

FACHHOCHSCHULE DORTMUND
University of Applied Science and Arts
Department of Information Technology

**DETECTION AND CLASSIFICATION OF
MOVING OBJECTS WITH A 24 GHZ FAST
CHIRP RADAR FOR PEOPLE COUNTING
APPLICATIONS**

MASTER THESIS

Submitted by:

XXX

Matriculation number: XXX

in fulfillment of the requirements for the degree of
Master Embedded Systems for Mechatronics

Supervised by

Prof. Dr. Andreas Becker, Fachhochschule Dortmund

Daniel Bonney, Fachhochschule Dortmund

Dortmund Feb 20, 2020

ABSTRACT

With the technology advancements, radar sensors have become cheaper and smaller. They have taken an important role in our automated life. In many situations it is very important to keep track of the people count.

The aim of this thesis is to use the selected 24GHz Infineon FMCW radar to recognize human walking pattern and find a way to distinguish walking human from other objects. Several tests were designed and conducted; the raw radar images are then recorded. These raw radar data are first processed to reduce noise clutter so that target detection is optimal. Then the targets are tracked using Kalman filter to extract the pattern sequences.

To further investigate the dynamic behavior of the walking pattern. The micro-Doppler effect of the walking pattern is studied. A simplified model of walking pattern is presented. Several features of the pattern are proposed and are extracted from the recordings of different test scenarios using a two pass processing technique. Then the results from the different tests are compared.

At last, a radar recording and tracking system is implemented to test the proposed approach for the walking pattern recognition.

CONTENTS

1	Introduction.....	1
	1.1 Motivation.....	2
	1.2 Problem Statement.....	2
	1.3 Research Method	3
2	Related work.....	4
	2.1 Infineon 24 GHz FMCW Radar kit.....	4
	2.2 Current Research.....	7
	2.3 Measurement Cases	8
3	Signal Preprocessing.....	13
	3.1 Clutter Removal.....	14
	3.2 Weak Signal Enhancement.....	19
	3.3 Signal Detection.....	22
	3.3.1 Angle Pre-separation.....	22
	3.3.2 Local maxima detector.....	25
	3.3.3 CA-CFAR detector.....	26
4	Target Tracking	30
	4.1 Kalman Filter	30
	4.2 Single target tracker	35
	4.3 Multi-target tracker	37
	4.3.1 Hungarian algorithm.....	39
	4.3.2 Multi-track management.....	43
	4.4 Tracking Results	44
5	Walking Pattern Recognition	49
	5.1 Pattern Definition and Extraction	51
	5.2 Feature Definition and Extraction.....	55
	5.3 Classification.....	65
6	Software System Implementation	72
	6.1 Radar data recording tool.....	72
	6.2 Live tracking tool.....	76

7	Conclusion and Future Work	79
8	References.....	80

List of Figures

Figure 1: Infinion 24 GHz automotive radar kit.....	5
Figure 2: Infinion 24 GHz automotive radar kit block diagram [3]	5
Figure 3: FMCW fast chirp scheme [4].....	6
Figure 4: inside a tent (left), outside FH-Dortmund main building (right).....	8
Figure 5: movement setup of test case 1.....	9
Figure 6: movement setup of test case 2.....	10
Figure 7: movement setup of test case 3.....	12
Figure 8: signal pre-processing blocks	13
Figure 9: noise image (left), noise image with stationary clutter cleared (right).....	14
Figure 10: mean Doppler bin SNR plot.....	15
Figure 11: test frame nr, d (left), max Doppler bin SNR (right).....	16
Figure 12: environment noises in Cartesian system	17
Figure 13: range-t and Doppler-t of recording [three].....	19
Figure 14: S_r of Figure 11 test frame	20
Figure 15: S_{maxr} of test 1.1, with different k	21
Figure 16: range-t and Doppler-t of test 1.1 with time gain $k=2$	21
Figure 17: angle distribution of test 2.2 (left), angle distribution peaks (right).....	22
Figure 18: angle distribution of one frame	23
Figure 19: angle distribution of test 3.....	24
Figure 20: CA-CFAR cell arrangement.....	27
Figure 21: Target detection of frame 150	28
Figure 22: Kalman filter work flow.....	31
Figure 23: coordinates overview	32
Figure 24: Single target tracker work flow.....	35
Figure 25: outliner C and D.....	36
Figure 26: track state diagram	37
Figure 27: Hungarian algorithm flow chart.....	40
Figure 28: Single target tracker work flow.....	43
Figure 29: movement trajectory of test 1.1.....	45
Figure 30: movement trajectory of test 2.1.....	46
Figure 31: movement trajectory of test 2.2.....	46
Figure 32: movement trajectory of test 2.3.....	47
Figure 33: movement trajectory of test 2.4.....	47
Figure 34: signal processing blocks overview.....	49
Figure 35: walking patterns	51

Figure 36: typical walking frames at different distances	52
Figure 37: pattern shape measurement of test 1.1	53
Figure 38: Walking pattern sequences	54
Figure 39: range-t and Doppler-t of the extracted pattern from test 1.1	54
Figure 40: Proposed feature set on range-Doppler frame	55
Figure 41: Doppler-t plot of walking.....	56
Figure 42: Proposed feature set on Doppler-t plot (positive velocity).....	57
Figure 43: peak extraction overview	59
Figure 44: Extreme point analysis of Figure 41	62
Figure 45: histogram of the three cycle length.....	62
Figure 46: NDS and PDS of test 1.1.....	63
Figure 47: mean radial Doppler profile (left), acceleration (right) of test 1.1	64
Figure 48: range-t and Doppler-t of test 1.2	66
Figure 49: range-t and Doppler-t of test 1.3	66
Figure 50: range-t and Doppler-t of test 1.4	67
Figure 51: range-t and Doppler-t of test 1.5	67
Figure 52: range-t and Doppler-t of test 1.10	68
Figure 53: mean Doppler profile and PDS of test 1.2-1.5	68
Figure 54: mean Doppler profile and PDS of test 1.2, 1.6, 1.7	69
Figure 55: range-t and Doppler-t of test 1.8	70
Figure 56: range-t and Doppler-t of test 1.9	70
Figure 57: mean Doppler profile and PDS of test 1.2, 1.8, 1.9	71
Figure 58: Radar recording/tracking setup	72
Figure 59: radar_viewer flow chart	73
Figure 60: radar_viewer state diagram	74
Figure 61: radar_viewer UI	75
Figure 62: radar_tracker, in single target tracking mode	77
Figure 63: radar_tracker, in multi-target tracking mode.....	78

List of Tables

Table 1: resolutions of Infineon radar kit.....	6
Table 2: specifications of Infineon radar kit.....	6
Table 3: Maxima count for test 2.2.....	22
Table 4: mean value of three cycle length of test 1.1-1.10.....	64

1 Introduction

Radar sensors have many advantages over traditional image sensors. They are robust, able to function in relative harsh environment. They are widely used in automotive industry, avionic industry military field and others. Another major benefit of radar sensors is privacy protection. Unlike traditional image sensor, radar image produced by current radar sensors usually doesn't contain enough details to identify individuals. Therefore they are the better candidate to work in privacy sensitive location where data protection is a major concern. Nowadays they can even fit inside mobile phones to provide gesture recognition [1].

In radar principle, moving targets have a frequency shift in the radar return signal due to Doppler Effect. For targets with rigid body in ideal cases, the movement pattern of different part of the target is similar; therefore the frequency shift of all parts is usually unanimous. Thus they often have a very narrow Doppler profile in radar range-Doppler frame, e.g. vehicles. However, most targets have many different movable parts, and each part has its own movement pattern relative to the radar, which means each part has its own Doppler profile. The variation of Doppler profile often results in weaker patterns around the main target detection in radar frames. This effect is named Micro-Doppler Effect [2].

In recent years, radar sensors have received a great amount of development. With the development of advanced manufacturing techniques, radar modules are becoming smaller and cheaper yet with higher performance. The resolution of radar sensor has increased greatly. The Micro-Doppler Effect now can be easily observed in radar frames. Since they are the direct result of object movement, with proper signal processing technique, the information regarding the target movement pattern can be obtained. Then further target classification is possible.

In this thesis, the Infineon 24 GHz radar kit is used to capture the micro-Doppler pattern of human walking. The pattern is then analyzed so that correct detection of human walking can be achieved for people counting applications.

1.1 Motivation

People counting can be especially useful and crucial in many applications. For example in big shopping malls, keeping track of the number of people in the building is critical for management purposes or in the case of emergency. When in car accidents, the number of people in vicinity is vital information for the ADAS system to react correctly and safely.

Since radar only detects moving objects, the walking pattern needs to be modeled so that the detection can be stable. Current researches often focus on the detection of the main target and ignore the Micro-Doppler pattern. Therefore a proper study of the Micro-Doppler pattern is important for recognizing human walking detections.

1.2 Problem Statement

The purpose of this thesis is to develop a system to detect and distinguish walking human from non-human objects. So that such information can be used in people counting applications. Since the Micro-Doppler pattern contains rich information regarding the movement pattern of the target, an approach to extract this information should be established.

To achieve this, the target signal first needs to be detected from the raw data samples, a proper method to reduce noise and detect target needs to be found. Then an approach to track the target needs be established, so that a continuous study of the interested target is possible. Afterwards the model of the walking pattern needs to be established and simplified so that the features of the walking patterns can be correctly extracted.

1.3 Research Method

The research method in this paper is primarily literature review and experiments. At first the existing solution of the encountered problems are researched, then the measurement cases are designed, and the radar raw frames are collected. Afterwards these data are analyzed in Matlab to test the signal processing algorithms found in current research. The results in the end validate the theory obtained in literature review. If the result is not up to satisfactory, modifications are made to the measurement cases and the corresponding signal processing steps, and then the process restart again. This makes sure that the theory and the implementation matched together to works on the Infineon radar platform.

2 Related work

This thesis is a continuation of a previous research work on the same Infineon 24GHz radar kit [4], in which the communication protocol between the radar and computer is explored. This chapter presents an overview of the Infineon automotive radar kit and some current research on detection, tracking and classification technique. Afterwards the measurement cases is described, where the participating people are moving in a specified way and corresponding radar frames are collected.

2.1 Infineon 24 GHz FMCW Radar kit

Electromagnetic (EM) waves can be absorbed and reflect when in contact with the surface of objects; the reflected signal always contains the range, velocity and angle information of the object due to the Doppler Effect. Radar is the sensor that sends a predefined EM wave and monitors the return wave to obtain the range, velocity and angle information. By the continuity of the transmitted EM wave, radar sensor can be classified into Continuous Wave (CW) radar and Pulsed radar. The CW radar can also be divided into unmodulated radar and modulated radar (FMCW), depending on modulation of the transmitted EM wave (modulation scheme). Compared with the unmodulated radar, The FMCW radar can detect range information while unmodulated radar can only detect velocity information. Angle detection can be achieved by implementing more than one received antennas. The angle information can be calculated from the time delay of the signal in different antennas [5].

The selected Infineon 24 GHz Automotive radar kit is a programmable platform to implement radar applications. Figure 1 shows the picture of the setup used in this thesis. Figure 2 is the block diagram of the Infineon 24 GHz automotive radar kit [3]. The main microcontroller TC264DA first modulates the radio signal and sends it to an external DAC DAC7552. Then the modulated analog signal is sent to the transceiver BGT24ATR12 and transmitted through a transmitter antenna. The same transceiver also receives the returned radio signal from two receiver antenna and sends them through LPF

(Low Pass Filter) and LNA (Low Noise Amplifier). Then the TC264DA samples the analog return signal to obtain the raw radar samples.

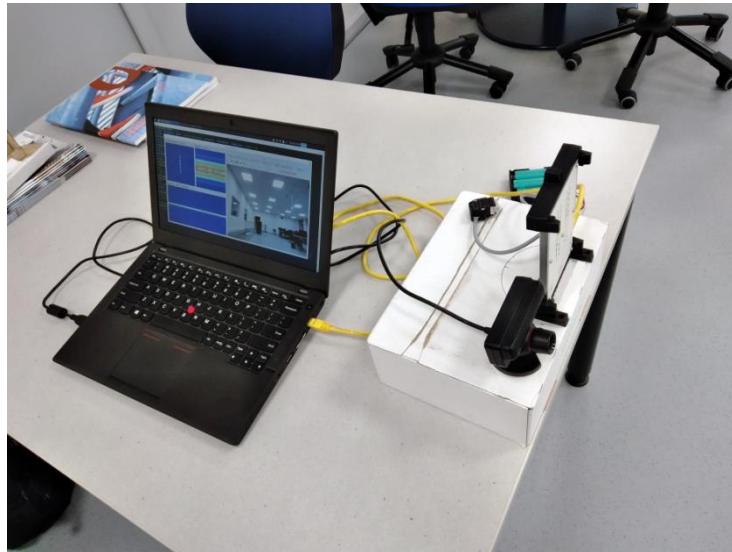


Figure 1: Infineon 24 GHz automotive radar kit

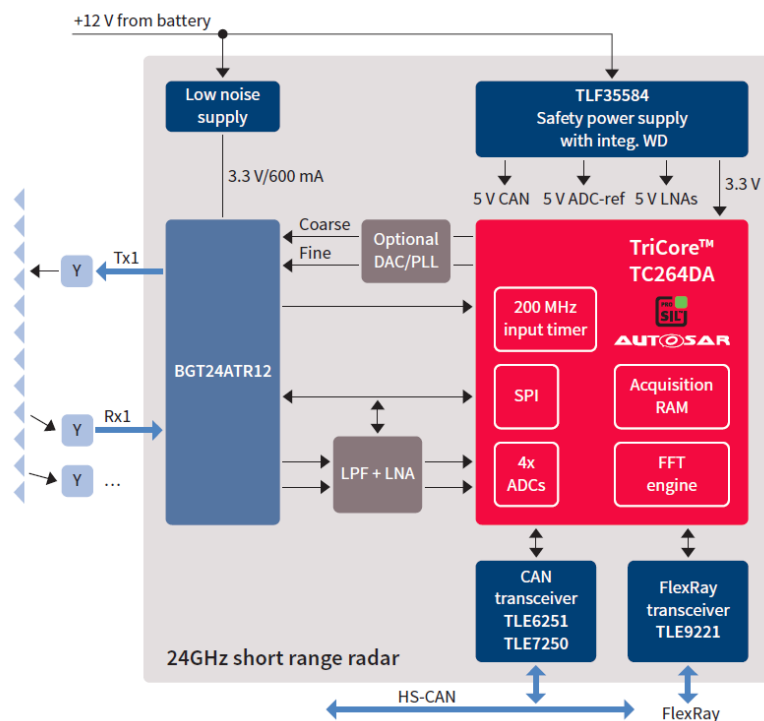


Figure 2: Infineon 24 GHz automotive radar kit block diagram [3]

The onboard microcontroller TC264DA has a built-in FFT engine. Further processing on the raw radar samples can be performed on board without the need of external computer. In practice the raw samples are transmitted through the Ethernet port to a computer so that several algorithms can be tested.

The modulation scheme of the Infineon radar kit is fast chirp. Compared with normal chirp rate, fast chirp improves the range resolution of the radar and in the meantime maintains the accuracy for the velocity measurement. Figure 3 shows the fast chirp scheme that used in the Infineon radar kit. Each radar frame corresponds to 128 chirps, each chirp contains 128 samples.

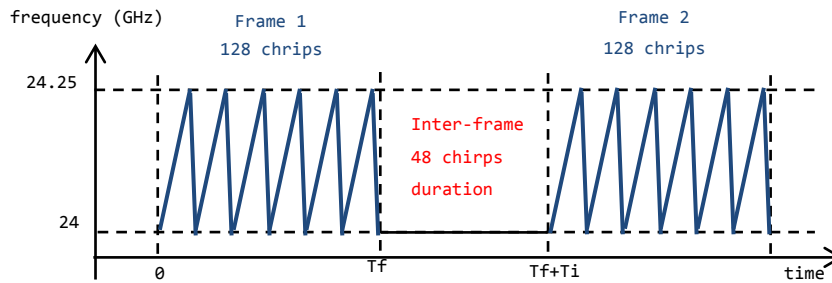


Figure 3: FMCW fast chirp scheme [4]

Table 1 and table 2 lists the resolutions and performance specifications of the Infineon radar kit.

	resolution	Max range
Range	0.6m,	38.4m
velocity	± 0.49 km/h	± 31.5 km/h
angle	1 degree	50 degree

Table 1: Resolutions of Infineon radar kit [4]

Carrier frequency	24GHz
Wavelength	1.25cm
Sweep bandwidth	250MHz
ADC sample frequency	366.3kHz
Number of samples per chirp	128
Number of chirps per measurement	128
Sweep period	354.9us

Table 2: Specifications of Infineon radar kit [4]

2.2 Current Research

The first step of radar image processing is noise removal. M. Hozhabri has presented a few clutter techniques in her work to remove clutters [6]. Then to obtain the correct detection from noise, several target detection method can be applied, including CFAR. Some clustering algorithms can also be applied for detection association. N. Carlström has used DBSCAN to assign correct detection to a target [7].

For target tracking, the Kalman filter can be used [8]. The extended Kalman filter [9] can also be applied here to make use of the radial velocity.

To classify different targets, S. Heuel and H. Rohling use SVM to classify the features of target range-Doppler profile [10]. Also they use the $\alpha\beta$ filter for target tracking and a two stage target classification technique to classify the range-Doppler profile of targets [11]. However, these two researches did not take into account the periodic movement pattern of human walking.

A. Sundaresan and his associates have presented a method to recognize human from gait sequences based on Hidden Markov Model [12], which shows that the movement pattern of walking could be utilized for human recognition.

2.3 Measurement Cases

To properly analyze the raw data samples from the radar. The raw data packets are captured with the data capture system `radar_viewer` developed in the previous work. Several test cases are conducted. In each test case, there may be more recordings made with different conditions. Recording made outdoor is done outside the FH-Dortmund main building. Recordings made indoor are done in a tent in Fredenbaum Park, Dortmund. Due to equipment limitations, one test is done inside the lab of FH-Dortmund. The specific test conditions (e.g., length and width of the test areas) are not provided, since these tests are to test the proof of concept implementation of the system. Figure 4 shows the two photos of the testing areas.



Figure 4: inside a tent (left), outside FH-Dortmund main building (right)

After the recording, the raw data files are processed in Matlab where various signal processing algorithms are tested.

Test case 1

Scenario: One people walking (longitudinal and lateral)

Objective: Test the basic measurement capabilities of the developed system.

Description: As shown in Figure 5 left image, the person is walking longitudinally away from the radar, AoA θ is 0. When the person reaches to line L2, he turns around and walks directly towards the radar, also with $\theta = 0$. The distance between L1 and L2 is D. In Figure 5 right image, the person starts at A and walking directly to the right side, AoA θ is continuously changing. When the person reaches location A', he then turns around and walks back to A. The distance between A/A' to radar center line is L.

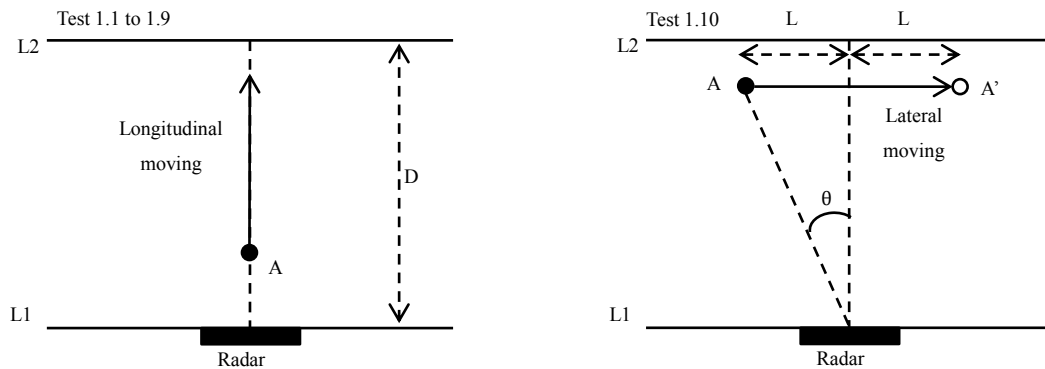


Figure 5: movement setup of test case 1

Tests 1.1 to 1.7 are the longitudinal walking of adults. Test 1.8 is the longitudinal walking of a child. Test 1.9 is the longitudinal moving of a cart tray, no walking pattern presents. Test 1.10 is the lateral walking of person A.

Test 1.1: person E is walking casually outdoor.

Test 1.2: person A is walking casually inside a tent.

Test 1.3: person A is walking slower inside a tent.

Test 1.4: person A is walking casually inside a tent with hands in pockets.

Test 1.5: person A is running inside a tent.

Test 1.6: person B is walking casually inside a tent.

Test 1.7: person C is walking casually inside a tent.

Test 1.8: person D (child) is walking casually inside a tent.

Test 1.9: person A is a moving cart tray in a lab.

Test 1.10: person A is walking casually inside a tent (lateral).

Test case 2

Scenario: Two people walking (longitudinal and lateral)

Objective: To test the two target measurement capabilities of the developed system.

Description: As shown in Figure 6, 4 tests are made with each individual movement pattern. Each test is described below:

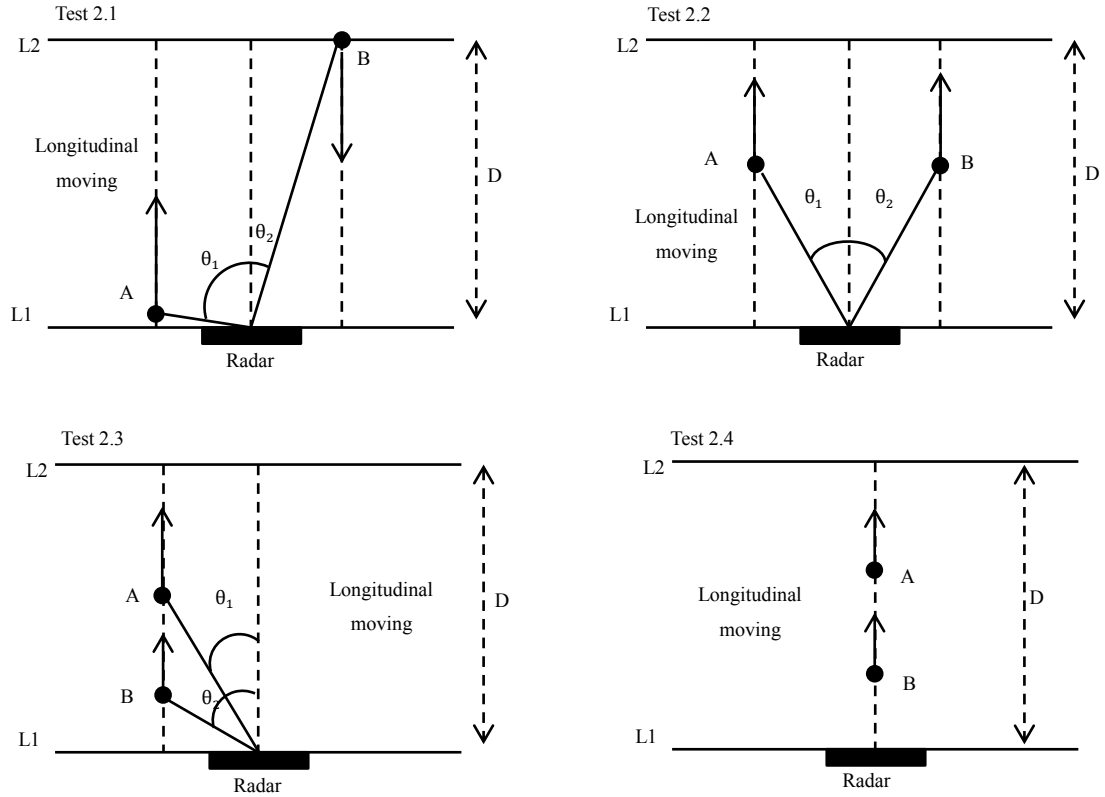


Figure 6: movement setup of test case 2

Test 2.1: different angle, different range-Doppler profile

There are two people. Person A is located at line L1 and walks longitudinal towards line L2. When A reaches line L2, he then turns around and walks back towards L1 also in a straight line. Similarly, the other person B starts at line L2 and walks towards line L1 at the same time. When B reaches line L1, he turns around and walks back to L2. θ_1 is the azimuth angle between person A and radar, θ_2 is for person B. This test is made to evaluate the basic two targets (different angle) tracking.

