Ph.D. DISSERTATION

RADAR SENSOR BASED SIGNAL PROCESSING TECHNIQUES FOR INDOOR AND OUTDOOR AUTONOMOUS DRIVING

실내외 자율주행을 위한 레이더 센서 기반의 신호 처리 기법

BY

SOHEE LIM

FEBRUARY 2023

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

Ph.D. DISSERTATION

RADAR SENSOR BASED SIGNAL PROCESSING TECHNIQUES FOR INDOOR AND OUTDOOR AUTONOMOUS DRIVING

실내외 자율주행을 위한 레이더 센서 기반의 신호 처리 기법

BY

SOHEE LIM

FEBRUARY 2023

DEPARTMENT OF ELECTRICAL AND
COMPUTER ENGINEERING
COLLEGE OF ENGINEERING
SEOUL NATIONAL UNIVERSITY

RADAR SENSOR BASED SIGNAL PROCESSING TECHNIQUES FOR INDOOR AND OUTDOOR AUTONOMOUS DRIVING

실내외 자율주행을 위한 레이더 센서 기반의 신호 처리 기법

지도교수 김 성 철

이 논문을 공학박사 학위논문으로 제출함

2023년 2월

서울대학교 대학원

전기·정보공학부

임 소 희

임소희의 공학박사 학위 논문을 인준함

2023년 2월

위 원 장:	김 남 수	(인)
부위원장:	김 성 철	(인)
위 원:	오 정 석	(인)
위 원:	최 정 환	(인)
위 원:	이 성 욱	(인)

Abstract

In recent years, research on autonomous sensors and signal processing techniques are being actively conducted with the increased public's interest on autonomous driving. Autonomous driving platforms such as autonomous robots, drones, and vehicles are being developed, and autonomous sensors such as lidar, camera, radar, and ultrasonic are being utilized. Among these sensors, the radar sensor is considered essential because it has a stable performance even under a lightless or adverse weather conditions and has a wide detectable range compared to other sensors. The radar sensor can be installed at robots, vehicles, and drones to perceive the surrounding environment or obtain information about detected targets such as the relative velocity, distance, and angle.

In this dissertation, I propose advanced signal processing techniques for radar sensor based indoor and outdoor autonomous driving. Simultaneous localization and mapping (SLAM) is becoming significant in autonomous driving. To accurately estimate the position of autonomous driving platform for SLAM, the ego-motion such as its rotation angle and velocity should be considered. A research on estimating the ego-motion of radar-equipped robot using a radar sensor without any additional device is proposed in this dissertation. In addition, when driving in a complex environment such as highway or urban road, various targets exist and unpredictable situations can occur. By predicting the path of surrounding vehicles, potential hazards can be prevented in advance, or it can be used to adjust the speed of the ego-vehicle or change lanes. Therefore, this dissertation proposes a method to estimate the heading direction of surrounding vehicles using a radar sensor. Lastly, there has been frequent accidents by leaving an infant or animal inside the vehicle, and it is becoming mandatory to install indoor passenger detection system in an autonomous vehicle. By monitoring the position of passengers, the safety can be improved and energy inside vehicle can be managed ef-

fectively. Also, by using a radar sensor, multiple people can be detected contactless and their position can be estimated even when they are stationary. Therefore, this dissertation proposes a method for detecting the number and location of passengers inside a vehicle using a radar sensor.

keywords: Autonomous driving, ego-motion estimation, frequency-modulated continuous-wave (FMCW) radar, impulse radio ultra-wideband (IR-UWB) radar, in-vehicle passenger detection, vehicle orientation estimation

student number: 2017-25081

Contents

Abstract **i**

Contents **iii**

List of Tables **vi**

List of Figures **vii**

1 INTRODUCTION **1**

2 PRINCIPLES OF RADAR SYSTEMS **5**

2.1 FMCW MIMO RADAR 5

2.2 IR-UWB RADAR 8

3 EGO-MOTION ESTIMATION USING FMCW RADAR **11**

3.1 APPLICATION TO AUTONOMOUS ROBOT 11

3.1.1 Motivation 11

3.1.2 Radar Sensor for SLAM 13

3.1.3 Proposed Method for Ego-Motion Estimation 16

3.1.4 Performance Evaluation 21

3.1.5 Summary 27

3.2 APPLICATION TO AUTONOMOUS VEHICLE 28

3.2.1 Motivation 28

3.2.2 Automotive Radar System Used in Experiments 29
3.2.3 Target Detection Result Through Radar Signal Processing . . 30
3.2.4 Stationary and Moving Targets in Angle-Velocity Domain . . 31
3.2.5 Hough Transform Based Ego-velocity Estimation 32
3.2.6 Summary 35

4 VEHICLE ORIENTATION ESTIMATION 36

4.1 RANGE-ANGLE MAP BASED APPROACH USING MIMO FMCW RADAR 36
4.1.1 Motivation 36
4.1.2 Radar Signal Measurement Scenarios 37
4.1.3 Proposed CNN-Based Heading Direction Estimation Method . 39
4.1.4 Performance Evaluation 43
4.1.5 Summary 47
4.2 POINT CLOUD DATA BASED APPROACH USING 4D IMAGING RADAR 48
4.2.1 Motivation 48
4.2.2 Basic Principles of Cascaded MIMO FMCW Radar System . 50
4.2.3 Measurement Using MIMO FMCW Radar 52
4.2.4 Vehicle Orientation Estimation Techniques 55
4.2.5 Performance Evaluation 60
4.2.6 Summary 65

5 IN-VEHICLE PASSENGER DETECTION 66

5.1 FEATURE EXTRACTION BASED APPROACH USING IR-UWB RADAR 66
5.1.1 Motivation 66
5.1.2 Basic Signal Processing in IR-UWB Radar System 69
5.1.3 Measurement of IR-UWB Radar Signal Inside Vehicle 71

5.1.4 Feature Extraction and Feature Selection 76
5.1.5 Performance Evaluation 79
5.1.6 Summary 87
5.2 DEEP NEURAL NETWORK BASED APPROACH USING IR-UWB RADAR 88
5.2.1 Motivation 88
5.2.2 Proposed DNN-Based People Localization 89
5.2.3 Performance Evaluation 90
5.2.4 Summary 95
5.3 DEEP NEURAL NETWORK BASED APPROACH USING FMCW RADAR 96
5.3.1 Motivation 96
5.3.2 Experimental Environment 98
5.3.3 Pre-processing of FMCW Radar Signal 99
5.3.4 Passenger Occupancy Detection Methods 105
5.3.5 Performance Evaluation 109
5.3.6 Summary 114

6 CONCLUSION **115**

Abstract (In Korean) **131**

List of Tables

3.1 Specifications of the FMCW radar 14
3.2 RMSE of the estimated angle when α is 0.19 26
3.3 Performance of various algorithms 27
3.4 Ego-velocity estimation results according to the velocity of the ego-vehicle 35

4.1 Radar system parameters 39
4.2 Confusion matrix derived from CNN 46
4.3 Specifications of RETINA 4F radar module 53
4.4 Comparison of various estimation methods 63
4.5 Comparison of other methods 63

5.1 Specifications of radar system 72
5.2 Experiment scenario 73
5.3 Personnel information of experiment participants 74
5.4 Weight values of each parameter using NCA algorithm 82
5.5 Classification accuracy for various machine learning algorithms . . . 85
5.6 Personnel information of experiment participants (Second experiment) 86
5.7 Performance of various machine learning algorithms and DNN 94
5.8 Comparison of classification methods 113

List of Figures

2.1 Basic principles of beat signal generation in FMCW radar 5
2.2 Gaussian pulse signal: (a) in time domain (b) in frequency domain . . 9
2.3 Block diagram of parallel sampling 10

3.1 Transmitted waveform of the FMCW radar 14
3.2 Analysis of detection results when the robot rotates 15
3.3 Analysis of detection results in angle-velocity domain 16
3.4 Matrix transformation of detected points in the angle-velocity plane . 17
3.5 Block diagram of the method for estimating the yaw rate 18
3.6 Experimental environment 21
3.7 Scenario of robot movement 22
3.8 Detection results in the angle-velocity domain (Case 4) 22
3.9 Results of matrix transformation (Case 4) 23
3.10 Yaw rate estimation result (Case 4) 23
3.11 Probability of detecting rotation for each case 24
3.12 RMSE of estimated angle 25
3.13 Estimation of ego-velocity using the detected points 26
3.14 Configuration of the 77 GHz automotive radar system 30
3.15 Target detection using the automotive radar: (**a**) radar signal measurement on the bridge (**b**) coordinate system conversion of the radar detection result 31

3.16 Target detection result on the bridge 32
3.17 Target detection result in the angle-velocity domain 32
3.18 Basic concept of the Hough Transform: (**a**) in the x-y domain; (**b**) in the α-β domain; (**c**) in the x-y domain; (**d**) in the ϕ-ρ domain 33
3.19 Result of applying the Hough transform to the detection result in the angle-velocity domain 34
3.20 Ego-velocity estimation results in the angle-velocity domain 34

4.1 AWR1642BOOST kit with DCA1000EVM module 38
4.2 Experimental environments in a parking lot 38
4.3 Experiment scenarios 39
4.4 3D data cube generated from the received radar signal 40
4.5 Range-angle map generation using the radar data cube 41
4.6 Data when the front vehicle turns left: Scenario D 42
4.7 CNN architecture of the proposed method 42
4.8 Classification accuracy according to the number of filters and blocks . 44
4.9 Loss and classification accuracy according to the number of iterations 45
4.10 Visualization of the data using the t-SNE algorithm 45
4.11 Point cloud data according to vehicle orientation 51
4.12 RETINA 4F radar module (Smart Radar System) 52
4.13 Experimental environment 53
4.14 Block diagram of signal pre-processing 54
4.15 Processed point cloud data (Orientation angle: 60°) 55
4.16 Block diagram of vehicle orientation estimation 56
4.17 Block diagram of bagging algorithm 58
4.18 Structure of CNN 59
4.19 Estimation results using PCA-based method 61
4.20 Estimation results using decision tree-based method 61
4.21 Estimation results using CNN-based method 62

4.22 Comparison of various estimation methods according to the number of accumulated scans 64

5.1 Block diagram of basic signal processing in the IR-UWB radar system 69
5.2 Signal transmitted by the IR-UWB radar 70
5.3 Comparison of signal before and after cross-correlation 71
5.4 IR-UWB radar module (NVA-R661) 71
5.5 Experiment environment 72
5.6 Actual photograph of experiment environment (Class E9) 74
5.7 Processed radar signal $\hat{\mathbf{p}}$ for 32 experiment scenarios 75
5.8 Overall signal processing chain of the proposed method 80
5.9 Block diagram of bagging with decision tree 81
5.10 Comparison of classification accuracy according to the number of features and the number of classifiers 83
5.11 Confusion matrix derived from bagging with decision tree (Unit: %) . 84
5.12 Actual photograph of experiment environment: Class E1 (Second experiment) 86
5.13 General structure of MLP network 89
5.14 Classification accuracy by changing the number of nodes in a hidden layer (Number of hidden layers: 1) 91
5.15 Classification accuracy by changing the number of hidden layers (Number of nodes: 50) 92
5.16 Classification accuracy by changing the number of nodes and hidden layers 92
5.17 Experimental environment 99
5.18 Experimental cases 99
5.19 FMCW radar detection results (Case 6) 101
5.20 Comparison of distance variation for stationary object and passenger . 103
5.21 Pre-processing of FMCW radar signal 104

5.22 Block diagram of proposed method 105
5.23 Passenger occupancy detection method with hyper parameter 106
5.24 Principal component analysis of two clutter reduction methods 110
5.25 Classification results using SVM method 111
5.26 Classification results using MLP method 112
5.27 Classification results using CNN method 112

Chapter 1

INTRODUCTION

In recent years, people's interest in autonomous driving has been rapidly increasing. To realize autonomous driving, autonomous platforms such as robot, drone, and vehicle need to perceive the surrounding environment and detect multiple targets with high precision. To achieve this task, sensors such as camera, lidar, and radar, each play a unique role and assist autonomous driving. Camera sensor can obtain RGB image data of target, but it is sensitive to adverse weather and light conditions and privacy issue can occur. Lidar sensor which uses laser beam can obtain 3D point cloud data at a high spatial resolution, but its detection performance is degraded in rain or foggy weather. Meanwhile, the radar sensor which uses radio wave has a robust detection performance under adverse environmental conditions and has outstanding velocity estimation performance. The radar sensor is considered essential in autonomous driving, and various types of radar sensor with different modulation techniques are used to estimate target information. In particular, the frequency-modulated continuous-wave (FMCW) radar and impulse radio ultra-wideband (IR-UWB) radar are widely used.

An FMCW radar uses a signal whose frequency increases linearly with time. This modulation technique is efficient for target detection because it enables the joint estimation of distance and velocity at a high resolution while consuming low power. The distance and velocity information of the target can be estimated by analyzing

the frequency domain received signal [1]. In addition, an FMCW radar with multiple-input–multiple-output (MIMO) antenna system is gaining significant attention because modern autonomous driving functions require obtaining an image of the front scene, which can be realized through a high-resolution radar data. The MIMO radar can provide more sophisticated target detection results because it enables the precise estimation of target's angle information [2]. Meanwhile, an IR-UWB radar uses a very short pulse with duration in nanoseconds [3], which has a low power consumption and high range resolution [4]. Studies related to IR-UWB radar are mainly focused on indoor detection because it has a relatively short detectable distance.

In this dissertation, I propose a signal processing technique for indoor and outdoor autonomous driving based on radar sensor. The main topics can be broadly divided as ego-motion estimation, orientation estimation of surrounding vehicles, and occupancy detection of passengers inside a vehicle. A 60 / 77 GHz FMCW radar and an IR-UWB radar are used in the dissertation. The remainder of this dissertation is organized as follows.

In Chapter 2, the basic principles of FMCW and IR-UWB radars are explained. Then, a technique for estimating the ego-motion using radar sensor is proposed in Chapter 3, where the focus is autonomous robot and autonomous vehicle. To perform radar-based simultaneous localization and mapping (SLAM), the ego-motion such as rotation angle and ego-velocity should be considered. To estimate the rotation angle, I use the distribution of detected points on a two-dimensional (2D) plane. The distributions of the detected points at successive time instants are correlated with each other. The detected points in the angle-velocity domain are transformed into matrix data, and the rotation angle is estimated by calculating the correlation between the matrices. In addition, the relative velocity of stationary targets is equal to minus sign of ego-velocity. In the angle-velocity domain, the moving targets are randomly distributed, whereas the stationary targets are distributed along a specific trend line. Therefore, the ego-velocity of the robot and vehicle is estimated by finding the trend line formed in

the angle-velocity 2D plane.

Next, a technique to estimate the heading direction of vehicle using radar sensor is proposed in Chapter 4. The estimated orientation information can help to predict the path of a surrounding vehicle, which can be used in vehicle re-identification task. Also, it can be used to perform appropriate actions such as changing lane or adjusting the speed of the ego-vehicle. In Chapter 4.1, the instantaneous movement of vehicle is estimated using the range-angle map obtained from MIMO FMCW radar. The automotive radar sensor data are accumulated for five different movements of the front vehicle (e.g., stop, going ahead, reversing, turning left, and turning right). Then, the radar data are converted into range-angle map by applying high-resolution angle estimation algorithm. The different movements of the front vehicle are classified by applying the convolutional neural network (CNN) algorithm. In Chapter 4.2, the vehicle orientation is estimated using the point cloud data obtained from imaging radar. The radar signal is collected by varying the orientation angle of the vehicle and the point cloud data corresponding to the vehicle are extracted through signal preprocessing. Because the processed point cloud data are distributed along the axis of vehicle orientation, the orientation angle is estimated by applying regression algorithms.

Finally, a technique to detect the occupancy of passengers inside vehicle using radar sensor is proposed in Chapter 5. The radar data are collected by installing the radar sensor at the rear-view mirror position inside the vehicle. In Chapters 5.1 and 5.2, the signals collected from an IR-UWB radar sensor are processed to extract signals from meaningful targets and reduce undesired signals from clutter and noise. Then, by applying either feature extraction method (Chapter 5.1) or non-feature extraction methods (Chapter 5.2), the number and position of passengers inside vehicle are estimated. In Chapter 5.3, the method using FMCW radar is presented to overcome the limitations of the IR-UWB radar-based method. The method is robust to indoor environmental changes regardless of whether the passenger is moving or not. By applying the phase-based clutter suppression method to the received signal, the signals from

undesired reflections are eliminated and signals from passengers are clearly identified. Then, by applying various machine learning algorithms, the number and position of passengers inside vehicle are accurately estimated.

Chapter 2

PRINCIPLES OF RADAR SYSTEMS

2.1 FMCW MIMO RADAR

The FMCW radar transmits a sinusoidal signal whose frequency increases linearly within the chirping duration. As shown in Fig. 2.1, the instantaneous frequency of a chirp signal can be expressed as

$$f_I(t) = f_c + \underbrace{\frac{B}{\Delta T}}_{\gamma} t \quad (0 \le t \le \Delta T), \tag{2.1}$$

where f_c is the carrier frequency, B is the bandwidth, ΔT is the chirping duration, and γ is the frequency slope, respectively. The instantaneous phase can be obtained by

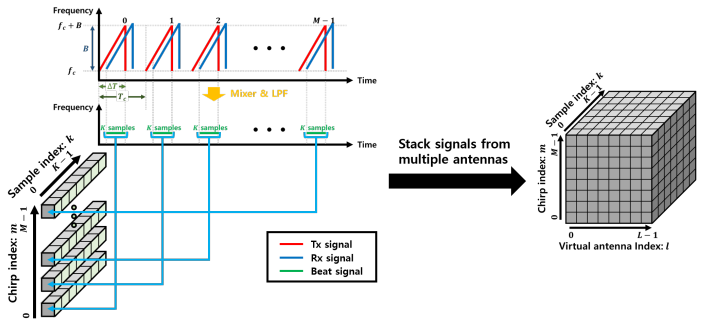


Figure 2.1: Basic principles of beat signal generation in FMCW radar

integrating the instantaneous frequency with respect to time,

$$\Theta_I(t) = 2\pi \int_0^t f_I(t)dt = 2\pi\Big(f_c t + \frac{\gamma}{2}t^2\Big), \tag{2.2}$$

and the transmitted chirp signal can be expressed as

$$S_T(t) = A_T \cos\big(\Theta_I(t)\big) = A_T \cos\left(2\pi\Big(f_c t + \frac{\gamma}{2}t^2\Big)\right), \tag{2.3}$$

where A_T is the amplitude of the transmitted signal. The time-delayed signals reflected from nearby scatterers are received at the receiving antenna (Rx), and can be expressed as

$$\begin{aligned} S_R(t) &= \sum_{n=1}^{N_t} A_n \cos\big(\Theta_I(t-\tau_n)\big) \\ &= \sum_{n=1}^{N_t} A_n \cos\left\{2\pi\left(f_c(t-\tau_n) + \frac{\gamma}{2}(t-\tau_n)^2\right)\right\}, \end{aligned} \tag{2.4}$$

where N_t is the number of scatterers, and A_n and τ_n are the amplitude and time delay of the n-th target, respectively. The received signals are mixed with the transmitted signal and low-pass filtering is applied, which can be expressed as

$$X(t) = LPF(S_T(t)S_R(t)), \tag{2.5}$$

where $LPF(\cdot)$ denotes the low-pass filtering operation.

The time delay τ_n is the two-way propagation delay of the radio wave, generally expressed as $\tau_n = \frac{2r_n}{c}$. In an FMCW radar system as shown in Fig. 2.1, multiple chirp signals are transmitted periodically; the range variation between different chirp signals must be considered. In addition, when an array antenna system with multiple transmitting and receiving antennas is used, the path difference between each antenna element must be considered. By incorporating these additional constraints, the modified time delay τ_n can be expressed as a function of chirp index m and antenna index l as

$$\tau_n(m,l) = \frac{2(r_n + m v_n T_c) + l d \sin\theta_n}{c}, \tag{2.6}$$

where r_n, v_n, and θ_n are the distance, relative velocity, and incident angle of the n-th target. In addition, T_c is the chirp interval, d is the antenna spacing, and c is the speed of light, respectively.

By substituting (2.6) into (2.4) and (2.5) and neglecting the high-frequency term, the resulting beat signal can be expressed as

$$\begin{aligned} X(t,\, m,\, l) \simeq \sum_{n=1}^{N_t} A'_n \cos\Bigg\{ 2\pi \Bigg(& \underbrace{\frac{2\gamma r_n}{c}}_{f_n^b} t + \underbrace{\frac{2v_n}{\lambda}}_{f_n^d} m T_c \\ & + \underbrace{\frac{d \sin\theta_n}{\lambda}}_{f_n^a} l + \underbrace{\frac{2r_n}{\lambda}}_{\psi_n} \Bigg) \Bigg\} \\ & (m = 0,\, 1,\, \cdots,\, M-1) \\ & (l = 0,\, 1,\, \cdots,\, L-1), \end{aligned} \tag{2.7}$$

where λ is the wavelength of the transmitted signal, M is the number of chirps, and L is the number of virtual antennas, respectively. When quadrature demodulation is used and analog-to-digital sampling is applied, the resulting beat signal can be expressed as

$$\begin{aligned} X[k,\, m,\, l] \simeq \sum_{n=1}^{N_t} A'_n \exp\left\{ j2\pi \left(f_n^b k T_s + f_n^d m T_c + f_n^a l + \psi_n \right) \right\} \\ (k = 0,\, 1,\, \cdots,\, K-1) \\ (m = 0,\, 1,\, \cdots,\, M-1) \\ (l = 0,\, 1,\, \cdots,\, L-1), \end{aligned} \tag{2.8}$$

where T_s is the sampling period, k is the sample index, and K is the number of samples, respectively. The beat signal in (2.8) is a 3D exponential signal whose frequency along each axis contains information about the target distance, velocity, and angle. By applying 3D fast Fourier transform (FFT) to (2.8), the target information can be

obtained from the frequency estimates as follows:

$$\begin{aligned}\hat{r}_n &= \frac{c}{2\gamma}\hat{f}_n^b \\ \hat{v}_n &= \frac{\lambda}{2}\hat{f}_n^d \\ \hat{\theta}_n &= \arcsin\left(\frac{\lambda}{d}\hat{f}_n^a\right).\end{aligned} \tag{2.9}$$

The processing of M chirp signals to estimate these three target parameters is defined as a frame in the remainder of dissertation.

2.2 IR-UWB RADAR

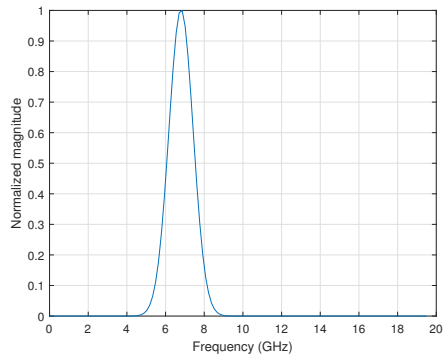
In an IR-UWB radar system, the radar transmits a sequence of narrow pulse signals that occupy a wide frequency spectrum. The main advantages of this system is the improved range resolution by using short pulses, and immunity to external narrowband noise by using a wide frequency spectrum [5]. The transmitted signal can be written as

$$s(t) = x(t)\cos(2\pi f_c t), \tag{2.10}$$

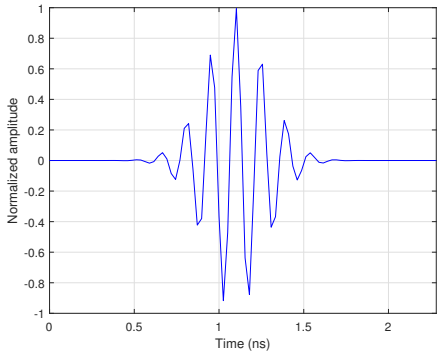
where $x(t)$ is the complex envelope of the pulse signal and f_c is the carrier frequency. Among the various types of pulse waveforms, the Gaussian pulse is most widely used due to its relative ease of implementation and excellent time-frequency product [6]. A modulated Gaussian pulse signal is shown in Fig. 2.2, for both time domain and frequency domain.

When the transmitted signal is incident on the target, the signal is backscattered and then received at the receiver. The received signals can be expressed as

$$\begin{aligned} r(t) &= \sum_{m=1}^{M} a_m\, s(t-\tau_m) + n(t) \\ &= \sum_{m=1}^{M} a_m\, x(t-\tau_m)\cos\{2\pi f_c(t-\tau_m)\} + n(t), \end{aligned} \tag{2.11}$$



(a)



(b)

Figure 2.2: Gaussian pulse signal: (a) in time domain (b) in frequency domain

where a_m and τ_m are the attenuation coefficient and time delay of the m_{th} path, M is the number of paths, and $n(t)$ is the noise added on the receiving antenna, respectively. Then, the received signals are digitalized through a sampling process, which can be expressed as

$$r[n] = r(nT_s) \quad (n = 1, 2, \cdots, N), \tag{2.12}$$

where T_s is the sampling period and N is the number of samples. In the IR-UWB radar system used in the dissertation, the time interval between adjacent samples is 26 ps, which corresponds to a range resolution of 4 mm. Thus, 256 samples are needed to display a range of 1 m. I define a set of 256 samples as a frame, and the number of

frames determines the range of observation. In addition, a set of frames is defined as a scan. If the number of frames is denoted as n_f, $256 \times n_f$ samples are obtained from each scan (i.e., $N = 256 \times n_f$), and a range of n_f m is observed.

Since the duration of a pulse is very short, sampling needs to be performed very quickly in an IR-UWB radar system, which is extremely difficult to implement in real time. To solve this issue, a parallel sampling is performed, which means that multiple samplers are used to sample the signal in parallel. The block diagram of parallel sampling is shown in Fig. 2.3. Each sampler samples the signal every $N_s T_s$ seconds, while the sampling offset between adjacent samplers is T_s seconds. By combining the results from each sampler, it has an equivalent effect of sampling per T_s seconds and fast sampling can be achieved.

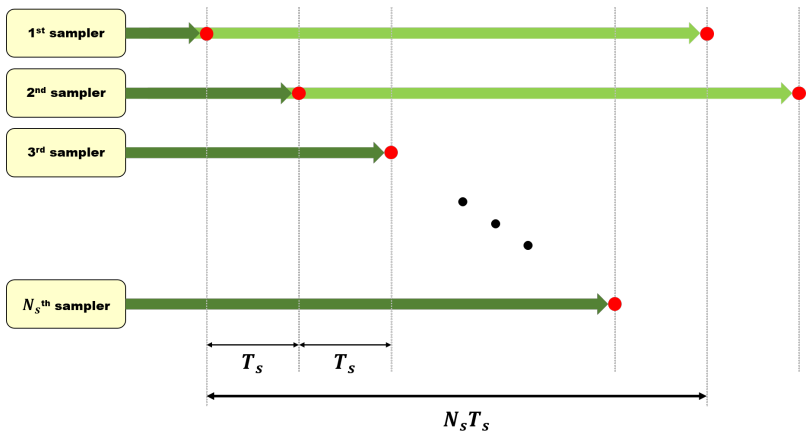


Figure 2.3: Block diagram of parallel sampling

Chapter 3

EGO-MOTION ESTIMATION USING FMCW RADAR

3.1 APPLICATION TO AUTONOMOUS ROBOT

3.1.1 Motivation

Recently, the importance of SLAM has been increasing due to the growing popularity of autonomous robots, vehicles and unmanned drones. SLAM is a technique in which a robot or vehicle moves in a new environment and simultaneously forms map and locates its position. To sense the surrounding environment, sensors such as laser or camera are widely used for SLAM [7]. These sensors have the advantage of detecting the distance to an object at high resolution. However, the detection performance of these sensors can be deteriorated in foggy or smoky environments [8, 9], which can significantly degrade the performance of SLAM.

Meanwhile, the radar sensor shows robust performance under low-light conditions. Also, the information about the target such as its relative velocity and radar cross section (RCS) can be obtained, which can be used to improve the performance of SLAM. As a result, studies on radar-based SLAM have been actively conducted by a number of researchers [10, 11, 12, 13]. For example, the authors in [10] used an UWB pulse-echo radar and performed SLAM by using the spectrogram of the received echo signal. Furthermore, research on SLAM using 24 GHz FMCW radar was conducted in

[11], giving the potential of radar-based SLAM to be used in an indoor environment. In [12], a millimeter-wave radar was used and the RCS information of the target was used to enhance the performance of SLAM. Recently, the authors in [13] developed a dual-mode FMCW radar sensor to perform SLAM in an indoor environment. To increase the performance of SLAM, the radar sensor used waveforms with two different bandwidths to detect short-range and long-range regions simultaneously.

