim4b/spruim4b.pdf

[168] Texas Instrument, “60ghz mmwave sensor evms,” 2018. [Online]. Available: https://www.ti.com/lit/ug/swru546e/swru546e.pdf

[169] Texas Instrument, “IWR1843 Single-Chip 76- to 81-GHz FMCW mmWave Sensor,” 2019. [Online]. Available: https://www.ti.com/lit/ds/symlink/iwr1843.pdf

[170] A. Farina and F. A. Studer, “A review of cfar detection techniques in radar systems,” *Microwave Journal*, vol. 29, p. 115, 1986.

[171] T. Instruments, “mmWave SDK user guide,” 2017.

[172] Texas Instrument, “TSW140x High Speed Data Capture/Pattern Generator Card,” 2018.

[173] Texas Instrument, “DCA1000EVM Data Capture Card,” 2018.

[174] Texas Instrument, “Mmwave Radar Device ADC Raw Data Capture,” 2017. [Online]. Available: https://www.ti.com/lit/an/swra581b/swra581b.pdf

[175] M. Dudek, D. Kissinger, R. Weigel, and G. Fischer, “A millimeter-wave fmcw radar system simulator for automotive applications including nonlinear component models,” in *2011 8th European Radar Conference*. IEEE, 2011, pp. 89–92.

[176] C. Stetco, B. Ubezio, S. Mühlbacher-Karrer, and H. Zangl, “Radar sensors in collaborative robotics: Fast simulation and experimental validation,” in *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2020, pp. 10 452–10 458.

[177] C. Schöffmann, B. Ubezio, C. Böhm, S. Mühlbacher-Karrer, and H. Zangl, “Virtual radar: Real-time millimeter-wave radar sensor simulation for perception-driven robotics,” *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4704–4711, 2021.

[178] J. Rissanen, “Modeling by shortest data description,” *Automatica*, vol. 14, no. 5, pp. 465–471, 1978.

[179] F. Bogo, J. Romero, M. Loper, and M. J. Black, “FAUST: Dataset and evaluation for 3D mesh registration,” in *Proceedings IEEE Conf. on Computer Vision and Pattern Recognition (CVPR)*. Piscataway, NJ, USA: IEEE, Jun. 2014.

[180] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: Unified, real-time object detection,” in *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun. 2016.

[181] W. Ruan, Q. Z. Sheng, L. Yao, X. Li, N. J. Falkner, and L. Yang, "Device-free human localization and tracking with uhf passive rfid tags: A data-driven approach," *Journal of Network and Computer Applications*, vol. 104, pp. 78–96, 2018.

[182] K. Qian, C. Wu, Z. Yang, Y. Liu, and K. 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*, 2017, pp. 1–10.

[183] X. Li, D. Zhang, Q. Lv, J. Xiong, S. Li, Y. Zhang, and H. Mei, "Indotrack: Device-free indoor human tracking with commodity wi-fi," *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, vol. 1, no. 3, pp. 1–22, 2017.

[184] V.-H. Nguyen and J.-Y. Pyun, "Location detection and tracking of moving targets by a 2d ir-uwb radar system," *sensors*, vol. 15, no. 3, pp. 6740–6762, 2015.

[185] C. Will, P. Vaishnav, A. Chakraborty, and A. Santra, "Human target detection, tracking, and classification using 24-ghz fmcw radar," *IEEE Sensors Journal*, vol. 19, no. 17, pp. 7283–7299, 2019.

[186] C. Wu, F. Zhang, B. Wang, and K. J. Ray Liu, "mmtrack: Passive multi-person localization using commodity millimeter wave radio," in *IEEE INFOCOM 2020 - IEEE Conference on Computer Communications*, 2020, pp. 2400–2409.

[187] N. Dahnoun and H. Cui, "Radar detection and tracking," U.K. Patent GB2 020 193.5, Dec. 18, 2020. [Online]. Available: https://www.ipo.gov.uk/p-ipsum/Case/ApplicationNumber/GB2020193.5

[188] B. Xiao, H. Wu, and Y. Wei, "Simple baselines for human pose estimation and tracking," in *Proceedings of the European conference on computer vision (ECCV)*, 2018, pp. 466–481.

[189] J. Huang, Z. Zhu, F. Guo, and G. Huang, "The devil is in the details: Delving into unbiased data processing for human pose estimation," in *The IEEE / CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2020.

[190] N. Dahnoun, *Digital signal processing implementation using the TMS320C6000 DSP platform*. Addison-Wesley Longman Publishing Co., Inc., 2000.

[191] N. Dahnoun, *Multicore DSP: from algorithms to real-time implementation on the TMS320C66x SoC*. John Wiley & Sons, 2018.

[192] Texas Instruments, "Deep learning inference for embedded applications reference design," 2018.

[193] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich, "Going deeper with convolutions," in *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015, pp. 1–9.

[194] C. Szegedy, S. Ioffe, V. Vanhoucke, and A. A. Alemi, "Inception-v4, Inception-ResNet and the impact of residual connections on learning," in *31st AAAI Conference on Artificial Intelligence, AAAI 2017*, 2017, pp. 4278–4284.

[195] G. Shafiq and K. C. Veluvolu, "Surface chest motion decomposition for cardiovascular monitoring," *Scientific reports*, vol. 4, no. 1, pp. 1–9, 2014.

[196] Z. Yang, P. H. Pathak, Y. Zeng, X. Liran, and P. Mohapatra, "Monitoring vital signs using millimeter wave," in *Proceedings of the 17th ACM international symposium on mobile ad hoc networking and computing*, 2016, pp. 211–220.

[197] Texas Instrument, "DCA1000EVM CLI Software User Guide," 2019.

[198] R. Gilgen-Ammann, T. Schweizer, and T. Wyss, "Rr interval signal quality of a heart rate monitor and an ecg holter at rest and during exercise," *European journal of applied physiology*, vol. 119, no. 7, pp. 1525–1532, 2019.

[199] Z.-K. Yang, H. Shi, S. Zhao, and X.-D. Huang, "Vital sign detection during large-scale and fast body movements based on an adaptive noise cancellation algorithm using a single doppler radar sensor," *Sensors*, vol. 20, no. 15, p. 4183, 2020.

[200] A.-J. Jang, I.-S. Lee, and J.-R. Yang, "Vital signal detection using multi-radar for reductions in body movement effects," *Sensors*, vol. 21, no. 21, p. 7398, 2021.

[201] Z. Chen, T. Zheng, C. Cai, and J. Luo, "Movi-fi: Motion-robust vital signs waveform recovery via deep interpreted rf sensing," in *Proceedings of the 27th Annual International Conference on Mobile Computing and Networking*, 2021, pp. 392–405.