Test 2.2: different angle, similar range-Doppler profile

Two people A and B are located at each side of the radar at line L1, and then they walk longitudinally from the line L1 towards L2 at a similar pace (with similar speed). When they reach line L2, they walk away from L2 and walk towards L1. θ_1 is the azimuth angle between person A and radar, θ_2 is for person B. This test is made to evaluate the basic angle separation of the system. Since A and B are of similar range-Doppler profile.

Test 2.3: similar angle, different range-Doppler profile

Two people A and B are both located at the left side of the radar at line L1. Person A first walk longitudinally from the line L1 towards L2. After a few seconds, person B follows the same path of A. When they reach line L2, they walk away from L2 and walk towards L1. θ_1 is the azimuth angle between person A and radar, θ_2 is for person B. This test is made to evaluate two targets (similar angle) tracking.

Test 2.4: same angle, blocked signal

Two people A and B are both located in front of the radar at line L1. Person A first walk longitudinally from the line L1 towards L2. After a few seconds, person B follows the same path of A. When they reach line L2, they walk away from L2 and walk towards L1. The AoA θ for person A and B is always 0. This test is made to evaluate the signal blocking effect.

Tests 2.1 to 2.4 are all done inside the big tent with two people walking casually.

Test case 3

Scenario: Three people walking (longitudinal).

Objective: To test the three target measurement capabilities of the developed system.

Description: As shown in Figure 7 below. Three people A, B, and C are located at line L1. A, C are located at each side of the radar, B is located in front of the radar. Then they walk away from the line L1 towards L2 longitudinally at a similar pace (with similar speed). When they reach line L2, they walk away from L2 and walk towards the start location at L1. θ_1 is the AoA of person A, θ_2 is the AoA of person C. The AoA for person B is 0. This test is made to evaluate the angle separation performance of the system further.

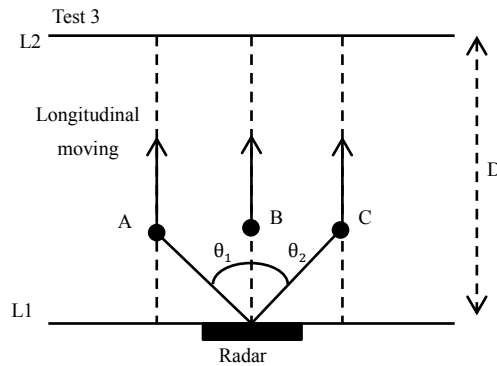


Figure 7: movement setup of test case 3

3 Signal Preprocessing

To extract desired target information from the received raw radar data, certain preprocessing must be performed on the radar raw images. Figure 8 shows the overview of all the blocks in the signal preprocessing; the complete signal preprocessing is done on the computer side. A short description is provided below:

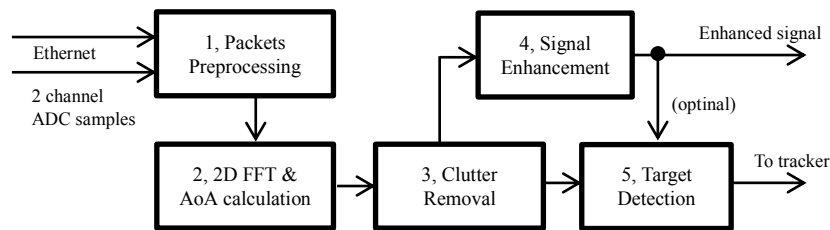


Figure 8: signal pre-processing blocks

In block 1, after the data packets are received from Ethernet, they are checked for packet loss, and then the raw data samples are extracted from these packets.

Block 2 performs the 2D FFT on the raw ADC samples to calculate the initial radar frame and the corresponding angle of arrival (AoA).

Block 3 reduces the environment noise of the raw radar frame to an acceptable level.

Block 4 is developed to amplify the weak signal. However, the noise is also increased; therefore this block is optional for target detection depending on actual requirement.

In block 5, firstly, an angle separation step is introduced. The primary purpose of the angle separation is to separate signals with angle differences so that target detection can be more accurate. Then based on the signal to noise ratio (SNR), target detection can be declared with certain algorithms in the target detection block. Two detectors are presented and tested in this chapter: the local maxima detector and the CA-CFAR detector.

Block 1 and 2 belong to the previous work on the Infineon radar module. This chapter is divided into three sections, each focuses on the three later blocks: clutter removal, signal enhancement, and target detection. Each section provides the theory and algorithms first and then the processed results for evaluation.

3.1 Clutter Removal

To properly process the signal noise, the averaged environmental noise is first measured. Figure 9 shows a complete noise frame calculated from 20 continuous frames (with no moving objects) of a saved recording measured indoor (about 1.1874s). To achieve this, the image of the stationary environment (without moving objects) is measured and considered to be the environment noise. The environment noise is measured for a certain duration and used to calculate a carefully arranged noise image, it is then be used for noise subtraction for all radar images at that same location.

The mirrored image to the left is not removed. The SNR scale is set to $1e3$, so that the detailed noise clutters are visible. There are two types of clutters presents, stationary clutter (marked as A) and correlated clutter (marked as B). Stationary clutter mainly located around Doppler bin number 64 (63 to 65) and spans across the whole range bin. Correlated clutter in this setup mainly stays at a specific range bin and spans multiple Doppler bins, but the number of Doppler bins varies. Stationary clutter could be easily removed by setting the entire clutter to zeros.

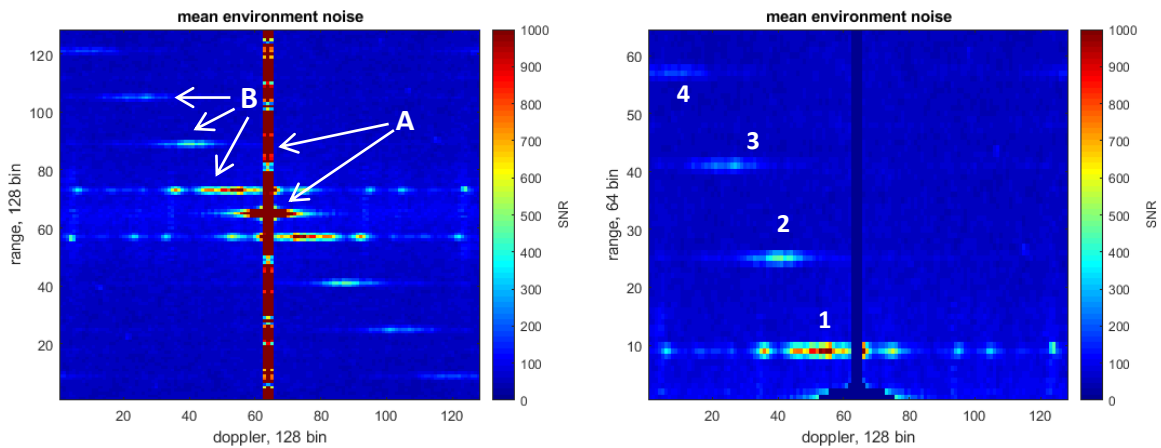


Figure 9: noise image (left), noise image with stationary clutter cleared (right)

As shown in Figure 9 right image, both the clutter in the middle and the bottom middle is cleared. The four correlated clutters seem evenly distributed across range bins. With the increasing of range bin, the clutter strength is also decreasing. However, the location of these correlated clutters is consistent during the operation.

Let $o_{r,d}$ be the range-Doppler frame produced by processing enough packets. In this case, $o_{r,d}$ is obtained by receiving and processing 128 packets from 128 fast chirps, $r = 1 \dots 128, d = 1 \dots 128$. Since the Infineon radar kit has angle measurement capabilities, let $\theta_{r,d}$ be the angle measurement frame of the corresponding $o_{r,d}$. Therefore each measurement consists of two frames, $\theta_{r,d}$ and $o_{r,d}$.

Let $n_{r,d}$ be the SNR value of the mean environment noise with stationary clutter cleared and the mirror image removed, the range bin r , and Doppler bin d , as shown in Figure 9 right side. $n_{r,d}$ can be calculated with Eq. 3.1 (There are some clutter cells at the bottom also removed, please see the source for detailed cell number). The mean Doppler bin SNR value \bar{S}_r is calculated using Eq. 3.2:

$$n_{r,d} = O_{r=(65:128),d=(1:62,66:128)} \quad \text{Eq. 3.1}$$

$$\bar{S}_r = \frac{1}{64} \cdot \sum_{d=1}^{128} n_{r,d} \quad \text{Eq. 3.2}$$

The original black line in the following Figure 10 shows the \bar{S}_r of each range bin across all Doppler bins. The four peaks represent the projection of four correlated clutters on the range bin axis. Each peak has a fixed width of w_{peak} , $w_{peak} = 3$ in this case.

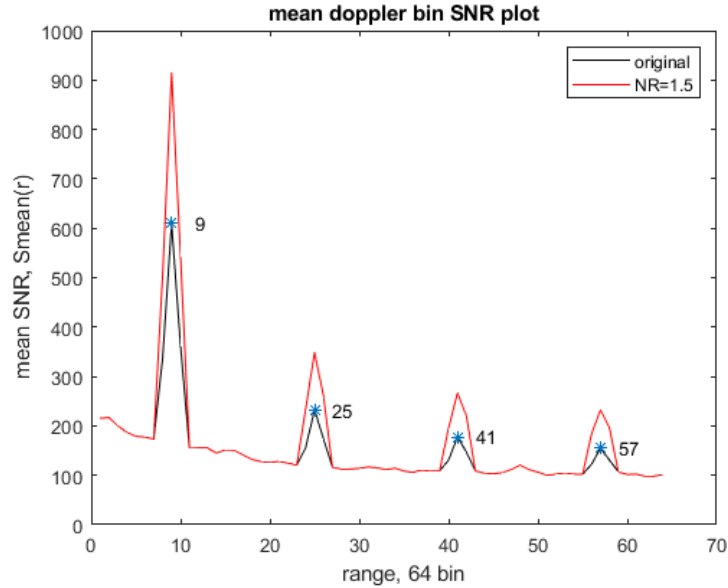


Figure 10: mean Doppler bin SNR plot

To reduce the averaged environment noise even more, the areas around each peak will be multiplied by a factor NR (noise ratio). The red line in Figure 10 is the adjusted Doppler bin SNR where only the peak value is multiplied with a factor NR=1.5. In this way, the correlated clutter will receive more reduction compared with other regions. The process is defined as follows:

Let p_r be the range bin of peaks found in the \bar{S}_r .

```

For each  $p_r$  do:
  For all r
    If  $r = \text{round}(p_r \pm \frac{w_{peak}}{2})$  then
       $n_{r,d} = n_{r,d} \cdot NR;$ 
    end
  end
end

```

With the calculated averaged environment noise $n_{r,d}$, for each actual measurement $m_{r,d}$, the processed signal $a_{r,d}$ can be written as Eq. 3.3:

$$a_{r,d} = \begin{cases} m_{r,d} - n_{r,d}, & \text{if } m_{r,d} > n_{r,d} \\ 0, & \text{otherwise} \end{cases} \quad \text{Eq. 3.3}$$

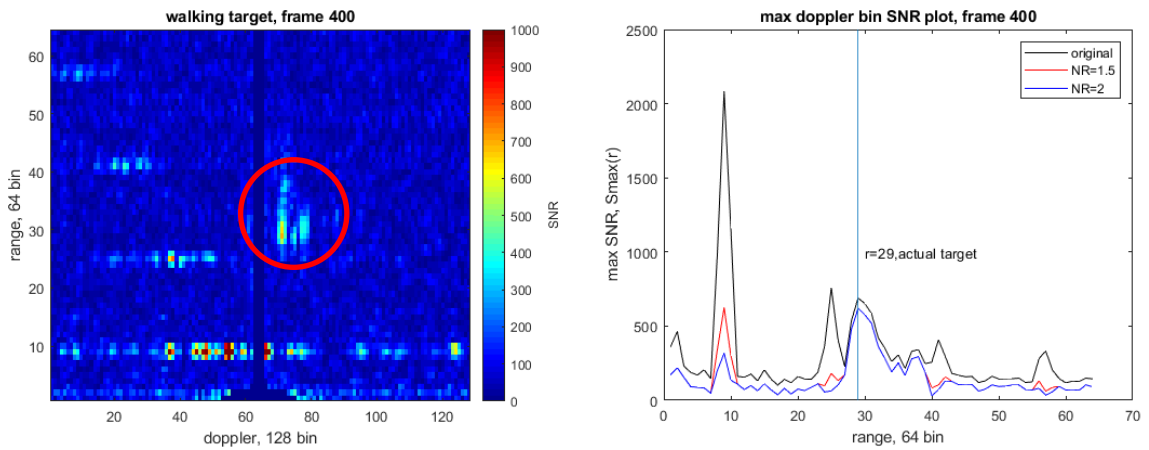


Figure 11: test frame $n_{r,d}$ (left), max Doppler bin SNR (right)

Figure 11 left shows a test frame $n_{r,d}$ from a saved recording, the target is clearly visible inside the marked red circle, The relative peak can be measured at $r = 29$, $d = 71$.

The right image shows the maximum SNR found in each range bin across all Doppler bins. The max Doppler SNR $Smax_r$ is calculated as Eq. 3.4:

$$Smax_r = \max_{d=1:128}(n_{r,d}) \quad \text{Eq. 3.4}$$

As can be seen, the original black plot has six major peaks, peak at $r=9$ has the largest SNR value. However, of all these peaks, only the peak at $r=29$ relates to actual target detection, the other peaks are background noises. The red plot and blue plot are two attempted $a_{r,d}$ to remove the noise $n_{r,d}$, one with $NR=1.5$ (red plot) and $NR=2$ (blue plot). The results of the 2 attempts are mostly the same except at the location where the noise peaks are. With $NR=1.5$, the noise peak at $r=9$ is still somehow comparable with actual detection at $r=29$. Only with $NR=2$, noise peaks are reduced further.

Although the clutter removal can remove most noises from the measurement frame $a_{r,d}$, In practice, there are still very few noise points left. When converting these noise points to Cartesian coordinates, these points tend to stay close to some specific points. The following method considered to reduce these noise points in Cartesian system.

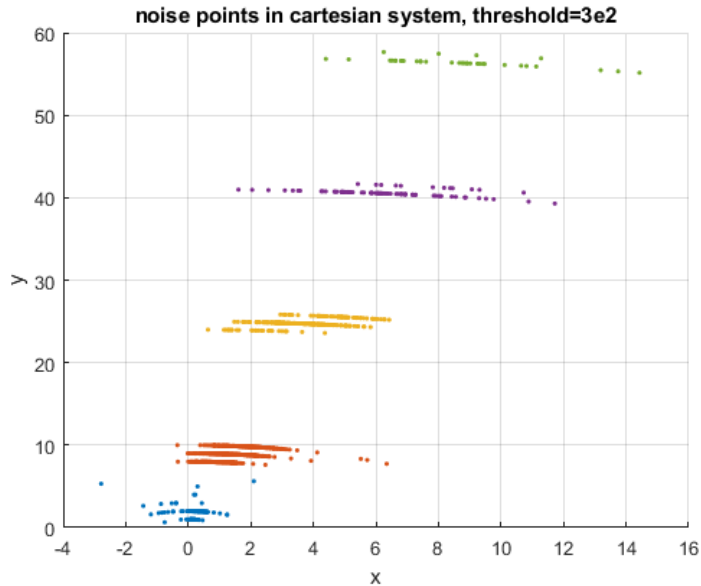


Figure 12: environment noises in Cartesian system

Figure 12 shows all the noise points in the above mentioned 20 continuous environment noise frames converted to Cartesian coordinates. There are mainly five noise clusters; each one has a relatively large x range spread. These clusters are first separated, and then the mean noise point for each cluster is calculated. If any detection point is within a specific range of these mean noise points, then it should be considered as a noise point.

The cause of the correlated clutters shown in the test frame is not clear, it may be related to antenna cross talk. They have a comparably large affected area. The advantage of environment noise subtraction shown in this section is the ability to remove the false peaks while keeping other signal areas intact. However, when the target detection is close to the noises, the actual signal is subtracted as well, making later detection difficult.

There is another type of noise clutter that can be found in the Infineon radar system, named ghost targets. When the actual target is at close proximity of the radar, the return radar EM waves are reflected again and received again by the radar. Then it manifests itself in the range-Doppler plot as a ghost target. The ghost target shares the same Doppler profile with the actual target, while the range profile doubles. To prevent the ghost targets, all test cases in chapter 2.3 were done in relatively open space and kept some distance away from the radar.

3.2 Weak Signal Enhancement

The following Figure 13 shows the range-t plot (left) and the Doppler-t plot (right) of test 1.1 described in chapter 2.3. A person is first walking away from the radar and then walking towards the radar for totally about 40 seconds. The SNR threshold = $5e2$, $NR = 2$. As the distance between the person and the radar module increases, the received signal at the radar becomes weaker. The Doppler spread of the target also decreases. If the SNR threshold is smaller, a lot of extra noises are introduced into the images. Therefore, a method to amplify the weak signal should be considered.

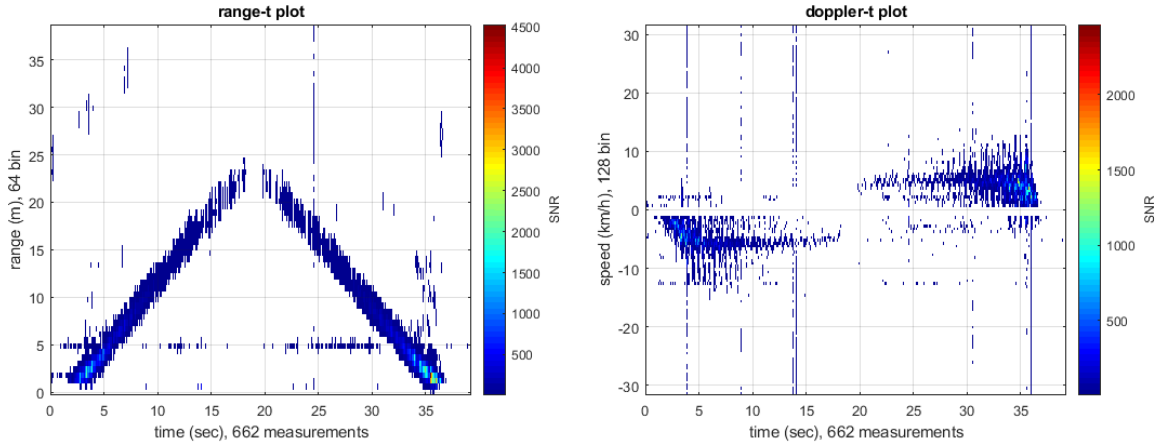


Figure 13: range-t and Doppler-t of recording [three]

The time gain method:

The time gain method rises from the idea that signal strength correlates well with traveled time from the radar system [13]. The traveled time correlates directly with the distance from the radar to the target. With the processed data set $a_{r,d}$, time gain factor k . $b_{r,d}$ can be calculated with Eq. 3.5.

$$b_{r,d} = a_{r,d} \cdot r^k, k = 1, 2, \dots, n \quad \text{Eq. 3.5 [13]}$$

By multiplying the distance factor to the signal, the signal that far away can be increased significantly, Figure 14 below shows the Mean Doppler bin SNR \bar{S}_r of the same image in Figure 11, with $NR=2$. The black plot is the original $a_{r,d}$, red and blue plots

represents the corresponding $b_{r,d}$ with $k=1$ and $k=2$. As can be seen, not only the target signal is amplified, the noise is increased dramatically. Especially for $k=2$.

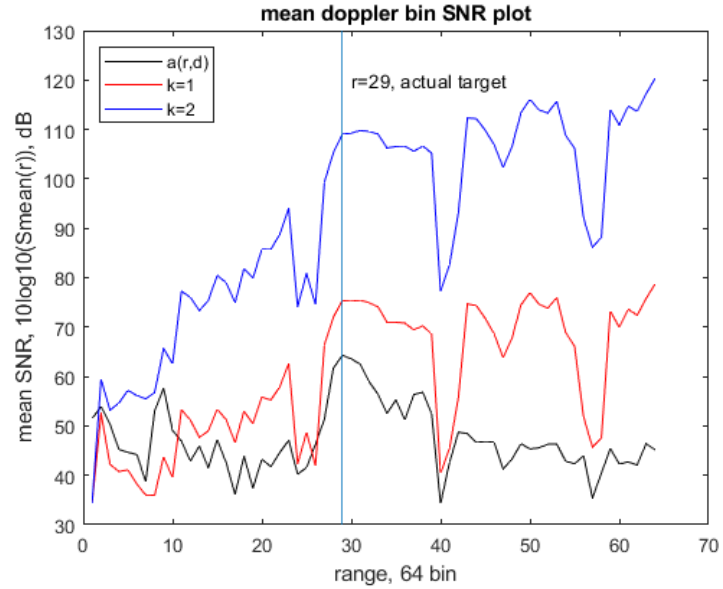


Figure 14: \bar{S}_r of Figure 11 test frame

To explore the time gain effect throughout time, Figure 14 shows the max Doppler SNR S_{max_r} of the entire recording of test 1.1, which consists of 662 frames, The threshold= $1e3$, NR=2 in this test case. As can be seen, without the time gain, the original (black plot) signal strength decreases rapidly with the increase of distance. With $k = 1$ (red plot), the SNR decreases slower. With $k=2$ (blue plot), the SNR becomes relatively stable over distance, which means a stable target pattern could be preserved with longer distance.

Figure 16 shows the result when applying the time gain method to the recording of test 1.1, time gain $k=2$, the threshold is adjusted to $3e5$. In the left image, the target image at around 20s is clearly visible. In the right image, the walking pattern is visible all the time and the Doppler spread is relatively stable.

The time gain method works very well for weak signals far away from the radar. However, it also amplifies existing noises, making later signal detection difficult. Therefore it mainly used in situations when signal detection is positive and target pattern extraction is desired.