In general, radar-based SLAM mainly comprises two steps. The first step is detecting objects through a radar sensor, and the second step is accumulating detection results of successive time instants. When accumulating the detection results, the ego-motion of the radar-equipped device (e.g., a robot or a vehicle) must be considered. In most of the previous studies, additional motors [14] or gyro sensors [15] were installed to estimate the ego-motion of the robot such as its rotation angle and velocity. However, installing devices for motion estimation incurs additional costs and increases the complexity of the overall system.

Therefore, in this study, I propose a method to estimate the ego-motion (e.g., rotation angle and ego-velocity) of a radar-equipped robot by using only radar sensor data without any additional devices. First, I mount a radar sensor on a small robot that performs linear and circular movements, and collect signals from the surrounding environment. Then, I use the detected points in the angle-velocity domain to estimate the rotation angle and velocity of the robot. In this domain, the signals from stationary and moving targets can be distinguished [16], [17]. To estimate the rotation angle, I transform the detected points in the angle-velocity domain into matrix data consisting of binary values. The matrix data acquired in successive time instants have similarities with each other, and the rotation angle of the robot can be estimated by calculating the correlation between the matrices. In addition, to estimate the ego-velocity, I note that the relative velocity of stationary targets is equal to the ego-velocity of the robot. As a result, the detected points of stationary targets form a specific trend line in the angle-velocity domain. I apply the line detection algorithm to the detected points in the

angle-velocity domain and the ego-velocity of the robot can be estimated by finding the y-intercept value of the trend line.

The estimated ego-motion results can help control the robot's movement when the motor or gyro sensor is not functioning properly. In addition, because the proposed method only requires data from the radar sensor, it can reduce the burden of installing additional equipment on small robots.

The remainder of this study is organized as follows. In Chapter 3.1.2, I introduce the radar sensor and robot used in the measurements. Then, a method for estimating the robot's ego-motion from the processed radar sensor data is discussed in Chapter 3.1.3. Next, in Chapter 3.1.4, I verify the estimation performance of the proposed method with the measurement data. Finally, conclusions are given in Chapter 3.1.5.

3.1.2 Radar Sensor for SLAM

Radar-Robot System Used in Measurements

In the experiment, I used an FMCW radar sensor with a center frequency of 62 GHz, which was developed in [13]. Unlike the pulse-Doppler radar, the FMCW radar can estimate the range of target at high range resolution through dechirp processing. The radar sensor used in this study transmits the waveform shown in Fig. 3.1, enabling both long-range and short-range detections. The bandwidths of 1.5 GHz and 3 GHz are used for long-range and short-range detection, respectively. Because the range resolution of the FMCW radar is inversely proportional to the bandwidth [18], objects placed nearby are detected with a higher resolution. The detailed specifications of the FMCW radar are listed in Table 3.1.

In the radar-robot system, the radar is mounted on a robot and the robot performs various linear and circular motions. The robot delivers the ego-motion information acquired from the motor to the radar, and the radar delivers the detection results to the robot. Then, the radar-based SLAM is generally performed by combining the detection results of the radar and the ego-motion information of the motor. To enhance the

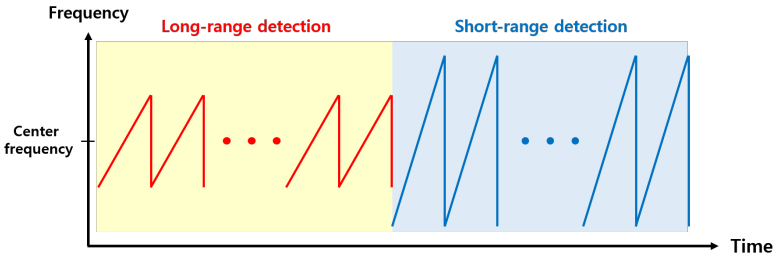


Figure 3.1: Transmitted waveform of the FMCW radar

estimation performance, estimating the yaw rate has been mainly considered in SLAM for autonomous robots and vehicles [19]. In this study, I will estimate the ego-motion of the robot by only using the radar sensor data, and the information from the motor and gyro sensor will be used as ground truth for comparison.

Analysis of Detection Results

The FMCW radar detects the distance, velocity, and angle of targets located in the field of view (FOV) of the radar. When the robot rotates, the antenna boresight changes with respect to the rotation angle, and the angles of the detected points also

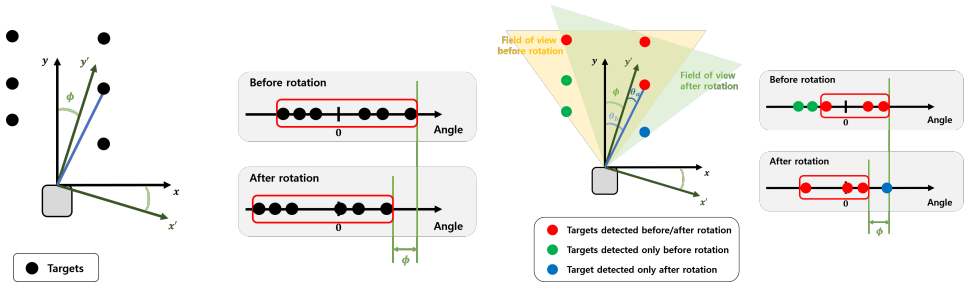
Table 3.1: Specifications of the FMCW radar

Radar parameters	**Long-rage detection**	**Short-range detection**
Center frequency (GHz)	62	62
Bandwidth (GHz)	1.5	3
Number of chirps	256	256
Chirp duration (μs)	150	150
Maximum detectable distance (m)	20	10
Maximum detectable velocity (m/s)	8	8
Range resolution (cm)	10	5
Velocity resolution (m/s)	0.315	0.315
Field of view (deg.)	-20 to 20	-60 to 60

change accordingly. Fig. 3.2a shows the detection results when the robot is rotating. In the figure, the boresight direction is changed from y to y', and the angle of the targets is shifted by ϕ. Therefore, the rotation angle can be estimated by measuring the angle difference before and after the rotation.

However, when the FOV of the radar is limited, the number of detected targets can change as the radar rotates. Fig. 3.2b shows the modified version of Fig. 3.2a by considering the FOV of the radar. In the figure, the red targets are inside the FOV of the radar before and after the rotation. On the other hand, the green targets are only detected before the rotation, and the blue target is only detected after the rotation. In this case, the rotation angle can still be estimated because the red targets are shifted by the rotation angle and there is a high correlation between the data. The detailed method for estimating the rotation angle will be explained in Chapter 3.1.3.

Meanwhile, when the robot moves forward with a velocity of v_e, the relative velocity of stationary targets is approximately detected as $-v_e$. As a result, the detected points of stationary targets form a straight line in the angle-velocity domain, as shown in Fig. 3.3. Therefore, the ego-velocity of the robot can be estimated by applying a line detection algorithm to the detected points in the angle-velocity domain.



(a) Detected points: FOV is not considered (b) Detected points: FOV is considered

Figure 3.2: Analysis of detection results when the robot rotates

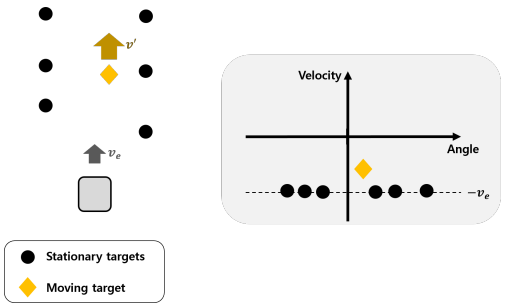


Figure 3.3: Analysis of detection results in angle-velocity domain

3.1.3 Proposed Method for Ego-Motion Estimation

In this study, I introduce a method for estimating the ego-motion of the radar-equipped robot. First, I present a method for converting the detection results into their matrix form. Then, I propose a method for estimating the rotation angle, followed by a method for estimating the ego-velocity.

Matrix Transformation of Radar Sensor Data

To estimate the ego-motion of the robot, I plot the detected points in the angle-velocity domain. When $\theta[k]$ and $v[k]$ are angle and velocity information of k_{th} detected points, the detected points can be expressed as $(\theta[k],\, v[k])\ (k = 1, 2, \cdots, N_t)$, where N_t is the number of detected points. To find the correlation between the data before and after the rotation, I convert the points in the angle-velocity domain into their matrix form. The overall process of the matrix transformation is summarized in Fig. 3.4. First, I quantize the detected velocity and angle values according to their resolutions. For the angle and velocity resolutions of θ_{res} and v_{res}, the quantized values can be expressed as

$$\begin{aligned} \tilde{\theta}[k] &= \lfloor \frac{\theta[k]}{\theta_{res}} + 0.5 \rfloor \cdot \theta_{res}, \\ \tilde{v}[k] &= \lfloor \frac{v[k]}{v_{res}} + 0.5 \rfloor \cdot v_{res}, \\ & \quad (k = 1,\, 2,\, \cdots,\, N_t), \end{aligned} \tag{3.1}$$

where $\lfloor \cdot \rfloor$ represents the floor function. Next, I create lists of quantized angle and velocity values, which can be expressed as

$$\begin{aligned} \boldsymbol{\theta}_{list} &= [-\theta_{max}, -\theta_{max} + \theta_{res}, \cdots, \theta_{max} - \theta_{res}, \theta_{max}], \\ \boldsymbol{v}_{list} &= [-v_{max}, -v_{max} + v_{res}, \cdots, v_{max} - v_{res}, v_{max}]. \end{aligned} \tag{3.2}$$

Here, θ_{max} and v_{max} denote the maximum values of angle and velocity, respectively. In addition, the number of points in $\boldsymbol{\theta}_{list}$ and $\boldsymbol{v}_{list}$ are N_θ and N_v, respectively. Through this process, the angle-velocity domain is divided into $N_\theta \times N_v$ regions, each with a size of $\theta_{res} \times v_{res}$. Then, a transformed matrix $\boldsymbol{M}^0$ is formed by comparing the quantized values of (3.1) and (3.2). The size of the matrix is $N_v \times N_\theta$, and the element is one if the quantized point exists inside the region and zero otherwise. I will use this transformed matrix $\boldsymbol{M}^0$ to estimate the rotation angle of the robot.

Yaw Rate Estimation

To estimate the yaw rate, the correlation between the matrices at two time instants is calculated. The calculation is performed in two directions using forward and backward cross-correlations, as illustrated in Fig. 3.5. In the figure, I use a right side

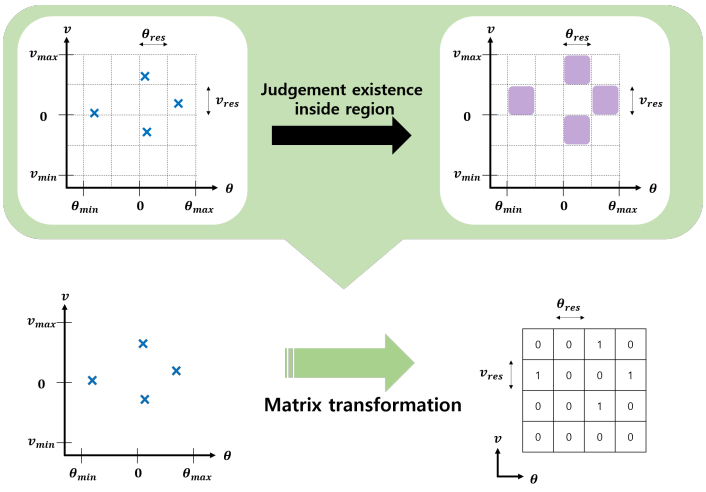


Figure 3.4: Matrix transformation of detected points in the angle-velocity plane

zero-padded matrix $\boldsymbol{M}^f$ for calculating the forward cross-correlation, and a left side zero-padded matrix $\boldsymbol{M}^b$ for calculating the backward cross-correlation. These matrices can be expressed as

$$\begin{aligned} \boldsymbol{M}^f &= [\boldsymbol{M}^0, \boldsymbol{\mathcal{O}}], \\ \boldsymbol{M}^b &= [\boldsymbol{\mathcal{O}}, \boldsymbol{M}^0], \end{aligned} \tag{3.3}$$

where $\boldsymbol{\mathcal{O}}$ is a zero matrix whose elements are all zero. The width of the zero matrix $\boldsymbol{\mathcal{O}}$ was set as $\tilde{N}_\theta = \lfloor \frac{N_\theta}{4} \rfloor$ by considering the FOV of the radar.

When the transformed matrix at the i_{th} frame is defined as $\boldsymbol{M}_i^0$, the cross-correlation between the transformed matrix at the i_{th} frame and the zero-padded matrix at the j_{th} frame can be calculated as

$$\begin{aligned} x^f_{(i,j)}[l] &= \sum_{\hat{q}=1}^{N_\theta} \sum_{\hat{p}=1}^{N_v} \boldsymbol{M}_j^f(\hat{p},\, \hat{q}+l-1) \boldsymbol{M}_i^0(\hat{p},\, \hat{q}), \\ x^b_{(i,j)}[\tilde{N}_\theta + 2 - l] &= \sum_{\hat{q}=1}^{N_\theta} \sum_{\hat{p}=1}^{N_v} \{\boldsymbol{M}_j^b(\hat{p},\, \hat{q}+\tilde{N}_\theta+1-l) \times \\ &\qquad \boldsymbol{M}_i^0(\hat{p},\, \hat{q})\} \\ &\qquad (l = 1,\, 2,\, \cdots,\, \tilde{N}_\theta + 1). \end{aligned} \tag{3.4}$$

The first vector $x^f_{(i,j)}$ which is the result of the forward cross-correlation represents the rotated angle in the clockwise direction. Similarly, the second vector $x^b_{(i,j)}$ which

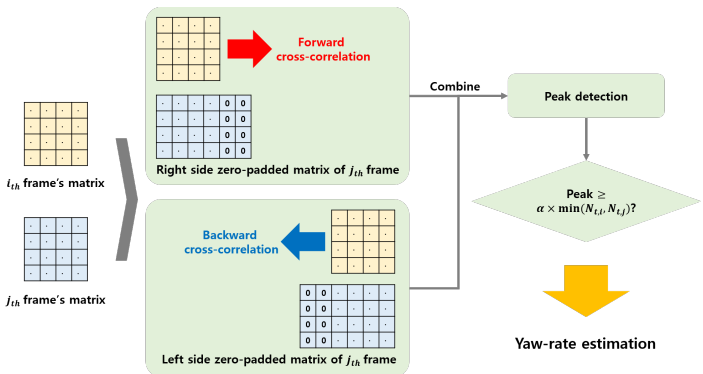


Figure 3.5: Block diagram of the method for estimating the yaw rate

is the result of the backward cross-correlation represents the rotated angle in the counterclockwise direction. These two vectors can be cascaded to form a single vector

$$\begin{aligned} x_{(i,j)} &= [x^b_{(i,j)}[1],\, x^b_{(i,j)}[2],\, \cdots,\, x^b_{(i,j)}[\tilde{N}_\theta + 1], \\ &\quad\; x^f_{(i,j)}[2],\, \cdots,\, x^f_{(i,j)}[\tilde{N}_\theta + 1]], \end{aligned} \tag{3.5}$$

which represents the rotated angle in both clockwise and counterclockwise directions. Because the components $x^b_{(i,j)}[\tilde{N}_\theta + 1]$ and $x^f_{(i,j)}[1]$ both represent the value at the zero angle, I remove the latter to avoid repetition. Then, to estimate the rotated angle, the maximum value and its index are extracted as

$$\begin{aligned} N_c &= \max\, x_{(i,j)}[l], \\ \hat{l} &= \arg\max_{l}\, x_{(i,j)}[l]. \end{aligned} \tag{3.6}$$

Finally, the rotated angle is estimated from the index of the maximum value, which can be expressed as

$$\phi = \begin{cases} (\hat{l} - \tilde{N}_\theta - 1)\,\theta_{res} & \text{if } N_c > \alpha \min\left(N_{t,i}, N_{t,j}\right) \\ 0 & \text{otherwise} \end{cases}, \tag{3.7}$$

where $N_{t,i}$ and $N_{t,j}$ represent the number of detected points in the i_{th} frame and j_{th} frame, respectively. When determining the rotation angle, I assume that the rotation angle is zero if the maximum value does not exceed a certain threshold. For this purpose, I use a threshold coefficient α having value between zero and one to set an appropriate threshold. By using the estimated rotation angle, the yaw rate can be determined as

$$\dot{\phi} = \frac{\phi}{T}, \tag{3.8}$$

where T is the duration between two time instants used to calculate the cross-correlation.

Ego-Velocity Estimation

Several studies have been conducted to estimate the ego-motion of vehicles in the automotive radar context [20, 21, 22]. Especially, in [20], the velocity of the ego-vehicle was estimated using the detected points in the angle-velocity domain. This

method can also be applied to the detection results of radar-based SLAM. In the angle-velocity domain, the velocity of stationary targets is equal to the ego-velocity with a reversed sign, as shown in Fig. 3.3. The detected points of stationary targets are concentrated on the velocity axis near the minus sign of ego-velocity, but randomly distributed on the angular axis. In other words, the detected points form a straight line parallel to the angular axis. In contrast, the detected points of non-stationary targets are randomly distributed without a particular trend line.

Because the number of stationary targets is generally higher than the number of non-stationary targets, I can estimate the ego-velocity by extracting the trend line of the detected points. The linear regression algorithm is applied to the detected points in the angle-velocity domain and a linear model is obtained as

$$\mathcal{Y} = \mathcal{A}\mathcal{X} + \mathcal{B}, \tag{3.9}$$

where $\mathcal{A}$ and $\mathcal{B}$ are slope and y-intercept value. Then, the ego-velocity can be estimated by finding the y-intercept value (i.e, $\mathcal{B} = -v_e$).

Various line estimation algorithms can be used to extract a trend line from the data. Linear least square (LLS) [23] is one of the representative algorithms for finding linear relationships between data. However, its performance is degraded when outlier or noise is present in the data [24]. The Hough transform [25] and random sampling consensus (RANSAC) [26] are well known as line detection algorithms that are robust to outlier. Hough transform algorithm converts the points in the image space into those in the parameter space and uses a voting procedure to detect straight lines. The RANSAC algorithm is an iterative method that randomly extracts a sample and detects an appropriate line through the voting process. In this study, I will use the LLS, Hough transform, and RANSAC algorithms to estimate the ego-velocity of the robot.

3.1.4 Performance Evaluation

Estimation Results of Yaw Rate

The measurement was conducted in a corridor, as shown in Fig. 3.6. There are obstacles in the hallway such as a chair and a cart. The robot moves along the corridor, turns left when it reaches the end of the corridor, and arrives at the entrance. Throughout the entire path, the robot encounters two stationary obstacles and a moving person. To draw a map of the surrounding environment, the robot continuously stops moving and rotates left and right. Therefore, I divided the movement of the robot into four scenarios. As shown in Fig. 3.7, the robot is stationary in case 1, going forward in case 2, and rotating left and right in cases 3 and 4. For example, when the robot is moving forward, the measured data is classified as case 2, and if it is rotating left and right without moving forward, it is classified as cases 3 and 4. The measured data was divided into intervals of 1 s, and this resulted in 217 non-overlapping data. In addition, θ_{res} and v_{res} were set as 0.1° and 0.1 m/s, and θ_{max} and v_{max} were set as 7 m/s and 70°, respectively. This resulted in 141 possible quantized values for both angle and velocity, indicating that N_v and N_θ were equally set as 141.

The detection results in the angle-velocity domain are shown in Fig. 3.8. Fig. 3.8a corresponds to the first frame of the data acquired for 1 s, and Fig. 3.8b corresponds to

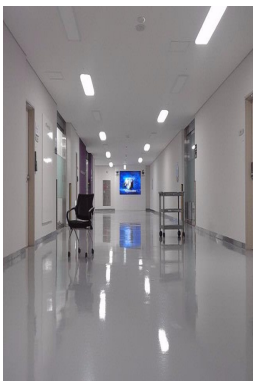


Figure 3.6: Experimental environment

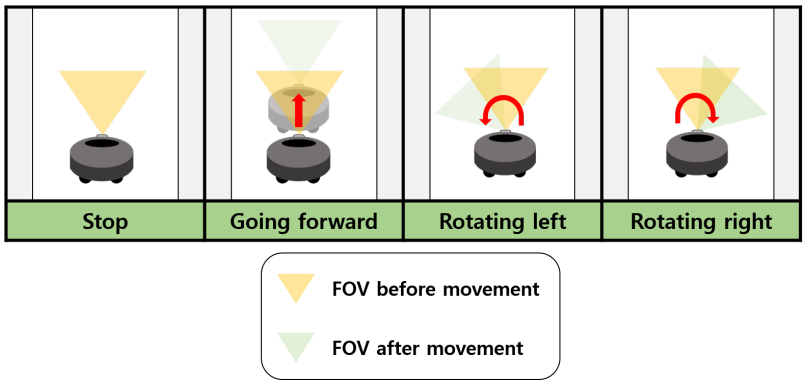
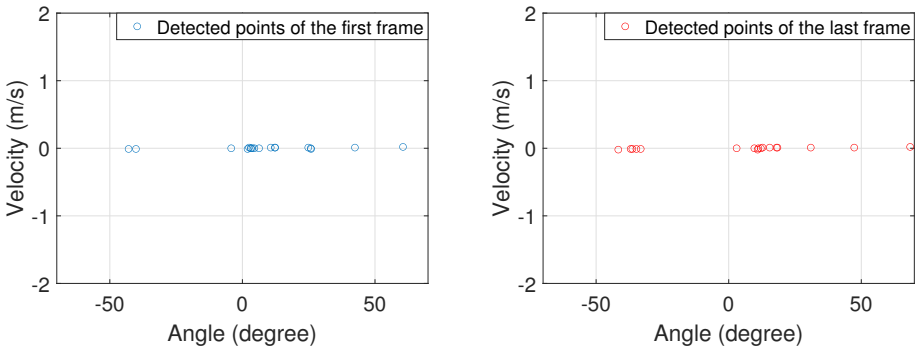


Figure 3.7: Scenario of robot movement

the last frame of the data acquired for 1 s. As the frame advances, most of the detected points are shifted to the right on the angular axis. However, the result in Fig. 3.8b cannot be simply expressed as the shifted version of the result in Fig. 3.8a because the number of detected points can change when the robot rotates. To determine the correlation between these two data, I applied the matrix transformation to the detected points. The resulting matrix is illustrated in Fig. 3.9, where the white area represents value 1 and the black area represents value 0. Then, I calculated the correlation between the transformed and zero-padded matrices. The estimated rotation angle obtained through



(a) First frame

(b) Last frame

Figure 3.8: Detection results in the angle-velocity domain (Case 4)

(a) First frame (b) Last frame

Figure 3.9: Results of matrix transformation (Case 4)

forward and backward cross-correlations is shown in Fig. 3.10. In the figure, the blue dashed line shows the threshold when the threshold coefficient α was set as 0.14. Since the maximum value is higher than the threshold, the rotation angle is estimated by reading the index of the maximum value, resulting in the yaw rate of $7°/s$. The motor and gyro sensor also showed that the robot rotated by 7° in 1 s, so the proposed method was able to estimate the yaw rate by only using the radar data.

Furthermore, I investigated the appropriate value of the threshold coefficient α by

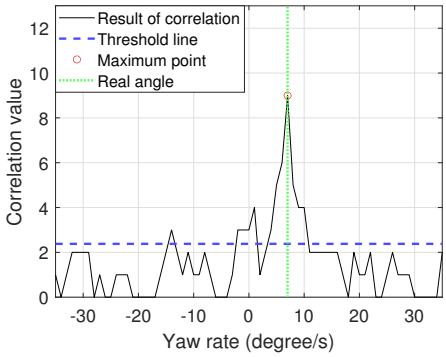


Figure 3.10: Yaw rate estimation result (Case 4)

varying the value from 0.1 to 0.4 in intervals of 0.01. I defined the probability of detecting rotation by deciding that a rotation is detected if the maximum value N_c exceeds the threshold line and vice versa. Fig. 3.11 shows the probability of detecting rotation for various values of α. For all four cases, the probability of detection decreased as the value of α increased. Specifically, for case 1, the probability of detection was 1 for most values of α because this case corresponds to the stationary case having high similarities between adjacent data. In contrast, the probability of detection was the lowest for case 2 because the robot is moving forward and the similarity between data is reduced. Cases 3 and 4 showed similar results, in which the probability of detection rapidly decreased with increasing value of α. Although a small value of α leads to a high probability of detection, the value of α cannot be simply set to a low value because it might lead to false alarms when the robot is actually not rotating. Therefore, I examined the root mean square error (RMSE) of the estimated angle for various values of α.

Fig. 3.12 shows the RMSE of the estimated angle for each of the four cases. In calculating the RMSE, I set the estimated angle as zero when rotation is not detected. As a result, in cases 1 and 2, the RMSE decreased with the increasing value of α because the true rotation angle is zero. Meanwhile, in cases 3 and 4, the RMSE increased with the increasing value of α. This is because the robot's rotation is not detected when the

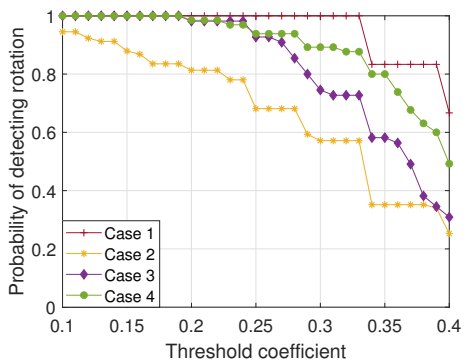


Figure 3.11: Probability of detecting rotation for each case

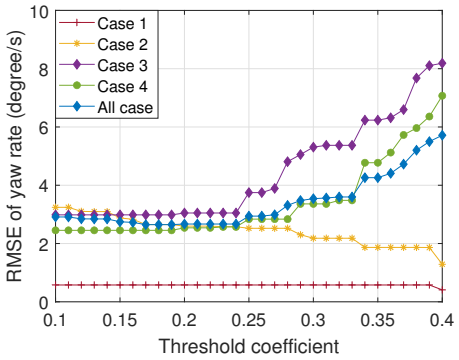


Figure 3.12: RMSE of estimated angle

threshold is higher than the maximum value, leading to large errors. The RMSE of the estimated angle for all cases is shown in Fig. 3.12. As the value of α increased, the RMSE decreased slightly and then increased rapidly. The minimum value of RMSE was 2.6482° when the value of α is 0.19. Therefore, I set the value of α as 0.19 to minimize the RMSE.

The RMSE of the estimated angle for all four cases is listed in Table 3.2, when α was set as 0.19. The rotation angle of the robot could be estimated within 3° of estimation error by using only the radar sensor data. The estimation error was particularly low for case 1 because the location and number of detected targets are relatively unchanging. In a dense clutter environment where there are multiple stationary targets, the rotation angle can be estimated more accurately because the correlation between the detection results increases. In addition, by using a high-resolution radar, it is possible to detect multiple points per target which can improve the estimation performance. On the other hand, when the signal-to-noise ratio is low, the number of detected points will decrease and the estimation performance is expected to degrade.

Estimation Results of Ego-velocity

To estimate the ego-velocity of the robot, the detected points in the angle-velocity domain were used to detect a line. I used the line detection algorithms mentioned

Table 3.2: RMSE of the estimated angle when α is 0.19

Case	**RMSE** (°)
Case 1	0.5774
Case 2	2.6499
Case 3	2.9841
Case 4	2.4528
All case	2.6482

in Chapter 3.1.3. After detecting the line, the robot's ego-velocity was estimated by finding the y-intercept value. Fig. 3.13 shows the detected points when the robot moves forward at a velocity of 0.12 m/s. The estimated ego-velocity was 0.09 m/s when the LLS and RANSAC methods were used. Moreover, when the Hough transform method was used, the ego velocity was estimated as 0.10 m/s, which was similar to the true ego velocity of the robot.

To compare the performance of these three algorithms, I used the data for cases 1 and 2 and calculated the RMSE of the estimated ego-velocity. The minimum velocity of the robot was 0 and the maximum value was 0.54 m/s. The RMSE values for each algorithm are listed in Table 3.3. The robot's ego-velocity was estimated within an

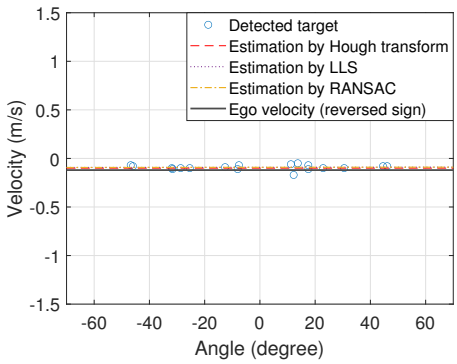


Figure 3.13: Estimation of ego-velocity using the detected points

Table 3.3: Performance of various algorithms

Type Algorithm	**RMSE (m/s)**	**Runtime (s)**
LLS	0.0728	0.0020
Hough transform	0.0702	0.0539
RANSAC	0.0706	0.0003

RMSE error of 0.073 m/s for all three algorithms. In particular, the error was the lowest when the Hough transform was used.

In addition, I calculated the runtime of each algorithm because the execution time is a significant factor in SLAM. As shown in Table 3.3, the runtime was relatively short, and the RANSAC algorithm was the fastest with a runtime of 0.0003 s. Therefore, I believe that the proposed algorithm can be implemented in radar-based SLAM.

3.1.5 Summary

In this study, I proposed a method to estimate the ego-motion of a robot by using only the radar sensor data. I first represented the detected points in the angle-velocity domain, and then transformed them into their corresponding matrix data with values of zero and one. Considering that the points detected in successive time instants have similarities to each other, the correlation between the matrix data was calculated. In addition, the velocity of the robot was estimated by applying a line detection algorithm to the detection results. The accuracy of the proposed estimation method was verified through the data acquired by mounting a radar sensor on a robot. Compared with the angle value obtained from the robot's motor, the proposed correlation-based yaw rate estimation method showed an estimation error of approximately $3^{\circ}/s$. Additionally, the proposed ego-velocity estimation method showed an estimation error of approximately 0.073 m/s. I expect that the proposed method can be used to complement or replace motor or gyro sensor in radar-based SLAM.

3.2 APPLICATION TO AUTONOMOUS VEHICLE

3.2.1 Motivation

To realize autonomous driving, data obtained from automotive sensors such as cameras, lidars, and radars must be used together. The main purpose of using these sensors is to effectively detect targets located around the vehicle. For accurate object detection, it is important to identify the absolute movement of the autonomous vehicle relative to the surrounding environments. For example, the authors in [27] estimated the ego-motion using a single monocular camera. In addition, laser range data is used to estimate the ego-motion of the vehicle in real time [28]. To obtain complete information on ego-motion, the position and speed of the ego-vehicle must be identified. In general, using radar sensor data, it is possible to estimate the relative distance to the target and the relative speed of the target. When considering the position of the ego-vehicle as the origin, the relative distance can be directly converted into the absolute distance. However, to convert the relative speed into the absolute speed, I need to know the speed of the ego-vehicle. Therefore, I propose a method for estimating the velocity of the ego-vehicle using only radar sensor data without the help of other vehicle sensors.

The most important thing in estimating the speed of the ego-vehicle is to identify the targets that are stationary among the detected targets because they can be used as reference points. In other words, the speed relative to the stationary target can be directly converted to the speed of the ego-vehicle. Thus, I use the detection result in the angle-velocity domain to classify the stationary targets. In this domain, the moving targets and the stationary targets exhibit different patterns. In other words, the detection points corresponding to stationary targets exist on a specific curve, but the points corresponding to moving targets randomly scattered on a two-dimensional plane. Therefore, if the curve formulated by points corresponding to the stationary targets is estimated, the absolute velocity of the ego-vehicle can be calculated.

When several points are distributed in a plane, methods such as LLS [23] and RANSAC [29], [30] are widely used to find the tendency of these points. In this work, I use the Hough transform [31] to extract only points corresponding to the stationary targets in the angle-velocity domain. This transformation can be effectively used to find inliers when points are scattered on the two-dimensional plane. Some studies have been conducted using Hough transforms on radar sensor data [32], [33]. For example, a vehicle moving at a constant speed in the time-distance domain was extracted through the Hough transformation in [32]. In addition, patterns of moving and stationary targets were classified by applying the Hough transform to the time-frequency spectrum [33]. When this transformation is applied to detection points located in the angle-velocity domain, the trend line corresponding to the stationary targets can be estimated. Then, from the estimated line, the velocity of the ego-vehicle is calculated directly. To evaluate the estimation performance of the proposed method, radar sensor data obtained from actual road environments are used. In the experiment, FMCW radar is mounted on the vehicle.

The remainder of this study is organized as follows. First, I introduce the automotive radar system used in the experiment. Then, the target detection result in a actual road environment is described. Next, I explain the Hough transform and how to apply it to estimate the absolute speed of the ego-vehicle. Finally, I conclude this study.

3.2.2 Automotive Radar System Used in Experiments

The automotive radar used in this study consists of Tx, Rx, waveform generator (WG), voltage-controlled oscillator (VCO), frequency mixer, LPF, analog-to-digital converter (ADC), and digital signal processor (DSP), as shown in Fig. 3.14. The WG generates a FMCW with a center frequency of f_c, a bandwidth of δf, and a sweep time of δt. As shown in Fig. 3.14, the waveform is composed of an up-chirp and a down-chirp whose frequency linearly increase and decreases with time, respectively. By extracting beat frequencies from each chirp, the relative distance to the target and

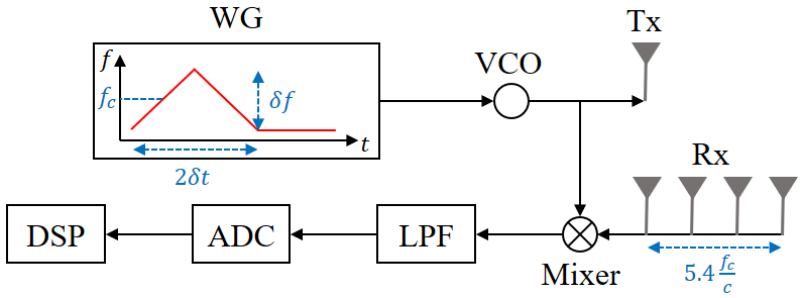


Figure 3.14: Configuration of the 77 GHz automotive radar system

the relative speed of the target can be estimated [34]. In addition, the antenna array is used to estimate the angle of the target, which consists of 4 antenna elements. The distance between the antenna elements is $1.8\frac{c}{f_c}$, where c denotes the speed of light. In the radar system, the Bartlett algorithm [35] is applied to the signals received at Rx to estimate the angle of the target.

3.2.3 Target Detection Result Through Radar Signal Processing

I mounted the automotive radar sensor described on the bumper of the vehicle and conducted experiments on the road. In the measurements, I set the values of f_c, δf, and δt to 76.5 GHz, 500 MHz, and 5 ms, respectively. Fig. 3.15 (**a**) shows one of the road environments where I acquired radar sensor data. In this case, the vehicle speed was maintained at 65 km/h. By processing the radar signal acquired in this environment, I can estimate the relative distance d_j, the relative speed v_j in the radial direction, and the angle θ_j information of the target, where j is the index of each detected target. Then, using the target information represented by the polar coordinate system (i.e., $(d_j,\, \theta_j)$), the positions of the targets can be converted into a two-dimensional Cartesian coordinate system (i.e., $(x_j,\, y_j) = (d_j \sin\theta_j,\, d_j \cos\theta_j)$, as shown in Fig. 3.15 (**b**). The target detection result corresponding to Fig. 3.15 (**a**) is shown in Fig. 3.16. In this figure, the targets including a vehicle located in front and road structures such as guardrails around the ego-vehicle are also detected by the radar sensor.

3.2.4 Stationary and Moving Targets in Angle-Velocity Domain

To estimate the ego-velocity, it is important to select only fixed targets among the detected targets. This is because the relative speed to a stationary target means the absolute speed of the ego-vehicle. Referring to Fig. 3.15 (**b**), the speed of the target vehicle in the y-axis $v_{j,y}$ can be expressed as $v_j \cos\theta_j$. I redraw the detection result in Fig. 3.16 in the angle-velocity (i.e., θ_j and $v_{j,y}$) domain, as shown in Fig. 3.17. In this domain, moving and stationary targets exhibit different characteristics. In other words, the moving targets are randomly placed, but the stationary targets exist around a specific line. Therefore, if the line formed by the stationary targets is estimated, I can discriminate the stationary targets, and it is possible to estimate the absolute velocity of the ego-vehicle.

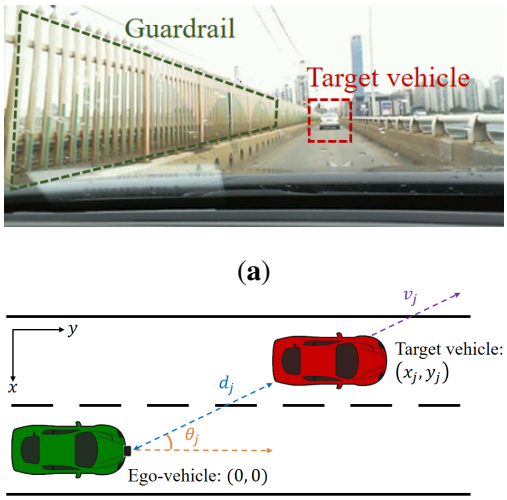


Figure 3.15: Target detection using the automotive radar: (**a**) radar signal measurement on the bridge (**b**) coordinate system conversion of the radar detection result

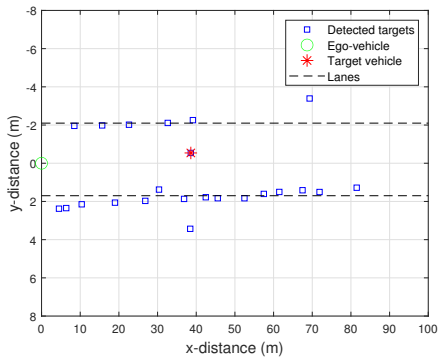


Figure 3.16: Target detection result on the bridge

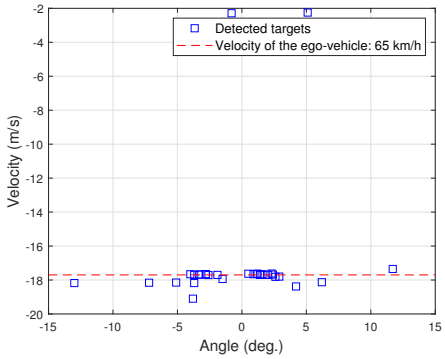


Figure 3.17: Target detection result in the angle-velocity domain

3.2.5 Hough Transform Based Ego-velocity Estimation

In this work, I use the Hough transform to find the tendency from the scattered points in the angle-velocity domain. The Hough transform is widely used to extract linear components from the image. For example, as shown in Fig. 3.18 (**a**), if there are two points in the image space, a straight line with the two points can be expressed in the slope-intercept form as $y = \alpha_0 x + \beta_0$, where α_0 is the slope and β_0 is the y-intercept of the straight line. This line can also be expressed as a single point (α_0, β_0) in the parameter space, as shown in Fig. 3.18 (**b**). In the parameter space, however, a vertical line parallel to the y-axis cannot be represented. Thus, in the Hough transform,

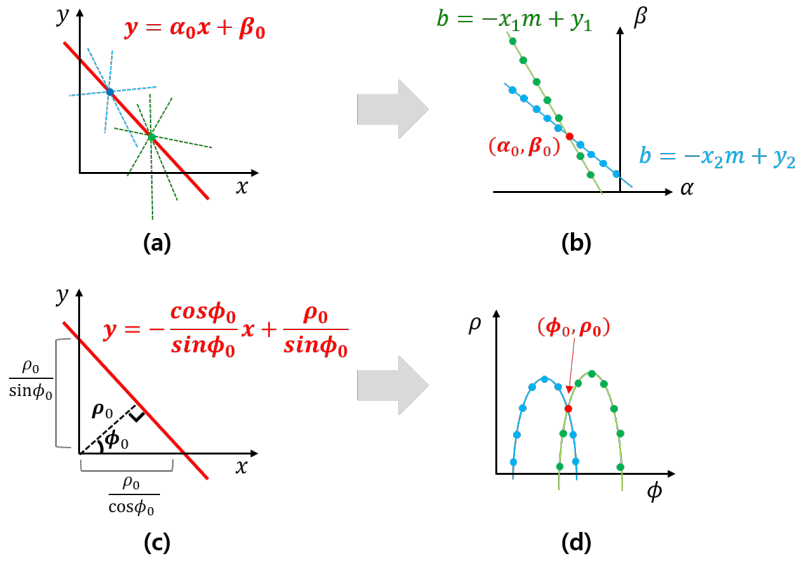


Figure 3.18: Basic concept of the Hough Transform: (**a**) in the x-y domain; (**b**) in the α-β domain; (**c**) in the x-y domain; (**d**) in the ϕ-ρ domain

this straight line is represented using the distance from the origin (ρ_0) and the angle between the x-axis (ϕ_0), as shown in Fig. 3.18 (**c**), which means each straight line in the x-y domain can be expressed as a point in the ϕ-ρ domain. The straight lines passing through each point in Fig. 3.18 (**a**) is expressed as points on each curve in Fig. 3.18 (**d**). Finally, the red points $(\phi_0,\ \rho_0)$ in Fig. 3.18 (**d**) indicates the point corresponding to a straight line passing through two points. Thus, from the point of intersection, I can estimate the straight line formed by the points through the Hough transform.

Fig. 3.19 shows the result of applying the Hough transform to the detection result in the angle-velocity domain of Fig. 3.17. In the Hough transform domain, several curves meet at one point $p_0 = (\phi_0, \rho_0) = (1.56, -17.7)$. Thus, the straight line corresponding to p_0 is the most obvious straight line, which can be expressed as $x\cos\theta_0 + y\sin\theta_0 = \rho_0$ (i.e., $y = -0.01x - 17.7$) in the original image space. I plot the estimated line back on the detection result in the angle-velocity domain, as shown in Fig. 3.20. Because the slope of this line is close to zero, the ego-velocity can

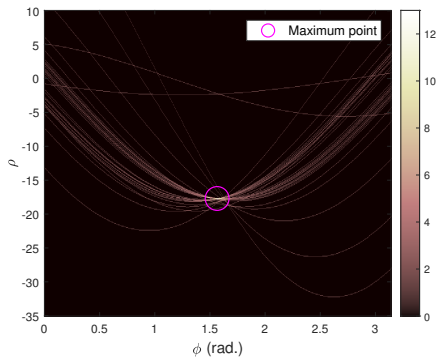


Figure 3.19: Result of applying the Hough transform to the detection result in the angle-velocity domain

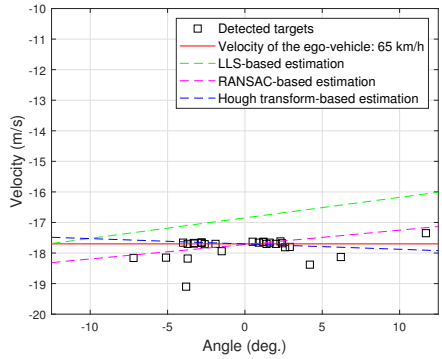


Figure 3.20: Ego-velocity estimation results in the angle-velocity domain

be estimated using the y-intercept. Thus, the absolute speed of the ego-vehicle is estimated as 63.7 km/h and it was estimated very accurately when compared to the actual speed of 65 km/h. Considering the concept of relative velocity, the absolute value of the y-intercept becomes the ego-velocity.

I also estimated the absolute speed of the ego-vehicle by applying the LLS method and the RANSAC algorithm proposed in [36] to the detection result in angle-velocity domain, as shown in Fig. 3.20. In the case of the LLS, the estimation result is not accurate compared to other methods because this method calculates a trend line using both detection results corresponding to stationary and moving targets. The RANSAC

algorithm finds the trend line quite accurately. However, in RANSAC-based methods, parameter values used in the algorithm, such as the minimum sample size and maximum distance, must be set empirically, which is not required in the Hough transform. Finally, I verified the performance of the proposed method by changing the speed of the ego-vehicle, as shown in Table 3.4. The estimation accuracy decreased slightly as the speed increased, but the maximum error was about 2.4% of the actual speed. Through the proposed method, it is possible to accurately estimate the ego-velocity by automatically selecting points corresponding to stationary targets among several detection points.

Table 3.4: Ego-velocity estimation results according to the velocity of the ego-vehicle

Actual velocity (km/h)	35.0	45.0	55.0	65.0	75.0	85.0
Estimated velocity (km/h)	34.7	44.3	54.0	63.7	73.5	83.0

3.2.6 Summary

In this study, I proposed a method for estimating the absolute speed of the ego-vehicle using the radar detection result. To estimate the speed of the ego-vehicle, it is important to identify stationary targets from the radar detection results. Thus, I used the distribution of targets in the angle-velocity domain. In this domain, stationary targets existed on a specific line, and the Hough transformation was used to estimate the trend line. I verified the performance of the proposed method with the radar data measured on the actual road. Through the Hough transform, inliers were effectively extracted among all the detection points. The proposed method has a maximum error of 2.4% in the speed range of 30-90 km/h.

Chapter 4

VEHICLE ORIENTATION ESTIMATION

4.1 RANGE-ANGLE MAP BASED APPROACH USING MIMO FMCW RADAR

4.1.1 Motivation

Among the various autonomous driving functions, it is important to predict the movement of the front vehicle. The information about the front vehicle's movement can be used to automatically adjust the speed of the ego-vehicle or to select appropriate actions. In this regard, the authors in [37] used a mono camera to estimate the heading direction of the front vehicle. Also, in [38], the heading information obtained from a lidar sensor was used for tracking multi-targets. However, to the best of my knowledge, no research has been performed to identify the motion of a front vehicle using a radar sensor alone.

Therefore, in this study, I propose a method for estimating the instantaneous heading direction of the front vehicle by applying a machine learning algorithm to the automotive radar sensor data. I use an FMCW MIMO radar system and collect data by measuring radar signals for various movements of the front vehicle. Specifically, I accumulate radar sensor data when the vehicle in front is stationary, going straight, go-

ing backward, turning left, and turning right. Then, I use a CNN [39] to classify these five different movements. The CNN is a widely used deep learning algorithm that has recently been applied to radar sensor data. For example, the authors in [40] attempted to monitor parking spaces by applying CNN to the radar-image data. In addition, in [41], various hand gestures were classified by applying 3D CNN to the processed radar data. Because the CNN uses the image format as an input data, it is essential to convert the radar signal received by the MIMO antenna into an image format. Therefore, I transform a 3D radar data cube composed of distance, velocity, and angle into an intuitive 2D range-angle map and use it as input to the network. The CNN structure was determined by testing the performance of the network for various combinations of the number of blocks and filters. Using the trained model, the proposed algorithm was able to estimate the heading direction of the vehicle with an accuracy higher than 94%.

The remainder of this study is organized as follows. In Chapter 4.1.2, the experiment environment is described. Then, in Chapter 4.1.3, I introduce how to convert radar data into an image format suitable for CNN input. In addition, the framework of the CNN is discussed. Next, the classification performance when the proposed network is trained with the acquired radar data is presented in Chapter 4.1.4. Finally, conclusions are given in Chapter 4.1.5.

4.1.2 Radar Signal Measurement Scenarios

In the experiment, I used the AWR1642BOOST [42] automotive radar sensor evaluation kit, manufactured by Texas Instruments. The radar kit is connected to the DCA1000EVM module to capture the data, as shown in Fig. 4.1. The radar system parameters are shown in Table 4.1. I installed the radar 1 m above the ground in an outdoor parking lot. As shown in Fig. 4.2b, the areas in front of the radar kit are divided into nine zones. The size of the zone is set as $4\,\mathrm{m} \times 4\,\mathrm{m}$ by considering the width of vehicle lanes and size of the vehicle, and safety corns are placed at the vertex of each

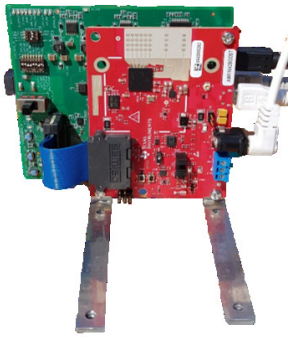
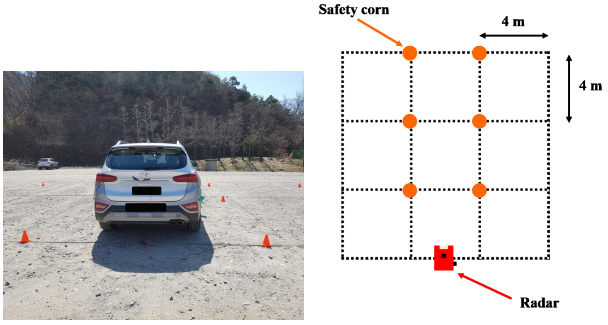


Figure 4.1: AWR1642BOOST kit with DCA1000EVM module

zone. The size of the zone did not affect the outcome provided that the full image of a front vehicle was obtained. To distinguish the different movements of the vehicle, the experiments are divided into five different scenarios, as shown in Fig. 4.3. In scenario A the vehicle is stationary, in scenarios B and C the vehicle is driving forwards and backwards, and in scenarios D and E the vehicle is turning left and right, respectively. In all measurements, the vehicle speed was in the range of 0-15 km/h. The total number of measurements for the five different scenarios was 60. A single measurement consists of 200 data cubes, so I collected a total of 12000 data cubes.



(a) Actual camera image (b) Bird's-eye view

Figure 4.2: Experimental environments in a parking lot

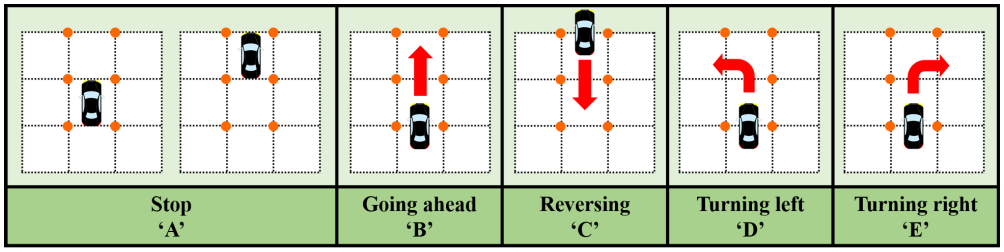


Figure 4.3: Experiment scenarios

Table 4.1: Radar system parameters

Parameter	Value	Parameter	Value
Number of Tx antennas (N_T)	2	Operating bandwidth (B)	1.8 GHz
Number of Rx antennas (N_R)	4	Sweep time (T_{sw})	60 μs
Tx antenna spacing (d_T)	2λ	Sampling period (T_s)	0.1 ns
Rx Antenna spacing (d_R)	0.5λ	Number of chirps (N_c)	128
Carrier frequency (f_c)	78 GHz	Number of samples per chirp (N)	256

4.1.3 Proposed CNN-Based Heading Direction Estimation Method

Generation of Input Data for CNN

Because the CNN uses an image data as input, it is important to convert the radar signal into an image suitable for learning. In this study, I use a 2D range-angle map as the network input. Therefore, it is important to obtain a range-angle information at a high resolution. I can express the radar signal as a 3D data cube for the n, p, and l axes, as shown in Fig. 4.4. The range and velocity information of the target can be estimated by applying FFT to the n and p axes. Similarly, by applying FFT to the l-axis, the angle of the target can be estimated.

However, when the angle of the target is estimated through FFT, the angular resolution is limited by the number of antenna elements. Each antenna element is regarded as a sampled value, and because the number of antenna elements is generally smaller

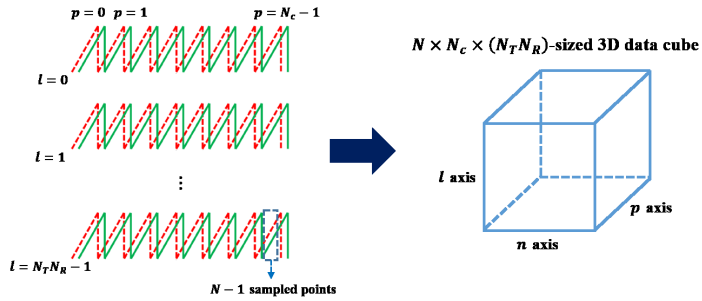


Figure 4.4: 3D data cube generated from the received radar signal

than the number of samples per chirp or the number of chirps, the angular resolution is considerably worse than the range or velocity resolution. For example, the number of virtual antenna elements is 8 in the MIMO system, whereas the number of chirps is 128 and the number of samples per chirp is 256. Therefore, instead of applying the FFT to the l-axis, I use the multiple signal classification (MUSIC) [43] algorithm, which is a high-resolution frequency estimation algorithm that uses an eigenspace method. The MUSIC algorithm first computes a covariance matrix using the received signal samples. Then, by performing eigen-decomposition to the covariance matrix, the eigenvectors corresponding to signals and noise are obtained. The MUSIC algorithm uses the idea that any signal vector belonging to the signal subspace should be orthogonal to the noise subspace. By using this orthogonality condition, a MUSIC pseudo-spectrum is formed such that the peak occurs at the target's angle.

The overall procedure of the proposed method is summarized in Fig. 4.5. First, I apply the FFT to the n-axis to extract the range information of the target. If the number of FFT points is N_n, the size of the data after FFT becomes $(N_T N_R) \times N_n$. Then, I apply the MUSIC algorithm to the l-axis for every range index. Through this process, the frequency along the n-axis and l-axis are extracted and I can create a high-resolution range-angle map. Figure 4.6 shows the range-angle map and converted x-y range map when the front vehicle turns left. The value of each pixel represents the

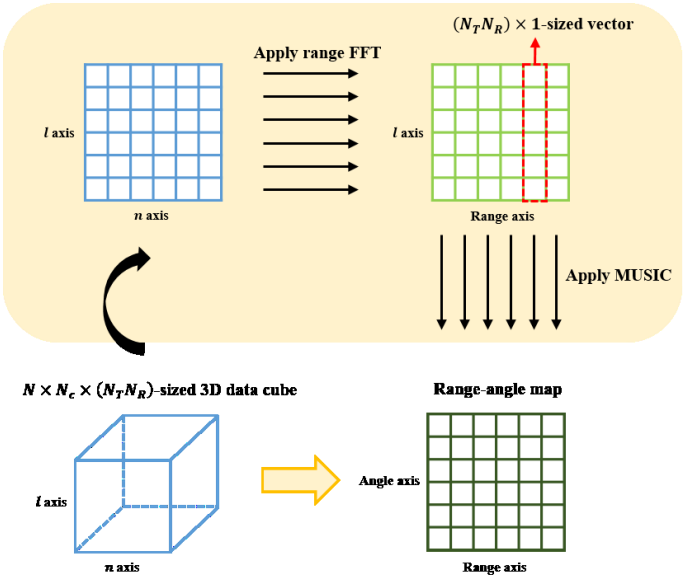
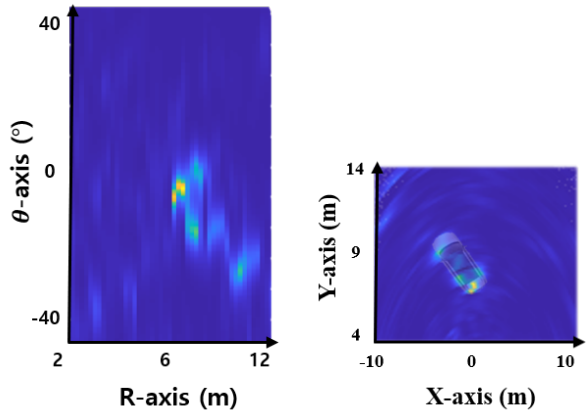


Figure 4.5: Range-angle map generation using the radar data cube

amplitude of the signal. As shown in the figure, the signals are strongly reflected by the wheels [44, 45], and I can identify the instantaneous heading direction of the vehicle through this image. Therefore, I use this single high-resolution range-angle map as the input to the CNN and train the network.