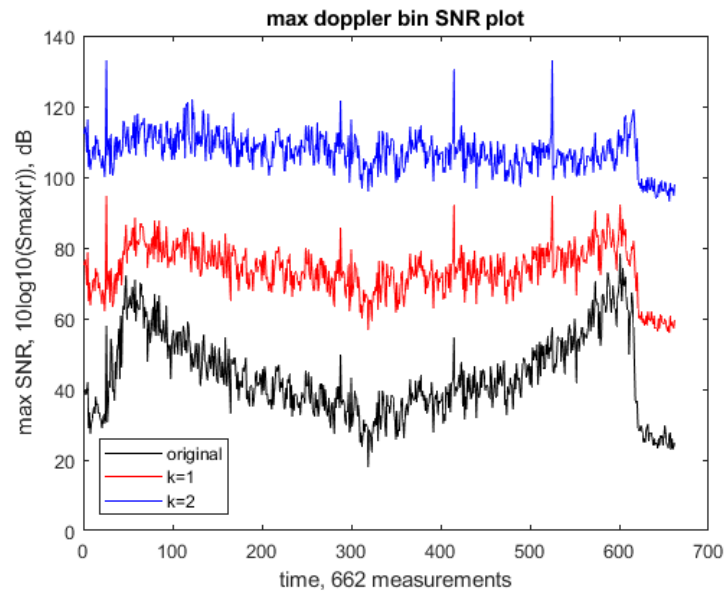


Figure 15: $S_{max,r}$ of test 1.1, with different k

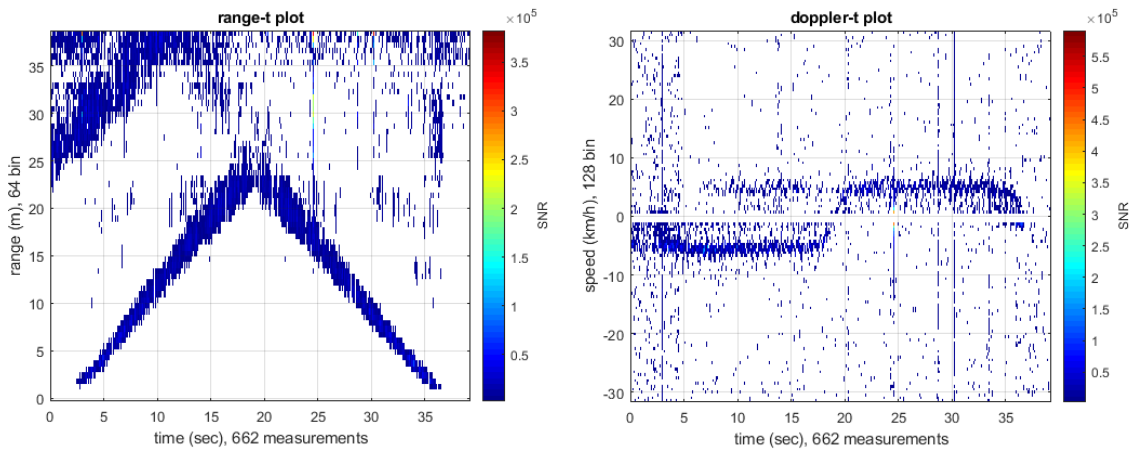


Figure 16: range-t and Doppler-t of test 1.1 with time gain k=2

3.3 Signal Detection

Once the signal has gone through environment noise subtraction and weak signal enhancement, a target signal detector can be used to pick out the correct signal out of noises. However, if two targets have similar range-Doppler profile, they will show up very close to each other in the radar measurement frame $a_{r,d}$, which makes it extremely difficult to separate these two targets. For radar modules with angle measurement capabilities, an angle analysis could be introduced to separate these targets easily.

In this section, an angle separation method is discussed first to explore the possibilities to separate $a_{r,d}$ based on the angle measurement $\theta_{r,d}$. Also two signal detection methods are presented and reviewed. They can work individually or combined to give a better result.

3.3.1 Angle Pre-separation

The range-Doppler plot does not contain angle information of the target. When targets share a similar range and velocity profile, they will share a similar location in the range-Doppler plot. The separation of these targets is challenging. For each $a_{r,d}$ in the radar frame, find the corresponding $\theta_{r,d}$. By counting the same $\theta_{r,d}$, a histogram plot of $\theta_{r,d}$ can be obtained.

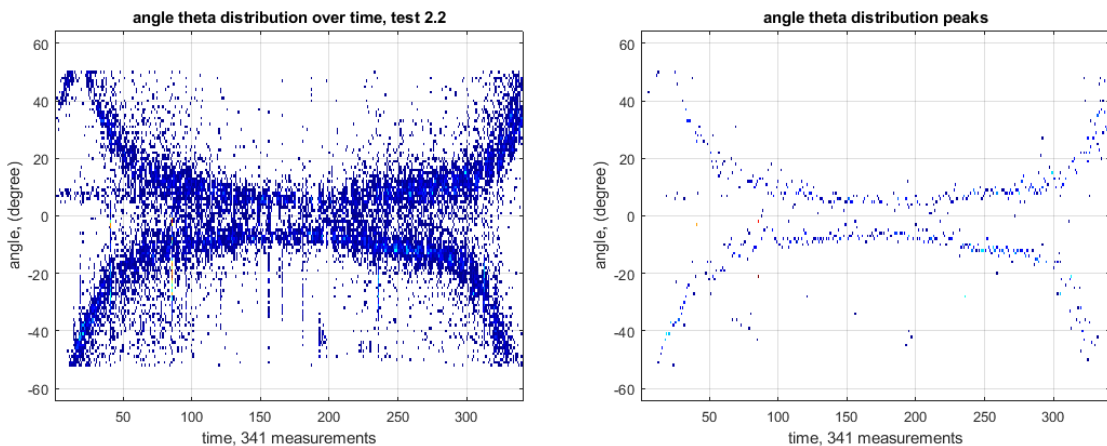


Figure 17: angle distribution of test 2.2 (left), angle distribution peaks (right)

Figure 17 image to the left shows the $\theta_{r,d}$ distribution over time of test 2.2. It can be seen that the two strong signals represent two persons; each signal has a wide angle range. A very intuitive way to separate the two strong signals is to find the angle value that has the highest histogram count (peaks). Then separate all data between these peaks. Figure 18 shows the angle distribution of frame 274 from test 2.2. Two peaks P_A and P_B are clearly visible, which indicates the two targets (A and B) have the highest possibility at -12 degree and 9 degree. To separate the two targets, the middle angle $P_M = \frac{P_A + P_B}{2} = -1.5$ can be chosen as the separation angle. All data to the left side of P_M belongs to target A, and the other side belongs to target B. With more targets, the number of P_M will also increase.

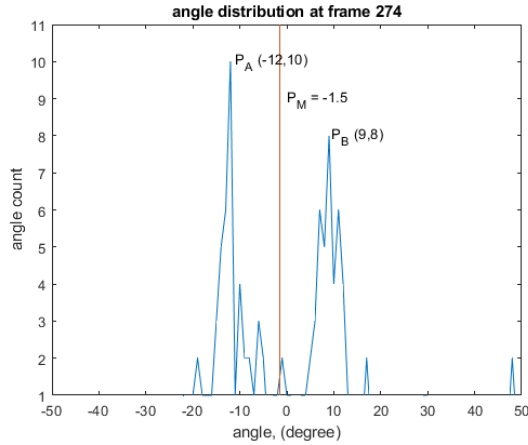


Figure 18: angle distribution of one frame

However, for the above mentioned angle separation method to work, the correct detection of the target number in the frame is the key. Figure 17 right image is an attempt to detect the number of targets by finding the 1D local maxima ($g=7$). The maxima with less than 2 angle occurrences are ignored. Table 1 shows the maxima count for all 341 frames, which can be an indication of the target number.

Number of maxima	0	1	2	3	4
Number of frames	43	121	154	22	1

Table 3: Maxima count for test 2.2

Although the number of the frame that has the correct number of targets (two targets) is the highest in Table 3, there are still more than half (54.8%) frames that have the incorrect number of the target numbers. Moreover, with the increase of targets, the difficulty of angle separation also increases dramatically. Figure 19 shows the angle distribution of test 3. With three people walking in a similar range-Doppler profile, only the walking pattern (person B) in the middle is distinct, the other two people do not show up clearly. These multiple targets share the same location on the range-Doppler plot, therefore the more similar targets there are, the fewer detection points each target can have. Finding the correct P_M to separate radar data, in this case, is very challenging.

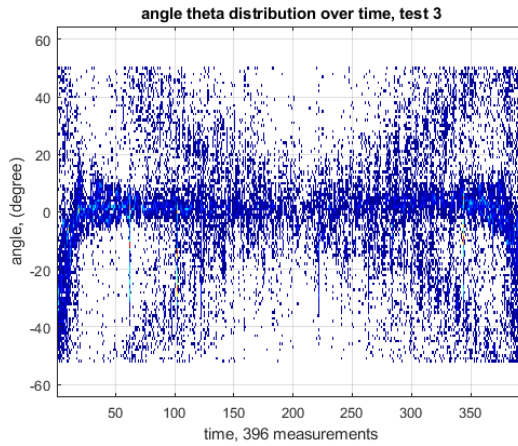


Figure 19: angle distribution of test 3

In this paper, the test cases can be divided into single target and multi-targets test case. For single target test cases, the angle separation is not necessary. In test cases involve two people, for a simple separation, $P_M = 0$ is used. The frame $a_{r,d}$ is separated by the signs of $\theta_{r,d}$, positive $\theta_{r,d}$ will have a right frame, and negative $\theta_{r,d}$ will produce a left frame. Then the two frames are sent to the detectors so that a less ambiguous detection can be achieved.

3.3.2 Local maxima detector

For a given processed 2d frame $a_{r,d}$ or $b_{r,d}$, there could be many strict local maxima within. These strict local minima are all possible valid target detection. The local maxima detector evaluates all local maxima within a neighboring area D. If a local maxima X is larger than other local maxima in area D, then only X should be the target detection. In this way, the relatively larger local maximum in the surrounding is selected to be the target detection. The objective of the local maxima detector can be simplified to the following:

Let $a_{r,d}(r = 1 \dots 128, d = 1 \dots 128)$ be the processed frame, find all local maxima $m_{mr,md}(mr = 1 \dots 128, md = 1 \dots 128)$ in $a_{r,d}$, such that the minimal range bin and Doppler bin distance between each local maxima is g .

The algorithm of local maxima detection is described as:

```
For each  $a_{r,d}$  do:  
  For each  $m_{mr,md}$  in area D of  $a_{r,d}$ ,  $mr = (r - g : r + g)$ ,  $md = (d - g : d + g)$   
    boundary check;  
    If  $m_{mr,md} > a_{r,d}$ , then  $a_{r,d}$  is not a local maxima,  
      clear  $a_{r,d}$ ;  
      continue to next  $a_{r,d}$ ;  
    end  
  end  
  If  $a_{r,d} > m_{mr,md}$ , for all  $m_{mr,md}$  in D, then  $a_{r,d}$  is a local maxima,  
    All elements in area D but  $a_{r,d}$  are cleared;  
  end  
end
```


The area D in the above algorithm is defined as a square with a width of $2g + 1$. It could also be defined as other shapes such as a circle or rectangular. The maxima at the boundaries are also evaluated in this algorithm, and they could be ruled out by adjusting the elements of $a_{r,d}$.

For a 1D array u_r , the algorithm can be adapted to the following to find peak m_{mr} :

```

for each  $u_r$  do:
    for each  $m_{mr}$  in area D of  $u_r$ ,  $mr = (r - g : r + g)$ 
        boundary check;
        if  $m_{mr} > u_r$ , then  $u_r$  is not a local maxima,
            clear  $u_r$ ;
            continue to next  $u_r$ ;
        end
    end
    if  $u_r > m_{mr}$ , for all  $m_{mr}$  in D, then  $u_r$  is a local maxima,
        All elements in area D but  $u_r$  are cleared;
    end
end

```

The local maxima detector is a simple yet effective signal detector; it does not rely on a fixed threshold for detection. Thus it can detect a relatively weak signal. However, it does not check for detailed signal quality. Therefore many noises can be treated as positive detection if the noise is has a strong peak. Also the complexity of the algorithm is $O(n^2)$, with a large g, the algorithm can be slow.

3.3.3 CA-CFAR detector

Cell-averaging Constant False Alarm Rate (CA-CFAR) is a scheme to detect weak signals by evaluating the surrounding cells. It calculates a dynamically changing threshold used for detection from the surrounding cells by maintaining a constant probability of false alarm P_{fa} [14].

For 2D CA-CFAR, Figure 20 shows the cell arrangement. The cell under test (CUT) is marked as ‘C’. The cells directly adjacent to CUT are guard cells (marked as ‘X’), they are excluded for the CUT evaluation. The outer layer cells (marked as ‘R’) are reference cells; they are used to calculate the surrounding energy of the CUT.

	R	R	R	R	R	
	R	X	X	X	R	
	R	X	C	X	R	
	R	X	X	X	R	
	R	R	R	R	R	

Figure 20: CA-CFAR cell arrangement

In evaluation, if the values of CUT is larger than threshold T_{cfar} (calculated with Eq. 3.6 to Eq. 3.8), then the CUT is consider to be a target detection. α is the threshold factor that depends on P_{fa} , P_N is the noise power estimation of the reference cells.

$$T_{cfar} = \alpha P_N \quad \text{Eq. 3.6}$$

$$\alpha = N \left(P_{fa}^{-\frac{1}{N}} - 1 \right) \quad \text{Eq. 3.7}$$

$$P_N = \frac{1}{N} \sum_{i=1}^N x_i \quad \text{Eq. 3.8}$$

The cell width of both guard cells and reference cells are flexible and depend on the pattern of the interested target signal. Similar to the local maxima detector, the CUT is evaluated with area D (area marked in R), here D is defined as a rectangular with width w and height h, which relates to the walking pattern of a human. The details of the target pattern are discussed in chapter 5.

In order to calculate the threshold, the energy of an area D is calculated, the cells at the boundaries of $a_{r,d}$ are not evaluated.

The algorithm of CA-CFAR detection is described as:

calculate alpha;

for each $a_{r,d}$ **do**:

for each $m_{mr,md}$ in area D of $a_{r,d}$, $mr = (r - w : r + w)$, $md = (d - h : d + h)$,

 sum up all elements in area D;

end

 calculate the threshold $Tcfa_{r,d}$;

if $a_{r,d} > Tcfa_{r,d}$, **then** $a_{r,d}$ is a valid detection,

 save $a_{r,d}$;

end

end

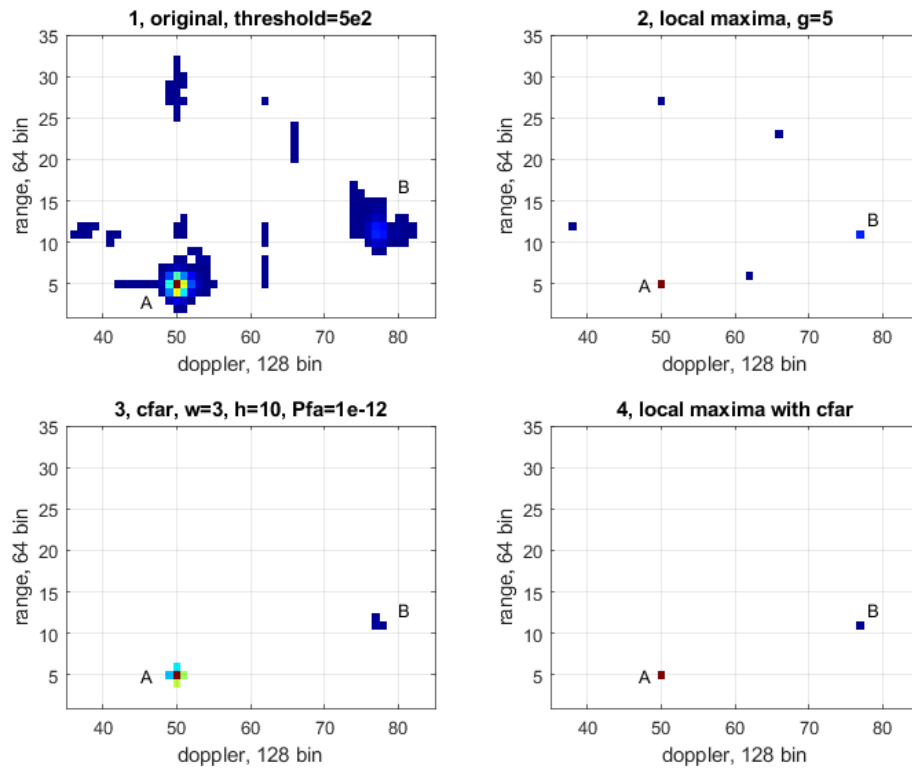


Figure 21: Target detection of frame 150

Figure 21 shows a test frame $a_{r,d}$ from a saved recording. It contains 4 subplots that demonstrate the results using the two detector method in this section.

Subplot 1 shows the original frame with a normal threshold $5e2$. Two targets (A and B) are clearly visible with several noise points.

Subplot 2 shows the result of local maxima detector with $g=5$. All targets including the noise points are treated as detections, which is not desirable.

Subplot 3 shows the result of CA-CFAR detector with $w=3$, $h=10$, and $P_{fa} = 1e^{-12}$. Now the actual targets are correctly detected. However, the CA-CFAR detector has two problems in this test. The tuning of P_{fa} is critical for a successful detection, a less optimal P_{fa} will result in either less detection or more noise detections. Also, since the cells around the detection points usually have low signal energy, the CA-CFAR detector tends to ignore them. For example, by comparing subplot 1 with subplot 3, the size of target A is reduced significantly, and the shape of A is also changed. This does not benefit the micro-Doppler study of the targets.

Subplot 4 shows the result of a combination of two detectors. By evaluating the local maxima in the CA-CFAR detector, all noise detections are filtered out. The remaining two points indicate the correct detection of the target core. By combining subplot 4 with subplot 1, the original shape of the target can be extracted and further processed.

In conclusion, both the local maxima detector and the CA-CFAR detector work well for the Infineon radar kit. With a local maxima detector, false detection often occurs due to noise points, which could be filtered out with CA-CFAR detector. However, the P_{fa} tuning is critical and depends on the setup. In most situations, certain calibration is required for the system to obtain a better performance.

4 Target Tracking

After the successful target detection, the polar coordinate that contains the range, radial velocity, and angle could be obtained for detected targets. Some other attributes, including Doppler spread and angle spread could also be analyzed. However, if the focus is on the dynamic behavior of the targets, the analysis of one static time frame is not sufficient, all related frames need to be analyzed to extract relevant information regarding each specific target. Therefore with valid detection points, a tracking method is necessary to associate the detection points to each target.

The objective of the target tracking in this paper is to locate the main target signal, so that the surrounding which contains the micro-Doppler information can be extracted for later analysis. All detection points have already gone through the clutter removal and CA-CFAR noise reduction in chapter 3. However, depending on the tracking range requirement, the time gain is optional for close range tracking experiences.

In this chapter, target tracking is divided into single-target and multi-target tracking depending on the use case. The popular Kalman filter is introduced to estimate the location of the target based on existing detections. In multi-target tracking, the Hungarian algorithm is used to solve the assignment problem between the target detection and current tracks. In the end some tests are discussed to show the result of the tracker.

4.1 Kalman Filter

Measurements (also known as the observations) of the system states usually contain statistical noises, they are thus cannot be used directly. The Kalman filter is an optimal estimator that estimates current system states by calculating the joint Gaussian distribution based on its last states. It minimizes the mean square error of the estimated system state by treating all noises as Gaussian. The recursive behavior that it only depends on the last states to makes the estimation contributes to the small memory footprint, which makes it especially suitable for embedded systems with less resource. Figure 22 below shows the typical workflow of a Kalman filter, X is the estimated system state variable, Z is the measurement and \bar{X} is the prediction [15].

Kalman filter works only with a linear system model with Gaussian distribution. However, the angle component of the radar measurement is nonlinear. There are two usual methods to perform Kalman based tracking on common radar signals. One method employs standard linear Kalman filter which uses Cartesian coordinates translated from the polar coordinates. Another uses the Extended Kalman filter to approximate the nonlinear system model with a linear system model, and therefore can use the polar coordinates directly.

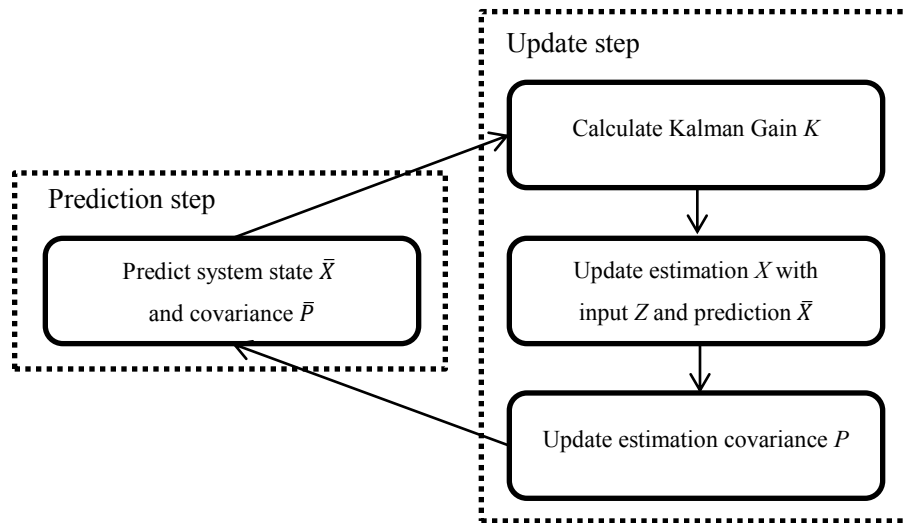


Figure 22: Kalman filter work flow

In principle, the radar measurement frames can also be viewed as pure video frames, and then the target tracking can be performed by apply Kalman filter on the pixels of the range-Doppler plot. The angle measurement can be used as the extra property of the target to help in the target association process. In practice, this method may not work well with targets that have similar range-Doppler profiles, because they may be detected as one target. However, if angle detection is not required, it should work well for the single channel FMCW radar.

In this paper, the standard linear Kalman filter is used. Below is the system model for the Kalman filter.

Since the measurement cycle T of the Infineon radar is about 60ms, the velocity of targets between each measurement can be seen as constant. Thus the target movement model can be simplified to a constant velocity model.

Each radar detection point $m_{r,d}$ has three components $[r, d, \theta]$ from the range-Doppler frame $a_{r,d}$ and angle map $\theta_{r,d}$. r it is the distance between the target and the radar. d is the radial velocity of the target. θ is the azimuth angle of the target. Figure 23 shows the coordinates overview of target A.