Framework of CNN

The structure of the CNN used in this study is shown in Fig. 4.7. The range-angle map data shown in Fig. 4.6a is used as an input to the network. First, a normalization is applied to the input data to center each data at the origin. Then, the input data is passed through multiple blocks of convolutional layers and pooling layers. In the convolutional layer, the input data is convoluted with filters to extract features of the image. In addition, a zero padding is applied to maintain the dimension of the output equal to that of the input. The size of the filter is 3×3, and the number of filters is 2^n. The appropriate number of blocks and filters will be discussed in Chapter 4.1.4. Next,



(a) Range-angle map (b) x-y range map

Figure 4.6: Data when the front vehicle turns left: Scenario D

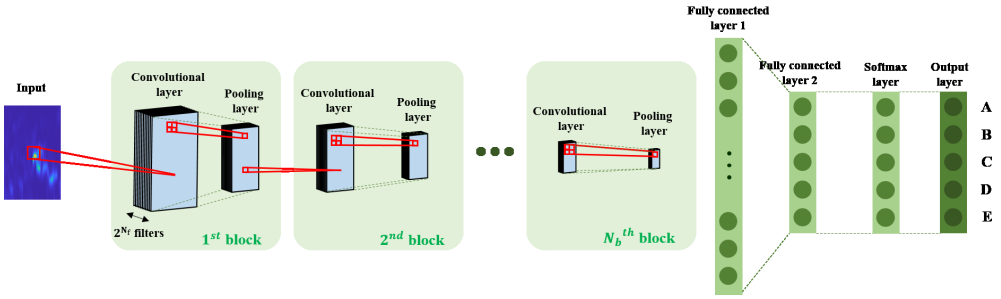


Figure 4.7: CNN architecture of the proposed method

the output of the convolutional layer passes through the rectified linear unit (ReLU) layer to provide nonlinearity to the network. Then, max pooling is performed, which is a down sampling technique to prevent over-fitting. Through this process, the size of the input data is reduced and features are extracted from the image data.

Next, a classification process is performed through fully connected layers and a softmax layer. I set the number of nodes as 1024 in the first fully connected layer and five in the second fully connected layer, representing the five different scenarios. Then, the real numbers of the output from the fully connected layer are converted

into probability values through the softmax layer. These probability values are used to predict the instantaneous heading direction of the vehicle at the output layer.

When the gradient of the loss function is calculated by backward propagation, the weight parameters are updated in the direction to minimize the loss function. Through an iterative update process, the optimal weight parameters are found where the loss function is minimum. In this study, the loss function is calculated by using the cross entropy, and gradient descent algorithm is used to minimize the loss function.

4.1.4 Performance Evaluation

In this study, I collected 12,000 data from experiments during two days. Among the 12,000 data, I extracted 11,748 data by removing the ones that do not belong to one of the 5 cases. A high-resolution range-angle map was obtained by applying the FFT to the n-axis of the data cube and then by applying the MUSIC algorithm to the l-axis, as explained in Chapter 4.1.3. I used the cell-averaging constant false alarm rate (CA-CFAR) [46] algorithm to detect the location of the target, and extracted the range-angle map around the target by applying a rectangular window function, as shown in Fig. 4.6a. The size of the windowed image was 127×51, which was used as input to the CNN. Then, I randomly extracted 70% of the total 11,748 data and used it as training set, 15% as validation set, and 15% as test set. The training set was equally extracted from each of the 5 cases so that the training data is not biased to a specific case. The training set was divided into multiple batches, each consisting of 300 data samples. A single batch represents a set of samples used to update the weight parameters. The learning rate was set as 0.001 and the number of epochs was set as 40. To reduce the variance of the estimated classification accuracy, this training process was repeated 10 times using the Monte Carlo method [47]. I trained the network with 10 independent data sets and derived the final classification accuracy by averaging the results. In this study, I used the MATLAB software for training the network.

First, to find the appropriate network parameters, I varied the number of blocks and filters and compared the performance of the network. The computational complexity of the CNN algorithm is affected by the number of blocks and filters. Therefore, these parameters should be appropriately set to lower the computational complexity while maintaining the classification accuracy high. The number of filters was increased from 2 to 128 in units of 2^n, and the number of blocks was increased from 1 to 4. The classification results are shown in Fig. 4.8. When the number of filters was 2, the classification accuracy was the lowest, and it tended to decrease as the number of blocks increased. In contrast, when the number of filters was higher than 2, the network showed the best performance when the number of blocks was 2. Therefore, I set the number of blocks as 2. In addition, the classification accuracy increased when more filters were used, but the increase was not significant when more than 32 filters were used. As a result, I set the number of filters as 32 considering the computational complexity.

Next, I examined the classification accuracy of the training and validation sets when the number of blocks and filters are 2 and 32, respectively. The classification accuracy and loss function according to the number of iterations are shown in Fig. 4.9. I set the maximum number of iterations as 1,080, and validation was performed every 20 iterations. As can be seen from the figure, the network was fully trained at the 780th

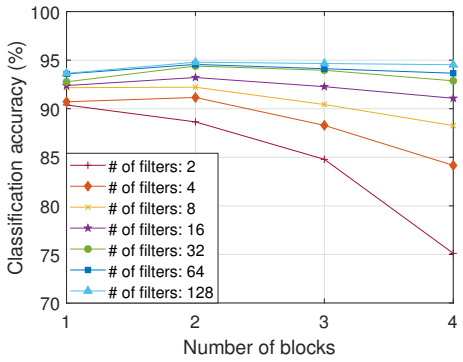
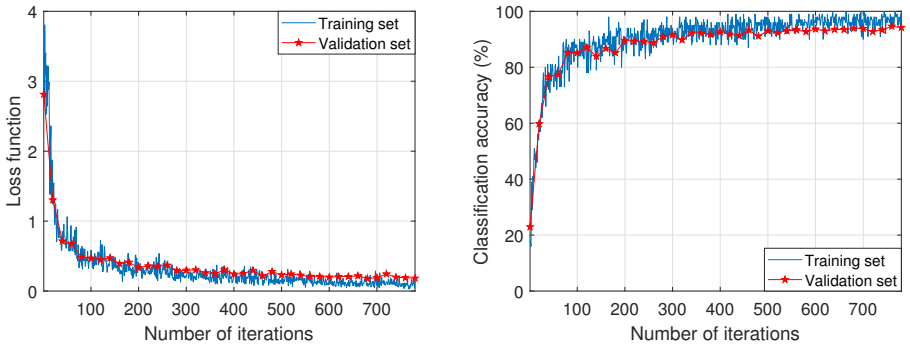


Figure 4.8: Classification accuracy according to the number of filters and blocks

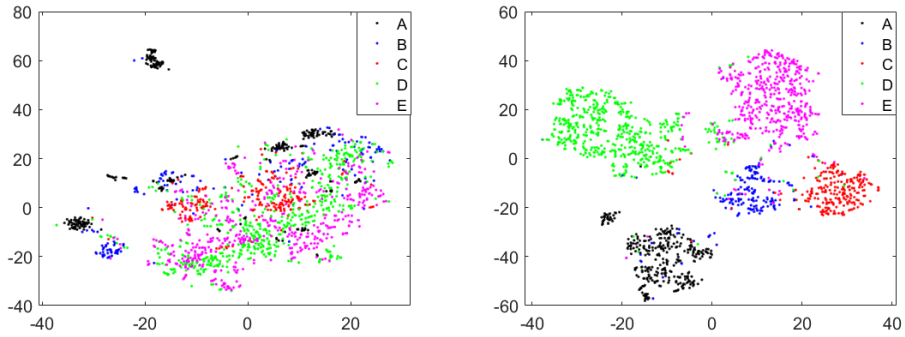


(a) Loss when cross entropy is used (b) Classification accuracy

Figure 4.9: Loss and classification accuracy according to the number of iterations

iteration and training has stopped at this point. Both graphs show that the training and validation sets have similar training curves. This indicates that the proposed network was well trained and over-fitting was prevented.

In addition, I analyzed the performance of the proposed network by using t-distributed stochastic neighbor embedding (t-SNE) [48] algorithm, which is a non-linear data reduction method. This method visualizes the high-dimensional data in low-dimensional subspace. The t-SNE representation of the raw data is shown in Fig. 4.10a, and that



(a) Projected data before applying CNN (b) Projected data at fully connected layer 2

Figure 4.10: Visualization of the data using the t-SNE algorithm

of the output at the second fully connected layer is shown in Fig. 4.10b. It is evident that the boundary between clusters becomes clear when using the output data from the trained network.

Moreover, Table 4.2 shows the confusion matrix indicating the classification results of the proposed network. Scenarios A, D, and E showed classification accuracy higher than 94%, whereas scenarios B and C showed low classification accuracy. Scenario A showed high classification accuracy since the vehicle is stationary and there is little fluctuations in the received signal. In addition, scenarios D and E resulted in high classification accuracy because the vehicle is rotating and more parts of the vehicle is illuminated by the radar. A high-intensity signal is reflected by the wheel, and the vehicle's motion can be easily detected. In contrast, when the vehicle is moving forwards or backwards, the area of illumination is narrow compared to when the vehicle is rotating, and there is little signal reflected by the wheel. As a result, the vehicle's movement was not easily recognizable and these cases showed low classification accuracy. The overall classification accuracy of the five scenarios was 94.44%. Therefore, I believe that this proposed algorithm can effectively estimate the instantaneous heading direction of the front vehicle.

Table 4.2: Confusion matrix derived from CNN

Estimated class / Actual class	**A**	**B**	**C**	**D**	**E**
A	99.39%	7.39%	0.87%	0.73%	0.63%
B	0%	82.95%	2.60%	0.73%	1.88%
C	0%	0.57%	90.91%	0.73%	0.63%
D	0.61%	6.82%	4.76%	96.35%	2.09%
E	0%	2.27%	0.87%	1.46%	94.78%

4.1.5 Summary

In this study, I proposed a CNN-based method for estimating the instantaneous heading direction of the front vehicle using an automotive radar sensor. The 77 GHz FMCW radar was installed in an outdoor parking lot, and experiments were conducted for five different scenarios by changing the heading direction of the front vehicle. I converted the received data into a 2D range-angle map by applying the FFT and MUSIC algorithm along the range and angle axes. Then, I used the CA-CFAR algorithm to estimate the location of the target, and extracted the range-angle map around the target by applying a rectangular window function. This windowed image was used as an input to the CNN. The structure of the CNN was determined by finding the number of blocks and filters that results in high classification accuracy while maintaining low computational complexity. The classification results showed that this proposed method can effectively estimate the instantaneous heading direction of the front vehicle with a high accuracy. For future work, a series of imgaes can be used as an input to improve the estimation accuracy, or a tracking algorithm can be used to obtain more information about how the front vehicle is moving. Furthermore, the estimation accuracy can be improved by using the velocity information as well as the range-angle information.

4.2 POINT CLOUD DATA BASED APPROACH USING 4D IMAGING RADAR

4.2.1 Motivation

In a complex environment such as a highway or urban road, various targets exist and unpredictable situations can occur. In such dynamic environments, predicting the path of a surrounding vehicle is important for identifying potential hazards in advance. When the surrounding vehicle is moving, its path can be predicted by applying a tracking algorithm to estimate the trajectory. However, when the vehicle is stationary, it is difficult to predict the path of the vehicle in advance.

Another method of predicting the path of the vehicle is by utilizing the vehicle orientation information, which can be applied to both stationary and moving vehicles. The vehicle orientation information can also be used to estimate the velocity vector of the target [49], or in vehicle re-identification task which is the problem of identifying the same vehicle from multiple images [50]. To this end, studies have been conducted to estimate the vehicle orientation using sensors such as camera [51] or lidar [52]. However, the main drawback of these light-based sensors is that their performance is significantly affected by ambient light or weather conditions. Conversely, radar sensors are robust against environmental changes while being cost-effective and are capable of measuring long distances. In addition, the target detection performance of radar can be significantly enhanced using a MIMO system [53, 54].

There has been ongoing research to estimate vehicle orientation using radar sensors. For example, the contour of a vehicle was estimated using synthetic aperture radar (SAR) algorithm in [55], where the possibility of estimating the vehicle orientation has been demonstrated. In addition, in [56], the velocity and orientation of the vehicle were estimated applying RANSAC to the detection results in the angle-Doppler domain. Further, the authors in [57] proposed a method to estimate the instantaneous heading direction of a vehicle using a range-angle image as input to the CNN. However, the

movement of the vehicle was restricted to going forward, backward, and turning left and right, and the orientation angle of the vehicle was not considered. In [58], the vehicle orientation angle was estimated with a RMSE of 9.77° by using two radar sensors. However, the stationary vehicles were not considered, and the method involves high computational complexity because the iterative method is used to determine the rotation angle.

Therefore, in this study, I propose a method of estimating the vehicle orientation using a 77 GHz cascaded MIMO FMCW radar system. The radar data is acquired by varying the orientation angle of the vehicle, and signal processing is applied to transform the raw data into point cloud data which reveals the shape and size of objects. Then, the generated point cloud data are further processed to remove redundant data which are irrelevant for vehicle orientation estimation. The processed point cloud data were distributed along the axis of vehicle orientation; therefore, I applied regression algorithms such as the principal component analysis (PCA), decision tree, and CNN to find the trend line of point cloud data. Moreover, the performance of the proposed method was compared with the iterative methods used in [58]. The comparison of various estimation methods revealed that the proposed method of using the CNN framework is most suitable for vehicle orientation estimation.

The contribution of this work can be summarized as follows. First, the proposed method can be applied regardless of whether the vehicle is moving or not. Unlike the tracking-based method which requires the movement of the vehicle, the proposed method is based on instantaneous reflections from the vehicle and can be applied to stationary vehicles as well. Next, the proposed method is robust to outliers resulting from noise or undesired reflections. This is because the vehicle orientation is determined using the overall distribution of the data rather than examining individual points. Moreover, the proposed method is a single radar solution for estimating the vehicle orientation and does not require additional sensors such as camera or lidar. The estimation accuracy can be further enhanced through sensor fusion; however, combining multiple

sensor data requires increased complexity. Thus, the use of single high-resolution radar sensor can reduce the overall complexity of the system. Also, the regression model can be trained in advance and estimation can be performed in real-time by using the pre-trained regression model with tuned hyperparameters. To the best of my knowledge, estimation of vehicle orientation using point cloud data with machine learning algorithms has not been sufficiently addressed in the related literatue. The proposed method was effectively applied to the existing commercial radar system and the vehicle orientation was reliably estimated.

The remainder of this study is organized as follows. In Chapter 4.2.2, the basic principles of the MIMO FMCW radar system are explained. Then, the experimental environment is described, and the method for processing the measured data is discussed in Chapter 4.2.3. In Chapter 4.2.4, three different regression methods for estimating vehicle orientation are introduced. Next, the estimation results obtained from each method are presented along with a comparison of the performances in Chapter 4.2.5. Finally, conclusions are given in Chapter 4.2.6.

4.2.2 Basic Principles of Cascaded MIMO FMCW Radar System

Using FMCW radar system, the range, velocity, and angle information of the target can be obtained, as mentioned in Chapter 2.1. The angle of the target can be estimated utilizing the phase information of the MIMO antenna system. The antenna elements placed in the horizontal direction are used to estimate the azimuth angle of the target, whereas those placed in the vertical direction are used to estimate the elevation angle of the target. The estimated azimuth and elevation angles are used to determine the location of the target as follows:

$$\begin{aligned} x_n &= R_n \sin\varphi_n \cos\theta_n, \\ y_n &= R_n \sin\varphi_n \sin\theta_n, \\ z_n &= R_n \cos\varphi_n, \end{aligned} \tag{4.1}$$

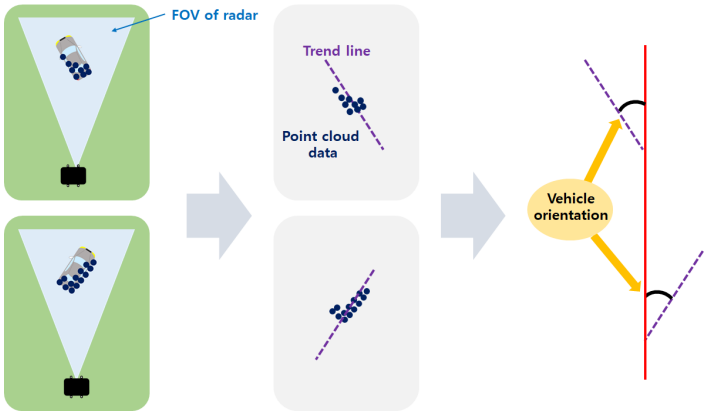


Figure 4.11: Point cloud data according to vehicle orientation

where φ_n and θ_n are the azimuth and elevation angles of the n_{th} target. The detection results in spherical coordinates are converted into Cartesian coordinates and the relative position between the radar and target is estimated.

Furthermore, multiple radar chips can be cascaded to increase the number of Tx and Rx antennas. The number of virtual antennas and aperture size is increased accordingly and the angular resolution is enhanced in both the azimuth and elevation angle domains. Using a cascaded MIMO FMCW radar system, a single object can be detected as multiple points and the point cloud data are formed which reveals the shape and size of the target. Even if commercial radar with fewer antenna elements are used, the point cloud data can be obtained by applying high-resolution angle estimation algorithms or by fusing the detection results of multiple radars [58]. Therefore, the orientation angle of the vehicle can be estimated using the distribution of point cloud data. For example, the point cloud data of a stationary vehicle is shown in Fig. 4.11 when it is oriented to the left and right. The point cloud data are distributed along a specific trend line depending on the vehicle orientation. Therefore, the orientation angle can be estimated by applying regression algorithms to the point cloud data, which will be discussed in Chapter 4.2.4.

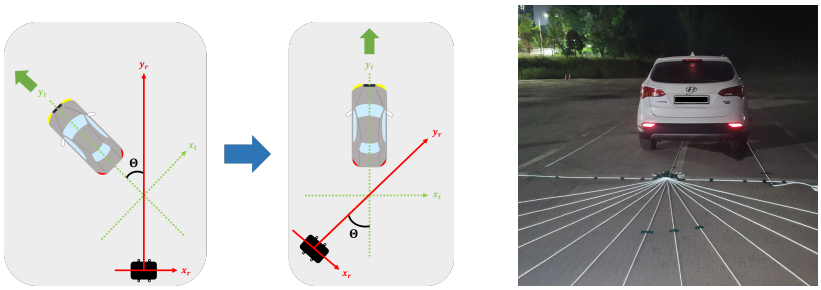
4.2.3 Measurement Using MIMO FMCW Radar

Experimental Environment

In the measurement, I used the RETINA 4F radar module manufactured by the Smart Radar System, as shown in Fig. 4.12. The detailed specifications of the radar system are listed in Table 4.3. The radar uses four cascaded AWR1243 chips manufactured by Texas Instruments. Each chip has three Tx antennas and four Rx antennas; therefore, the total number of Tx and Rx antennas is 12 and 16, respectively. These MIMO antennas form 192 virtual antennas, resulting in angular resolutions of 2° and 4.7° in the azimuth and elevation directions, respectively. The data capture rate is eight scans per second, and the maximum number of points per scan is 6,114 points.

When conducting the measurements, the orientation angle of the vehicle needs to be varied while the boresight direction of the radar is fixed. However, it is difficult to precisely adjust the orientation angle of a vehicle in a real environment. Therefore, I fixed the orientation angle of the vehicle and varied the boresight direction of the radar, as shown in Fig. 4.13. Specifically, the angle was varied from -60° to 60° in units of 10°. For each measurement, the vehicle repeated two round-trip movements (stop - going forward - stop - going backward). The velocity of the vehicle was in the range of 0-10 km/h during the measurements. I conducted five measurements for each orientation angle; therefore, the total number of measurements was $5 \times 13 = 65$.

Figure 4.12: RETINA 4F radar module (Smart Radar System)



(a) Measurement principle

(b) Photo of experimental environment

Figure 4.13: Experimental environment

Signal Pre-processing of Measured Data

Fig. 4.14 shows the overall block diagram of signal pre-processing which transforms the data into a format suitable for vehicle orientation estimation. The first step of the proposed framework consists of generating point cloud data from the raw data. A two-dimensional FFT, which consists of range-FFT and Doppler-FFT, is applied and a range-Doppler spectrum is obtained. Thereafter, CFAR algorithm [59] is used to extract the peak values corresponding to the target. This process is repeated for all virtual

Table 4.3: Specifications of RETINA 4F radar module

Parameter	**Value**	**Parameter**	**Value**
Mode	Short range radar	Number of Tx antennas	12
Center frequency	77 GHz	Number of Rx antennas	16
Bandwidth	400 MHz	Azimuth FOV	$\pm 50°$
Chirp duration	42 μs	Azimuth angle resolution	2.0°
Maximum range	118 m	Elevation FOV	$\pm 12°$
Range resolution	0.46 m	Elevation angle resolution	4.7°
Maximum velocity	82.9 km/h	Update-rate (per second)	8 scans
Velocity resolution	0.22 km/h	Point cloud output (per scan)	max 6,144 points

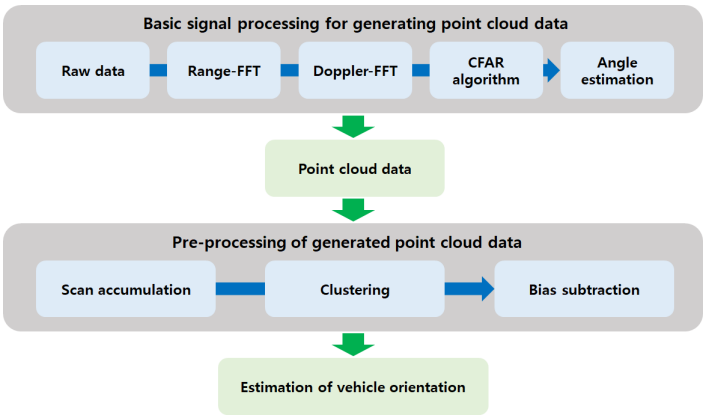
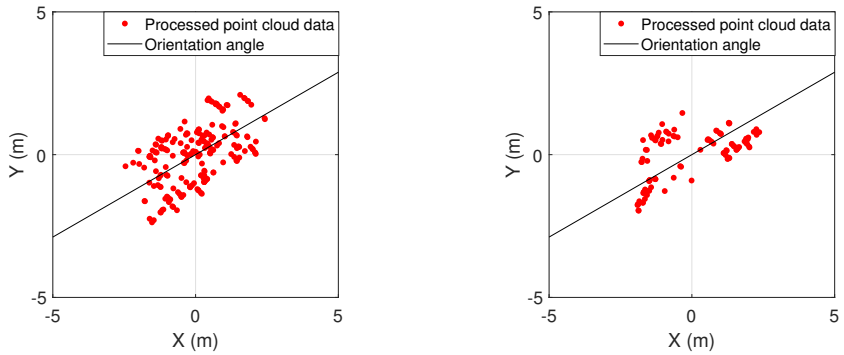


Figure 4.14: Block diagram of signal pre-processing

antennas and the azimuth and elevation angles are estimated for each target. Then, the point cloud data are generated using the coordinate transformation described in (4.1).

The next step involves processing the generated point cloud data to remove redundant data which are irrelevant for vehicle orientation estimation. First, I accumulate the data for multiple scans prior to applying the clustering algorithm. In other words, the point cloud data obtained for consecutive scans are gathered together to form a single data, which significantly improves the clustering performance compared to the case of using single scan. Next, a clustering algorithm is applied to remove ghost targets which result from noise and multipath reflections. I used the density-based spatial clustering of applications with noise (DBSCAN) algorithm [60], which is an unsupervised algorithm that is robust to outliers. The algorithm does not require prior information about the number of clusters and all points do not need to be allocated within the cluster. The hyperparameters of the DBSCAN algorithm, which are the search radius and minimum number of points, were set as 2 m and 5 by considering the size of the vehicle. Lastly, the mean value is subtracted to remove the range bias between the vehicle and radar. The resulting processed data are shown in Fig. 4.15 when the orientation angle of the vehicle is 60°. The figure clearly reveals that the point cloud data are distributed along the axis of vehicle orientation regardless of whether the target vehicle is stationary or



(a) Stationary case

(b) Non-stationary case

Figure 4.15: Processed point cloud data (Orientation angle: 60°)

not. Therefore, this processed point cloud data will be used to estimate the orientation angle of vehicle in Chapter 4.2.4.

4.2.4 Vehicle Orientation Estimation Techniques

In this study, a method of estimating the orientation angle of vehicle using the processed point cloud data is presented. Among the x, y, and z coordinates of the point cloud data, the x and y coordinates are used for estimation because those coordinates are crucial for determining the orientation angle. The x and y values are extracted from the point cloud data and regression algorithms such as PCA, decision tree, and CNN are applied, as illustrated in Fig 4.16. The PCA-based method will be explained, followed by the decision tree-based method and the CNN-based method.

PCA-based Orientation Estimation

PCA [61] is a statistical algorithm used to analyze multi-dimensional data, which has found applications in areas such as dimension reduction [62], denoising [63], and regression [64]. In this study, the PCA algorithm is employed to find the regression line of point cloud data. The algorithm finds a set of principal axes that are perpendicular

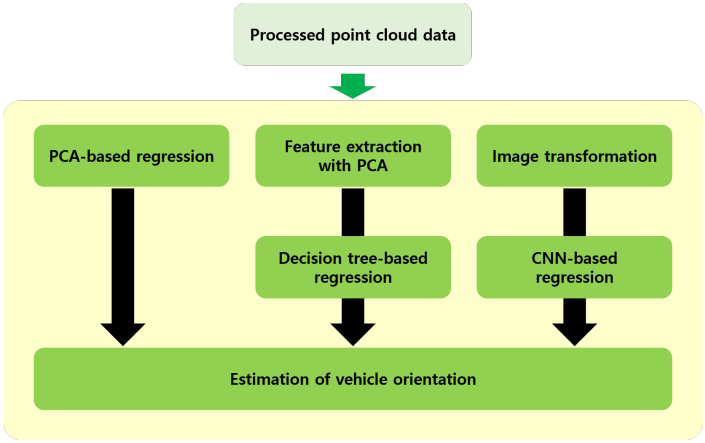


Figure 4.16: Block diagram of vehicle orientation estimation

to each other. The first principal axis is determined by finding the axis that accounts for the largest variance of the data. The second principal axis accounts for the largest variance of the data under the condition of being perpendicular to the first principal axis. This process is repeated until the N_d principal axes are found, where N_d is the dimension of the data.

The principal axes can be determined by applying eigen decomposition to the covariance matrix of the data. When the x and y values of the point cloud data with N_d points are arranged in matrix form as

$$\mathbf{D} = \begin{bmatrix} x_1 & y_1 \\ x_2 & y_2 \\ \vdots & \vdots \\ x_{N_d} & y_{N_d} \end{bmatrix}, \tag{4.2}$$

the covariance matrix of (4.2) can be written as

$$\mathrm{Cov}(\mathbf{D}) = \mathrm{E}[(\mathbf{D} - \mathrm{E}[\mathbf{D}])^T(\mathbf{D} - \mathrm{E}[\mathbf{D}])], \tag{4.3}$$

where $\mathrm{E}[\cdot]$ and $(\cdot)^T$ denote the expectation and transpose operators, respectively. By

applying eigen decomposition to (4.3), the covariance matrix is diagonalized as

$$\text{Cov}(\mathbf{D}) = \begin{bmatrix} \mathbf{v}_1^T & \mathbf{v}_2^T \end{bmatrix} \begin{bmatrix} \lambda_1 & 0 \\ 0 & \lambda_2 \end{bmatrix} \begin{bmatrix} \mathbf{v}_1 \\ \mathbf{v}_2 \end{bmatrix}$$
$$(\lambda_1 \geq \lambda_2). \tag{4.4}$$

Here, λ_1 and λ_2 are the eigenvalues of the covariance matrix in descending order, and $\mathbf{v}_1$ and $\mathbf{v}_2$ are the corresponding eigenvectors. The eigenvectors correspond to the principal axes and the eigenvalues indicate the extent to which the variance is explained by the corresponding eigenvector. Consequently, $\mathbf{v}_1$ is the axis that contains the largest variance of the data and can be regarded as a linear regression result. Therefore, the vehicle orientation can be estimated using the first principal axis as follows:

$$\Theta_p = \arctan\left(\frac{\mathbf{v}_1(1)}{\mathbf{v}_1(2)}\right), \tag{4.5}$$

where $\arctan(\cdot)$ is the arctangent operator, and $\mathbf{v}_1(1)$ and $\mathbf{v}_1(2)$ are the first and second components of vector $\mathbf{v}_1$, respectively. Because the orientation angle is measured from the y-axis, the arctangent function is applied to $\mathbf{v}_1(1)/\mathbf{v}_1(2)$ instead of $\mathbf{v}_1(2)/\mathbf{v}_1(1)$.