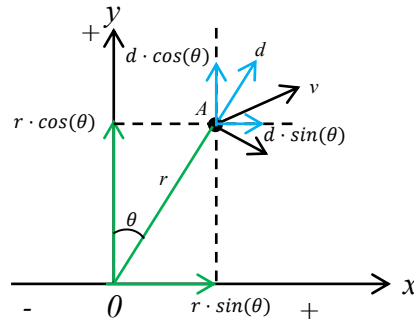


Figure 23: coordinates overview

The converted Cartesian coordinates is shown as:

$$\begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} r \cdot \sin(\theta) \\ r \cdot \cos(\theta) \end{bmatrix} \quad \text{Eq. 3.9}$$

Consider the current system states at t as:

$$X_t = [x \ y \ v_x \ v_y]_t' \quad \text{Eq. 3.10 [8]}$$

The state prediction equation for Kalman filter is shown in Eq. 3.11, B is the control matrix and u is the control input vector. Since the system has no external control input, $B \cdot u = 0$. w is the process noise with Gaussian distribution, which correlates to the covariance matrix E_x , A is the state transition matrix.

$$X_t = A \cdot X_{t-1} + B \cdot u + w_{t-1} \quad \text{Eq. 3.11 [8]}$$

With

$$A = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Eq. 3.12 [8]}$$

Eq. 3.11 can be written as below:

$$\begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_t = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ v_x \\ v_y \end{bmatrix}_{t-t} + \begin{bmatrix} w_x \\ w_y \\ w_{vx} \\ w_{vy} \end{bmatrix}_{t-1} \quad \text{Eq. 3.13 [8]}$$

The measurement matrix Z_t can be written as below; n is the measurement noise, which correlates to the covariance matrix E_z :

$$Z_t = H \cdot X_t + n_t \quad \text{Eq. 3.14 [8]}$$

Since only location of the target is measured, the observation matrix H contains only the Cartesian coordinates x and y , shown below:

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \quad \text{Eq. 3.15 [8]}$$

In the prediction step, the state prediction \bar{X}_t and covariance prediction \bar{P}_t can be calculated as:

$$\bar{X}_t = A \cdot X_{t-1} \quad \text{Eq. 3.16 [8]}$$

$$\bar{P}_t = A \cdot P_{t-1} \cdot A^T + E_x \quad \text{Eq. 3.17 [8]}$$

In the update step, the Kalman gain K_t , the state estimation X_t and covariance estimation P_t can be calculated as:

$$K_t = \bar{P}_t \cdot H^T \cdot (H \cdot \bar{P}_t \cdot H^T + E_z)^{-1} \quad \text{Eq. 3.18 [8]}$$

$$X_t = \bar{X}_t + K_t \cdot (z_t - H \cdot \bar{X}_t) \quad \text{Eq. 3.19 [8]}$$

$$P_t = (I - K_t \cdot H) \cdot \bar{P}_t \quad \text{Eq. 3.20 [8]}$$

The state noise covariance matrix E_x is provided as:

$$E_x = \begin{bmatrix} \frac{T^4}{4} & 0 & \frac{T^3}{2} & 0 \\ 0 & \frac{T^4}{4} & 0 & \frac{T^3}{2} \\ \frac{T^3}{2} & 0 & T^2 & 0 \\ 0 & \frac{T^3}{2} & 0 & T^2 \end{bmatrix} \quad \text{Eq. 3.21 [8]}$$

The measurement noise covariance matrix E_z is written as:

$$E_z = \begin{bmatrix} z_x & 0 \\ 0 & z_y \end{bmatrix} \quad \text{Eq. 3.22 [8]}$$

z_x and z_y are chosen experimentally.

4.2 Single target tracker

If only one target is assumed in the radar field of view, then the objective of target tracking is to find the most possible location for that target in each frame. Only one track needs to be maintained. For a more simplified tracking experience, the following assumptions are made for the single target tracker:

1. There is at most one active target in the current measurement frame, hence one track.
2. The detection that is closest to the prediction is assumed to be the current target.
3. The start location of the target is known, which is the initial condition for the Kalman filter.
4. The target has movement limitations (e.g., velocity limitation); it is utilized to remove outliers.

Figure 24 below shows the core work flow overview of the single target tracker.

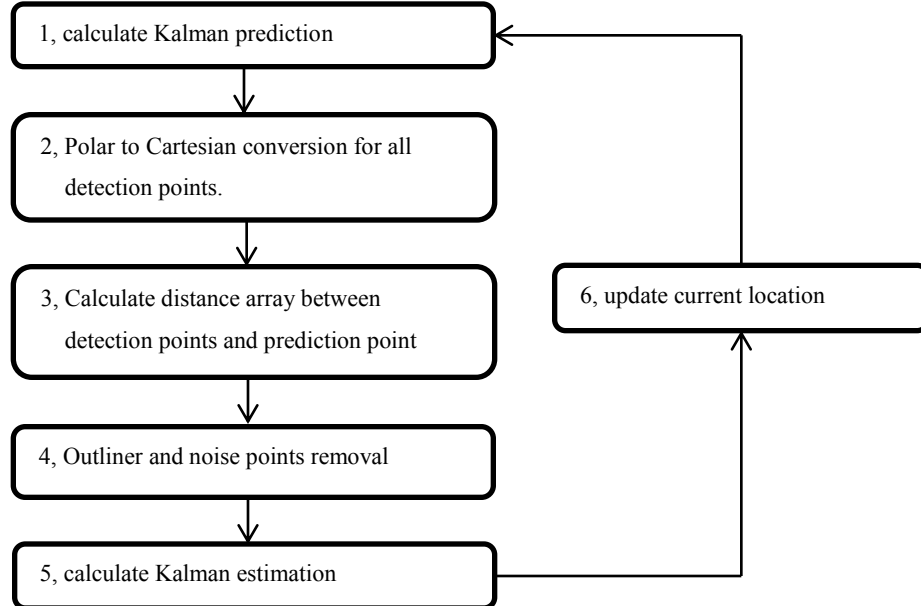


Figure 24: Single target tracker work flow

In Step 1, the initial location of the target is assumed to be $(0, 0)$, then the location prediction is calculated. Here z_x and z_y are chosen to be $1e-2$.

In Step 2, the polar coordinates of all input detections are converted to Cartesian coordinates.

In step 3, in order to find the most possible location of the target. The distance between all detections and the prediction point is calculated. Moreover, the detection that has the smallest distance value is chosen to be the target measurement.

In step 4, before the target can be used in the Kalman update process in step 5, the validity of the target measurement needs to be checked. Unqualified target measurements are considered to be outliers and are ignored for further process. There are usually two types of outliers: the first type is noise point (described in chapter 3.1). The second type is that it exceeds the movement limitations of a normal target (e.g. moving too fast). In this case if the measurement is still considered too far away from the prediction (distance is larger than g), it is considered to be an outlier. If there is no qualified measurement input, then the current prediction is used as the measurement input for the Kalman filter.

Figure 25 shows an overview of the two types of outlier. Consider point A is the current prediction, point D is the noise, and point C is the rejected outlier that is too far away. Point B is the qualified measurement input. If C or D shows up in step 3 as the input candidates, then both points will be rejected, and current input is still chosen to be point A.

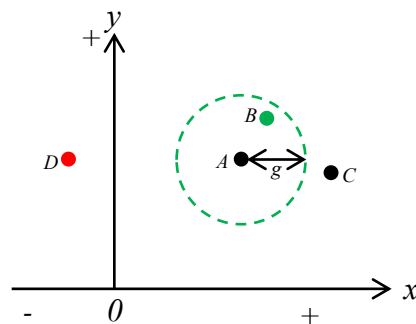


Figure 25: outlier C and D

Step 5 calculates the estimated target location.

Step 6 updates the estimated target location to the track for further processes.

The single target tracker is a simple yet effective tracker for one target scenario. In this paper it is mostly used to extract the multiple feature sets of people's walking patterns and later for measurements.

4.3 Multi-target tracker

In situations of the multi-target scenarios, most of the assumptions of the single target tracker do not hold. Different detections need to be associated with different tracks; the multi-target assignment is more complicated than the single target case. A track management system needs to be established to maintain each track.

Consider the following properties of a track: alive (a), penalty (p) and error (e). Also three states of each track: untracked (state: 0), tracked (state: 1) and lost (state: -1)

a : the number of valid target inputs of a track.

p : the number of continuous updates without valid target input.

e : the mean error between the target input and estimation, calculated as the mean Euclidean distance between input Z_t and the estimation X_t with the formula Eq. 4.1 below, N is the number of time steps:

$$e = \frac{1}{N} \sum_{t=1}^N \|Z_t - X_t\| \quad \text{Eq. 4.1}$$

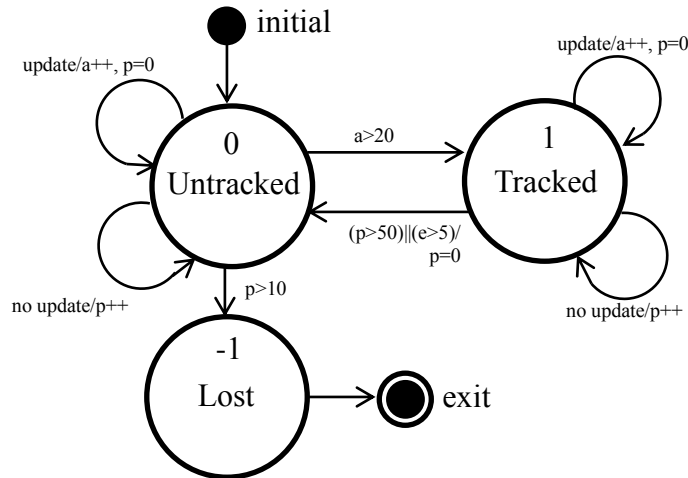


Figure 26: track state diagram

Figure 26 shows the proposed states diagram for the track management system. All tracks initially start in untracked state 0. The following described the transition between each state:

In untracked state 0:

1. If a track receives a valid update, $a=a+1$, $p=0$.
2. If a track has no update at the current time step, then $p=p+1$.
3. If $a > 20$, it is considered to be a valid track and transfer into tracked state 1.
4. If $p > 10$, then it is considered to have lost target and downgrade to lost state -1.

In tracked state 1:

1. If a track receives a valid update, $a=a+1$, $p=0$.
2. If a track has no update at the current time step, then $p=p+1$.
3. If $p > 50$, then it is considered to have lost target and downgraded to untracked state 0.
4. Similarly, if $e > 5$, then the quality of the current track has become too low, it will also be downgraded to state 0.

In lost state -1:

1. All tracks in state -1 will no longer receive updates and are subject to removal.

All tracks are estimated path for the target, the primary purpose of the state transition is to separate the low-quality and high-quality tracks. Tracks with different quality should be treated differently, because some early stage low quality track may compete with the high-quality track for inputs, which often results in false track detection. The proposed state manage system solves such issue.

A weighting factor could also be considered at the tracked and untracked tracks. The tracked track should have a higher chance than the untracked track for input candidates. The factor could be based on the similarity between the estimated target and the input. However, such factor is not implemented in this paper.

In track management, the assignment problem is of vital importance.

4.3.1 Hungarian algorithm

The Hungarian algorithm [16] is a combinatorial optimization algorithm that solves the assignment problems. It has two ways to formulate the problem: as a matrix or as a bigrah. In this paper, the matrix formulation is discussed.

Assume the following radar tracking problem: consider there are two radar frames contain target detections, there are M targets in the previous frame A and N targets in current frame B. Find the most likely matching that the M targets are associated with the N targets, so that targets can be tracked. To simplify the case, assume M=N.

To qualify the most likelihood matching, a matching score table needs to be calculated. Each element in the table describes how likely a target in A is matched to a target in B. Similar to the single target tracking; the distance between the two targets is chosen to be the table element. The closer two targets are in the two frames, the more likely they are the same target. Total cost E is all the distances from all targets added together, it is the total assignment cost for current frame B.

The assignment problem can be formalized as:

Given a $N \times N$ matrix of cost $\{X_{ij}\}$, determine a $N \times N$ perturbation matrix M_{ai} that minimizes the total cost E:

$$\text{minimize: } E = \sum_{i=1}^N \sum_{j=1}^N M_{ij} X_{ij} \quad \text{Eq. 4.2}$$

$$\text{subject to: } \forall i \sum_{j=1}^J M_{ij} = 1, \forall j \sum_{i=1}^I M_{ij} = 1, M_{ij} \in \{0,1\} \quad \text{Eq. 4.3}$$

The matrix formulation if the Hungarian algorithm describes several steps of operations that performed on the cost matrix $\{X_{ij}\}$, so that the total cost E can be minimized when the algorithm terminates. Figure 27 illustrates the algorithm overview. The algorithm is implemented in Matlab. Since step 2 and step 4 have more complicated processes, they are described separately.

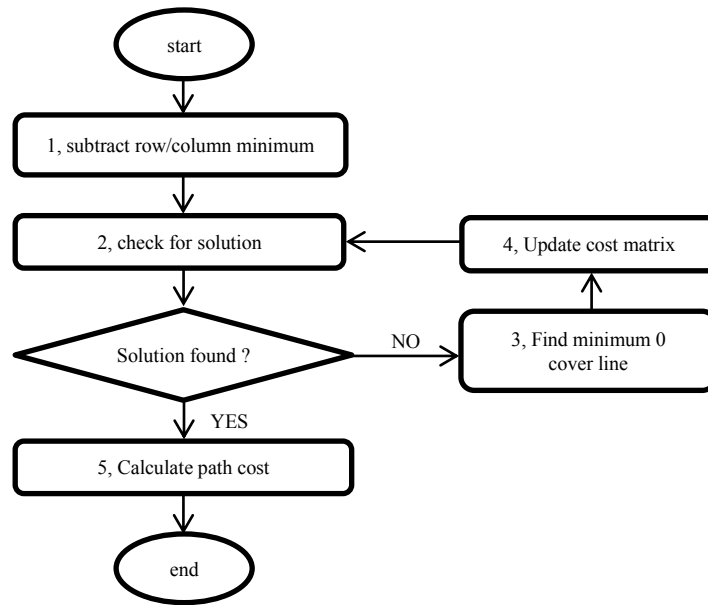


Figure 27: Hungarian algorithm flow chart

Step 1; for each row, find the minimum element of each row, and subtract it from that row. For each column, find the minimum element and subtract it from that column. Now each row and column has at least one zero

Step 2; decide if a solution can be found using the existing zeros. If yes, the solution is then found. It has already been proven that after step 1, the solution does not change. The detailed algorithm to check if the cost matrix has a solution is described later.

Step 3; if a solution cannot be found, find the minimal number of lines that can cover all the existing zeros.

Step 4; if the number of total cover lines is less than the number of rows (or columns); find the minimum element of all elements that are not covered by any lines. All elements that are not covered by any line need to subtract the minimum element. All elements that are covered by both row line and column line need to add the minimum element. If the number of total cover lines equals the number of rows (columns), there exists a solution from all the existing zeros in the cost matrix. Then it goes to step 2 to check if a solution can be found.

Step 5; if a solution can be found, the algorithm terminates. The path cost is then calculated.

The detailed algorithm of step 2:

```
while finish = 0:  
    if there is only one 0 in each row, then  
        replace the 0 as -2;  
        replace all 0s in the same column as -1;  
    end  
    if there is only one 0 in each column, then  
        replace the 0 as -2;  
        replace all 0s in the same row as -1;  
    end  
    if there is no more 0s in current matrix, then  
        solution is found, finish = 1;  
    else  
        if more than one row has more than one 0, and more than one column has more  
            than one 0. Then it's not possible for the iteration steps to replace all 0s,  
            manual intervene is therefore required.  
            find the row or column with least number of 0s;  
            replace the first 0 in that row/column with -1, and continue iteration;  
        end  
    end  
end
```


The detailed algorithm of step 3 is shown as following, the list_r, and list_c contains the row number and the column number that corresponds to the lines that cover all 0s in the cost matrix.

execute step 2 to replace all prime 0s to -2, and replace normal 0s to -1;

mark the rows that does not contain -2 in list_r;

while finish = 0:

for each row in list_r,

 find all -1s, and find the -2 in the same column;

 save the column number of the found -1 in list_c;

 save the row number of the found -2 in list_r;

end

if all -1s in the cost matrix is processed, **then**

 Finish = 1;

end

end

reverse select all rows that are not marked, save in list_r;

The detailed algorithm of step 4 is shown as following,

find the minimum element that is not covered by the 0 cover line;

all elements that are not covered by any line need to subtract the minimum element;

all elements that are covered by both row line and column line need to add the minimum element;

However, the above mentioned implementation of the Hungarian algorithm does not guarantee a successful assignment when N is a large number. The root cause lies in step 2, when enough 0s in the cost matrix is formed, an assignment always exists, but current implementation does not guarantee the solution can be found.

4.3.2 Multi-track management

Multi-track management is the core of the multi-target tracker. Figure 28 below shows the eight main steps for the track management in one radar measurement frame. A detailed explanation of each step is provided later.

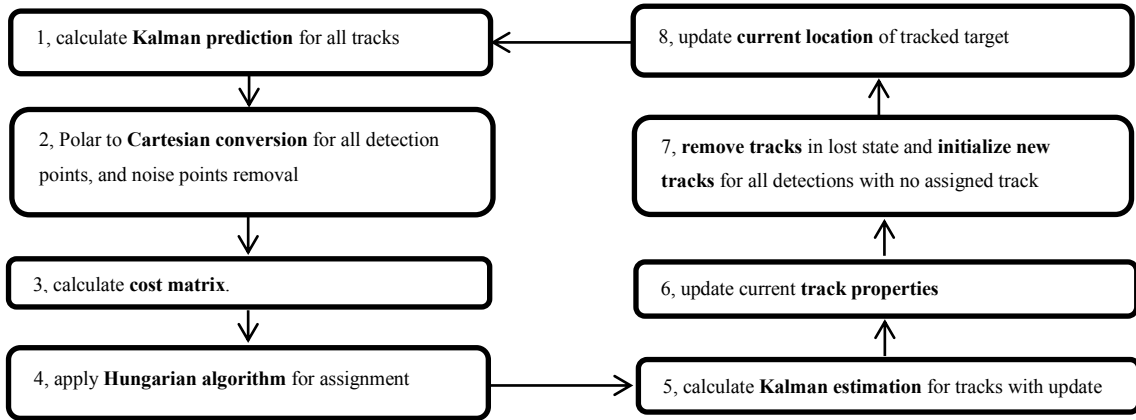


Figure 28: Single target tracker work flow

Step 1; use the Kalman filter to find the next predicted track location.

Step 2; extract the valid target detection points from the current radar frame, and then convert their coordinates from polar to Cartesian. The default format of the target detection is polar coordinates with the Doppler shift. They are converted to Cartesian coordinates and used as the Kalman filter input.

Step 3; to get the cost matrix, the distance matrix between detected target points and predicted track locations is calculated first. The Euclidian distance is calculated between each detection (row) and estimation (column). To ease the assignment process, if the distance value is larger than the predefined gate value, the distance value is set to a higher value.

The distance matrix may contain new detection points that do not fit into any existing tracks. These detection points will not participate in the following association process. Also, existing tracks may not have any valid detection inputs, and then these tracks will not take part in the association process. Thus it is a good practice to slim down the distance matrix into a smaller cost matrix beforehand.

Before using the Hungarian associator to calculate optimal assignments, the cost matrix needs to be made into a square matrix. If the number of target detections (rows) is larger than the number of tracks (columns), extra columns with maximum cost value are added. If the target detection number is smaller than the number of tracks, then extra dummy targets with maximum cost value are added.

Step 4; apply Hungarian associator to get the solution matrix for all existing tracks

Step 5; calculate the Kalman estimations for all tracks with detection points according to the solution matrix. The solution matrix may contain invalid assignments to indicate the current track has no update input.

Step 6; update the above mentioned track properties, if a track has no detection update, it will use the prediction as input, and suffer a penalty.

Step 7; remove any tracks that do not meet predefined criteria. If the penalty of a track is too high, or the track consistently has low quality detection updates. The track needs to be terminated. In the meantime, for all the remaining detection points that are not associated with any track, a new track is initialized for each one.

Step 8; update the current location of all tracked objects, and then continue to the next radar frame.

In this paper, the multi-target tracker is developed to separate the walking pattern between different people. So that later feature extraction can be applied to each pattern.

4.4 Tracking Results

This section describes the tracking results of some tests in chapter 2.3. The test 1.1 is used to test the single target tracker. Tests 2.1 to 2.4 are used to test the performance of the multi-target tracker.

At the end of the tracking, all state 0 tracks are ignored and removed.

Single target tracking results,

Choosing the appropriate parameter is crucial for successful tracking. The following parameters for each processing block are chosen:

Threshold, $Th1 = 1e2$;

Clutter removal, $NR = 2$;

Local maxima detector, $g = 5$;

CFAR, $w = 5, h = 15, Pfa = 1e-8$;

Single tracker, $gate = 5$;

Figure 29 shows the single target tracking result of test 1.1. The distance D in this test is around 25m. x and y do not reflect the real scale in the distance. With the increase of distance y , the number of detection points is decreasing. In the end, the tracker is able to pick out the complete track correctly.

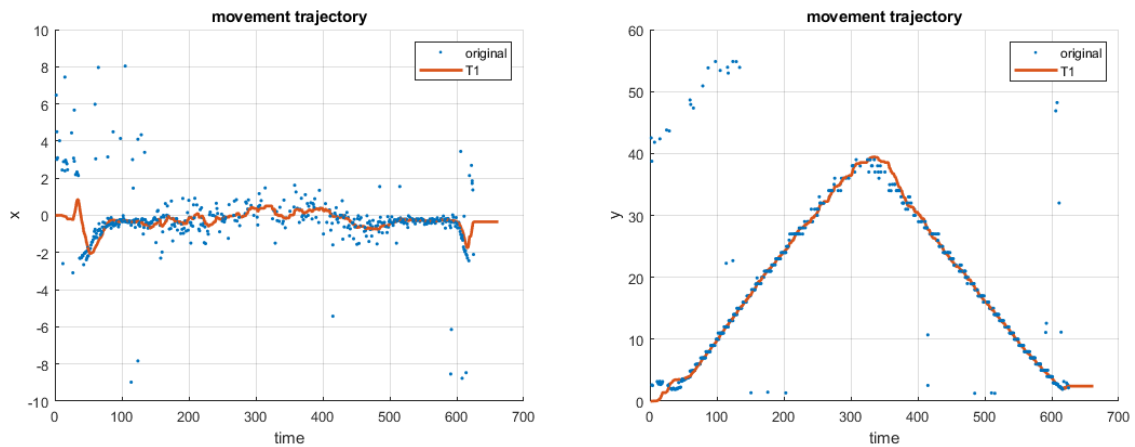


Figure 29: movement trajectory of test 1.1

Two targets tracking results,

The following parameters for each processing block are chosen:

Threshold, $Th1 = 3e2$;

Clutter removal, $NR = 2$;

Local maxima detector, $g = 5$;

CFAR, $w = 5, h = 15, Pfa = 1e-8$;

Multi tracker, $gate = 5, penalty = 50, error = 3$;

Figure 30 shows the multi-target tracking result of test 2.1. The two tracks T1 and T2 can be clearly seen. Since the range-Doppler profile of the two targets does not

interfere with each other, more valid detection points are present. Therefore both tracks have enough update candidates for successful tracking.

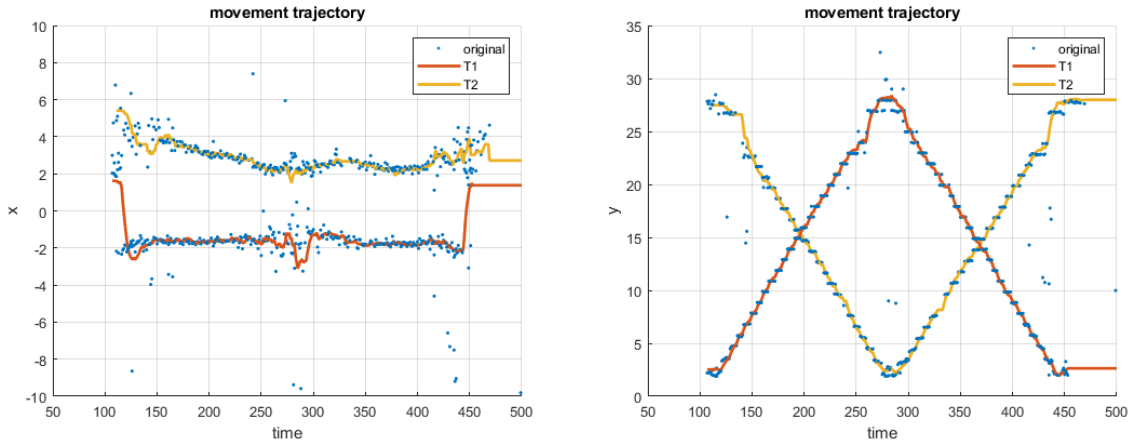


Figure 30: movement trajectory of test 2.1

Figure 31 shows the multi-target tracking result of test 2.2. Since the two targets are interfering with each other, even with angle separation, the detection points still have a relatively large deviation, causing the instability of the tracks. However, the multi-target tracker is still able to correctly recognize the two tracks T1 and T2.