Decision Tree-based Orientation Estimation

Among the various machine learning algorithms used for data prediction, the decision tree [65] has the advantage of being fast and simple. In addition, by combining multiple decision trees, the error caused by the variance of data can be reduced, which is known as the ensemble algorithm. The bagging algorithm is a widely used ensemble algorithm that improves stability and prediction accuracy. The block diagram of the bagging algorithm using the decision tree as the basic block is shown in Fig. 4.17. The dataset is divided into multiple sample data through random sampling with replacement. These sample data are used to train multiple decision trees in parallel, and the final prediction result is derived by averaging or voting the prediction results from each decision tree.

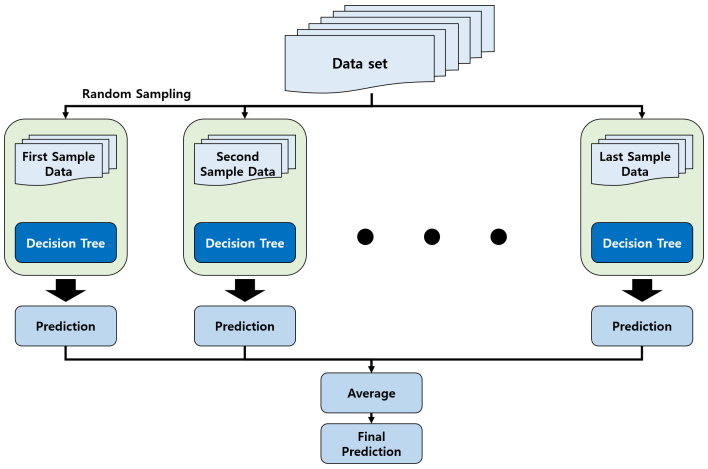


Figure 4.17: Block diagram of bagging algorithm

When applying machine learning algorithms such as the decision tree, the feature parameters need to be extracted to represent the characteristics of the data. As a candidate parameter, I considered seven parameters obtained through the PCA algorithm in Chapter 4.2.4, which can be expressed as

$$\mathbf{f} = [\bar{\lambda}_1,\, \bar{\lambda}_2,\, \mathbf{v}_1(1),\, \mathbf{v}_1(2),\, \mathbf{v}_2(1),\, \mathbf{v}_2(2),\, \Theta_p], \tag{4.6}$$

where $\bar{\lambda}_1$ and $\bar{\lambda}_2$ denote the normalized eigenvalues. Among these candidate parameters, I performed the feature selection to eliminate redundant parameters that are irrelevant to data prediction. Then, the extracted parameters are used as inputs to the decision tree.

CNN-Based Orientation Estimation

CNN is a representative deep learning algorithm that does not require feature extraction. Because the CNN uses image data as input, I converted the point cloud data into an image format suitable for the CNN input. In image transformation, I used a 2D histogram to convert each point cloud data into a matrix of equal size. For both x and y values, the range was set from -5 m to 5 m and the bin size was set as 0.1 m.

Consequently, a matrix with a size of 101×101 was generated, where the elements of the matrix represent the number of points in each bin. Then, I normalized the matrix such that the elements are in the range of zero to one,

$$\mathbf{M}_{\text{normalized}} = \frac{\mathbf{M} - \min(\mathbf{M})}{\max(\mathbf{M}) - \min(\mathbf{M})}. \tag{4.7}$$

I used this normalized matrix as the input to CNN.

The structure of the CNN is shown in Fig. 4.18. The input data are passed through multiple blocks that consist of a convolution layer, batch normalization layer, ReLU layer, and an average pooling layer. In the convolution layer, a 2D convolution is applied to the input to extract meaningful features. The size and number of filters were set as 3×3 and 2^n, respectively. In addition, zero padding is used in the convolution process to prevent the size of the data from decreasing. Then, a batch normalization layer is applied to the output of the convolution layer for fast and stable training. Further, a nonlinearity is applied to the network through the ReLU layer, and the average pooling layer is applied to downsample the data and prevent overfitting. After passing through multiple blocks, the output is passed through a fully connected layer with 64 nodes, ReLU layer, and fully connected layer with one node. Finally, a regression is performed and the loss function is calculated using the mean squared error.

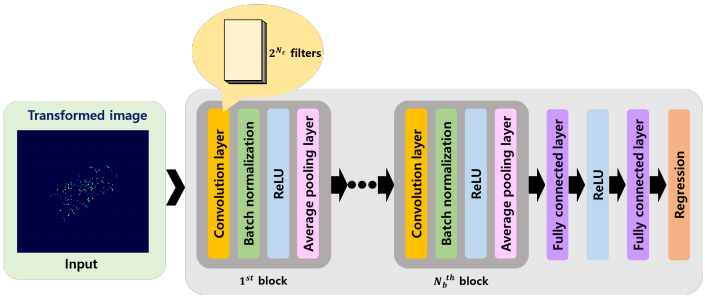


Figure 4.18: Structure of CNN

4.2.5 Performance Evaluation

In this study, the performance of the proposed method is evaluated using measurement data obtained through experiments with stationary and moving vehicles. As mentioned in Chapter 4.2.3, measurements were conducted for 13 different orientation angles ranging from -60° to 60°, and a total of more than 16,000 data were obtained. When training the decision tree and CNN model, I divided the data into training, validation, and test sets using the five-fold cross-validation method. For example, in the first iteration, the first measurement data were used to test the model trained and validated using the remaining measurement data. Then, in the second iteration, the second measurement data were used as a test set and the rest of the data were used as training and validation sets. This process was repeated five times until all the measurements were used as the test set. The final result was derived by averaging the prediction results of the five iterations.

First, I used a PCA-based regression method to estimate the orientation angle of the vehicle. The estimated orientation angles for the 13 different cases are shown in Fig. 4.19. For each orientation angle, I calculated the mean value of the estimated orientation angle, which is plotted as a red line. In addition, the one-sigma and two-sigma deviation values are plotted in the figure. Although the mean value was close to the ground truth value, the sigma deviation values were far from the mean, indicating that the estimation variance was high and the performance was relatively unstable. This is because the PCA algorithm assumes the distribution of the data to be Gaussian, which can degrade the regression performance in certain situations.

Next, I applied the decision tree-based regression method. When using the decision tree-based method, the number of features and decision trees must be determined. To this end, I used the neighborhood component analysis algorithm [66] to determine the importance of the seven features in (4.6). The seven features in the order of importance were $\bar{\lambda}_2$, Θ_p, $\mathbf{v}_1(2)$, $\bar{\lambda}_1$, $\mathbf{v}_2(1)$, $\mathbf{v}_1(1)$, and $\mathbf{v}_2(2)$. Based on this observation, I analyzed the performance of the network using all seven features or by selecting only a few of

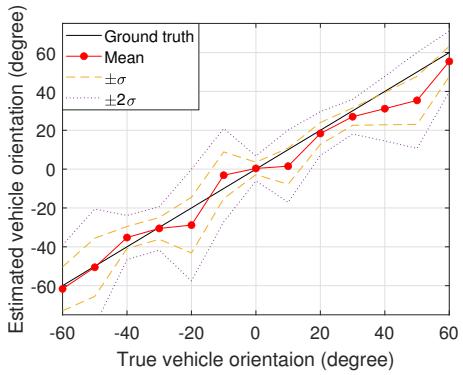


Figure 4.19: Estimation results using PCA-based method

the important features. In addition, to investigate the performance according to the number of decision trees, I varied the number of decision trees from 10 to 200 in intervals of 10.

The estimation results are shown in Fig. 4.20. As shown in the figure, the RMSE generally decreased as the number of decision trees increased. However, the rate of decrease was not noticeable when the number of decision trees was higher than 100. Therefore, I set the number of decision trees as 100 by considering the complexity of the network. In addition, the RMSE of the estimated orientation angle was the lowest when all seven features were used; therefore, I used all the seven features for regres-

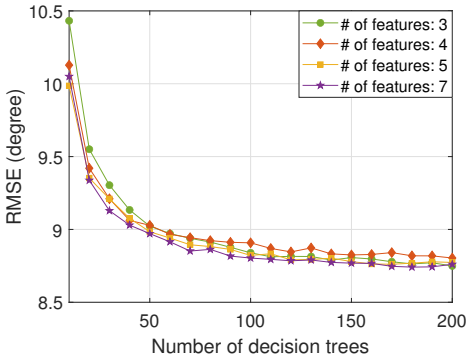


Figure 4.20: Estimation results using decision tree-based method

sion. Under these conditions, the RMSE of the estimated orientation angle was 8.8°.

Further, the CNN-based regression method described was used to estimate the orientation angle. When using the CNN-based method, the number of blocks and filters are the main parameters that should be determined appropriately. To determine these parameters, I investigated the performance of the network by varying the number of blocks from two to six and the number of filters from two to 256 in units of 2^n. The estimation results are shown in Fig. 4.21. When the number of blocks was two or three, the RMSE diverged as the number of filters increased. By contrast, when the number of blocks was four, five, or six, the RMSE decreased with an increase in the number of filters. Because the decreasing rate was not prominent when the number of filters was higher than 32, I set the number of filters and blocks as 32 and 5, respectively. Using these parameters, the RMSE of the estimated orientation angle was found to be 3.34°.

A comparison of various regression methods is presented in Table 4.4. The performance of the PCA-based method was the lowest with an RMSE of 10.96°, whereas that of the CNN-based method was the highest with an RMSE of 3.34°. The PCA algorithm showed unstable performance because it is a non-parametric algorithm in which there are no parameters to adjust [61]. Using the results of the PCA algorithm as an input to the decision tree, the estimation performance was slightly improved. However, the RMSE of the decision tree-based method was still higher than that of

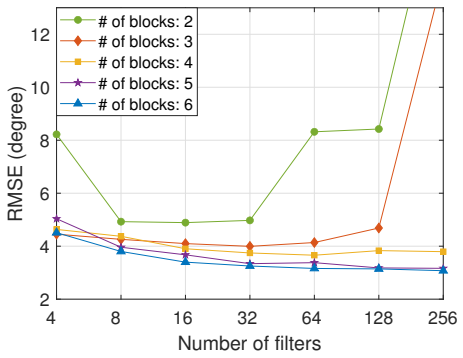


Figure 4.21: Estimation results using CNN-based method

the CNN-based method. This is because the decision tree requires a feature selection process in which information can be lost, whereas the CNN does not require feature selection. Consequently, the CNN-based method showed better performance than the PCA and decision tree-based methods.

Moreover, the comparison of various methods which can be used for vehicle orientation estimation is summarized in Table 4.5. Because the method in [58] uses the point cloud data similar to the method used in the proposed algorithm, I compared the performance with the orientation estimation methods used in [58], which were L-fit, enhanced orientated bounding box (EOBB), and brute-force methods. The L-fit is a method of randomly extracting three points and finding two orthogonal lines. However, this method is hard to apply to cascaded MIMO FMCW radar system because the

Table 4.4: Comparison of various estimation methods

Type	**RMSE**
PCA-based method	10.96°
EOBB method	9.92°
Brute-force method	8.56°
Decision tree-based method	8.80°
CNN-based method	3.34°

Table 4.5: Comparison of other methods

Paper	**Radar type**	**Frequency**	**Bandwidth**	**No. of antenna**	**Data type / method**	**Main result**
[55]	FMCW	24 / 77 GHz	0.5 - 7 GHz	1×1	SAR image / SAR processing	Detection of scattering centers
[56]	CW	76 GHz	Not specified	Not specified	Velocity profile / RANSAC	Lateral velocity estimation
[57]	FMCW	78 GHz	1.8 GHz	2×4	Range-angle map / CNN	Heading direction classification
[58]	FMCW	77 GHz	500 MHz	2×10	Point cloud data / Bounding box-based iterative method	Orientation estimation
Proposed method	FMCW	77 GHz	400 MHz	12×16	Point cloud data / Regression method	Orientation estimation

point cloud data comprises thousands of points and the computational complexity is extremely high. The EOBB method is based on rotating calipers algorithm [67], which creates a bounding box from the extreme values of the convex hull. The RMSE of the estimated orientation angle was 9.92° when the EOBB method was applied to the measured point cloud data. The brute-force method finds the main direction of points using the random sample consensus algorithm [29]. When I applied this method to the measured point cloud data, the RMSE of the estimated orientation angle was 8.56°. The methods used in [58] were susceptible to outlier data and required iterative approach which resulted in lower performance compared to the CNN-based method.

Lastly, I compared the performance of various estimation methods by varying the number of accumulated scans. The default number of accumulated scans was set as eight in the previous discussions, which corresponds to accumulating the data for one second. To analyze the effect of accumulating more scans, I investigated the RMSE of various regression methods by varying the number of accumulated scans from 4, 8, 12, and 16. As shown in Fig. 4.22, the RMSE decreased by accumulating more scans because the correlation of the data is higher and the clustering performance is enhanced. However, the number of accumulated scans cannot be set to a large value when an instant decision is required in a dynamic road environment. This is a design

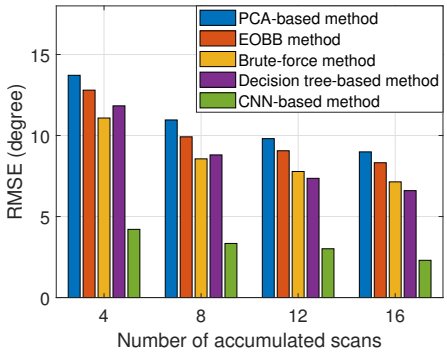


Figure 4.22: Comparison of various estimation methods according to the number of accumulated scans

parameter that should set appropriately depending on desired automotive applications. By using the CNN-based method, the vehicle orientation angle was reliably estimated with an RMSE of 3.34°. Therefore, I believe that the proposed method of estimating the orientation angle of a vehicle using machine learning-based algorithms will assist in autonomous driving.

4.2.6 Summary

In this study, I proposed a method to estimate the orientation angle of a surrounding vehicle using a cascaded MIMO FMCW radar system. Measurements were conducted for 13 different orientation angles and the point cloud data corresponding to the vehicle were extracted through data pre-processing. To estimate the orientation angle of the vehicle, I applied three regression algorithms such as PCA, decision tree, and CNN. The appropriate structure of machine learning algorithms was determined by varying the hyperparameters of the network and analyzing its performance. Further, I compared the RMSE of each regression method, which revealed that the CNN-based method was the most accurate within an RMSE of 4°. The future focus concerning this research will be to improve the estimation performance by combining the detection results of multiple sensors. Moreover, when the processing speed of radar is further improved, it is expected that the proposed algorithm can be used to estimate the vehicle orientation in a dynamic environment as well.

Chapter 5

IN-VEHICLE PASSENGER DETECTION

5.1 FEATURE EXTRACTION BASED APPROACH USING IR-UWB RADAR

5.1.1 Motivation

Every year, several children and pets die of heat stroke and hyperthermia after being left alone in a locked car [68]. The temperature inside a closed vehicle increases rapidly, even in moderate weather [69]. If the temperature becomes too high, lives in the vehicle are put in a dangerous situation. Therefore, when a child or a pet is left unattended in a car, the driver should be aware of the situation and should be alerted quickly to prevent tragedy. Some studies have been carried out on combining multiple sensors (e.g., temperature, motion, GPS, or ultrasonic sensor) along with smart phone applications to avoid death from heat stroke [70], [71]. However, the sensors had to be attached to the body in these methods, and the complexity was high because information from multiple sensors should be integrated.

Monitoring of people can also be conducted by using a single, non-contact passive sensor, such as camera, ultrasonic, and radar sensors. However, the performance of a camera sensor degrades in a poorly-lit environment and it can invade the privacy

of individuals. The radar sensor has advantages over other sensors because it is robust to environmental conditions [72] and has high detection performance (e.g., range resolution) [73], [74], but does not invade the privacy of an individual [75]. Studies that address human detection based on radar system were conducted in [76], [77]. For example, the authors in [76] introduced a 2.4 GHz Doppler radar system for cardiopulmonary function. In [77], research was conducted to detect the breathing of a human using a 24 GHz continuous wave radar.

However, few studies have focused on the position or number of people inside the vehicle. The existing outdoor algorithms cannot be applied inside the car because the propagation characteristics for indoor and outdoor are different. Also, ultrasonic sensors are being used in commercial vehicles to determine the presence of people inside the vehicle [78], but this method only works when a motion is detected and the number and position of people cannot be determined. Meanwhile, because IR-UWB radar is remarkable for detecting subtle movements of objects, it can be applied on estimating the location and the number of people inside the vehicle. Therefore, in this study, I propose an effective method for estimating the position and number of people inside a vehicle using IR-UWB radar. The received IR-UWB radar signals consists of reflected and scattered signals from various parts of people or fixed objects, whose distribution varies significantly depending on the arrangement of people. Hence, the position and number of people inside the vehicle can be estimated by identifying the distribution of the received radar waveforms. I use a method of extracting parameters that represent the statistical characteristics of the signal distribution, which can then be used as criteria for recognizing the pattern of each distribution. Furthermore, I use ensemble learning with a decision tree [79], [80] as a base classifier to classify the various arrangements of people. However, when a single decision tree is used, the performance of the algorithm becomes unstable because it is vulnerable to small changes of input. Therefore, I use multiple decision trees and then combine the results of each decision tree through ensemble learning. In addition, five-fold cross-validation

is used, and the classification accuracy calculated for each validation is averaged. The proposed method is demonstrated to successfully recognize the arrangement of people inside the vehicle with a classification accuracy higher than 90%. To make the proposed method suitable for real-time applications, I also derive the proper number of features and classifiers to be used in classification.

The proposed method is simple yet efficient method for monitoring people inside a vehicle, as it requires only a single IR-UWB radar sensor and is a passive type of recognition method that does not require attaching sensors to the human body. The classification accuracy is expected to increase when multiple radar sensors are used, but the complexity of the algorithm will also increase. Therefore, I used a single radar in the experiment, and confirmed that the classification accuracy is higher than 90%. In addition, the proposed algorithm reduces the real-time computational load because the training stage is done in advance. The pre-trained classification function can be embedded in commercial radars' microcontrollers inside the vehicle. Then, when the radar receives a signal, the proposed algorithm can be applied to the signal and I can immediately determine the number and position of people inside the vehicle. Also, I used the feature selection algorithm to reduce the number of features, so the training time is further reduced.

The remainder of this study is organized as follows. In Chapter 5.1.2, basic signal processing in the IR-UWB radar system is explained. Then, along with the experiment environment, the analysis of IR-UWB radar signals measured in that environment is introduced in Chapter 5.1.3. In Chapter 5.1.4, the feature extraction and selection methods are presented. Next, a classification method using ensemble learning and classification results using it are presented in Chatper 5.1.5. Finally, conclusions are given in Chapter 5.1.6.

5.1.2 Basic Signal Processing in IR-UWB Radar System

The sampled signal reflected in (2.12) is the raw signal of IR-UWB radar that has not been processed. To extract signals from meaningful targets and suppress the clutter signal, additional signal processing has to be performed. Fig. 5.1 shows a block diagram of the overall signal processing procedure. First, the direct current (DC) component of the raw signal is removed by subtracting its average value [81]. First, the DC bias of the signal is removed by subtracting its mean value, which results in

$$\hat{r}[n] = r[n] - \frac{1}{N}\sum_{k=1}^{N} r[k]. \tag{5.1}$$

Then, the signal is passed through a bandpass filter to remove the undesired frequency components. The bandpass filter is designed by calculating the half-power bandwidth of the transmitted signal from Fig. 5.2, and then passing only those frequency components. The bandpass filtering operation can be expressed as

$$\hat{r}_{BPF}[n] = \mathscr{F}^{-1}\big\{\mathscr{F}\{\hat{r}[n]\} \cdot H[z]\big\}, \tag{5.2}$$

where $\mathscr{F}$ and $\mathscr{F}^{-1}$ denote the Fourier transform and inverse Fourier transform operation, and $H[z]$ denotes the bandpass filter in the frequency domain. To suppress the

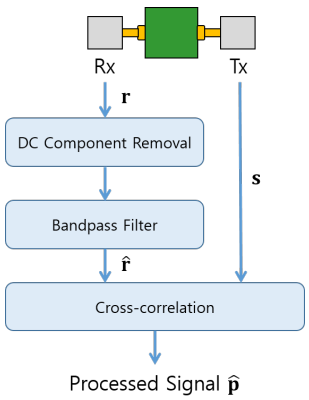


Figure 5.1: Block diagram of basic signal processing in the IR-UWB radar system

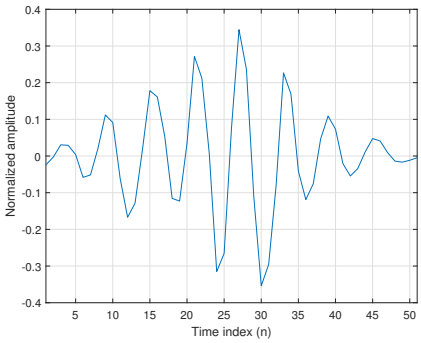


Figure 5.2: Signal transmitted by the IR-UWB radar

noise component remaining in the passband of the filter, the signal is cross-correlated with the transmitted signal shown in Fig. 5.2. By cross-correlating with the transmitted signal, the signals from targets become more prominent, and the effect of noise is mitigated. The cross-correlated signal is expressed as

$$p[n] = \sum_{j} \hat{r}_{BPF}[j]\, s[N_t - n + j], \quad (n = 1,\, 2,\, \cdots,\, N + N_t - 1), \tag{5.3}$$

where the summation is performed over all possible values of j, and N_t is the number of samples in the transmitted signal. After cross-correlation, the length of the signal becomes longer than the original signal (i.e., from N to $N + N_t - 1$), so the signal is truncated to match the length of the original signal. The truncated signal is given as

$$\begin{aligned} \hat{\mathbf{p}} = [\hat{p}[1],\, \hat{p}[2],\, \cdots,\, \hat{p}[N]] \quad &= \quad [p[\tilde{N}],\, p[\tilde{N}+1],\, \cdots,\, p[\tilde{N}+N-1]], \\ &\quad (\tilde{N} = \lceil N_t/2 \rceil), \end{aligned} \tag{5.4}$$

where $\lceil \cdot \rceil$ is an operator that rounds down to the nearest integer. The effect of applying cross-correlation is shown in Fig. 5.3. The amplitude of the signal from the target becomes larger, so the boundary between the target region and the non-target region becomes more clear. Therefore, in this study, I mainly use this processed signal $\hat{\mathbf{p}}$ for the feature extraction and the classification.

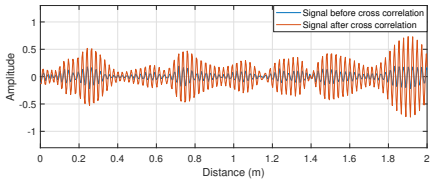


Figure 5.3: Comparison of signal before and after cross-correlation

5.1.3 Measurement of IR-UWB Radar Signal Inside Vehicle

Experiment Environment

In the experiment, I used the NVA-R661 IR-UWB radar module manufactured by Novelda (Xethru), as shown in Fig. 5.4. The radar is connected to laptop via SPI to USB cable (C232H) to store the data. The radar kit has one Tx and one Rx, which are 15 cm apart. A dielectric lens is mounted on each antenna to narrow the radiation pattern and increase the antenna gain. The specifications of the radar system are summarized in Table 5.1. The center frequency of the radar signal, bandwidth, and antenna gain are 6.8 GHz, 2.3 GHz, and 6.0 dBi. The pulse repetition frequency is set as 100 MHz, which means that the radar transmits a pulse every 10 ns. However, the actual data capture rate, denoted as the frame rate, is only 120 Hz which is much slower than the pulse repetition frequency. This is due to coherent processing, where multiple pulses are combined and averaged to increase the processing gain. In addition, the time it takes to transfer the data over SPI to USB link further limits the frame rate.

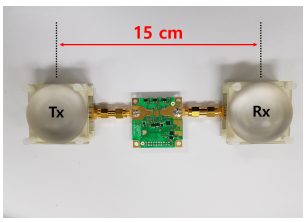


Figure 5.4: IR-UWB radar module (NVA-R661)

Table 5.1: Specifications of radar system

Radar type	**IR-UWB radar**
Carrier frequency, f_c	6.8 GHz
Bandwidth	2.3 GHz
Pulse repetition frequency	100 MHz
Frame rate	120 Hz
Number of samplers, N_s	256
Number of samples, N	512
Sampling period, T_s	26 ps
Antenna gain without dielectric lens	6 dBi
Antenna gain with dielectric lens	6.7 dBi

Moreover, the sampling period is set as 26 ps, so the time difference between adjacent samples is 26 ps. Since the time delay τ can be expressed as $2R/c$ where R is the distance and c is the velocity of light, the distance difference between adjacent samples is $c\tau/2 \approx 4$ mm.

As shown in Fig. 5.5, I installed the radar in the rear view mirror. This position is regarded as an appropriate position to simultaneously monitor the front and rear seats. The vertical distance from the radar to the floor of the vehicle is approximately 1 m, and the diagonal distance from the radar to the side rear seat is about 1.6 m. Therefore,

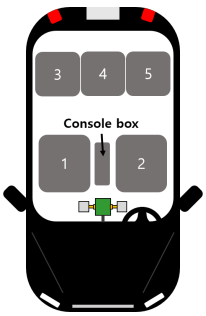


Figure 5.5: Experiment environment

in the experiment, the number of frames is set as 2 to observe a distance of $0 \sim 2$ m. In addition, the front and rear seats are numbered from 1 to 5, as shown in Fig. 5.5. The experiment is conducted for 32 different scenarios depending on the presence of people in each seat. These 32 experiment scenarios will be labeled from class E1 to E32. The position of people for each class is shown in Table 5.2, where 'O' and 'X' indicate whether a person is in the corresponding seat or not. For example, Fig. 5.6 shows the experiment environment for the class E9 where only one person is sitting in the front left seat. The heights and weights of the experiment participants are given in Table 5.3.