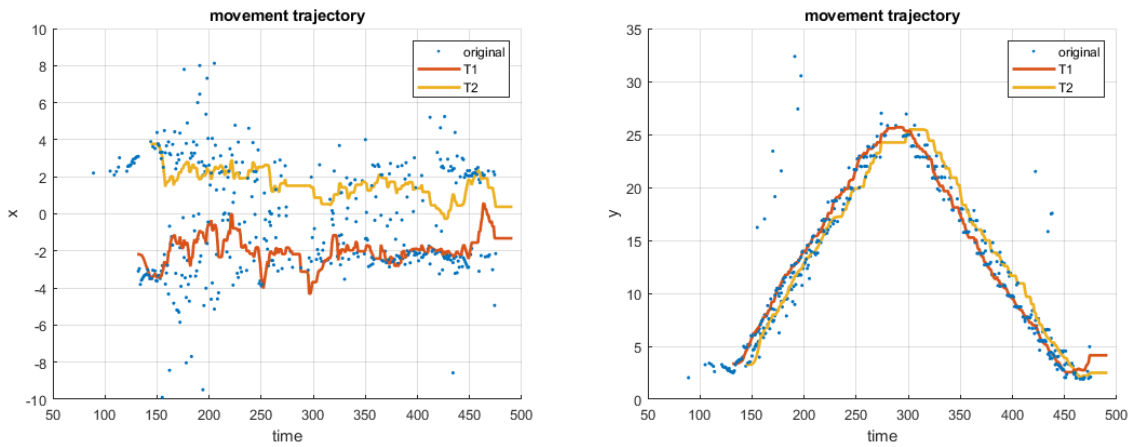


Figure 31: movement trajectory of test 2.2

Figure 32 shows the multi tracking result of test 2.3. As can be seen, the two tracks T1 and T2 have been correctly recognized. Even they are very close at around time frame 260, there are still no interferences.

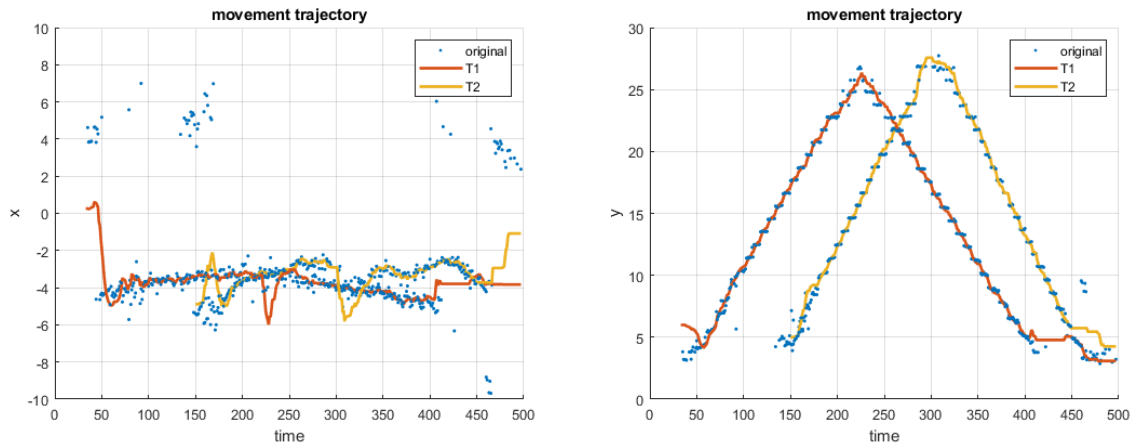


Figure 32: movement trajectory of test 2.3

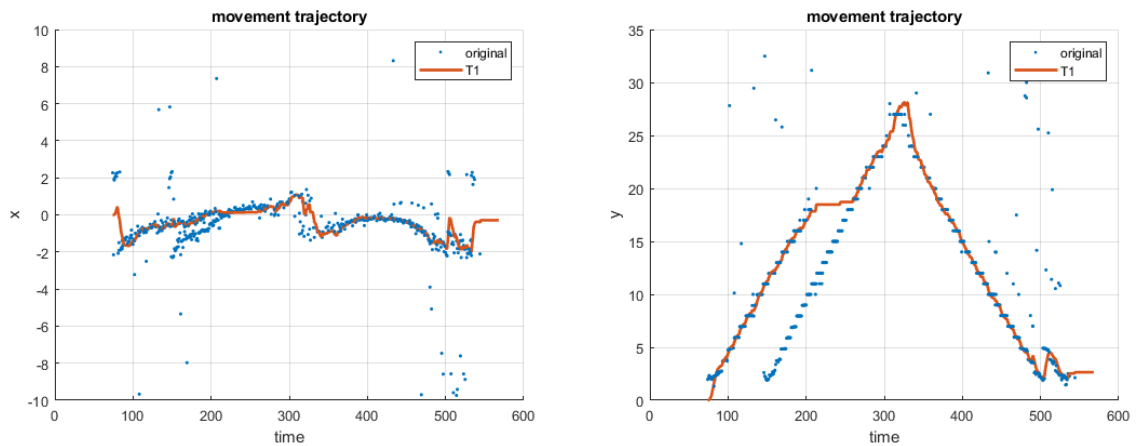


Figure 33: movement trajectory of test 2.4

Figure 33 shows the multi tracking result of test 2.4. This time only one false track T1 is shown. As can be seen from the right image, the track of person A started fine until it reaches time frame 200, when no more detection of A can be found. This is because the radar signal is blocked by person B. When person A returns, the detection appears again

while the signal of B is blocked. Therefore the middle part (around frame 250 to 300) of the track T1 belongs to person B.

To recognize the track in test 2.4 correctly, the simple assignment with the closest detection is not enough. The similarity of the different tracks needs to be taken into account. To achieve this, the signature of the target can be established and used as the weight for the track assignment process; this signature is discussed in chapter 5.

In conclusion, the performance of the tracker depends on the quality of the valid detection points. Too many detection points will overwhelm the tracker and result in more unwanted tracks. Too few detections will result in lost tracks. In situations when targets can interfere with each other or block each other's signals, false target detection or missed detection often occur. This will significantly affect the success of target tracking.

There are some additional notes on the tracking range of the radar. In single target tracking, the angle separation is not necessary. The signal is more focused and clear, even in long range tracking. The CFAR detector can perform well in these cases and is recommended. However, in multi-target tracking, different objects can interfere with each other, The CFAR detector often does not have enough energy to declare detection due to the angle separation, and causes miss detection in long range tracking. Therefore the multi-target tracking range is often short compared to single target tracking. To increase the performance, the CFAR evaluation can be switched off in close range tracking situations.

5 Walking Pattern Recognition

From the research of V. C. Chen [17], the pattern of human walking can be seen as periodic. For a successful walking recognition, the walking movement pattern must be modeled first. In this chapter, a simplified pattern model of human walking is presented. Several features of such walking pattern are also defined. Based on the tracking results from chapter 4, the walking pattern and the define features are extracted over time from the recording files of test case 1 in chapter 2.3. These extracted features are then compared together to find a simple classification method to separate the human walking pattern and non-human movement pattern. The following Figure 34 shows the overview of the complete signal processing blocks of the human walking pattern recognition.

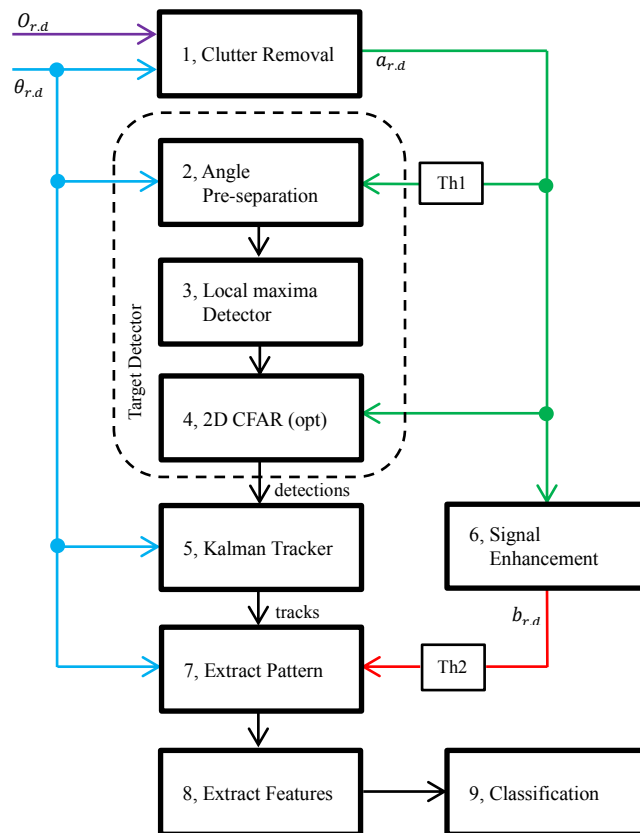


Figure 34: signal processing blocks overview

In the complete signal processing, a two pass processing technique is used in the target detection and pattern extraction. The targets is first detected and tracked in each radar frame, then the corresponding pattern and features are extracted from the enhanced radar frame. In this way, the details of the enhanced frames are mostly preserved, while the noises in the enhanced frames are ignored.

At first, the raw data image $O_{r,d}$ is sent to block 1 for clutter removal. Block 1 also uses the angle information to clear some fixed position noise points in Cartesian coordinate system. After the clutter removal, the processed image $a_{r,d}$ (applied with threshold Th1) is sent to block 2 for angle pre-separation. For single target cases, the angle separation is optional. Then the image data of different angles are sent to block 3 for target detection. Then block 4 uses image $a_{r,d}$ to evaluate the detections from block 3. However, if the target is far away, the target signal becomes weak, and detections become rare. The block 4 in this case can be turned off to give better result. Afterwards the target detector sends all detections to tracker in block 5 to get estimated target location.

Then block 7 extract the pattern of the target from the enhanced signal (applied with threshold Th2) provided by block 6 and the target track estimation provided by block 5. With the extracted pattern data, the block 8 can extract the walking features that can be used in block 9 for simple classification.

This chapter focuses on the signal processing block 7, 8 and 9.

5.1 Pattern Definition and Extraction

Different targets have different detection patterns on the range-Doppler plots. Generally speaking, the pattern of human walking has wide micro-Doppler spread and less range spread. The pattern of moving vehicle usually has wide range spread but less Doppler spread, since vehicle is longer and the velocity of all parts is unanimous. The difference in the micro-Doppler pattern can be used to separate rigid moving objects and human beings.

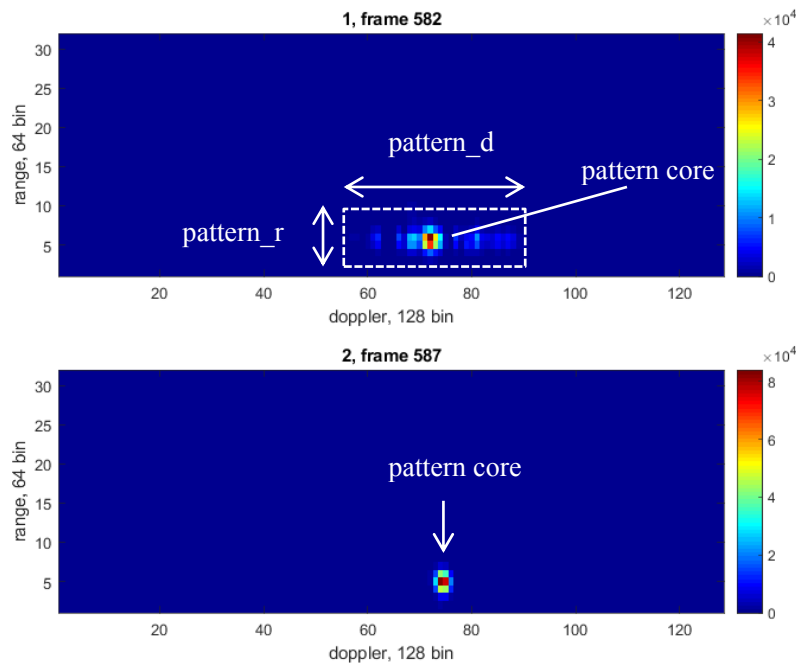


Figure 35: walking patterns

However in practice the Doppler spread of a walking person is not always more noticeable than range spread. Figure 35 shows two walking patterns in test 1.1. The pattern consists of two parts: the main pattern core and the micro-Doppler pattern. The above frame 582 shows the typical characteristics of human walking pattern. The pattern core spans around Doppler bin 70, range profile spans around bin 5. The micro-Doppler pattern spans from Doppler bin 60 to 90, which indicate that the person has moving parts (arms and legs) that reached high radial velocity. Frame 587 also shows walking pattern of the same person. In this frame, only the core pattern is shown, no more micro-Doppler

pattern can be found. This indicates that the limb of the person stay relative still to the radar.

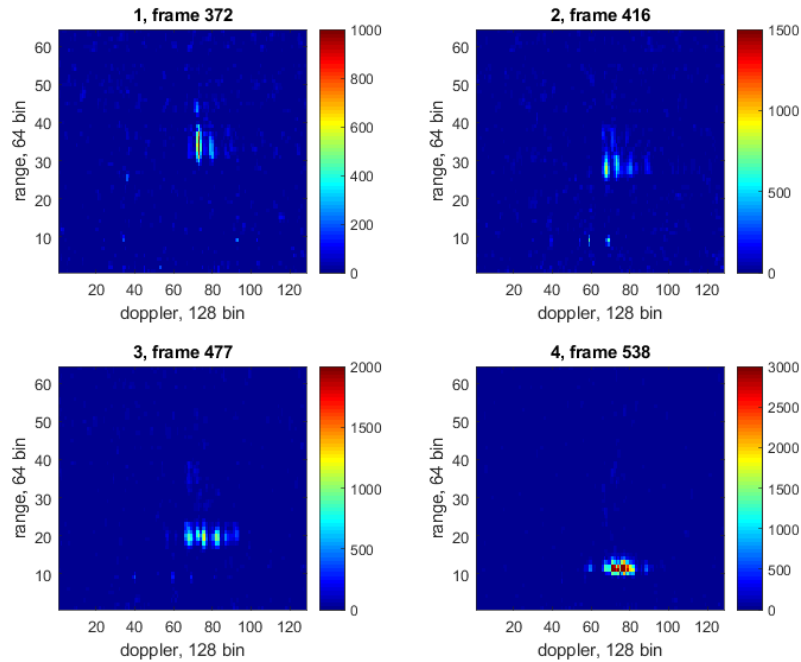


Figure 36: typical walking frames at different distances

Figure 36 shows four more raw frames in test 1.1. Each frame shows the pattern at a different range, with frame 372 at furthest distance and frame 538 at closest distance. The scale of the image is adjusted accordingly to show the details of the pattern. In the micro-Doppler patterns that are clearly visible on all frames, the range span of the pattern also increases when range increases. In frame 372, the target range spans from 30 to 40, larger than the other 3 frames. The reason of the increase in range spread is unclear; it indicates that the range spread of targets from the Infineon radar frame cannot be utilized directly without correction.

Both cases listed in Figure 35 and Figure 36 has shown that differentiate targets with only micro-Doppler spread and range spread may not be sufficient, because the walking pattern may exhibit different characteristics in different situations. A more thorough approach is required here.

In Figure 35 frame 582 up, a method to extract the walking pattern is proposed. First, the tracked local maxima points in each frame are chosen to be the pattern core.

Then the surrounding area including pattern core are extracted as pattern. The shape of this pattern depends on the micro-Doppler pattern of the target. It needs to cover most of the pattern in most frames, while stay as small as possible to ignore most noises. For walking pattern, a rectangular shape with width `pattern_d` and height `pattern_r` is chosen. `pattern_d` is usually larger than `pattern_r`. To set the value of `pattern_d` and `pattern_r`, some measurements are made as follows:

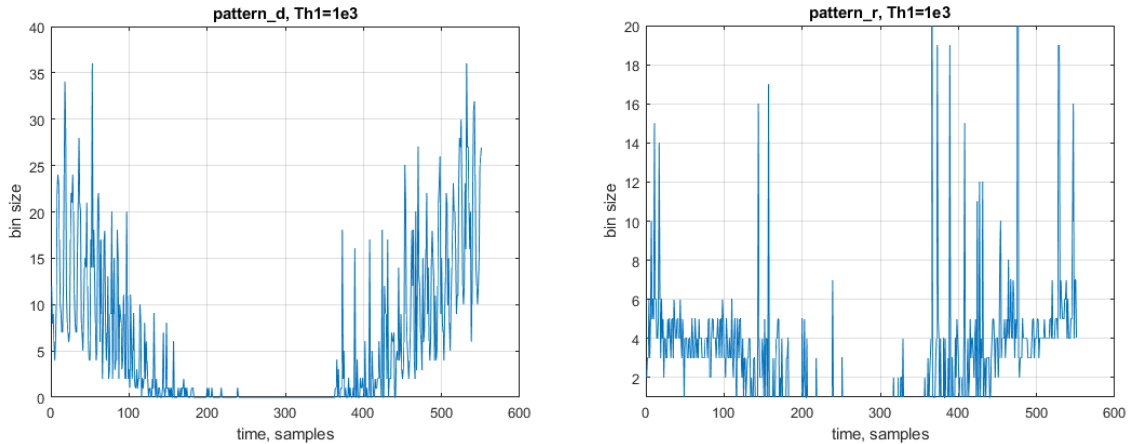


Figure 37: pattern shape measurement of test 1.1

In Figure 37, the `pattern_d` and `pattern_r` of each frame in test 1.1 are measured with threshold `Th1=1e3`. In the left image, when `pattern_d=30`, most of the walking pattern can be covered. And on the right image, apart from some irregular shape, if `pattern_r=7`, most of the pattern are covered. Thus the `pattern_d=30`, `pattern_r=7` is a good estimation of the rectangular shape so that most micro-Doppler signature are confined within. The original radar frames are used in this measurement because the enhanced frames are too noisy.

With the defined pattern shape, all data within this shape can be extracted as pattern sequence for further processing. Since block 7 in Figure 34 extracts all walking pattern from the enhanced signal with the track provided by the tracker, the angle information is also required. Because only the micro-Doppler pattern close to the main target is extracted. To separate targets with similar range-Doppler profile, another parameter `pattern_a=5` is proposed. If the angle of the evaluated pattern point is within 5 degree with the pattern core, it is considered to be a part of the pattern. Figure 38 shows a sliced

3D view of an extracted pattern sequences from 10 frames in test 1.1. In some frames, the extracted micro-Doppler pattern clusters to the side of the pattern core are clearly visible.

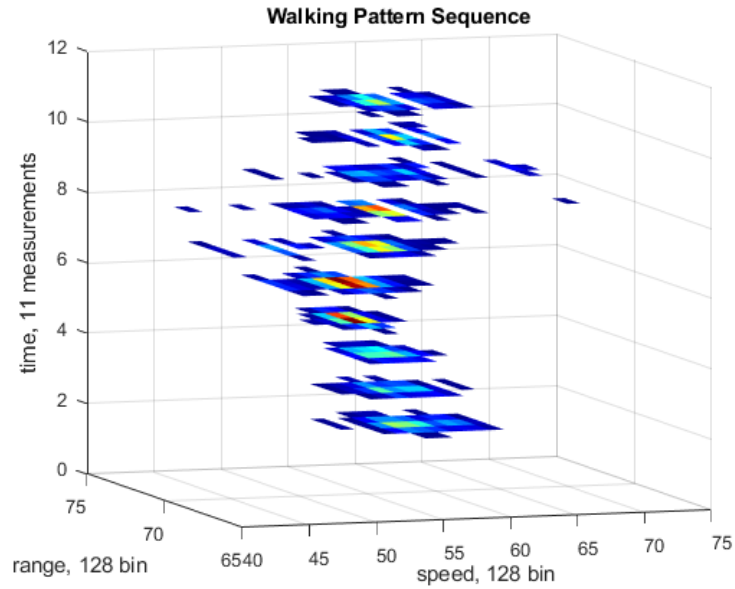


Figure 38: Walking pattern sequences

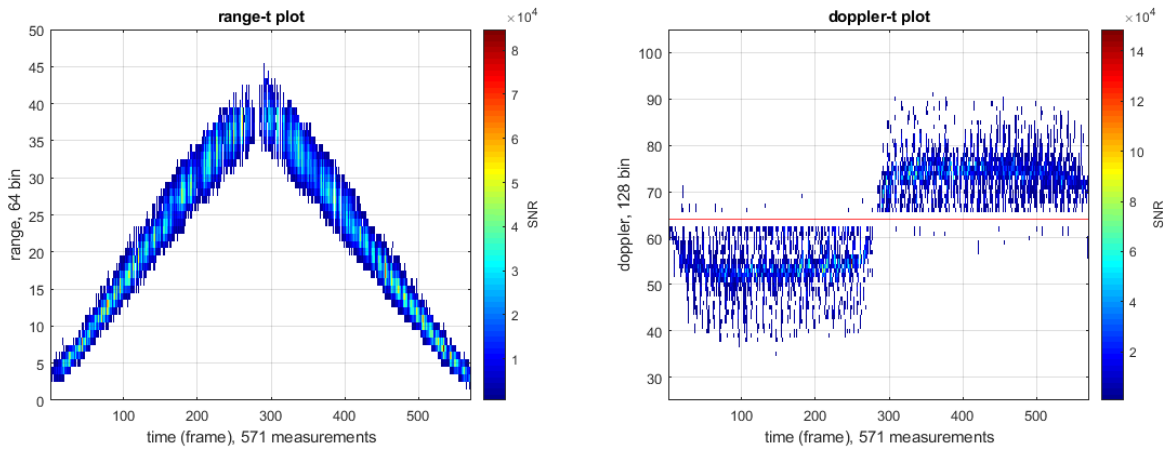


Figure 39: range-t and Doppler-t of the extracted pattern from test 1.1

Figure 39 shows the result of the extracted pattern sequence from enhanced radar frames of test 1.1. The pattern shape is set as $\text{pattern}_d=30$, $\text{pattern}_r=7$, and $\text{pattern}_a=5$. The threshold is chosen as $\text{Th}_2=8e4$. Blank measurements at the beginning and the end are removed. Compared with Figure 13 and Figure 16, almost all clutter noises are filtered out. Please note that in the range-t plot, with the increase of range, the range span also increases. This corresponds with the findings in Figure 36. It however does not show

clearly in Figure 37 pattern_r with unenhanced radar image. Therefore the choice of pattern_r is set more or less experimentally. With this result, a more thorough investigation of the pattern can be conducted.

5.2 Feature Definition and Extraction

With extracted pattern sequences, features of human walking can be extracted. In Figure 40, a set of features are proposed on the frame 582 from Figure 35. The tracked 2D local maxima detection is considered as the center of pattern core. These features are described as following:

minDS (Minimum Doppler Spread): the Doppler bin minimum

maxDS (Maximum Doppler Spread): the Doppler bin maximum

NDS (Negative Doppler Spread): the maximal difference between minDS and core

PDS (Positive Doppler Spread): the maximal difference between maxDS and core

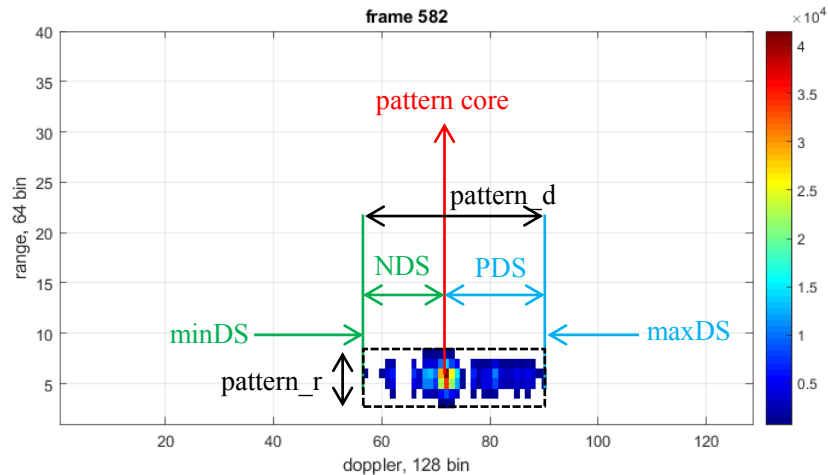


Figure 40: Proposed feature set on range-Doppler frame

In the Doppler-t plot from Figure 39, the human walking pattern can be viewed as periodic. Figure 41 shows the partial data frame 500 to 550 from Figure 39. The pattern corresponds to human walking towards the radar; therefore most patterns are above the Doppler middle bin 64. The main pattern core is jumping around Doppler bin 75 in zigzag pattern. This is because the person is constantly in accelerating and decelerating

state. By carefully measuring the pattern core, the acceleration rate and the deceleration rate could be obtained.