Table 5.2: Experiment scenario

Class	**Seat 1**	**Seat 2**	**Seat 3**	**Seat 4**	**Seat 5**	**Class**	**Seat 1**	**Seat 2**	**Seat 3**	**Seat 4**	**Seat 5**
E1	X	X	X	X	X	E17	X	O	X	X	X
E2	X	X	O	X	X	E18	X	O	O	X	X
E3	X	X	X	O	X	E19	X	O	X	O	X
E4	X	X	X	X	O	E20	X	O	X	X	O
E5	X	X	O	O	X	E21	X	O	O	O	X
E6	X	X	O	X	O	E22	X	O	O	X	O
E7	X	X	X	O	O	E23	X	O	X	O	O
E8	X	X	O	O	O	E24	X	O	O	O	O
E9	O	X	X	X	X	E25	O	O	X	X	X
E10	O	X	O	X	X	E26	O	O	O	X	X
E11	O	X	X	O	X	E27	O	O	X	O	X
E12	O	X	X	X	O	E28	O	O	X	X	O
E13	O	X	O	O	X	E29	O	O	O	O	X
E14	O	X	O	X	O	E30	O	O	O	X	O
E15	O	X	X	O	O	E31	O	O	X	O	O
E16	O	X	O	O	O	E32	O	O	O	O	O

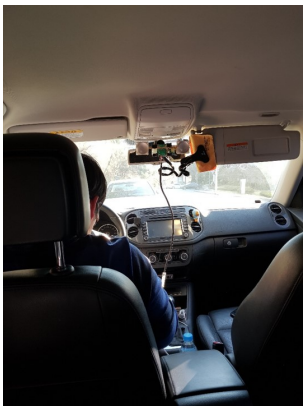


Figure 5.6: Actual photograph of experiment environment (Class E9)

Table 5.3: Personnel information of experiment participants

Participant	**Height (cm)**	**Weight (kg)**
A	173	82
B	156	52
C	156	58
D	163	52
E	176	97

Analysis of Measured IR-UWB Radar Signal

The radar signal obtained in one experiment scenario can also vary depending on the participant since each person has different physical characteristics. Therefore, measurements were conducted several times while changing the participant and the order of people sitting. The total number of measurements for all classes was 240. Because one measurement consists of more than 500 scans, a total of more than 120,000 raw radar signals are collected. The processed signal $\hat{\mathbf{p}}$ for each class is shown in Fig. 5.7. Since the participants' movement is almost stationary, there is no periodic component of the signal. As a result, the processed signals are analyzed in the time domain. The signal waveforms show different trends depending on the arrangement of

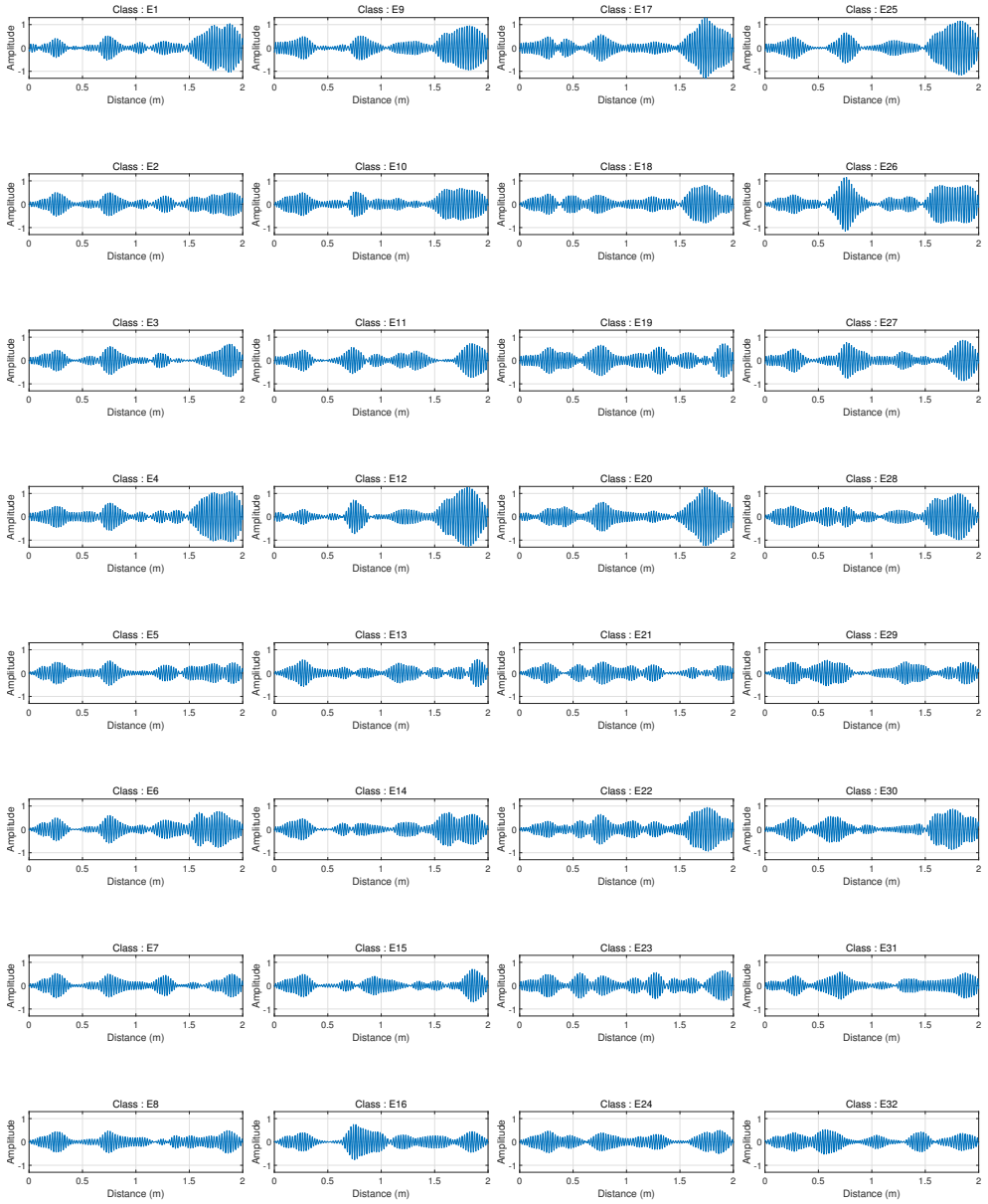


Figure 5.7: Processed radar signal $\hat{\mathbf{p}}$ for 32 experiment scenarios

people. Thus, it is confirmed that the given 32 classes can be classified by analyzing the characteristics of the received radar signals. The signal waveforms from E2 and E4 or E5 and E7 should be theoretically identical because people are sitting at the same

distance from the radar. However, these symmetrical classes exhibit slightly different signal waveforms. This is because the distance between the transmitter and receiver is approximately 15 cm, so the measurement is not conducted symmetrically. Thus, even though one radar development kit is used, I can distinguish people sitting on the left from people sitting on the right.

5.1.4 Feature Extraction and Feature Selection

Feature Extraction

To analyze the signals received from various classes, I choose parameters that represent the statistical properties of the signal. Generally, mean, variance, coefficient of variance, kurtosis, and skewness are frequently used parameters [82]. The variance is a measure of how the data is spread around its mean. The coefficient of variance is a standardized measure of the dispersion of data around its mean, which is known as relative standard deviation. The kurtosis is a measure of how the data is concentrated around the mean, and the skewness is a measure of the asymmetry of the data. In addition to these parameters, I use the maximum value of the signal and the argument of the maximum value in the signal (i.e., argmax). These parameters are used as features representing the characteristics of the signal.

Because the distance between the radar and console box in the vehicle is 1 m, signal changes owing to human position in the front seat are reflected by $0 \sim 1$ m portion of the signal, whereas the changes due to human position in the rear seat are reflected by the $1 \sim 2$ m portion. Therefore, I divide the entire signal into intervals of $0 \sim 1$ m and $1 \sim 2$ m, and extract the suggested parameters at each interval. The

parameters from the i_{th} interval can be expressed as

$$
\begin{aligned}
f_{\mu}^{i} &= \frac{1}{N'} \sum_{n=N_{s,\,i}}^{N_{e,\,i}} \hat{p}[n] \quad (i = 1,\, 2), \\
f_{v}^{i} &= \frac{1}{N'-1} \sum_{n=N_{s,\,i}}^{N_{e,\,i}} \left(\hat{p}[n] - f_{\mu}^{i}\right)^{2} \quad (i = 1,\, 2), \\
f_{c}^{i} &= \frac{\sqrt{f_{v}^{i}}}{f_{\mu}^{i}} \quad (i = 1,\, 2), \\
f_{k}^{i} &= \frac{\frac{1}{N'} \sum_{n=N_{s,\,i}}^{N_{e,\,i}} \left(\hat{p}[n] - f_{\mu}^{i}\right)^{4}}{\left(\frac{1}{N'} \sum_{n=N_{s,\,i}}^{N_{e,\,i}} \left(\hat{p}[n] - f_{\mu}^{i}\right)^{2}\right)^{2}} \quad (i = 1,\, 2), \\
f_{s}^{i} &= \frac{\frac{1}{N'} \sum_{n=N_{s,\,i}}^{N_{e,\,i}} \left(\hat{p}[n] - f_{\mu}^{i}\right)^{3}}{\left(\frac{1}{N'} \sum_{n=N_{s,\,i}}^{N_{e,\,i}} \left(\hat{p}[n] - f_{\mu}^{i}\right)^{2}\right)^{3/2}} \quad (i = 1,\, 2), \\
f_{m}^{i} &= \max_{n \in [N_{s,\,i}, \cdots, N_{e,\,i}]} \hat{p}[n] \quad (i = 1,\, 2), \\
f_{\tilde{n}}^{i} &= \arg\max_{n \in [N_{s,\,i}, \cdots, N_{e,\,i}]} \hat{p}[n] \quad (i = 1,\, 2), \\
& N' = \frac{N}{2}, \\
& \begin{pmatrix} N_{s,\,1} & N_{e,\,1} \\ N_{s,\,2} & N_{e,\,2} \end{pmatrix} = \begin{pmatrix} 1 & N' \\ N'+1 & N \end{pmatrix},
\end{aligned}
\tag{5.5}
$$

where f_{μ}^{i}, f_{v}^{i}, f_{c}^{i}, f_{k}^{i}, f_{s}^{i}, f_{m}^{i}, and $f_{\tilde{n}}^{i}$ represent the mean, variance, coefficient of variance, kurtosis, skewness, maximum value, and argument of the maximum value, respectively. Because the entire signal is divided into two intervals, a total of $7 \times 2 = 14$ parameters are extracted from each scan. Consequently, the feature vector can be expressed as

$$
\begin{aligned}
\mathbf{f} &= [f_{\mu}^{1},\, f_{v}^{1},\, f_{c}^{1},\, \cdots,\, f_{\tilde{n}}^{2}], \\
&= [f[1],\, f[2],\, \cdots,\, f[N_f]],
\end{aligned}
\tag{5.6}
$$

where N_f denotes the number of features, which is 14 in this study. Since 32 classes have to be classified using 14 features, the dimension is relatively large. Therefore, the number of features should be reduced to use only meaningful features.

Feature Selection Using NCA Algorithm

Before applying the classification algorithm, the importance of each feature should be investigated because duplicate or irrelevant features only increase complexity, without affecting the classification accuracy. A training set with L samples can be expressed as

$$T = \{(\mathbf{f}_1,\, y_1),\, (\mathbf{f}_2,\, y_2),\, \cdots,\, (\mathbf{f}_L,\, y_L)\}, \tag{5.7}$$

where $\mathbf{f}_l$ is a N_f-dimensional feature vector of the l_{th} sample, and $y_l \in \{1,\, 2,\, \cdots, 32\}$ is a class label of the l_{th} sample. Among the feature selection methods, neighborhood component analysis (NCA) is used to reduce the dimensions of features and to increase the execution speed [66]. The NCA algorithm finds feature weights by solving a classification problem. A feature vector can be thought of as a point in the N_f-dimensional space, and the algorithm assumes that feature vectors which are close to each other are likely to belong to the same class. Therefore, the algorithm calculates the distance between all feature vectors. The measure of closeness is determined by the Mahalanobis distance, which is mainly used in the K-nearest neighborhood algorithm [83]. The Mahalanobis distance between two feature vectors $\mathbf{f}_l$ and $\mathbf{f}_r$ can be expressed as

$$D_w(\mathbf{f}_l,\, \mathbf{f}_r) = \sum_{q=1}^{N_f} w_q^2 |f_l[q] - f_r[q]|, \tag{5.8}$$

where w_q is the weight of q_{th} feature. To predict the case label of $\mathbf{f}_l$, the algorithm selects a reference point $\mathbf{f}_r$ and labels $\mathbf{f}_l$ as the label of $\mathbf{f}_r$. The probability of $\mathbf{f}_l$ choosing $\mathbf{f}_r$ as its reference point becomes higher if two points are close to each other. Therefore, a kernel function $k(z) = \exp(-\frac{z}{\sigma})$ is adopted to have high probability for a small distance. Then, the probability of $\mathbf{f}_l$ selecting $\mathbf{f}_r$ as its reference point can be expressed as

$$P_{l,\,r} = \begin{cases} \frac{k(D_w(\mathbf{f}_l, \mathbf{f}_r))}{\sum_{r=1, r\neq l}^{L} k(D_w(\mathbf{f}_l, \mathbf{f}_r))} & \text{if } r \neq l \\ 0 & \text{if } r = l \end{cases}. \tag{5.9}$$

Here, the denominator acts as a normalizing factor so that the sum of $P_{l,r}$ for all r becomes one, and $P_{l,r}$ can be considered as the probability. In addition, the probability of $\mathbf{f}_l$ choosing $\mathbf{f}_l$ as its reference point is zero since NCA is a leave-one-out algorithm which excludes itself in the prediction process. The class label of $\mathbf{f}_l$ is accurately estimated if the reference point $\mathbf{f}_r$ has the same class label as $\mathbf{f}_l$. Thus, the probability that $\mathbf{f}_l$ is accurately classified can be expressed as

$$\begin{aligned} P_l &= \sum_{r=1, r\neq l}^{L} P_{l,r} Y_{l,r}, \\ Y_{l,r} &= \begin{cases} 1 & \text{if } y_l = y_r \\ 0 & \text{if } y_l \neq y_r \end{cases}. \end{aligned} \tag{5.10}$$

The average leave-one-out probability of correct classification is the average of P_l ($l = 1, 2, \cdots, L$) over all samples. As a result, an objective function, with the regularization term to prevent over-fitting, can be expressed as

$$F(\mathbf{w}) = \frac{1}{L} \sum_{l=1}^{L} P_l - \lambda \sum_{q=1}^{N_f} w_q^2, \tag{5.11}$$

where $\mathbf{w} = [w_1, w_2, \cdots, w_{N_f}]$, and λ is a regularization parameter which has a positive value. The argument $\mathbf{w}$ that maximizes the objective function maximizes the average leave-one-out probability of correct classification and indicates the importance of each feature. Therefore, I perform the feature selection based on this weight vector.

5.1.5 Performance Evaluation

Ensemble Learning

Machine learning algorithms such as a decision tree or a support vector machine (SVM) [84] are commonly used for the classification of data. In addition, instead of using a single classifier, multiple classifiers such as multiple decision trees can be used to yield better generalization performance. Moreover, ensemble learning [85] combines multiple weak classifiers into a strong classifier, where the weak classifier is

slightly correlated to the true classifier and the strong classifier is strongly correlated to the true classifier. Ensemble learning can achieve better classification results than using a single model, by improving reliability and accuracy.

The most commonly used algorithms in the ensemble learning are boosting [86] or bagging [87]. A boosting algorithm sequentially converts weak classifier into strong classifier. In contrast, a bagging algorithm, also referred to as bootstrap aggregating, trains multiple classifiers independently. Multiple training sets are generated by sampling with replacement of the total data, and each classifier is trained in parallel. Then, the final output is determined by aggregating the prediction results from each classifier, and by averaging or voting the results. Through the bagging algorithm, the variance of data is reduced, so it is more robust to an over-fitting problem than the boosting algorithm.

Classification Results

The overall signal processing chain of the proposed method is summarized in Fig. 5.8. First, basic signal processing is applied to the raw radar signal obtained by the IR-UWB radar. A total of more than 120,000 processed signals are obtained for 32 ex-

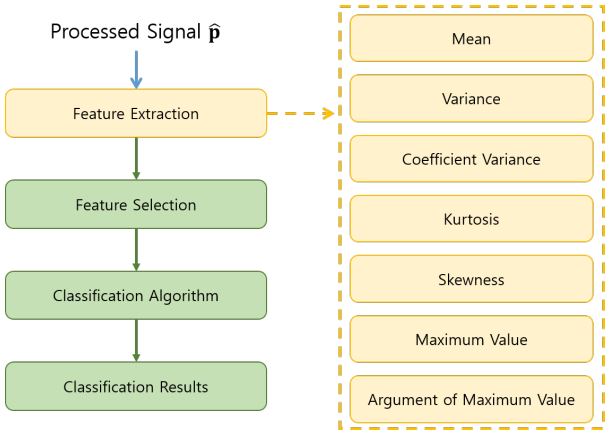


Figure 5.8: Overall signal processing chain of the proposed method

periment scenarios. Then, approximately 30% of the processed signals are randomly selected to prevent unintended biases. As a result, 34,749 processed signals are selected, and 14 features are extracted from each processed signal. These features are used as inputs for classification, so a data set consists of 34,749 14-dimensional feature vectors. For classification, I use the decision tree as a base classifier since it is a fast and simple algorithm, and I use the bagging algorithm for the ensemble of decision trees. The block diagram of the classification algorithm is shown in Fig. 5.9. The data set is divided into multiple training sets, and each training set trains its own decision tree. The prediction result from each decision tree is combined by averaging the individual prediction results, and I obtain an output label from 1 to 32. In addition, when training an individual decision tree, five-fold cross-validation is used to prevent a data over-fitting problem. The training set is divided randomly into five subsets of equal size. Then, four of the five partitions are used to train the classifier, and the remaining partition is used to validate the performance of the trained model. This process is repeated five times until each partition is used for the validation.

As mentioned in Chapter 5.1.4, among the 14 parameters of the feature vector, parameters that are crucial for classification are identified through the NCA algorithm. The weight values of each parameter are shown in Table 5.4. The parameters written

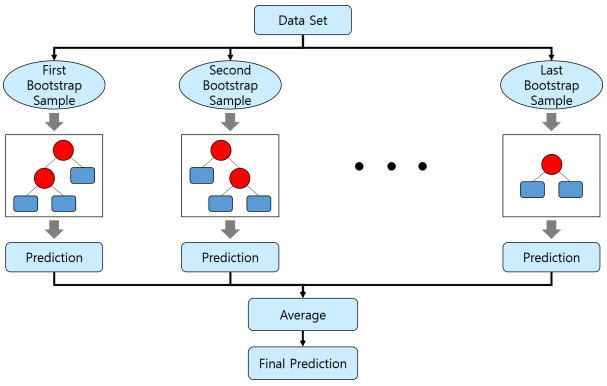


Figure 5.9: Block diagram of bagging with decision tree

Table 5.4: Weight values of each parameter using NCA algorithm

Feature weight	**Weight value**	**Feature weight**	**Weight value**
$w_1\,(f_\mu^1)$	0.75	$w_8\,(f_\mu^2)$	0.75
$w_2\,(f_v^1)$	0.76	$w_9\,(f_v^2)$	0.82
$w_3\,(f_c^1)$	16.33	$w_{10}\,(f_c^2)$	14.68
$w_4\,(f_k^1)$	1.13	$w_{11}\,(f_k^2)$	1.63
$w_5\,(f_s^1)$	0.75	$w_{12}\,(f_s^2)$	0.75
$w_6\,(f_m^1)$	0.80	$w_{13}\,(f_m^2)$	1.17
$w_7\,(f_{\tilde{n}}^1)$	19.61	$w_{14}\,(f_{\tilde{n}}^2)$	17.44

in the order of importance are $f_{\tilde{n}}^1$, $f_{\tilde{n}}^2$, f_v^1, f_v^2, f_c^2, f_m^2, and f_c^2, where only the parameters having weight value higher than 1 are considered. In other words, the most crucial parameter for classification is the argument of maximum value, followed by the coefficient of variance, kurtosis and maximum value.

Meanwhile, two important factors have a great impact on classification performance: the number of classifiers (i.e., the number of decision trees) and the number of features. Computational cost is mainly dependent on the number of classifiers and the complexity of the base classifier [88]. To utilize the proposed method in real-time applications, the computational cost should be reduced by choosing a proper number of classifiers and features. First, the number of classifiers is increased from 10 to 100, and its effect on classification accuracy is investigated. As can be seen from Fig. 5.10, the classification result becomes more accurate when more classifiers are used. However, when there were more than 50 classifiers, there was no prominent performance enhancement regardless of the number of features used as input. Thus, the number of classifiers is set to 50.

Next, to determine the appropriate number of features used for classification, I examine the classification accuracy by changing the number of features, while maintaining the number of classifiers at 50. First, the four most crucial features are used as

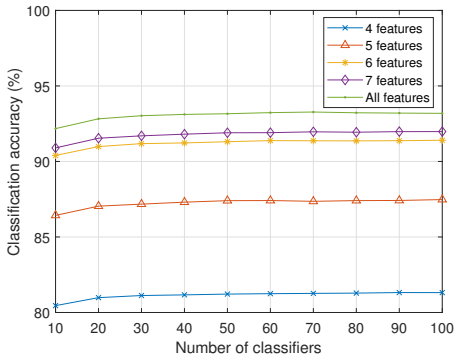


Figure 5.10: Comparison of classification accuracy according to the number of features and the number of classifiers

inputs for the classification. Then, the performance is also evaluated when the number of features is 5, 6, 7, and 14. Fig. 5.10 shows that the classification accuracies are 81.22%, 87.40%, 91.31%, 91.9%, and 93.16% when the number of features is 4, 5, 6, 7 and 14, respectively. As expected, the classification accuracy tends to increase when more features are used in training the model. However, although the classification accuracy is increased with the number of features, the increment in each case is decreased. When the number of features was increased from 4 to 5, 5 to 6, and 6 to 7, the classification accuracy increased by 6.16%p, 3.97%p, and 0.52%p, respectively. Therefore, I set the number of features as 7 since using 14 features only increases the classification accuracy by 1.26%p. When the number of classifiers was 50 and the number of features was 7, the classification accuracy was 91.9%. In terms of computation cost, ensemble learning consists of multiple decision trees, so the computational cost of each decision tree determines the total computational cost. The computational complexity of a decision tree is $O(\mathcal{D}\mathcal{N}\log\mathcal{N})$ [89], where $\mathcal{N}$ is the number of training data and $\mathcal{D}$ is the number of features. Since the number of training data is fixed, computational complexity is only dependent on the number of features. Therefore, by reducing the number of features from 14 to 7, the computational cost is expected to

reduce by half.

The confusion matrix showing the classification result for each class is given in Fig. 5.11. The matrix shows that the lowest classification accuracy is 86.35% corresponding to the class E25 and the highest classification accuracy is 97.02% corresponding to the class E20. In addition, I examined how the presence of people in the front seat affects the classification accuracy. I divided the total experiment classes into groups of E1 to E8 (no one is sitting in the front seat), E9 to E16 (one person is sitting in the right front seat), E17 to E24 (one person is sitting in the left front seat), and E25 to E32 (two people are sitting in each front seat). The average classification accuracy for each group was 91.74%, 92.87%, 92.21%, and 89.77%, respectively. In other words, the classification accuracy was the lowest when two people were sitting in the front seat. This is because people in the front seat can block the path of the radar and narrow the detectable space in the rear seat. In this case, multiple radars can be employed to further improve the detection performance.

Furthermore, I compared the proposed method with other classification algorithms, such as a decision tree, boosting with a decision tree, and an SVM. The SVM is one of the most popular machine learning algorithms used for classifying data. It was orig-

Actual class/ Estimated class	E1	E2	E3	E4	E5	E6	E7	E8	E9	E10	E11	E12	E13	E14	E15	E16	E17	E18	E19	E20	E21	E22	E23	E24	E25	E26	E27	E28	E29	E30	E31	E32
E1	96.26	0	0.04	2.06	0	0	0	0	0.31	0	0.19	0	0.23	0	0	0	0.27	0	0.27	0	0.04	0	0	0	0	0	0	0	0.08	0	0	0.04
E2	0	92.18	1.12	0.07	1.56	1.04	0.74	0.22	0	0.3	0	0	0	0.15	0	0	0	0	0.07	0	0	0	0	0.07	0	0.22	0	0	0.22	0	0	0
E3	0.31	1.36	92.44	0.19	1.98	0	3.59	0.56	0	0	0	0	0.06	0	0.06	0	0.06	0	0.25	0	0	0	0	0	0.06	0	0	0	0.06	0	0	0
E4	5.14	0	0.24	88.77	0	2.06	0	0	0.71	0	0	0.16	0	0	0	0	0.32	0	0	0	0	0	0	0	0.08	0	0	0	0	0.08	0	0
E5	0	2.2	2.62	0	90.1	0	2.45	0.51	0	0	0	0	0.08	0	0	0	0	0	0	0	1.02	0	0	0.59	0	0	0	0	0.59	0	0	0
E6	0	1.27	0	4.19	0	94.8	0.13	0.06	0.06	0.44	0.13	0.06	0	0.32	0	0	0	0.13	0	0	0	0.25	0	0	0.06	0.13	0	0	0	0.38	0	0
E7	0	0.66	3.28	0.07	2.55	0.07	87.15	2.99	0	0	0	0	0.15	0	0.29	0.07	0	0	0.07	0	1.17	0	0.07	0.51	0	0.07	0	0	0.66	0	0.22	0
E8	0	0	0.26	0	0.1	0.05	1.45	92.19	0	0	0	0	0.1	0.1	0.26	0.88	0	0	0	0	0.26	0	0.26	0.83	0	0.05	0.05	0	0.1	0	0.1	0.83
E9	0.53	0	0	0.7	0	0.09	0	0	89.08	0.09	0	1.23	0	0.26	0	0	0.35	0	0	0	0	0	0	0	6.51	0	0	1.14	0	0.35	0	0
E10	0	0.49	0	0	0	0.33	0.33	0	0	92.12	0	0	0	2.3	0	0	0	0.49	0	0.16	0	0	0	0.16	0	3.78	0.16	0	0	0.16	0	0
E11	0.41	0	0	0	0	0	0	0.1	0	0	95.54	0.1	0.1	0	0.52	0.41	0	0	0	0	0	0	0	0	0	0.1	1.66	0.31	0.41	0	0.1	0.1
E12	0	0	0	0.14	0	0	0	0	1.56	0	0	94.63	0	0	0	0	0	0	0	0.14	0	0	0	0	3.39	0	0	0.14	0	0.71	0	0
E13	0	0	0.48	0	0	0	0.24	0.48	0	0	0.36	0	93.44	0	0	1.79	0	0	0.6	0	0.24	0	0	0.48	0	0	0.12	0	0.36	0	0.36	0.48
E14	0	1.49	0	0	0	0.75	0	0.3	0	1.34	0	0	0	91.79	0.45	0	0	0.15	0	0	0	0.3	0.15	0	0	3.73	0	0	0	0.15	0	0
E15	0	0	0.61	0	0	0	0.25	1.23	0.25	0	0	0	0.25	0	91.29	0	0	0	1.72	0	0	0	0.61	0.25	0	0.86	1.1	0.74	0	0	0.25	0.74
E16	0.13	0.06	0	0	0.06	0	0.19	0.25	0	0	0.06	0	0.44	0	0.06	95.09	0	0	0.13	0	0.5	0	0.13	0.31	0	0.13	0.06	0	0.19	0	0.63	0
E17	0.5	0	0.07	0.35	0	0	0	0	0.28	0	0	0	0	0	0	0	96.95	0.35	0	0.64	0	0.21	0	0	0.07	0	0	0	0	0	0	0
E18	0	0	0	0	0	0.7	0	0	0.12	0.12	0	0	0	0.23	0	0	1.17	94.17	0	0.58	0	3.15	0	0	0	0.58	0	0	0	0.23	0	0
E19	0.89	0.44	0.44	0	0	0	0.55	0.11	0	0	0.33	0.11	0.66	0.11	0.66	0.22	0	0	93.02	0	0.55	0	1.33	0.44	0.11	0.22	0.33	0	0.78	0	0.22	0.22
E20	0	0	0	0	0	0	0	0	0	0	0	0.11	0	0	0	0	1.15	0.69	0	97.02	0	0.46	0	0	0.11	0.23	0	0	0	0.23	0	0
E21	0	0	0	0	1.63	0	1.51	0.81	0	0	0	0	0.12	0	0	0.23	0	0	0.47	0	86.73	0	0.7	3.03	0	0	0.12	0	0.58	0	0	0.23
E22	0	0	0	0	0	1.03	0	0	0	0	0	0	0	0.26	0	0	0.39	3.36	0	0.26	0	93.14	0	0	0	0.26	0	0	0	1.16	0	0
E23	0.12	0	0	0	0	0	0.58	2.09	0	0	0	0	0.35	0.12	1.51	0.23	0	0	0.93	0.12	1.74	0	90.14	2.9	0	0.23	0.58	0	0.12	0	0.7	1.04
E24	0	0	0	0	0.12	0	0.18	0.86	0	0	0.12	0	0.06	0	0.06	0.37	0	0	0.12	0	2.33	0	1.16	86.52	0	0.12	1.04	0	0.25	0	0.61	2.39
E25	0	0	0	0.12	0	0	0	0	9.29	0.12	0	2	0	0	0.24	0	0.35	0	0	0.35	0	0	0	0	86.35	0.71	0.24	0.94	0	0.35	0	0
E26	0	0.55	0.14	0.14	0	1.11	0.28	0.14	0	3.18	0	0	0	2.9	0.69	0.14	0.14	0.55	0.69	0.55	0	0	0.69	0.14	0.28	86.86	0.55	0	0	0.97	0.41	0.41
E27	0	0.19	0	0	0.39	0	0.19	0.78	0	0	2.53	0	0.39	0	3.7	1.75	0	0	1.17	0	0.19	0	2.14	4.09	0	0.19	87.35	0.39	0	0	1.17	0.78
E28	0	0	0	0	0	0	0	0	0.72	0	0.36	0	0	0	0	0	0	0	0	0	0	0	0	0	0.97	0.12	0	95.17	0	0.24	0	0
E29	0	0.49	0.12	0	0.98	0	0.98	0.73	0	0	0.24	0	1.1	0	0	0.86	0	0	0	0	0.98	0	0.37	0.98	0	0	0	0	89.11	0	1.96	1.47
E30	0	0	0	0.14	0	1.42	0	0	0.43	0.28	0	0.14	0	0.28	0	0	0	0.43	0	0	0	1.85	0	0	0.28	0.71	0	1	0	93.87	0	0.28
E31	0	0.12	0.62	0	0	0	0.12	0.12	0	0	0.25	0	0.12	0	0.25	1.12	0	0	0.37	0	0.37	0	1.36	1.98	0	0.12	0.12	0	2.97	0	90.83	3.1
E32	0	0	0.09	0	0	0	0.18	1.63	0	0	0.63	0	0.9	0	0.36	0.27	0	0	0.09	0	0	0	0.36	6.87	0	0.09	0.27	0	1.26	0	0.9	88.62

Figure 5.11: Confusion matrix derived from bagging with decision tree (Unit: %)

inally used as a linear classifier, but it can be used as a non-linear classifier by introducing slack variables or kernel functions [23]. In this study, I used a Gaussian kernel since the number of classes and the number of features are relatively high. For every classification algorithm, the number of features was set as 7 and five-fold validation was performed. A comparison of classification accuracy for various machine learning algorithms are shown in Table 5.5. A single decision tree resulted in low classification accuracy because the structure of a single decision tree is too simple to classify the complex experiment classes. In addition, boosting with a decision tree showed poor performance since it is prone to over-fitting problems. Although SVM showed a relatively high classification accuracy of 80.4%, bagging with decision tree was the most effective method because it has the highest classification accuracy and a relatively fast computation time.