Even though the pattern sequence in Figure 41 seems periodic, it roughly contains 6 main peaks (Peak1 to Peak6). However, the number of actual cycles in Figure 41 depends on actual human movement. Normally, two neighboring peaks are the micro-Doppler pattern caused by different hand/leg. In this case, Peak1, Peak3 and Peak5 come from the same hand/leg, Peak2, Peak4 and Peak6 come from the other hand/leg. If one cycle is defined as the time of hand/leg moving to the same place with the same velocity (relative to the person), then Peak1 and Peak2 forms one cycle, Peak3 and Peak4 forms another cycle and so on. So there are 3 cycles in Figure 41. In practice the two peaks in each cycle may not always be correctly detected. In this paper, the even more simplified pattern cycle is used, which is defined as the half cycle between each peak. In the meantime, the main walking velocity is defined as the mean pattern core Doppler value of each pattern cycle.

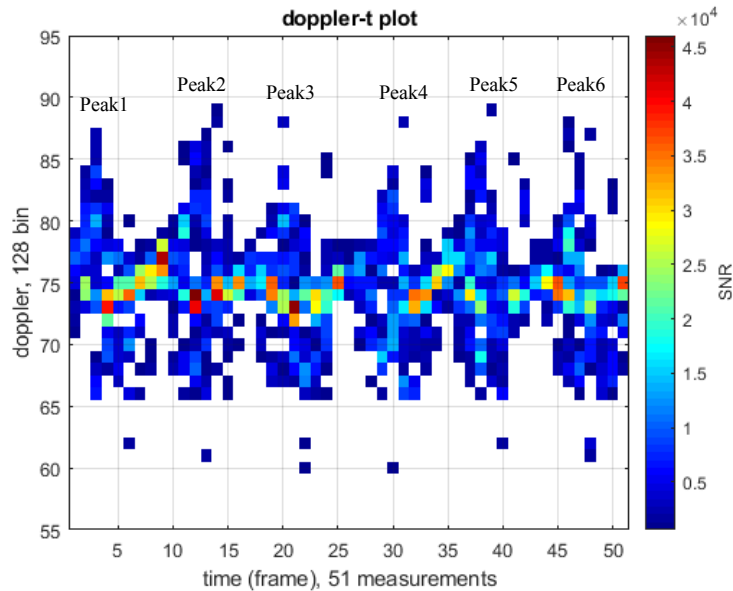


Figure 41: Doppler-t plot of walking

Figure 42 shows the simplified model of the Doppler-t plot in Figure 41. The whole pattern sequence is simplified to 3 sub-sequences: maxDS, pattern core and minDS. The maxDS plot is the maximal Doppler bin of all frames. The focuses of maxDS are the peaks. There are 4 example peaks shown: P1 to P4. These peaks correlate to the maximal

micro-Doppler pattern, which indicates the limb of the human is moving at maximal radial velocity. The time frame of the peaks and the value of such peaks can be measured as (P1, PDS). The maxDS_cycle is defined as the interval between each maxDS peaks.

Similarly, the minDS plot is the minimal Doppler bin of all frames. For minDS, the troughs are of interests. Four troughs are shown: T1 to T4. They represents the limb of the human is moving at maximal radial velocity in another direction. The minDS_cycle is defined as the interval between each minDS trough.

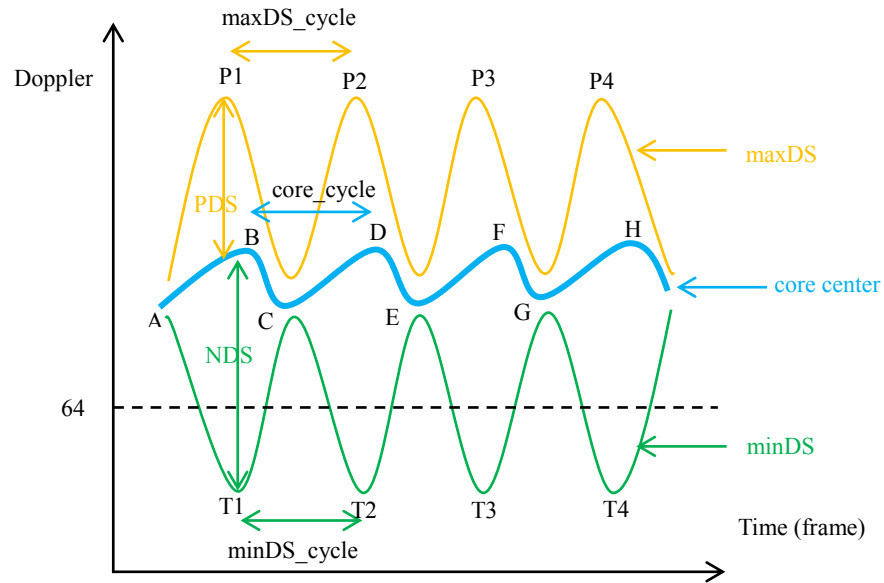


Figure 42: Proposed feature set on Doppler-t plot (positive velocity)

The pattern core center is the tracked target detection. Both peaks and troughs are of interests in this case (B, D, E, and H are peaks, A, C, E, and G are troughs). While walking towards the radar, each up ramp (A to B, C to D ...) represents the person is accelerating. The deceleration path (B to C, D to E...) is also important. However, in practice the velocity drop of the person often result in weak detection points. The deceleration path often does not show up very clearly. The core_cycle is defined as the interval between each core center peaks (in positive velocity above bin 64). If the person is walking away from the radar, the core_cycle is defined as the interval between each core center troughs (in negative velocity below bin 64). If the core center jumps around bin 64, the core_cycle measured may not be accurate.

These proposed features can be extracted easily from saved recordings using the local maxima detection technique in chapter 3.3.2. However, to extract all the proposed features live, several extra steps are involved. Below describes the algorithm to extract all features from processed radar patterns, these features include:

1. Pattern cycles: core_cycle, max_doppler_cycle, min_doppler_cycle
2. Average walking radial velocity: walk_mean_v
3. NDS and PDS
4. Walking acceleration: walk_acceleration

The algorithm of live feature extraction is described as:

```

for each pattern image do:
    get the tracked detection as pattern core;
    sum up range bin;
    find maxDS and minDS;
    for each core peak/trough do:
        save peak/trough;
        calculate walk_mean_v;
        calculate core_cycle;
        calculate walk_acceleration;
    end
    for each maxDS peak do:
        save maxDS peak;
        calculate max_doppler_cycle;
        calculate PDS;
    end
    for each minDS trough do:
        save minDS peak;
        calculate min_doppler_cycle;
        calculate NDS;
    end
end

```

} online core peak/trough extraction
} online maxDS peak extraction
} online minDS trough extraction

The online peak/trough extraction is different from the offline version described in chapter 3.3.2, because live data stream presents data one by one instead of all at once. To find the peak/trough of live pattern sequences, a temporary FIFO buffer is used. A certain number of sequences are stored inside the buffer so that the peak/trough can be extracted.

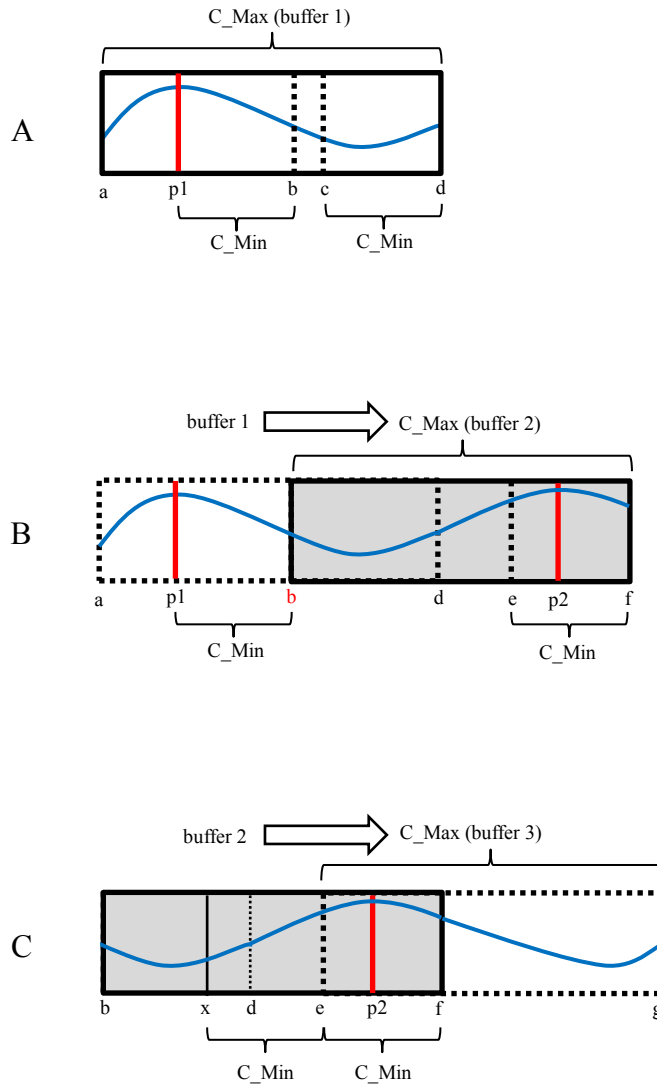


Figure 43: peak extraction overview

Figure 43 shows the overview of the detailed maxDS peak extraction process. The data stream is marked in blue, while the peaks are marked with a red vertical line.

In Figure 43 A, when the buffer of size C_Max is filled (as buffer 1), then a process to find the first peak $p1$ is started: for each element inside buffer 1 from a to c , if an element is larger than C_Min elements after itself, then it is considered to be a peak element. If a peak $p1$ is found ($p1$ is larger than all elements between $p1$ and b), then $p1$ is returned so that all elements between a and b ($b = p1 + C_Min$) are deleted from the buffer. Afterwards the new data elements are again added at the back of the buffer (new data as d to f in Figure 43 B).

In Figure 43 B, when the buffer is full again (as buffer 2). The same process starts. Although this time there is no peak elements between b and e , the first element b can be falsely recognized as a peak if it is larger than all C_Min elements after it. To solve this problem, the ramp of the data stream is considered. The peak check process is only performed on the up ramp data stream. The down ramp data stream is ignored for peak finding.

In Figure 43 C, if no peak is found in the current buffer, an adjusted x is returned, $x = C_Max - C_Min \times 2$. So that all elements between b and e are deleted from the buffer. Then the peak extraction restarts when f to g is filled with new data. Also, to keep track of the distance between each peaks ($p1$ and $p2$ in this case), the number of buffer that has no peaks must be noted.

After the peak extraction, the PDS also needs to be extracted. To calculate the PDS, the pattern core center is required. However, the core value buffer from the online core peak/trough extraction cannot be used directly, because they are in a different pace than the maxDS peak extraction. Therefore a second FIFO buffer is used to store the core center value in sync with the maxDS buffer, so that the PDS can be calculated.

For minDS trough and NDS extraction, the procedure is similar as maxDS with small variations, instead of finding the largest element, the smallest element is searched in the buffer. Also the trough checking process is only performed on the down ramp data stream.

Below is the detailed pseudo code for the online maxDS peak extraction, the pseudo code for the minDS trough extraction and the core peak/trough extraction are similar, and thus not repeated:

Set C_Max as the maximum size of FIFO buffer buf and C_Min as the minimum peak distance;

```

for each incoming sequence  $S_f$  do:
    push  $S_f$  into the back of buffer  $buf$ ;
    if  $sizeof(buf) = C\_Max$ ,  $buf$  is full, then
        for each element  $buf_x$  in  $buf$ , with  $x=0$  to  $C\_Max-C\_Min-1$ 
            if  $buf_x$  is in down ramp, then
                continue to next element in  $buf_x$ ;
            end
            check  $C\_Min$  elements  $buf_{x+i}$  after  $buf_x$ , with  $i=0$  to  $C\_Min$ 
            if  $buf_{x+i}$  is the largest, then
                set  $peak = i$ ;
                break;
            end
        end
        if peak is not found in  $buf$ , then
            set  $peak = C\_Max - C\_Min \times 2 - 1$ ;
        else
            save peak as cycle;
            calculate and save PSS;
        end
        update ramp direction for  $buf_{peak+C\_Min}$ ;
        remove all elements before  $buf_{peak+C\_Min}$  from  $buf$ ;
    end
end

```

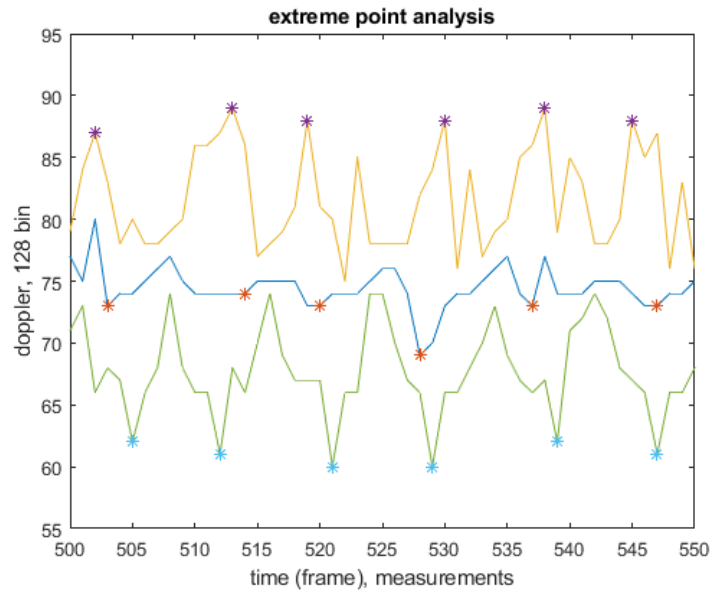


Figure 44: Extreme point analysis of Figure 41

Figure 44 shows the extreme point analysis of Figure 41 (frame 500 to 550) with the setting of $C_Max=16$, $C_Min=4$. Each peak of $maxDS$ is marked as stars, each trough of $minDS$ and pattern core are also marked in stars. Then the differences between each peaks/troughs are calculated as the three cycles. The difference between each peak/trough and the pattern core is calculated as PDS and NDS .

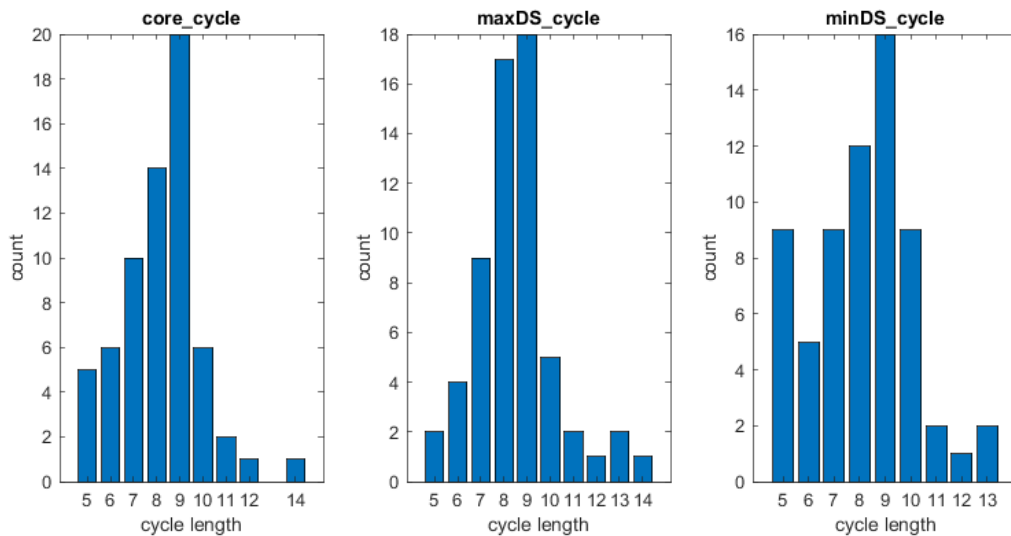


Figure 45: histogram of the three cycle length

Figure 45 shows the histograms of the `core_cycle`, `maxDS_cycle` and `minDS_cycle` of the complete dataset from test 1.1 ($Th1 = 5e2$, $Th2 = 3e5$). One cycle equals to one measurement frame $T=60ms$. The mean values of these three cycles are 8.17 (`core_cycle`), 8.51 (`maxDS_cycle`) and 8.12 (`minDS_cycle`). In the `core_cycle` plot, cycle length 9 has the largest count number. In the `maxDS_cycle` plot, cycle length 9 also has the most counts. In the `minDS_cycle` plot, cycle length 9 still has the most counts. Of all these three histograms, cycle length 8 and 9 account for the majority of the cycle counts. This indicates that the cycle length 8 and 9 are good estimations for the walking pattern in test 1.1, the equivalent time is about $8 \times T = 480ms$, which corresponds to the study of gait step time in table 2 from Koo and Lee's research [18].

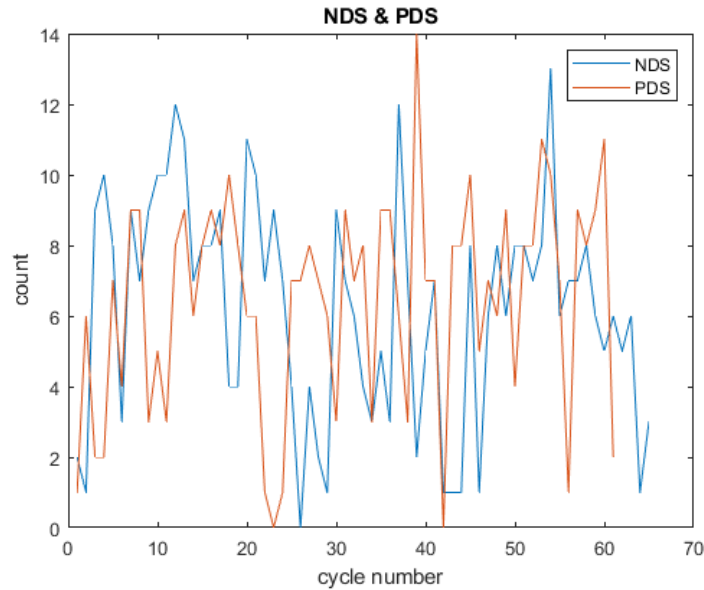


Figure 46: NDS and PDS of test 1.1

Figure 46 shows the extracted NDS and PDS of test 1.1. They vary between 0 and 16 with variance as $\text{var}(\text{PDS})=9.4820$ and $\text{var}(\text{NDS})=10.1841$. With the chosen `pattern_d` and `pattern_r`, the NDS and PDS are within expected range. There are no significant differences between NDS and PDS, although PDS seems a little smaller in this case, thus in later sections only the PDS is compared between tests. To calculate NDS and PDS, the $Th2$ must be carefully adjusted so that the pattern stays well within the extraction shape. If $Th2$ is set too high, the micro-Doppler pattern will be significantly reduced, if set too low, the NDS and PDS will likely be capped by the extraction shape.

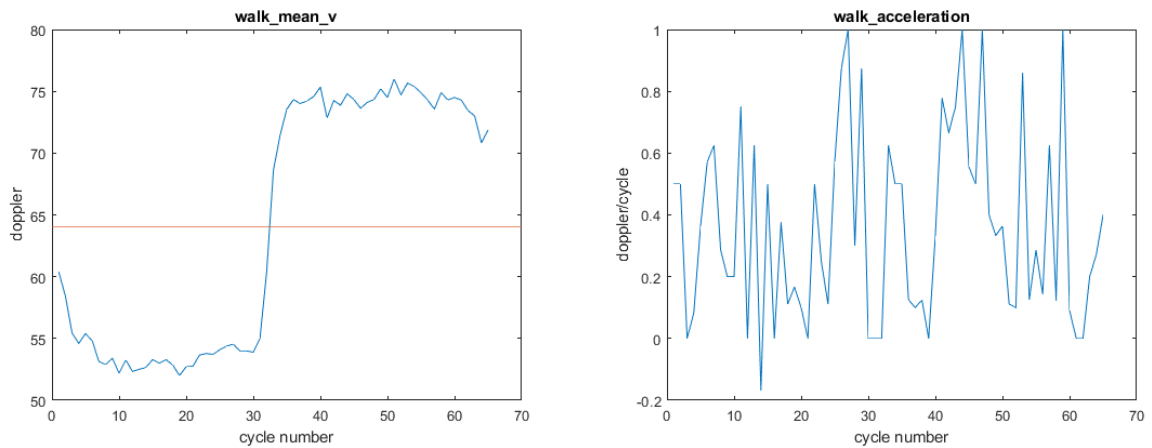


Figure 47: mean radial Doppler profile (left), acceleration (right) of test 1.1

Figure 47 image to the left shows the mean walking radial Doppler profile of test 1.1. The middle line at bin 64 indicates radial velocity is zero. Thus the actual relative radial velocity can be calculated as: $D_{act} = 64 - \text{walk_mean_v}$. The direction of the person walking is the sign of D_{act} . With positive D_{act} , the person is walking towards the radar. With negative D_{act} , the person is walking away from the radar. From around cycle 10 to 30 and cycle 40 to 60, the person is walking in rather stable velocity. The D_{act} in both cases are comparable, which corresponds to the consistency of human walking. In the image to the right walk_acceleration plot, the acceleration profile of each cycle varies between -0.2 and 1. Since no obvious relationship is found between the calculated profile and the walking gait, it is no longer discussed in later sections.

The proposed features of walking are successfully extracted for test 1.1. To be able to correctly recognize the walking pattern, some other different types of walking patterns are compared in the section below.

5.3 Classification

For people counting application, the correctly detection of walking pattern is the key. With the features of the moving pattern extracted, a simply classification can be applied to estimate if the target is a walking human.

The structure of this section is arranged as follows, the Doppler-t and the range-t plot of the moving pattern are first presented, and then the features of each pattern are compared and discussed together. Totally 4 types of moving patterns are compared.

The following parameters are used for the processing blocks:

Threshold, $Th_1 = 5e2$, $Th_2 = 1e5$; Clutter removal, $NR = 2$;
 Local maxima detector, $g = 5$; CFAR, $w = 15$, $h = 5$, $Pfa = 1e-8$;
 Single tracker, $gate = 5$; Pattern extraction, $pattern_d=30$, $pattern_r=7$;
 Feature extraction, $C_Max=14$, $C_Min=4$;

Test	core_cycle	maxDS_cycle	minDS_cycle
Test 1.1	8.1061	8.1194	8.0735
Test 1.2	7.5556	7.3191	7.4255
Test 1.3	8.3800	8.7174	9.2500
Test 1.4	8.0435	8.7857	8.4750
Test 1.5	6.5667	6.8929	6.4545
Test 1.6	8.2642	8.0189	8.3269
Test 1.7	8.4000	8.0357	8.6346
Test 1.8	6.8621	6.9808	6.8571
Test 1.9	8.7143	6.7719	7.2340
Test 1.10	8.3676	8.0896	7.3788

Table 4: mean value of three cycle length of test 1.1-1.10

Table 4 lists the mean value of the 3 main cycle length, it is discussed later.

Compare Set 1: test 1.2-1.5, 1.10

In this set, there are 5 types of patterns are compared. The purpose is to show the proposed features of different type of movement pattern of the same person. The range-t and Doppler-t plots are provided in Figure 48 to Figure 52:

Test 1.2, normal walking longitudinal:

This test is similar as test 1.1 outdoor.

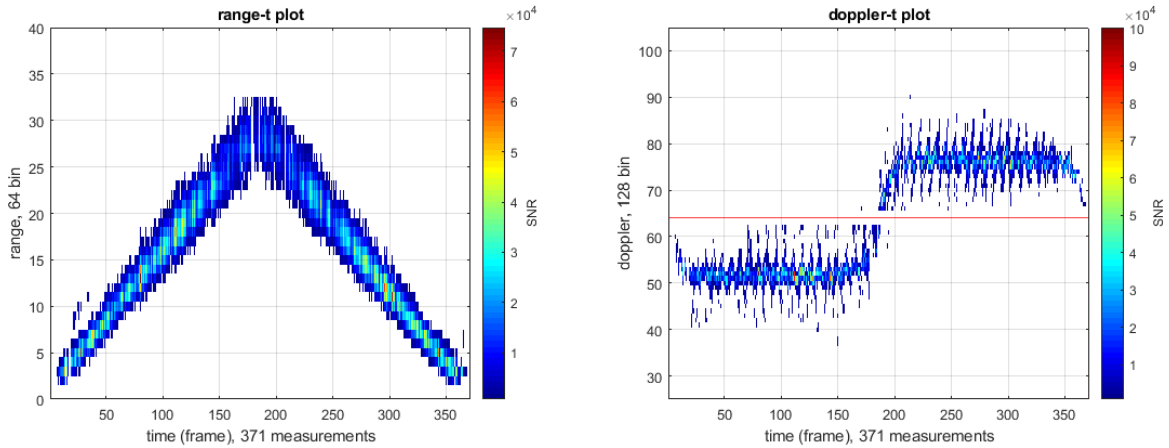


Figure 48: range-t and Doppler-t of test 1.2

Test 1.3, slower walking:

Compared with test 1.2, this test takes slightly longer; it has 446 frames (larger than the 370 of test 1.1).

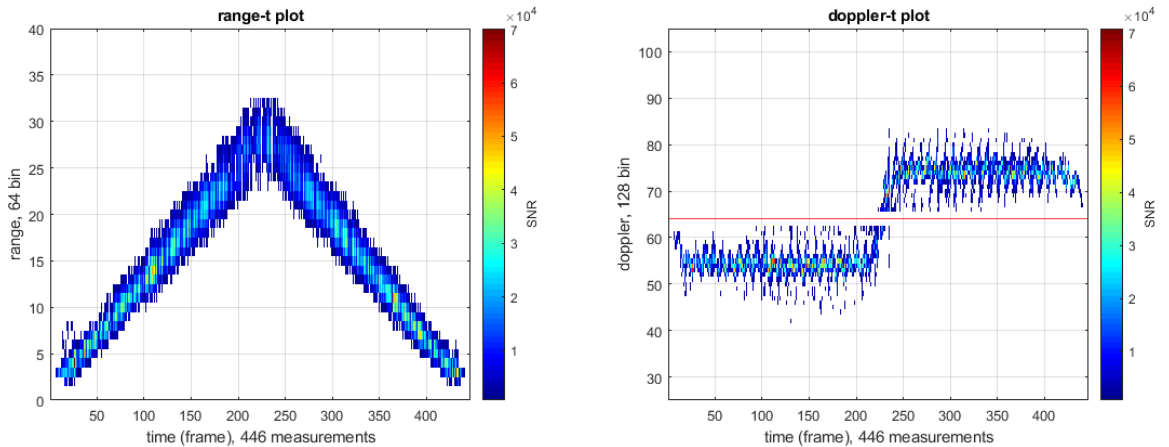


Figure 49: range-t and Doppler-t of test 1.3

Test 1.4 is walking casually with hands in pockets:

The Doppler-t plot of this test is very similar to test 1.1; a detailed comparison is discussed later.

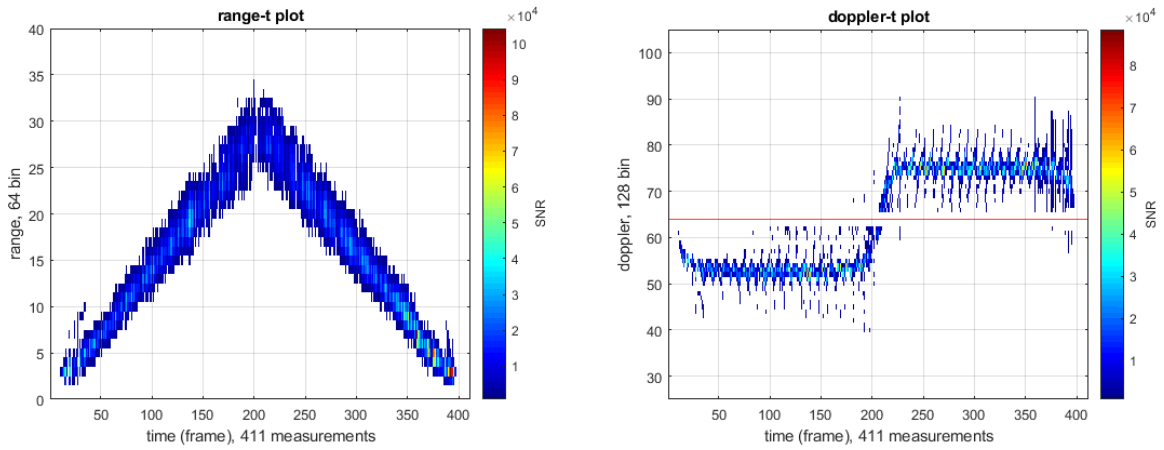


Figure 50: range-t and Doppler-t of test 1.4

Test 1.5 is running:

This test take even less time, and the Doppler profile is larger than test 1.1, since the person is faster.

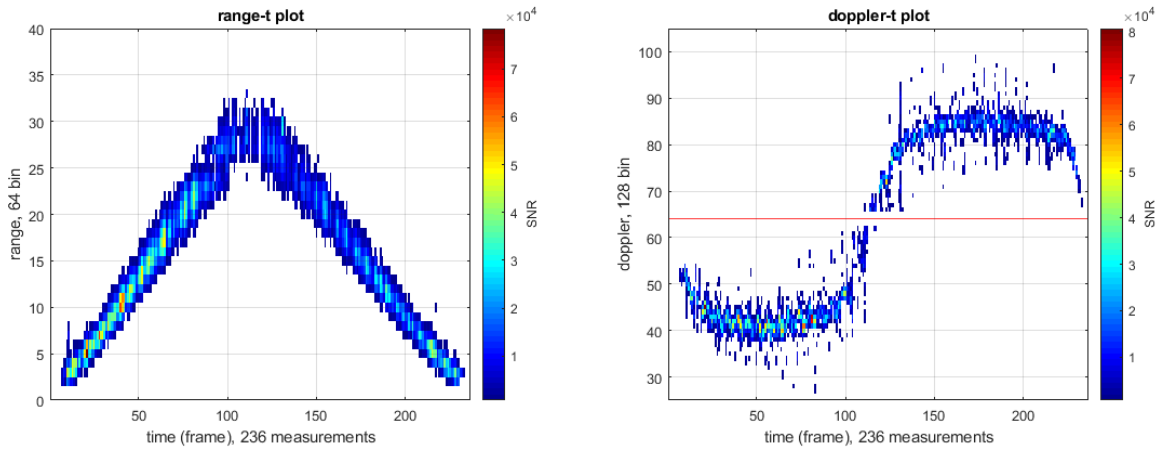


Figure 51: range-t and Doppler-t of test 1.5

Test 1.10 is casual walking lateral

The modeled walking pattern does not show clearly in the Doppler-t plot.

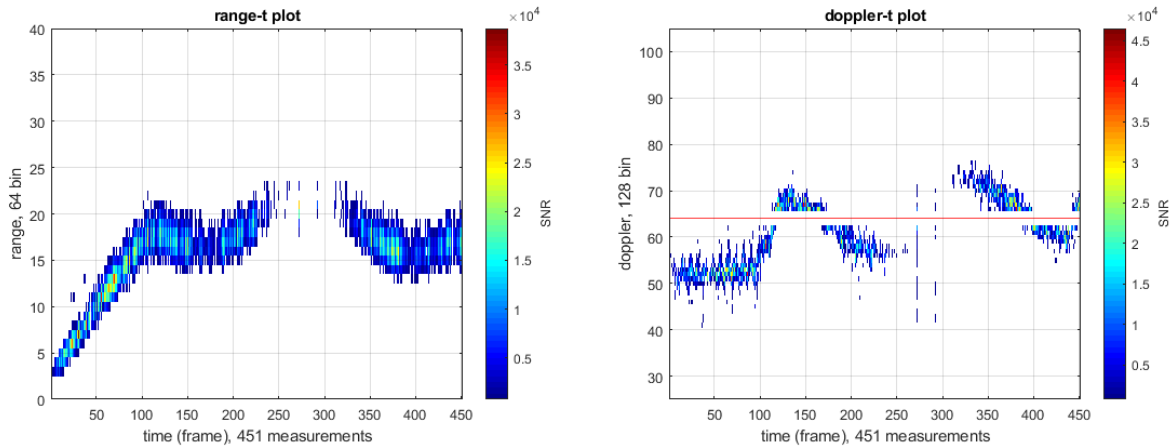


Figure 52: range-t and Doppler-t of test 1.10

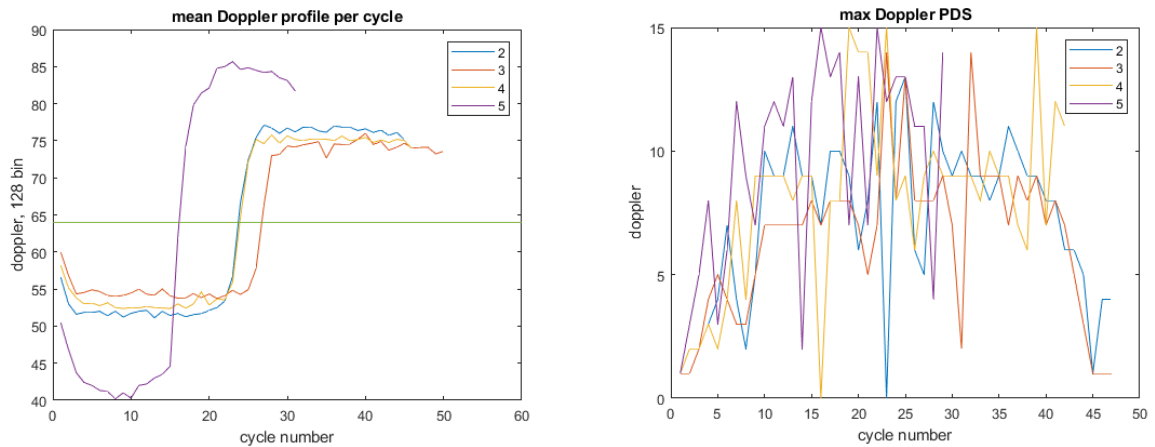


Figure 53: mean Doppler profile and PDS of test 1.2-1.5

Figure 53 left shows the mean Doppler profile of each detected cycle of test 1.2 to 1.5. The middle bin 64 marks the 0 radial velocities. Each test has different number of cycles. In test 1.5, the subject is running, which corresponds to the largest Doppler bin value (relative to bin 64). In test 1.3 and 1.4, the subject is moving slower than in test 1.2.

In the right image, the difference of the max Doppler PDS of test 1.2 to 1.5 cannot be clearly seen. They all range from 0 to 17, most of them always stay above 6 bin. However, in test 1.4, the subject has both hands in pockets, the caused PDS is still

comparably larger as the other tests, this means that both hand swing and leg swing contributes to the micro-Doppler walking pattern.

The mean cycle length of test 1.2 to 1.5 can be found in table 4. For walking patterns, the three cycles are considered to be relatively similar. In test 1.3 and 1.4, the mean cycles are larger than test 1.2, which indicates the cycle length is longer and the person is moving slower than test 1.2. In test 1.5, the person is running, therefore it has the smallest mean cycle length.

Compare Set 2: test 1.2, 1.6, 1.7

In this set, the patterns of three different people are compared. Since the Doppler-t and the range-t plot are very similar in this case, they are omitted here.

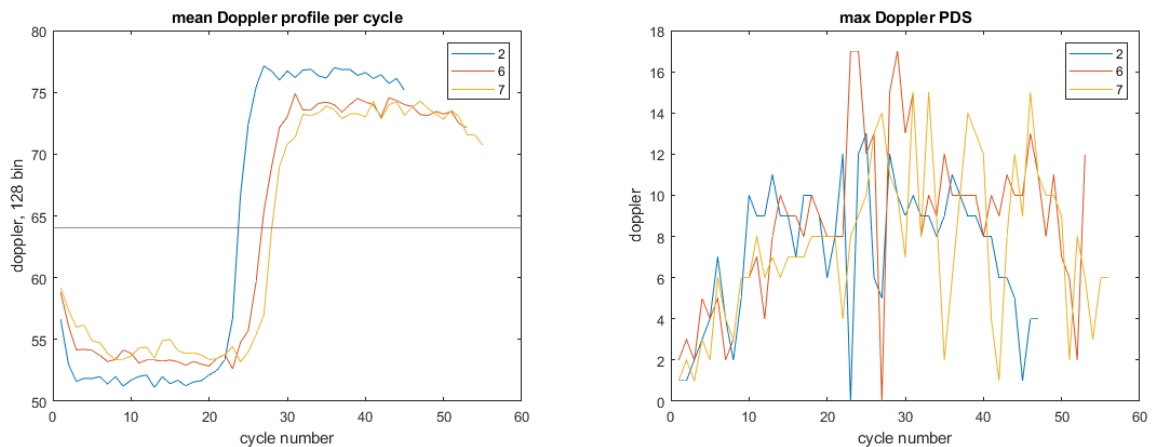


Figure 54: mean Doppler profile and PDS of test 1.2, 1.6, 1.7

As can be seen in Figure 54 left image, person B in test 1.6 and person C in test 1.7 has similar Doppler profile, while person A in test 1.2 has a relatively higher radial velocity. This is confirmed in the cycle length in table 4, where the three main cycles are comparable for test 1.6 and 1.7, and larger than 1.2.

In the right image, similar to the compare set 1, most of the cycles have a noticeable PDS above bin 6.

Compare Set 3: test 1.2, 1.8, 1.9

In this set, three movement patterns are compared. Test 1.2 shows the normal walking pattern of an adult. In test 1.8, the pattern of a child is shown in Figure 55. In test 1.9, a moving cart tray is shown in Figure 56.

Test 1.8 is a child walking casually

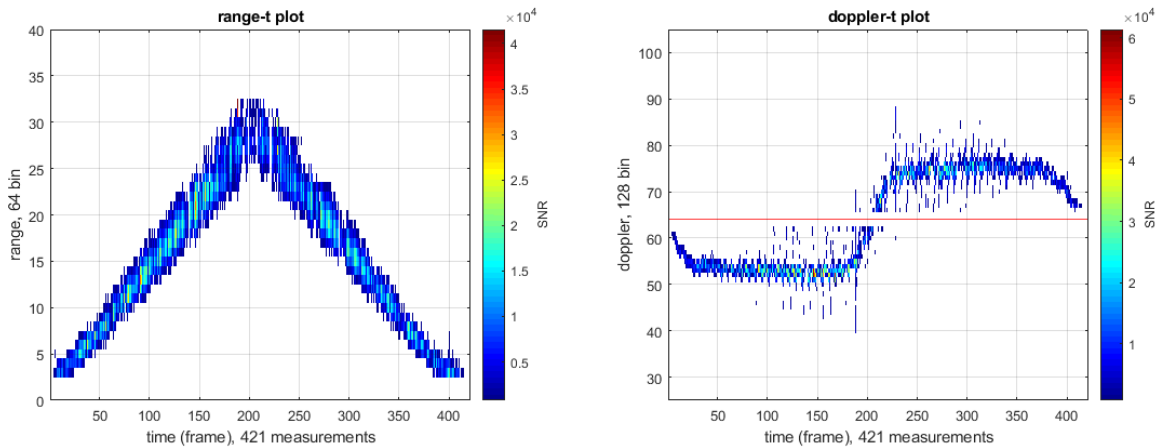


Figure 55: range-t and Doppler-t of test 1.8

Test 1.9 is a moving cart tray in a lab

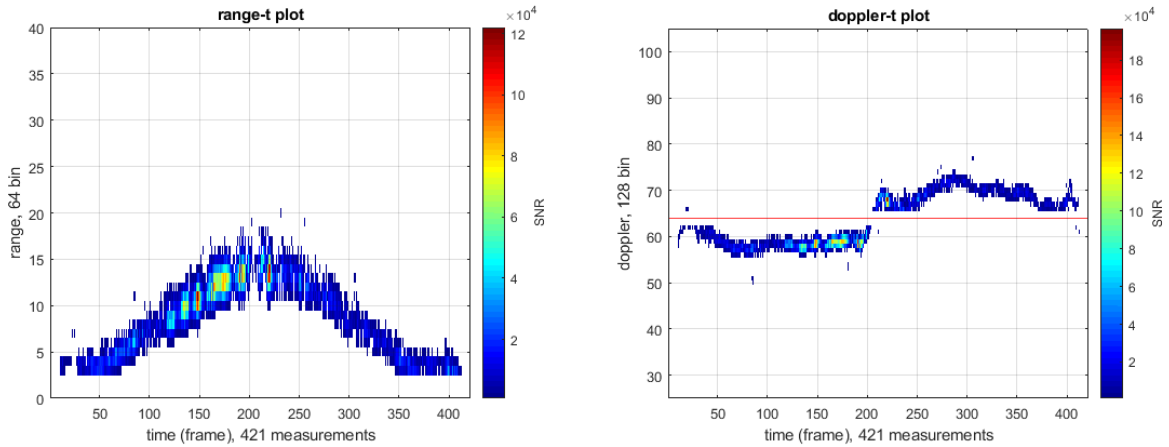


Figure 56: range-t and Doppler-t of test 1.9

Compared with normal sized adults, child usually has smaller body size with shorter hand and leg. Therefore the signal for child in range-Doppler plot should be weaker. In Figure 55, the walking pattern of the child D seems thinner than test 1.2 with adult. In Figure 57 left, the radial profile of the child is comparable to person A in test 1.2. Also, test 1.8 in table 4 indicates the pattern of a child has a shorter cycle length. This corresponds with the sense that shorter hand and leg requires faster movement for the same walking speed. In Figure 57 right, the PDS of the child in test 1.8 seems to be smaller than the adult in test 1.2. This also correlates to the fact that child usually has a smaller of hand/leg swing amplitude.

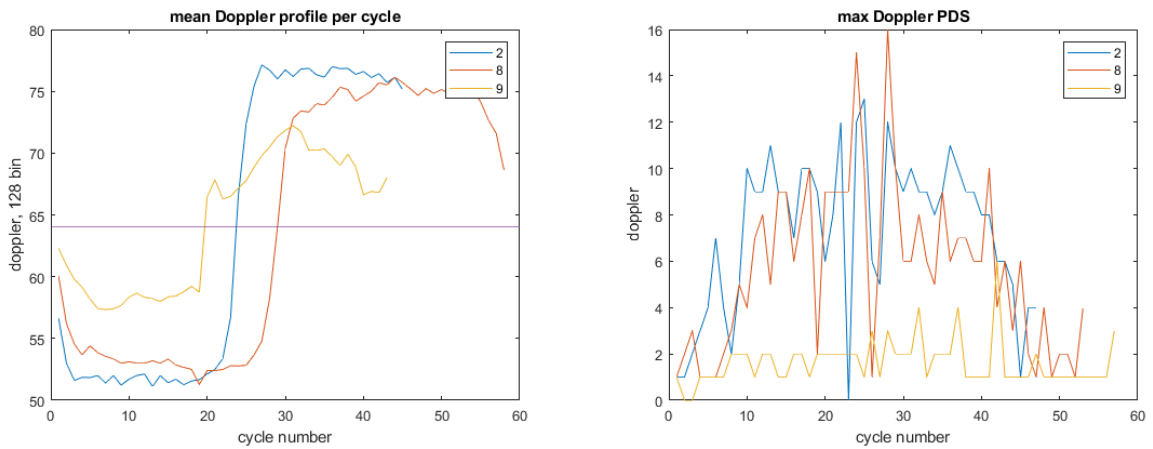


Figure 57: mean Doppler profile and PDS of test 1.2, 1.8, 1.9

Compared with normal human walking, the moving pattern of rigid object is completely different, shown in Figure 56. The pattern is no longer cyclic anymore, and the micro-Doppler pattern is not very clearly shown. In Figure 57, the PDS of test 1.9 stays below 3 for most of the time. And in the left image, the mean Doppler profile is not stable compared to normal walking.

In summary, to detection a walking human, the most intuitive indicator would be the radial Doppler profile. The PDS/NDS could also be two important indicators as they measure the micro-Doppler pattern directly. The cycle length could be used to further estimate whether the person is running or not. However, to further distinguish between different scenarios between casual walking, running, and different body size. All of the features must be utilized.

6 Software System Implementation

To apply the human walking recognition process live, a simple radar client system is developed. The whole system consists of mainly two parts: recording tool and live tracking tool. The recording tools is developed to save radar images from each session and used for later Matlab analysis. The live analysis tool can be used to track the targets and display the extracted features of walking pattern. In this chapter, the design and implementation of such PC side software system is discussed. This radar system is merely a proof of concept implementation of the tracking and recognition process discussed in this thesis and therefore not a main focus.

Figure 58 shows the radar system setup. The antenna in the middle of the radar should be placed to a similar height to the target to obtain the optimal return signal. This setup is used in both recording and live tracking. In live tracking however, the webcam is removed.

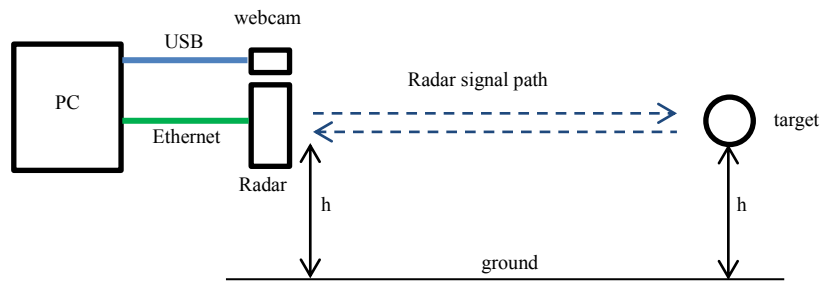


Figure 58: Radar recording/tracking setup

6.1 Radar data recording tool

The independent tool named “radar_viewer” is developed in C and C++ to save recordings from each measurement test. It displays the radar signal and the picture of the field of interest at the same time. It also has the capability to save the radar data and the pictures for later analysis. Although “radar_viewer” is mostly implemented in the earlier research work, in this paper, however, a few extra functions are added for convenience

purposes. Here the “radar_viewer” is only capable of radar session recording; the recording playback function is moved to another tool named “radar_play_record”, thus no longer supported in the “radar_viewer”.