To show the robustness of the proposed algorithm under diverse environment, I performed an additional experiment in a different car. Similarly, the radar is installed in the rear view mirror position, as shown in Fig. 5.12. The distance between the radar and the middle rear seat is 1.63 m, and the distance between the radar and the side rear seat is 1.8 m. The personnel information of participants is given in Table 5.6.

I applied the same classification procedure as in the first experiment. The total number of measurements was 194, and a total of more than 100,000 processed signals were obtained for 32 experiment scenarios. Then, I randomly selected 20,858 processed signal and used for classification. The classification result was 92.02% when

Table 5.5: Classification accuracy for various machine learning algorithms

Machine learning algorithm	**Classification accuracy**
Decision tree	64.8%
Boosting	42.8%
SVM	80.4%
Bagging	91.9%

all features were used, and 90.16% when 7 features were used. Therefore, I can conclude that the proposed method can be applied to vehicles of different size.

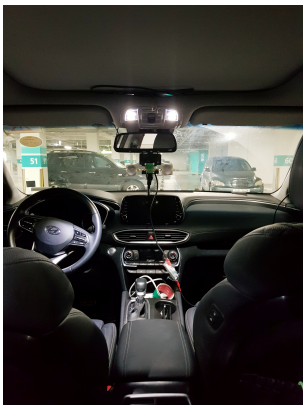


Figure 5.12: Actual photograph of experiment environment: Class E1 (Second experiment)

Table 5.6: Personnel information of experiment participants (Second experiment)

Participant	**Height (cm)**	**Weight (kg)**
A	173	82
D	163	52
F	170	62
G	176	72
H	181	70

5.1.6 Summary

In this study, I proposed a method for monitoring people inside a vehicle by using the statistical characteristics of IR-UWB radar signals. The radar was mounted in front of the rear view mirror, and experiments were conducted for 32 different scenarios depending on the presence of people in each seat. From the received signals, I extracted features representing the statistical characteristics of the signals and used the features as criteria to classify the various arrangements of people. In addition, I investigated the importance of each feature through the NCA algorithm to remove insignificant features and reduce the computational cost. For classification, a bagging algorithm with a decision tree was used to achieve a better generalization performance than using a single classifier. I examined how the classification accuracy changes as the number of features and classifiers increased, and determined the proper number of features and classifiers that resulted in high classification accuracy while keeping the complexity relatively low. The classification results demonstrated that the proposed method can effectively estimate the position and number of people inside a vehicle. To achieve more reliable and accurate performance, measurements in various experiment environments including various objects (e.g., paper boxes, car seats) need to be conducted.

5.2 DEEP NEURAL NETWORK BASED APPROACH USING IR-UWB RADAR

5.2.1 Motivation

In Chapter 5.1, the location of people sitting inside the vehicle by IR-UWB radar sensor was estimated, but the method required extracting feature from the received data. In this study, I propose a deep learning-based method for estimating the location of people inside vehicle using a single IR-UWB radar sensor. The application of deep learning for the classification of radar sensor data have been reported in [90, 91, 92]. The pre-processed time-sampled radar signals are used as the input to the DNN. At this time, the pre-processed signal itself becomes the input to the classifier without extracting any features from it. Then, to find an appropriate network structure for classification, I evaluate the performance by changing the number of hidden layers, the number of nodes in each hidden layer, and the type of activation function. I use 70%, 15%, and 15% of the total data as training, validation, and test sets to verify the performance of the proposed method. The proposed method shows remarkable performance in recognizing the arrangement of people sitting inside the vehicle with an accuracy of 99%. In addition, I also compare the performance of the proposed method with conventional machine learning algorithms such as SVM and decision tree-based methods. The classification results show that the proposed method outperforms the conventional methods. Unlike the method of [93], because the proposed method is not based on feature extraction, a deep understanding of radar signals is not needed. Also, it has the advantage of being able to monitor people inside the vehicle without compromising people's privacy.

The remainder of this study is organized as follows. In Chapter 5.2.2, the proposed DNN-based people localization method is presented. Then, localization results are presented and the classification performance with other machine learning algorithms are compared in Chapter 5.2.3. Additionally, I compare the classification per-

formance with other machine learning algorithms. Finally, conclusions are given in Chapter 5.2.4.

5.2.2 Proposed DNN-Based People Localization

Multi-layer perceptron (MLP) is one of the simplest class of DNN, in which each layer is fully connected to its neighboring layers [94]. The general structure of the MLP network is shown in Fig. 5.13. It consists of an input layer, multiple hidden layers, and an output layer. In addition, each layer is comprised of multiple nodes and nodes are connected to each other through edges. The network is trained through a repeated process of forward propagation and backward propagation. In the forward propagation stage, each layer passes its value to the following layer by using weights and an activation function. Let $\mathbf{x}^{(k)}$ and $\mathbf{y}^{(k)}$ denote the input and output vector at layer k, and $\mathbf{W}^{(k)}$ denote the weight matrix between layer k and $k+1$. Then, the input vector at layer $k+1$ can be expressed as

$$\mathbf{x}^{(k+1)} = f(\mathbf{W}^{(k)}\mathbf{y}^{(k)}), \tag{5.12}$$

where f denotes the activation function that gives nonlinearity to the network. In the backward propagation stage, the weight values are updated by computing the gradient

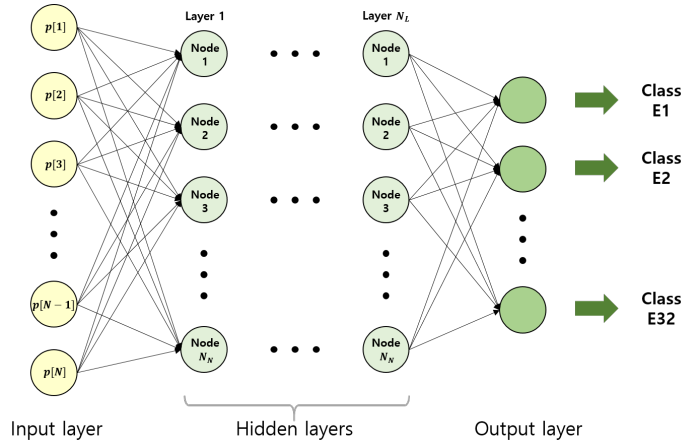


Figure 5.13: General structure of MLP network

of the loss function with respect to each weight. If the weight value before backward propagation is $\mathbf{W}_{before}$, the updated value after backward propagation is

$$\mathbf{W}_{after} = \mathbf{W}_{before} - \alpha \frac{\partial J}{\partial \mathbf{W}_{before}}, \tag{5.13}$$

where α is the learning rate that determines the speed of the learning process and J is the loss function that indicates the error between the estimated and true values. In this study, I used the cross entropy as the loss function, which is commonly used in classification problems [95]. This process of forward and backward propagation, denoted as an epoch, is repeated multiple times to train the weight parameters properly.

In this system, I used the processed radar signal $\hat{\mathbf{p}}$ in (5.4) as the input to the network. Thus, each time-sampled points of $\hat{\mathbf{p}}$ becomes the input to the network, and the number of nodes in the input layer is 512. Also, the number of nodes in the output layer was set as 32 to classify the 32 different scenarios. I used the one hot encoding method, which means that class E1 corresponds to $[1000 \cdots 0]$, class E2 corresponds to $[0100 \cdots 0]$, and so on. The important parameters that determine the performance of the network are the number of hidden layers, the number of nodes in each hidden layer, and the type of activation function. Therefore, I compared the classification accuracy by changing the above-mentioned parameters to find an appropriate network structure for the system.

5.2.3 Performance Evaluation

The measurements were conducted on two different vehicles in Chapter 5.1, and the total number of measurements was 434. Since one measurement contains more than 500 raw radar signals, I obtained more than 220,000 radar signals. In this study, I randomly selected 15% of the total data to avoid the data being biased. This resulted in 35,981 processed signals among a total of more than 220,000 signals. Then, I used 70% of the data as training set, 15% of the data as validation set, and 15% of the data as test set. The input is a 512×1 vector and the output is 32×1 vector. The number of epochs

was set as 1000, and the learning rate was set as 0.01. In addition, I considered two types of activation functions: the sigmoid function and the hyperbolic tangent function. The sigmoid function can be expressed as $1/(1+\exp(-x))$ and the hyperbolic tangent function can be expressed as $(\exp(x) - \exp(-x))/(\exp(x) + \exp(-x))$.

Fig. 5.14 shows the classification accuracy as a function of the number of nodes in a hidden layer. The number of nodes in a hidden layer was increased from 10 to 100 in intervals of 10, while the number of hidden layers was fixed as 1. As can be seen from the figure, the classification accuracy generally increases with an increase in the number of nodes. However, when the number of nodes is higher than 50, there is no prominent increase of classification accuracy regardless of the type of activation function. Therefore, I set the number of nodes in a hidden layer as 50. Moreover, since the classification accuracy is generally higher when hyperbolic tangent function is used, I used this type of activation function for the network. Next, I examined how the classification accuracy changes according to the number of hidden layers, while maintaining the number of nodes in a hidden layer as 50 and using hyperbolic tangent activation function. As shown in Fig. 5.15, the classification accuracy was highest when the number of hidden layers was 3. Therefore, I set the number of hidden layers as 3, and this resulted in a high classification accuracy of 99.5%.

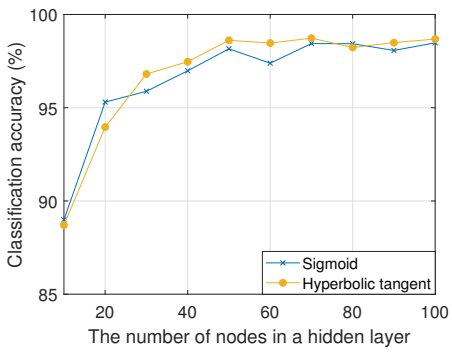


Figure 5.14: Classification accuracy by changing the number of nodes in a hidden layer (Number of hidden layers: 1)

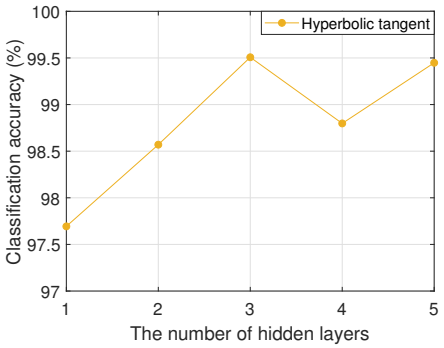


Figure 5.15: Classification accuracy by changing the number of hidden layers (Number of nodes: 50)

Furthermore, I investigated the performance of the network by changing both parameters, the number of hidden layers and the number of nodes, since fixing one parameter might lead to inaccurate results. The number of hidden layers was changed from 1 to 10 and the number of nodes in each hidden layer was changed from 10 to 100 in intervals of 10, resulting in a 10×10 combination of the network structure. The results are shown in Fig. 5.16. In deriving the classification accuracy, a Monte Carlo technique was used to average the results for multiple iterations. In other words, a pseudorandom generator was used to extract 35,981 processed signals randomly and

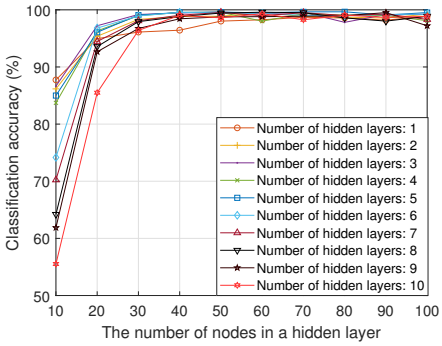


Figure 5.16: Classification accuracy by changing the number of nodes and hidden layers

this process was repeated 10 times, which results in 10 data sets. Then, each data set was used to train the network in parallel, and the classification accuracy was derived by averaging the results. From the figure, I confirmed that the classification accuracy showed a similar trend regardless of the number of hidden layers. Since the computational complexity increases when more nodes and layers are used, I concluded that setting the number of hidden layers as 3 and the number of nodes as 50 is appropriate for the network.

Moreover, I compared the proposed method with other machine learning algorithms such as decision tree, SVM, and bagging with decision tree. The decision tree is a simple classification algorithm that uses a set of hierarchical rules [79]. The model is trained by dividing the data into smaller subsets based on the features of the data. The SVM is a method that divides the input data by finding the maximum margin hyperplane [96]. It can be applied to nonlinear classification problems by using kernel trick, in which the input data is mapped into high-dimensional feature space. Bagging with decision tree combines multiple decision trees and makes a decision by aggregating the prediction results from each decision tree [97]. The variance of the data is reduced through the bagging algorithm, making it robust to over-fitting problem. These algorithms have a similar property in that they all require extracting meaningful features from the data.

The comparison results are summarized in Table 5.7. A single decision tree was not effective in classifying the data due to its simple structure. The SVM with a Gaussian kernel function also showed poor performance because the dimension of the data is high. When bagging with decision tree is used, it resulted in a relatively high classification accuracy of 91.3%, but the proposed DNN method showed the most superior performance of all algorithms. The proposed DNN method also has advantages from other machine learning algorithms because it does not require extracting features from the data.

Furthermore, I investigated how the classification accuracy changes by reducing

Table 5.7: Performance of various machine learning algorithms and DNN

Algorithm	**Classification accuracy**
Decision tree	44.3%
Gaussian SVM	74.6%
Bagging with decision tree	91.3%
DNN	99.5%

the size of the input. To reduce the input size by half, I collected only the odd index data from the processed radar signal $\hat{\mathbf{p}}$ in (5.4). In other words, I used the signal $\hat{\mathbf{p}}' = [\hat{p}[1], \hat{p}[3], \cdots, \hat{p}[511]]$ as input to the network, which has the same effect as doubling the sampling period of the radar system. The other parameters and settings were the same as before; only the input size was changed from 512×1 to 256×1. As a consequence of reducing the input size, the classification accuracy decreased from 99.5% to 99.2%. Since sampling is performed very quickly in an UWB radar system (26 ps), doubling the sampling period did not seriously affect the performance of the proposed network model.

5.2.4 Summary

In this study, I proposed a deep learning-based method to estimate the location and number of people inside the vehicle. First, I accumulated the received IR-UWB radar signals in 32 different measurement scenarios by changing the subjects, the number of subjects, the location of subjects, and the type of vehicle. Then, I trained the classifier using the DNN with the received radar signals. Here, pre-processed radar signals were used as an input to the classifier. Unlike feature extraction machine learning techniques, the proposed method does not require a feature extraction stage. To design a suitable DNN structure for classification, I evaluated the performance of the network by changing the number of hidden layers, the number of nodes in each layer, and the activation function. The classification results showed that the proposed method can be used as an effective in-vehicle localization technique. In addition, the classification performance was improved compared to the conventional machine learning techniques.

5.3 DEEP NEURAL NETWORK BASED APPROACH USING FMCW RADAR

5.3.1 Motivation

Numerous studies have been carried out to detect the occupancy of passengers inside a vehicle using a radar sensor [93, 98, 99, 100]. The location and number of passengers were estimated by extracting features from the impulse radio ultra-wideband (IR-UWB) radar data and using a decision tree as the classifier in [93]. In addition, the authors in [98] proposed a deep learning-based approach that did not require extracting features. A 60 GHz FMCW radar was used to identify the location of passengers in the rear seat using spectral power and Winner entropy [99]. Furthermore, the presence of the driver and breathing rate were estimated using a 60 GHz pulse radar [100]. However, these methods were based on single-channel radar with signal strength information according to distance. When multiple targets are located at the same distance, the signals from these targets can interfere with each other and result in performance degradation.

To solve this problem, a multi-channel radar system that effectively captures spatial information inside a vehicle has been proposed [101, 102]. The features extracted from 77 GHz FMCW radar were used as input to the machine learning algorithm to classify passengers in five different seats [101]. A 77 GHz FMCW radar signal was converted into point cloud data which were then used to identify each zone [102]. However, these methods using the 77 GHz band is difficult to use in practice because it is designated for automotive applications such as adaptive cruise control. The radar inside a vehicle can cause interference with other radar-equipped vehicles, leading to an increased noise floor and false alarms [103]. Therefore, interference problems with the existing radar system must be resolved to successfully identify the passengers inside a vehicle.

In this study, we propose a method for detecting the occupancy of passengers inside a vehicle using multi-channel 60 GHz FMCW radar. The radar is installed at the rear-

view mirror position to provide a line-of-sight (LOS) path between the radar and all seats. The raw radar signal is converted into a range-angle map that produces an image of the in-vehicle environment. Then, to suppress clutter signals caused by stationary objects such as the seat, wall, and roof, two clutter suppression methods are proposed by capturing the phase variation of micro-movements. The resulting range-angle map is used to classify various arrangement of passengers inside the vehicle by applying three classification algorithms: SVM, MLP, and CNN. Because each algorithm uses a different type of input, additional signal processing is required to transform the data structure. The appropriate structure of each network is determined by analyzing the classification accuracy for different hyperparameters; the results of the proposed algorithm are compared with the results reported in a preceding study [101]. The classification results demonstrated that the proposed method can successfully estimate the location and number of passengers inside a vehicle, which is robust to environmental changes.

The main contributions of this study can be as follows. First, the proposed method can be applied inside a vehicle without causing interference with the existing automotive radar systems. By using a 60 GHz band designated for short range radar by the Federal Communications Commission (FCC) in 2021 [104], the proposed method avoids interference by automotive radars using 24 GHz and 77 GHz bands [105, 106]. In addition, the 60 GHz radar can transmit a narrow beam in the desired direction through a miniaturized, high-directivity antenna system [107]. Next, the proposed method can monitor passengers in all seats regardless of their movement. By applying a clutter suppression method based on mean-subtraction (MS) and variance, the signals from passengers can be separated from those of stationary objects. Furthermore, by capturing the micro-movements of human that result from breathing and heartbeat, it is possible to detect both stationary and non-stationary passengers. Lastly, an analysis of various classification algorithms is presented, along with comparison with existing studies. By examining the classification accuracy of various algorithms and

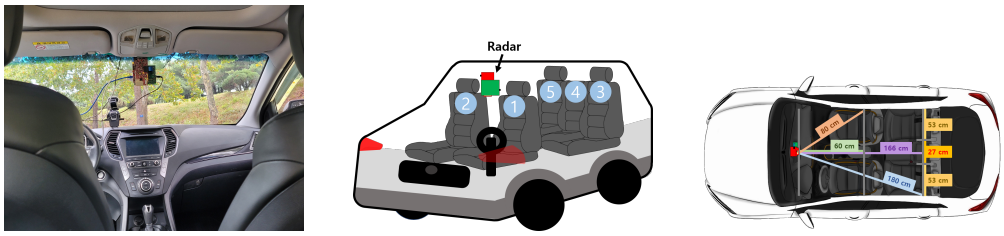
network structures, it is verified that the proposed method using CNN classifier with variance-based clutter suppression method can detect passenger occupancy with the highest accuracy.

The remainder of study is organized as follows. In Chapter 5.3.2, the FMCW radar signal model and the experimental environment are presented. Then, the preprocessing of FMCW radar signal and clutter suppression methods are introduced in Chapter 5.3.3. In Chapter 5.3.4, three passenger occupancy detection methods are presented, which are based on SVM, MLP, and CNN. Next, the performance of the proposed method is evaluated in Chapter 5.3.5 by comparing the performance of various classification algorithms according to network structure. Finally, the conclusions are given in Chapter 5.3.6.

5.3.2 Experimental Environment

For the measurement, an IWR6843ISK FMCW radar sensor manufactured by Texas Instruments was used. The radar sensor was connected to MMWAVEICBOOST and DCA1000EVM modules to capture the raw data. The carrier frequency of the radar was 60 GHz, and a bandwidth of 4 GHz was used to obtain a distance resolution of 3.75 cm [108]. The chirping duration and repetition interval between chirps were 60 μs and 160 μs, respectively. In addition, an array antenna system with two Tx antennas and four Rx antennas is used, which results one Tx antenna and eight Rx antennas having the angle resolution of 15° in the boresight direction [109]. The frame duration was set to 100 ms; the data capture rate was ten frames per second.

The experimental environment is described in Fig. 5.17a. The Santa Fe (DM) manufactured by Hyundai Motor was used which has two front seats and three rear seats. The radar sensor was installed at the rear-view mirror position to provide a LOS path between the radar and all seats. The seats were numbered from 1 to 5, as shown in Fig. 5.17b. The total number of experimental cases was 32 depending on whether a passenger occupied each seat, as shown in Fig. 5.18. Seven people participated in the



(a) Vehicle interior (b) Side view (c) Bird's-eye view

Figure 5.17: Experimental environment

experiment; the experiment was performed by randomly changing the arrangement of participants. Fig. 5.17c shows a bird's-eye view of the experimental environment. The minimum and maximum distances between the radar and front seats were 60 cm and 80 cm, and those between the radar and rear seats were 166 cm and 180 cm, respectively.

5.3.3 Pre-processing of FMCW Radar Signal

Analysis of FMCW Radar Signal

As mentioned in Chapter 2.1, the FMCW radar signal can be formulated as a 3D exponential form, which is commonly referred to as a radar data cube. By applying

Case	Seats					Case	Seats				
	1	2	3	4	5		1	2	3	4	5
C1	X	X	X	X	X	C17	X	O	X	X	X
C2	X	X	O	X	X	C18	X	O	O	X	X
C3	X	X	X	O	X	C19	X	O	X	O	X
C4	X	X	X	X	O	C20	X	O	X	X	O
C5	X	X	O	O	X	C21	X	O	O	O	X
C6	X	X	O	X	O	C22	X	O	O	X	O
C7	X	X	X	O	O	C23	X	O	X	O	O
C8	X	X	O	O	O	C24	X	O	O	O	O
C9	O	X	X	X	X	C25	O	O	X	X	X
C10	O	X	O	X	X	C26	O	O	O	X	X
C11	O	X	X	O	X	C27	O	O	X	O	X
C12	O	X	X	X	O	C28	O	O	X	X	O
C13	O	X	O	O	X	C29	O	O	O	O	X
C14	O	X	O	X	O	C30	O	O	O	X	O
C15	O	X	X	O	O	C31	O	O	X	O	O
C16	O	X	O	O	O	C32	O	O	O	O	O

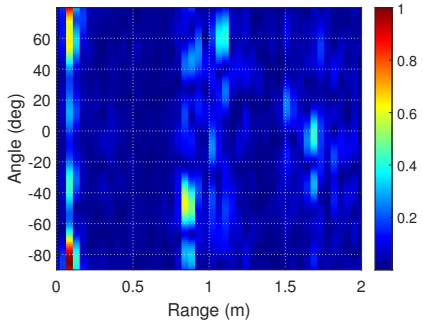
Figure 5.18: Experimental cases

FFT to each dimension of the radar data cube, the range, velocity, and angle spectrum can be obtained. The 2D spectrum in range-angle domain and range-velocity domain are denoted as the range-angle map and range-velocity map, respectively. The range-angle map and range-velocity map are shown in Figs. 5.19a and 5.19b, respectively, with passengers located in seats 3 and 5 (case 6). In Fig. 5.19a, there are undesired signals caused by the antenna coupling effect and background clutters, and the target signals at 1.7 m cannot be clearly identified. The reflected signals from stationary objects such as the seat, wall, and roof can obscure the desired signal because the reflectivity of a passenger is relatively weak. In addition, the range-velocity map in Fig. 5.19b indicates that there are only zero-Doppler components because the passengers are quasi-stationary. Stationary objects are also detected in the zero-Doppler bin; the target signal at 1.7 m can be masked by signals from other stationary objects. Thus, the clutter signals must be suppressed to extract the passenger signals from overlapped signals.

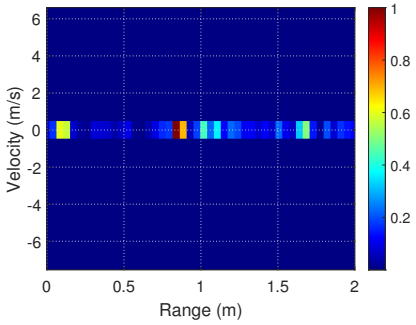
Clutter Reduction

Clutter suppression is generally performed using the Doppler component [110], based on the assumption that the desired target is moving and the clutter signal is stationary. However, for an in-vehicle environment, the passengers are usually in a quasi-stationary state and the foregoing assumption is not valid. The passenger signals can be detected in the nonzero-Doppler bin only if the passengers are intensely moving; this is usually not the case. Thus, a clutter suppression technique that does not use the Doppler component is required so that the method can be applied regardless of passenger movement.

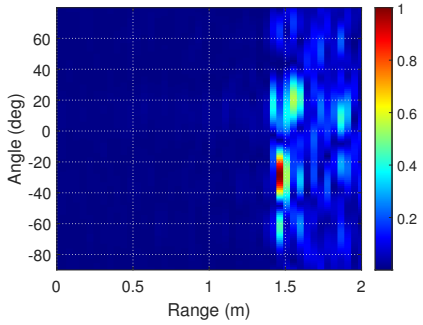
To this end, the phase value ψ_n in (2.8) is used, which is a function of the distance between the radar and target. As shown in Fig. 5.20, the distance to a stationary object does not change over time, whereas a slight variation of distance to a passenger occurs due to movement of the chest caused by breathing. The chest cavity changes in mm-



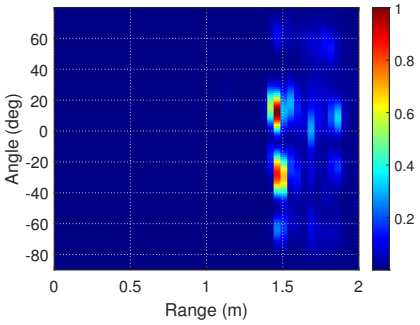
(a) Range-angle map without clutter suppression



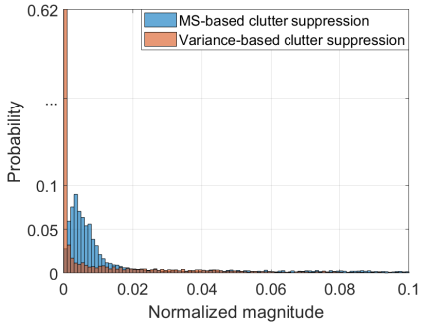
(b) Range-velocity map without clutter suppression



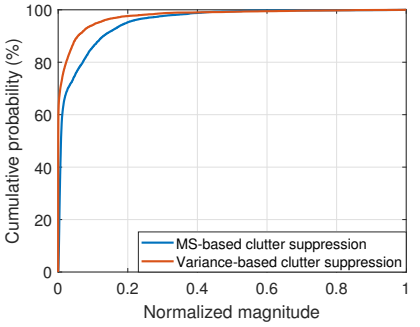
(c) Range-angle map with clutter suppression (MS-based)



(d) Range-angle map with clutter suppression (Variance-based)



(e) Probability distribution of two clutter suppression methods



(f) Cumulative probability distribution of two clutter suppression methods

Figure 5.19: FMCW radar detection results (Case 6)

scale as the diaphragm contracts or expands during inhalation and exhalation [111]. By considering these micro-movements, the distance to the target can be modified as

$$r_n[p] = \bar{r}_n + \Delta r_n[p], \tag{5.14}$$

where p is the frame index, $\bar{r}_n$ is the large-scale distance with a scale greater than cm, and Δr_n is the small-scale distance in mm, respectively. Then, the phase of the target can be re-expressed as

$$\psi_n[p] = \frac{2r_n[p]}{\lambda} = \frac{2(\bar{r}_n + \Delta r_n[p])}{\lambda}. \tag{5.15}$$

By using a 60 GHz millimeter wave FMCW radar sensor with a mm wavelength, the micro-movements of passenger in mm scale can be detected using the phase value. Also, it is possible to separate stationary and non-stationary targets using the phase variations in (5.15) for multiple frames [112]. For example, the large-scale and small-scale distances of stationary objects are both constant; thus, the phase of the corresponding signal does not change over time. In contrast, the large-scale distance of a stationary passenger is constant, but there are small-scale distance variations due to micro-movements and the phase of the passenger changes over time. Moreover, when a passenger is moving, the phase changes more abruptly because the large-scale distance also changes over time. Thus, by analyzing the phase variations of the received signal, it is possible to distinguish passenger signals from stationary clutter signals.