Figure 59 displays the flow chart of the “radar_viewer”. After the initialization, it receives the data packets using libpcap API [19] and extract raw ADC samples from two antennas. Then the 2D FFT are performed on the ADC samples to produce the radar image using FFTW library [20]. The radar images are displayed on the UI with the help of SDL library [21]. If session recording saving is required, the pcap packets are saved to file. In the meantime, the photos from webcam are saved to separate file with timestamps as name using opencv [22].

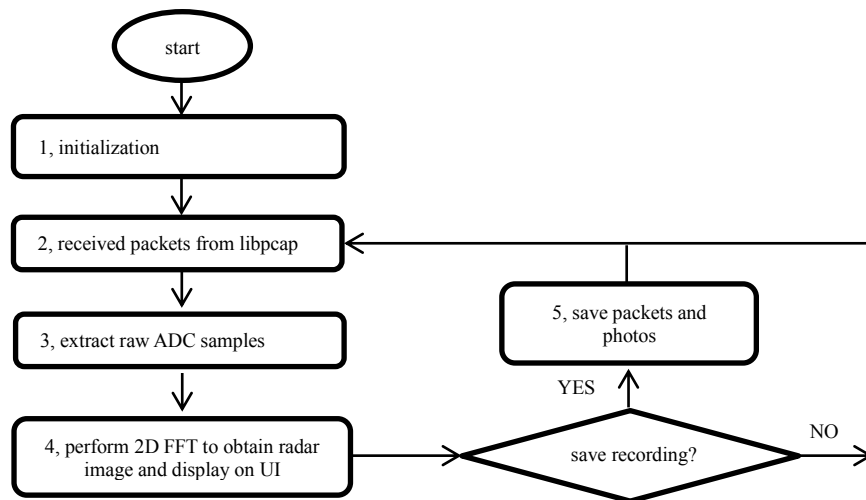


Figure 59: radar_viewer flow chart

The “radar_viewer” now has 3 operation modes: viewing (visualize the radar return signal), recording (save radar return image), paused. To ease the switch between each mode, the key presses are used. Below describes some of the main requirements of the “radar_viewer”.

- 1, the default mode is viewing mode.
- 2, in viewing mode, the processed frame and the corresponding camera picture are displayed on the window.

3, in recording mode, the radar frames and photos from webcams are not only displayed on the windows, but also saved as files.

4, in paused mode, the system continues to receive data packets but no frame processing is done, no data is saved. The window shows the last frame prior to the 'p' key press.

5, in viewing mode, if 'r' key is pressed, the system changes to recording mode. If 'p' key is pressed, the system changes to paused mode.

6, in recording mode, if 'f' key is pressed, the system changes to viewing mode. If 'p' key is pressed, the system changes to paused mode.

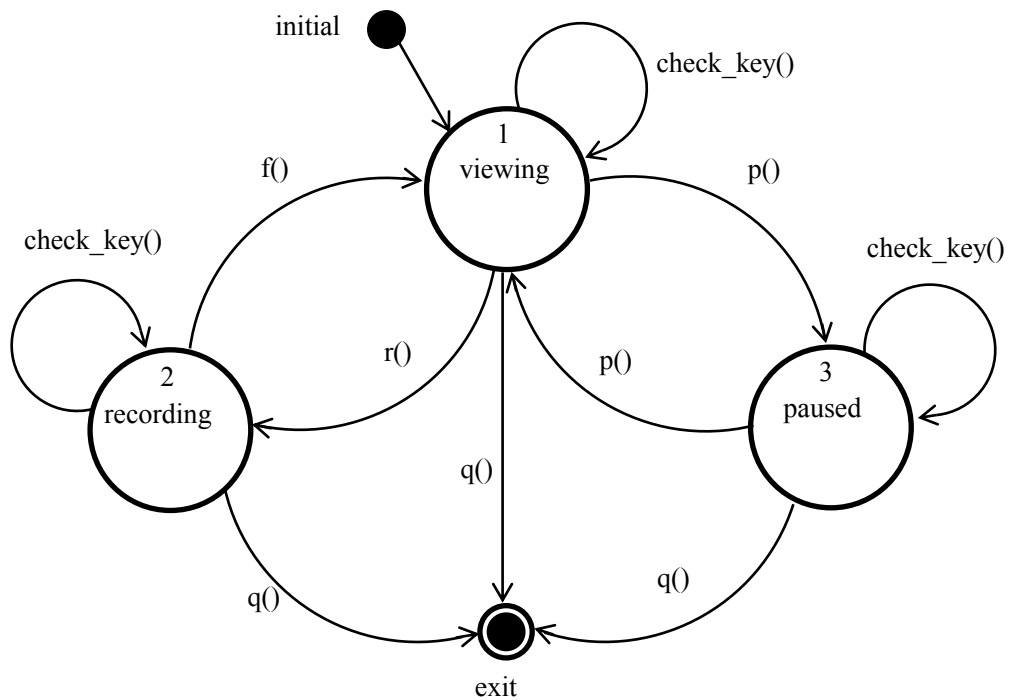


Figure 60: radar_viewer state diagram

In Figure 60, the state diagram of the “radar_viewer” is drawn according to the requirements. The system has 3 states: viewing, recording and paused. The default state of the system is viewing state. The transition between each state is initiated by key press events.

In viewing state, the system captures data packets and calculates each frame, then displays the frame on the spectrogram window. After which it checks SDL key press events. If 'r' key (record) is pressed, the state is transited to recording state. If 'p' key

(pause) is pressed, the state is transited to paused state. If ‘q’ key (quit) is pressed, the state is transfer to final state and exit.

In recording state, apart from displaying the frame in spectrogram window, the system also saves the data packets and the corresponding picture of the field of interest at the same time. Then it checks SDL key press events. If ‘f’ key is pressed, the state is transited to recording state. If ‘f’ (finish) key is pressed, the state is transited back to viewing state. If ‘q’ key is pressed, the state is transited to final state and exit. However, recording state could not be paused; the transition between recording state and paused state is not defined.

In paused state, no data packet processing is done; it checks SDL key press events for an un-paused key press event ‘p’, which will transit it to viewing state. If a ‘q’ key press event is captured, it will transfer to final state and exit.

Figure 61 shows the UI interface of the “radar_viewer”. Left panel shows the range-Doppler plot, range-t plot and the Doppler spectrogram of the target. The right panel shows the photo in front of the radar module.

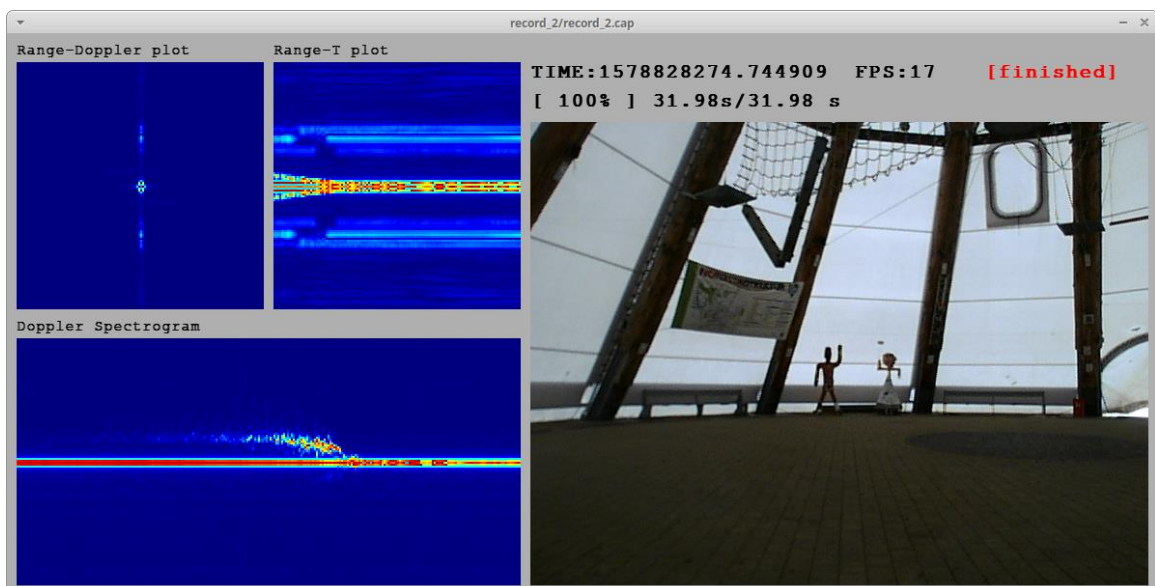


Figure 61: radar_viewer UI

6.2 Live tracking tool

A SDL client named “radar_tracker” is developed in C and C++ to track targets and view the extracted features from the walking pattern. Two tracking modes are supported: single target tracking and multi-target tracking.

In single target tracking, the spectrogram of the single target and the live location of the target are displayed. The extracted features are displayed as well. However, in multi-target tracking mode, the displays of the walking pattern features are not implemented.

The work flow of the “radar_tracker” is linear, no multi-thread functionality is developed. Below describes the work flow of the “radar_tracker”. The work each step corresponds to an individual module:

1. Packet extraction; extract Ethernet packets from libpcap API.
2. Initialization; including Kalman filter initialization, fft initialization, SDL initialization, pcap initialization.
3. Extract two channel raw ADC samples.
4. 2D FFT; when 128 chirps are received for each channel, the 2D FFT are performed on two channel ADC samples to get two complex radar frames.
5. fftshift; the fftshift is done separately on the two radar frames after 2D FFT.
6. Calculate radar frame and angle; the raw radar frames and angles are calculated from the two complex radar frames.
7. Remove main clutters; remove the stationary clutters.
8. Calibration; try to find the averaged environment noise.
9. Remove environment noise; remove the correlated clutters.
10. Apply threshold Th1.
11. Apply local maxima detector; detect the local peaks.
12. Apply CFAR detector; evaluate the local peaks.
13. Apply single target Kalman tracker/multi-target Kalman tracker; depends on the tracking mode, the corresponding tracker is selected.
14. Enhance radar image; amplify weak signal using the time gain method.
15. Apply threshold Th2.

16. Extract walking pattern features; extract the patterns using the tracking result from the tracker.
17. Calculate pattern features; obtain the features from the extracted pattern.
18. Display features and coordinates; show the features and coordinates on the UI.
19. Draw Range-Doppler window; show the range-Doppler plot.
20. Calculate and draw Range-T plot and spectrogram show the range-t and Doppler-t plot on the UI.
21. Update SDL GUI; update each frame.

Figure 62 shows a screen shot of a single target tracking mode of test 1.1 in the “radar_tracker”. The red dots in the right side map view indicate the tracked target. Below the map view, the coordinates and features of the walking pattern are displayed.

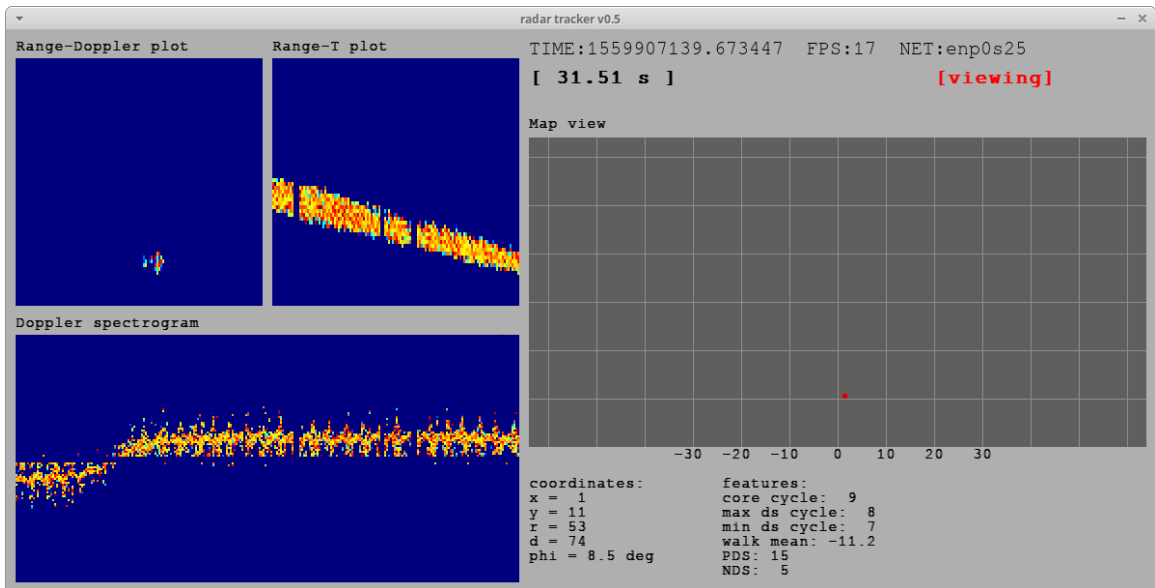


Figure 62: radar_tracker, in single target tracking mode

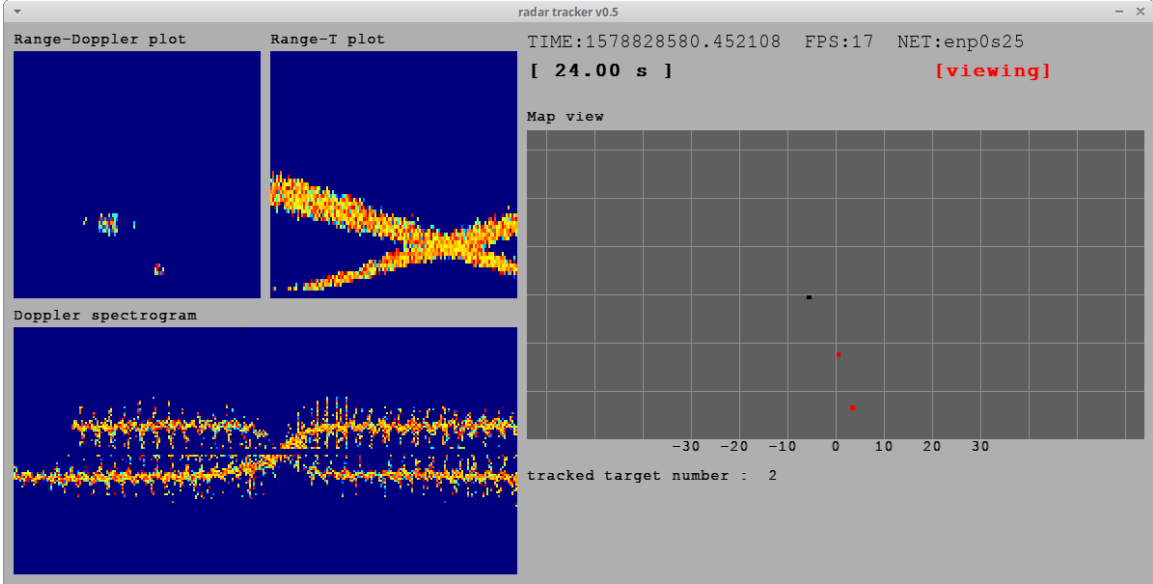


Figure 63: radar_tracker, in multi-target tracking mode

Figure 63 shows a screen shot of a multi-target tracking mode of test 2.1 in the “radar_tracker”. The two red dots in the right side map view indicate two tracked targets in state 1. The surrounding black dots indicate untracked target in state 0. Below the map view, the number of tracked targets is displayed.

In conclusion, the single target works very well for single target case, the extracted walking pattern features can be observed. The multi-target tracking mode works well for two target cases.

7 Conclusion and Future Work

This thesis focuses on the detection, tracking and recognition of human walking pattern. In detection stage, the combination of 2D maxima and CA-CFFAR detector ensures that the correct target is detected. In tracking stage, the multi-target Kalman tracker is developed to make sure the correct target is focused for later pattern extraction. In the recognition stage, the two pass sampling strategy is employed to make sure the extracted walking pattern has enough details for a successfully features extraction.

However, the system still has less optimal issues. First the performance of angle separation in situations described in test 2.2 and test 3 is low. With low quality target detection, the tracker may not be able to find the target. The tweaking of an optimal parameter for processing blocks also requires a lot of try and error. Also, in feature extraction, enough frames must be provided to find the peak/trough of the pattern, which means it may require a relatively long time to extract correct features. For casual walking pattern, the time of gathering data is one step cycle, which is around half second.

For future work, lots of optimization can be done on the existing processing blocks. Apart from people counting application, the walking pattern recognition can be applied to other fields. For people have difficulty in walking, they usually have an abnormal walking patterns. Successful detection of such pattern is important, as they may need additional assistances. Other features of walking pattern can also be proposed, for example, the detection of dominant hand, since in walking the dominant hand may move first before the other hand. This can be applied to gesture recognition applications.

Another way to improve the system is by utilizing the radar module. Since the radar kit has built in FFT engine, the initial 2D FFT and angle calculation can be done completely in the microcontroller onboard the radar. This will require some modification to the communication protocol to increase the Ethernet throughput, because the radar image and the angle image are transmitted at the same time. Also, if the application is less complex, then it is possible to port all software components to the onboard microcontroller, so that the PC is no longer required.

8 References

- [1] Google, "Project Soli", Accessed on: Feb. 20, 2020. [Online]. Available: <https://atap.google.com/soli/>
- [2] Victor C. Chen: The micro-doppler effect in radar, pp.18. ARTECH HOUSE (2011).
- [3] Infineon, "KIT_ATV_24GHZ_RADAR", Accessed on: Feb. 20, 2020. [Online]. Available: https://www.infineon.com/cms/en/product/evaluation-boards/kit_atv_24ghz_radar/
- [4] Y. X, Micro-Doppler effect capture system with Infineon automotive 24 GHz radar kit, 2019.
- [5] radartutorial.eu, "Radar Basics", Accessed on: Feb. 20, 2020. [Online]. Available: <https://www.radartutorial.eu/index.en.html>
- [6] M. Hozhabri, Human Detection and Tracking with UWB radar, pp. 32-33, 2019. Accessed on: Feb. 20, 2020. [Online]. <https://www.diva-portal.org/smash/get/diva2:1346671/FULLTEXT02.pdf>
- [7] N. CARLSTRÖM, Target Identification Using Low Level Radar Measurements, pp.18, 2018. Accessed on: Feb. 20, 2020. [Online]. Available: <https://odr.chalmers.se/bitstream/20.500.12380/254978/1/254978.pdf>
- [8] A. Macaveiu, A. Campeanu, and I. Nafornta, Kalman-Based Tracker for Multiple Radar Targets, 2014 10th International Conference on Communications (COMM), Accessed on: Feb. 20, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/6866690>
- [9] J. Wang, P. He and T. Long, Use of the radial velocity measurement in target tracking, IEEE Transactions on Aerospace and Electronic Systems, vol 39, no. 2, April 2003. Accessed on: Feb. 20, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/1207253>
- [10] S. Heuel and H. Rohling, Pedestrian Classification in Automotive Radar Systems, 2012 13th International Radar Symposium, Accessed on: Feb. 20, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/6233285>
- [11] S. Heuel, H. Rohling, Two-Stage Pedestrian Classification in Automotive Radar Systems, 2011 12th International Radar Symposium (IRS), Accessed on: Feb. 20, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/6042174>
- [12] A. Sundaresan, A. RoyChowdhury and R. Chellappa, A Hidden Markov Model based framework for recognition of humans from gait sequences, Proceedings 2003 International Conference on Image Processing (Cat. No.03CH37429), 2003. Accessed on: Feb. 20, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/1246624>
- [13] Weak signal enhancement in radar signal processing, 20th International Conference Radioelektronika 2010. Accessed on: Feb. 20, 2020. [Online]. Available: <https://ieeexplore.ieee.org/document/5478571>
- [14] radartutorial.eu, "False Alarm Rate", Accessed on: Feb. 20, 2020. [Online]. Available:

- <https://www.radartutorial.eu/01.basics/False%20Alarm%20Rate.en.html>
- [15] H. You, X. Jianjuan, G. Xin: Radar data processing with Applications, chapter 3. John Wiley & Sons Singapore Pte. Ltd 2016.
- [16] J. Munkres, Algorithms for the Assignment and Transportation Problems, Journal of the Society for Industrial and Applied Mathematics, 5(1), 32–38. Accessed on: Feb. 20, 2020. [Online]. Available: <https://doi.org/10.1137/0105003>
- [17] V. C. Chen: The micro-doppler effect in radar, chapter 5. ARTECH HOUSE 2011.
- [18] H. Koo, S. Lee, Gait analysis on the condition of arm swing in healthy young adults, Physical Therapy Rehabilitation Science 2016;5:149-54. Accessed on: Feb. 20, 2020. [Online]. Available: <https://doi.org/10.14474/ptrs.2016.5.3.149>
- [19] TCPDUMP&LIBPCAP. Accessed on: Feb. 20, 2020. [Online]. Available: <https://www.tcpdump.org/>
- [20] FFTW Homepage. Accessed on: Feb. 20, 2020. [Online]. Available: <http://www.fftw.org/>
- [21] SDL Homepage. Accessed on: Feb. 20, 2020. [Online]. Available: <http://www.libsdl.org/>
- [22] OpenCV Homepage. Accessed on: Feb. 20, 2020. [Online]. Available: <https://opencv.org/>

APPENDIX A: ABBREVIATIONS

ADAS	Advanced Driver-Assistance Systems
ADC	Analog-to-Digital Converter
AoA	Angle of Arrival
CA-CFAR	Cell-Averaging Constant False Alarm Rate
CUT	Cell Under Test
CW	Continuous Wave
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
EM	Electromagnetic
FFT	Fast Fourier transform
FMCW	Frequency-Modulated Continuous-Wave
HMM	Hidden Markov Model
LNA	Low Noise Amplifier
LPF	Low Pass Filter
maxDS	Maximum Doppler Spread
minDS	Minimum Doppler Spread
NDS	Negative Doppler Spread
NR	Noise Ratio
PDS	Positive Doppler Spread
SDL	Simple DirectMedia Layer
SNR	Signal to Noise Ratio
SVM	Support Vector Machine

APPENDIX B: CONTENTS OF DVD

- [1] internet sources/ : this directory stores the archived references pdf files

- [2] matlab/ : this directory stores all the Matlab scripts used in this thesis
 - record.cap/ : this directory stores all recordings from record_1.1.cap to record_3.cap
 - main_plot_*.m : Matlab scripts to generate figures in this thesis
 - my*.m : Matlab function scripts
 - process_walk_3d_play.m : this Matlab script display 3D structure of recordings
 - process_walk_kalman_rd_single.m : single target kalman tracker with rd tracking
 - process_walk_kalman_xy_single.m : single target kalman tracker
 - process_walk_kalman_xy_multi.m : multi target kalman tracker
 - record_*.mat : processed Matlab recordings from *.cap files

Notes: the pcap *.cap recordings under record.cap directory cannot be used in Matlab directly. Please run the process_walk_all.m script to convert them into *.mat files and copy them to the root of Matlab folder, the processed *.mat files in this case are already provided. The scripts are tested under Matlab R2018a

- [3] radar/ : this directory stores source code for the implemented PC client
 - radar_tracker/ : radar_tracker source directory
 - radar_viewer/ : radar_viewer source directory

Notes: before compiling, please make sure the following libraries are installed and can be liked: -lm -lpcap -lfftw3 -lSDL2 -lSDL2_image -lSDL2_ttf.

To compile both PC clients, copy source to Linux directory and execute makefile. They are tested on Ubuntu 18.04.4 LTS with gcc version 7.4.0.