Prior to applying the clutter suppression method, the FMCW radar signal should be modified in an appropriate structure to reflect the phase variation characteristics over time. Because the velocity term is not required for in-vehicle passenger detection, the frequency f_d^n along the chirp index axis in (2.8) is not considered. A range-angle map is generated by applying 2D-FFT to the beat signal in (2.8), which can be expressed as

$$\begin{aligned}\mathbf{X}[f_k, f_l] = &\sum_{k=0}^{N_r-1}\sum_{l=0}^{N_a-1} X[k,l]e^{-j\frac{2\pi}{N_r}kf_k}e^{-j\frac{2\pi}{N_a}lf_l} \\ &(f_k = 0, 1, \cdots, N_r - 1) \\ &(f_l = 0, 1, \cdots, N_a - 1),\end{aligned} \tag{5.16}$$

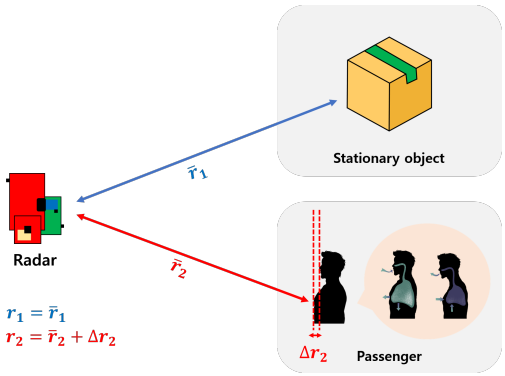


Figure 5.20: Comparison of distance variation for stationary object and passenger

where N_r and N_a are the number of FFT points along the k and l indices, respectively. Because the size of the vehicle is limited and it is not necessary to use all data along the range axis, the size of the 2D matrix is truncated from $\mathbb{R}^{N_r \times N_a}$ to $\mathbb{R}^{\hat{N}_r \times N_a}$. The resulting range-angle map obtained at the p-th frame is denoted as $\widetilde{\mathbf{X}}_p$, which represents the image of passengers inside the vehicle. This process is repeated for P consecutive frames; multiple range-angle maps are accumulated to form a 3D data. The overall procedure of 2D-FFT, truncation, and frame accumulation is summarized in Fig. 5.21.

Two clutter suppression methods based on MS and variance are considered to extract the signal from the desired target. The MS method can be regarded as a high-pass filtering operation that calculates the mean of the accumulated range-angle map for multiple frames and subtracts it from the original range-angle map. The resulting signal after applying the MS method can be expressed as

$$\mathbf{C}^M = \left| \widetilde{\mathbf{X}}_P - \frac{1}{P} \sum_{p=1}^{P} \widetilde{\mathbf{X}}_p \right|, \tag{5.17}$$

which indicates the extent to which the signal fluctuates with respect to its average value. The variance-based method calculates the average of the squared deviation, which can be expressed as

$$\mathbf{C}^V = \frac{1}{P-1} \sum_{p=1}^{P} \left(\widetilde{\mathbf{X}}_p - \frac{1}{P} \sum_{p=1}^{P} \widetilde{\mathbf{X}}_p \right)^2. \tag{5.18}$$

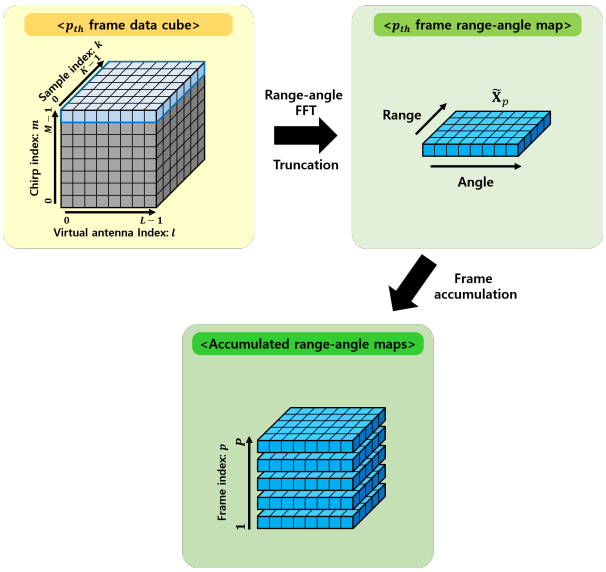


Figure 5.21: Pre-processing of FMCW radar signal

Unlike the MS-based method, the variance-based method captures the characteristics of the range-angle map for multiple frames.

The detection results after applying the two clutter suppression methods are shown in Figs. 5.19c and 5.19d. Compared to the raw range-angle map in Fig. 5.19a, both methods successfully eliminate the undesired signals from the antenna coupling effect and stationary clutter signals; two passengers located at 1.5 m are clearly detected. The difference between the two clutter suppression methods is examined in Figs. 5.19e and 5.19f using the probability distribution of the normalized magnitude of the range-angle map. The probability distribution of normalized magnitude indicates the degree of concentration and can be used to show the prominence of peak value. For the MS-based method, the magnitude of the peak value is smaller than that for the variance-based method, and multiple local peaks with intermediate magnitude are detected around the maximum peak value. Thus, a number of multipath components are present near the passenger location. For the variance-based method, more than 60% of the signal is concentrated near the normalized magnitude of zero, and signals with intermediate

magnitude are less likely to be detected. As a result, the passenger signal is clearly identified in the area where the passenger is located. The range-angle maps obtained from both clutter suppression methods are used as an input to various machine learning algorithms in Chapter 5.3.4.

5.3.4 Passenger Occupancy Detection Methods

In this study, a method of detecting the occupancy of passengers inside a vehicle using the processed range-angle map is presented. The overall block diagram of the proposed method is shown in Fig 5.22. To estimate the location and number of passengers, various machine learning algorithms including the SVM, MLP, and CNN are used. Prior to applying the machine learning algorithms, additional signal processing is applied to transform the range-angle map into a data structure suitable for each machine learning algorithm. The SVM-based method is presented, followed by the MLP-based and CNN-based methods.

SVM-based Method

The SVM is a representative machine learning algorithm that is widely used for classification of multi-dimensional data. The algorithm finds a hyperplane in multi-dimensional space that can separate each individual class with the largest margin. As

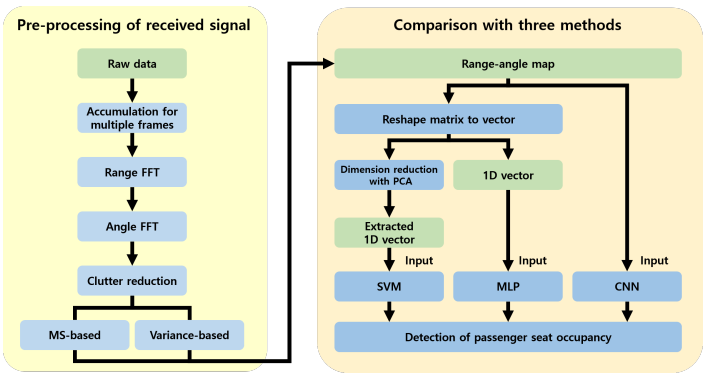
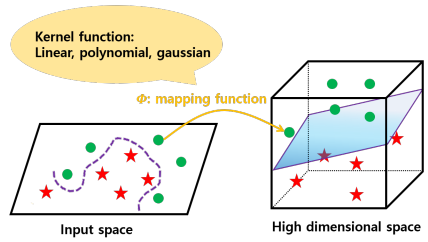
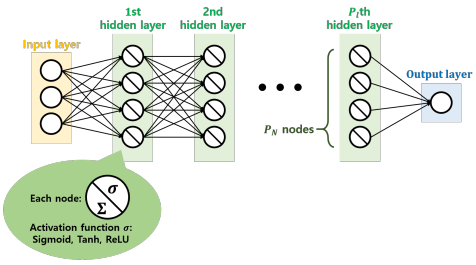


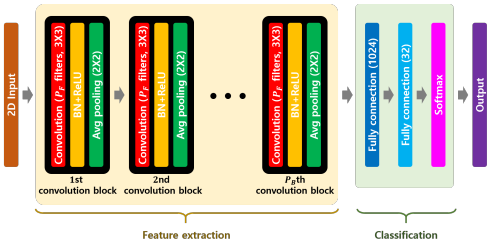
Figure 5.22: Block diagram of proposed method



(a) Structure of SVM



(b) Structure of MLP



(c) Structure of CNN

Figure 5.23: Passenger occupancy detection method with hyper parameter

shown in Fig. 5.23a, the input data is transformed into high-dimensional space using a mapping function. Linear, polynomial, and Gaussian kernel functions are commonly used as mapping functions to reduce the computational complexity. In this study, the SVM algorithm is used to classify various arrangement of passengers inside the vehicle.

To apply the range-angle map as an input to the SVM, the 2D matrix of size $\mathbb{R}^{\hat{N}_r \times N_a}$ must be vectorized as a 1D vector of size $\mathbb{R}^{1 \times \hat{N}_r N_a}$. However, because the size of the vector is too large, using this vector as the input increases the computational

complexity in the classifier; a phenomenon known as the 'curse of dimensionality' can occur [113]. Thus, the dimension of the input data must be reduced to be suitable as an input to the SVM.

The PCA [61] algorithm can be used to reduce the dimension of input data by finding a set of principal axes that contain as much information as the original data. The principal axes are found by applying eigen-decomposition to the input data. Because the size of the vectorized 1D range-angle map is $\mathbb{R}^{1\times\hat{N}_r N_a}$, $\hat{N}_r N_a$ eigenvalues and eigenvectors are generated. The eigenvector and eigenvalue represent the principal axis and the data variance explained by the corresponding eigenvector, respectively. To reduce the dimension of data, a subset of principal axes with the highest eigenvalues is selected by solving the following optimization problem,

$$\begin{aligned} &\text{minimize } \alpha \\ &\text{subject to } \frac{\sum_{i=1}^{\alpha}\lambda_i}{\sum_{i=1}^{\hat{N}_r N_a}\lambda_i}\times 100 \geq T, \end{aligned} \tag{5.19}$$

where λ_i $(i = 1, \cdots, \hat{N}_r N_a)$ are the eigenvalues and α is the optimization parameter representing the number of principal axes to be used. After α principal axes are determined, the original input data are projected onto these axes and the size of the 1D vector is reduced from $\mathbb{R}^{1\times\hat{N}_r N_a}$ to $\mathbb{R}^{1\times\alpha}$. The appropriate value of T determining the dimension of data is discussed in Chapter 5.3.5.

MLP-based Method

The MLP is a basic algorithm in a deep neural network where each layer is fully connected to the adjacent layers. Unlike machine learning algorithms such as the SVM, feature extraction is not necessary. The structure of the MLP network is shown in Fig. 5.23b, which consists of an input layer, multiple hidden layers, and an output layer. For the input layer, the vectorized 1D range-angle map of size $\mathbb{R}^{1\times\hat{N}_r N_a}$ is used; the output layer consists of 32 nodes to classify 32 different arrangement of passengers inside the vehicle.

The weight parameters of the network are updated through a process of forward and backward propagation, which is defined as an epoch. In forward propagation, the weighted sum of the input is passed through an activation function and sent to the next layer. Activation functions such as the sigmoid, hyperbolic tangent, and ReLU functions are used to introduce non-linearity to the data. In backward propagation, the weight parameters are updated using the gradient descent algorithm to minimize the loss function between the actual and estimated value. The cross-entropy is used as the loss function, and the softmax function is used in the output layer to convert the output value into a probability distribution.

CNN-based Method

The CNN is one of the most widely used algorithms in a deep neural network which is mainly used to analyze image or video data. Because the CNN uses high-dimensional data in forward and backward propagation, the spatial information of high-dimensional input data such as an RGB image can be preserved. In this study, the 2D range-angle map of size $\mathbb{R}^{\hat{N}_r \times N_a}$ is used as input to the CNN; the input is normalized to have a value between zero and one. The overall structure of the proposed CNN network is shown in Fig. 5.23c, which can be divided into the feature extraction stage and the classification stage. In the feature extraction stage, multiple convolution blocks are used to extract features representing the properties of the input data. Each block consists of multiple layers that perform convolution, batch normalization (BN), non-linear transformation, and average pooling. In the convolution process, the size of the filter is set as 3×3, and zero-padding is used to maintain the size of the input. Also, an average pooling with a filter size of 2×2 and a stride of 2 is applied to the output of the ReLU layer to downsample the data and prevent overfitting. In the classification stage, the output of the feature extraction stage is passed through multiple fully connected layers. Similar to the structure used in the MLP, the number of nodes in the final fully connected layer is set as 32 to classify 32 different arrangement of

passengers inside the vehicle. Using this CNN structure, the weight parameters are updated through forward and backward propagation.

5.3.5 Performance Evaluation

In this study, the performance of the proposed in-vehicle passenger occupancy detection method is evaluated using experimental data. The experiment was conducted inside the vehicle for 32 different cases. For each experimental case, measurement was conducted ten times with randomly changed participants. The number of frames per measurement was set as 600; each measurement lasted for a minute. In accumulating multiple range-angle maps as shown in Fig. 5.21, the number of frames to accumulate (P) was determined by considering the micro-movement cycle of human arising from respiration. For an adult, the respiration rate is approximately 12-20 breaths per minute; a child or elderly person has a higher respiration rate [114]. To capture at least half of the respiration cycle and identify the phase change arising from respiration, the number of frames to accumulate (P) was set as 20. Thus, 600 frames of radar data obtained for each measurement were divided into 30 non-overlapping sections with 20 frames. In addition, when training the SVM, MLP, and CNN models, the five-fold cross-validation method was used to evaluate the performance of the trained model. The entire dataset was divided into five groups; one group was used as the test set and the remaining groups were used as the training set. This process was repeated five times, such that each group is used as a test set; the final classification accuracy was calculated by averaging the results from each test set.

First, I investigated the performance of the PCA algorithm to analyze the importance of each principal component and reduce the size of the input. The eigenvalue distributions using two different clutter suppression methods are shown in Fig. 5.24. In the figure, the eigenvalues are plotted in descending order starting from the first principal component, along with the cumulative eigenvalues. When using the variance-based method, the eigenvalues of a few upper principal components accounted for most

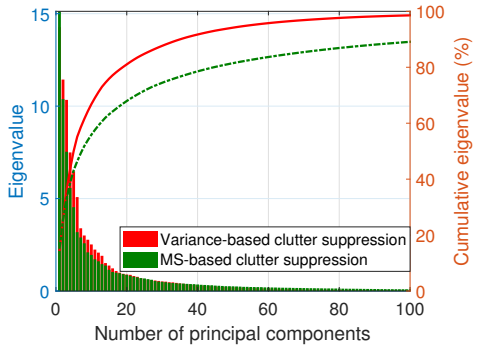


Figure 5.24: Principal component analysis of two clutter reduction methods

of the data. The cumulative eigenvalues corresponding to 70%, 80%, 90%, and 99% were obtained using 12, 19, 36, and 117 upper principal components, respectively. This means that the data are distributed along a few principal axes, which is consistent with the probability distribution results in Chapter 5.3.3. The MS-based method showed a similar trend, but the degree of concentration was slightly less than that of the variance-based method. The cumulative eigenvalues corresponding to 70%, 80%, 90%, and 99% were obtained using 23, 45, 111, and 423 upper principal components, respectively. Compared to the variance-based method, more principal components were required to explain the same amount of data. The number of principal components to be used was determined by solving the optimization problem in (5.19) by varying the threshold parameter T.

After reducing the dimension of data using the PCA algorithm, I applied the SVM algorithm to classify various arrangements of passengers inside the vehicle. Fig. 5.25 shows the classification accuracy by varying the threshold coefficient T and kernel functions. For both clutter suppression methods, a linear kernel function produced a higher classification accuracy than polynomial and Gaussian kernel functions. For the MS-based method, the classification accuracy was highest with a threshold value of 75, which corresponds to using 32 principal components. The variance-based method produced the highest classification accuracy when the threshold value was 90, which

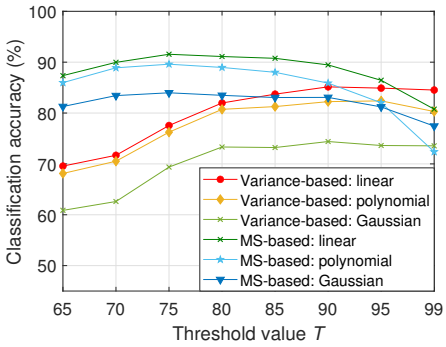
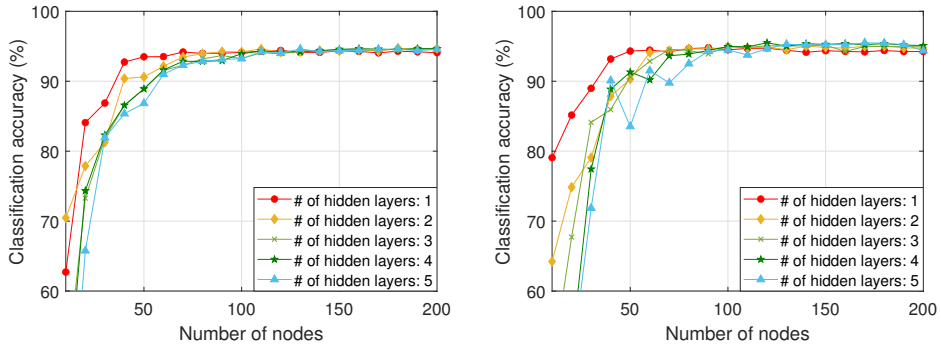


Figure 5.25: Classification results using SVM method

corresponds to using 36 principal components. The classification accuracy in these conditions was 91.56% for the MS-based method and 85.15% for the variance-based method.

Next, I analyzed the performance of the MLP algorithm using the vectorized 1D range-angle map as the input. The hyperparameters that must be considered in the MLP algorithm are the type of activation function and the number of layers and nodes. Therefore, the network performance was evaluated by changing these hyperparameters. Among the ReLU, sigmoid, and hyperbolic tangent activation functions, the sigmoid function showed the best training performance. Then, to find the appropriate number of layers and nodes, I varied the number of layers from one to five and the number of nodes from 10 to 200 in intervals of 10. Fig. 5.26 shows the network performance with the sigmoid function used as the activation function. For both clutter suppression methods, the classification accuracy tended to increase and converge with an increasing number of nodes. For the highest classification accuracy, 2 layers and 180 nodes were used for the MS-based method, resulting in a classification accuracy of 94.74%. Also, for the variance-based method, a classification accuracy of 95.50% was obtained with 5 layers and 170 nodes.

Furthermore, I applied the CNN algorithm using the 2D range-angle map as the input. Similar to the MLP-based method, the CNN network performance was analyzed

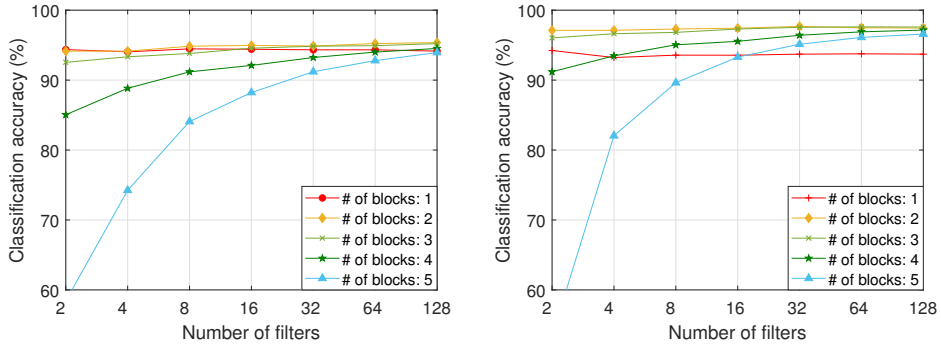


(a) MS-based method (b) Variance-based method

Figure 5.26: Classification results using MLP method

by changing hyperparameters such as the number of filters and blocks. Fig. 5.27 shows the classification accuracy by changing the number of filters from 2 to 128 in units of 2^n and the number of blocks from 2 to 5. In both cases, the CNN network performance was highest when the number of blocks was 2. The number of filters resulting in the highest classification accuracy was 128 and 32, respectively, with classification accuracies of 95.36% and 97.68%.

The classification accuracy of various methods is summarized in Table 5.8. The



(a) MS-based method (b) Variance-based method

Figure 5.27: Classification results using CNN method

Table 5.8: Comparison of classification methods

Type	Accuracy
SVM method (MS-based) [101]	91.56%
SVM method (Variance-based)	85.15%
MLP method (MS-based)	94.74%
MLP method (Variance-based)	95.50%
CNN method (MS-based)	95.36%
CNN method (Variance-based)	97.68%

method in [101] used the MS-based clutter suppression method; the PCA algorithm was used to reduce the dimension of data before applying the SVM algorithm. The method is similar to the MS-based SVM method in this study, resulting in a classification accuracy of 91.56%. The MLP method resulted in higher classification accuracy than the SVM method for both clutter suppression methods because the MLP method uses the entire range-angle map as the input, whereas the SVM method is based on feature extraction and information can be lost. The highest classification performance was achieved using the CNN method. The CNN method can learn the non-linear characteristics of the data using the 2D range-angle map as the input. Unlike the SVM and MLP methods that vectorize the 2D range-angle map, the CNN method can preserve the spatial information and correlation between distance and angle. Moreover, the variance-based method demonstrated higher classification performance than the MS-based method when a deep neural network was used. This is because the variance-based method has prominent peaks at passenger location and few intermediate values, as shown in Figs. 5.19 and 5.24. As a result, the variance-based CNN method demonstrated the highest classification accuracy of all methods.

5.3.6 Summary

In this study, I proposed a method to detect passenger occupancy inside a vehicle using multi-channel 60 GHz FMCW radar sensor. The measurement was conducted for 32 experimental cases by changing the location and number of passengers. The received radar signal was converted into a range-angle map, and clutter suppression was performed using MS-based and variance-based methods. To detect the occupancy of passengers in each seat, three classification algorithms (SVM, MLP, and CNN) were used. Because each algorithm uses a different type of input, additional signal processing was performed to transform the data into a form suitable for input. The appropriate structure of each algorithm was determined by varying the network hyperparameters and analyzing the classification accuracy. Of all classification algorithms and clutter suppression methods, the CNN algorithm using variance-based clutter suppression produced the highest classification accuracy (97.68%). Thus, it is verified that the location and number of passengers inside a vehicle can be accurately estimated regardless of whether passengers are moving. Moreover, the performance of the proposed method can be further improved using high-resolution imaging radar with enhanced angular resolution.

Chapter 6

CONCLUSION

In this dissertation, I proposed various radar signal processing techniques to solve issues that must be considered in autonomous driving. The research topic was divided into three categories: ego-motion estimation, orientation estimation of surrounding vehicles, and occupancy detection of passengers inside the vehicle. The research was conducted by using an IR-UWB and FMCW radar.

First, I proposed a method to estimate the ego-motion of the radar-equipped platform by only using radar sensor data. The rotation angle was estimated by using the distribution of the detected points at successive time instants and correlating with each other. In addition, the moving velocity of the robot was estimated from the trend line formed by the detected points on the angle-velocity 2D plane. The estimation results showed that the proposed method can estimate the ego-motion of the robot with high accuracy. Also, the proposed method can be applied to estimate the ego-velocity of the vehicle using automotive radar.

Next, I proposed a method to estimate the heading direction of vehicle using automotive radar. The received radar data were converted into range-angle map or point cloud data. By using a high-resolution range-angle map, various movements of the front vehicle were classified with high accuracy. Also, by using the point cloud data, the orientation angle was accurately estimated by applying regression algorithm.

Finally, I proposed a method to detect the occupancy of passengers inside vehicle using IR-UWB or FMCW radar sensor. The received radar signal was processed to separate target signal from clutter and noise. Two methods were proposed based on whether the features are extracted or not. The performance was compared by using various machine learning algorithms, and the number and position of people sitting inside vehicle was accurately estimated.

Bibliography

[1] S. M. Pateole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Process. Mag.*, vol. 34, no. 2, pp. 22-35, Mar. 2017.

[2] J. Bechter, M. Rameez, and C. Waldschmidt, "Analytical and experimental investigations on mitigation of interference in a DBF MIMO radar," *IEEE Trans. Microw. Theory and Techn.*, vol. 65, no. 5, pp. 1727-1734, May 2017.

[3] Y.-J. Yoon, S. Lee, B.-h. Lee, S.-C. Kim, and C. Lee "Enhanced clutter removal and peak detection methods for localization using IR-UWB radar," in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2017, pp. 313-317.

[4] B.-h. Lee, S. Lee, Y.-J. Yoon, K.-M. Park, and S.-C. Kim, "Adaptive clutter suppression algorithm for human detection using IR-UWB radar," in *Proc. IEEE SENSORS*, Nov. 2017, pp. 1-3.

[5] I.I. Immoreev and P.G.S.D.V. Fedotov, "Ultra wideband radar systems: advantages and disadvantages," in *Proc. IEEE Conf. Ultra Wideband Syst. Technol.*, May 2012, pp. 201-205.

[6] H. Kanada and N. Aoshima, "Analog Gabor transform filter with complex first order system," in *Proc. SICE*, Jul. 1997, pp. 925-930.

[7] F. Roos, J. Bechter, C. Knill, B. Schweizer, and C. Waldschmidt, “Radar sensors for autonomous driving: Modulation schemes and interference mitigation,” *IEEE Microw. Mag.*, vol. 20, no. 9, pp. 58-72, Sep. 2019.

[8] S. Liang, Z. Cao, C. Wang, and J. Yu, ‘’A novel 3D LiDAR SLAM based on directed geometry point and sparse frame,” *IEEE Robot. and Autom. Lett.*, vol. 6, no. 2, pp. 374-381, Apr. 2021.

[9] T.-j. Lee, C.-h. Kim, and D.-i. D. Cho, “A monocular vision sensor-based efficient SLAM method for indoor service robots,” *IEEE Trans. Ind. Electron.*, vol. 66, no. 1, pp. 318-328, Jan. 2019.

[10] G. Schouten and J. Steckel, “RadarSLAM: Biomimetic SLAM using ultra-wideband pulse-echo radar,” in *Proc. Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2017, pp. 1-8.

[11] J. W. Marck, A. Mohamoud, E. vd Houwen, and R. van Heijster, “Indoor radar SLAM: A radar application for vision and GPS denied environments,” in *Proc. Eur. Radar Conf.*, Oct. 2013, pp. 471-474.

[12] Y. Li, Y. Liu, Y. Wang, Y. Lin, and W. Shen, “The Millimeter-Wave Radar SLAM Assisted by the RCS Feature of the Target and IMU,” *Sensors*, vol. 20, no. 18, pp. 1-27, Sep. 2020.

[13] S. Lee, S.-Y. Kwon, B.-J. Kim, H.-S. Lim, and J.-E. Lee, “Dual-mode radar sensor for indoor environment mapping,” *Sensors*, vol. 21, no.7, pp. 1-16, Apr. 2021.

[14] Y. S. Park, J. Kim, and A. Kim, “Radar localization and mapping for indoor disaster environments via multi-modal registration to prior LiDAR map,” in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2019, pp. 1307-1314.

[15] M. Holder, S. Hellwig, and H. Winner, “Real-time pose graph SLAM based on radar,” in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2019, pp. 1-7.

[16] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous full-motion estimation of arbitrary objects using dual Doppler radar," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2014, pp. 324-329.

[17] H.-S. Lim, J.-E. Lee, H.-M. Park, and S. Lee, "Stationary target identification in a traffic monitoring radar system," *Sensors*, vol. 10, no. 17, pp. 1-15, Aug. 2020.

[18] Z. Xu, C. J. Baker, and S. Pooni, "Range and Doppler cell migration in wideband automotive radar," *IEEE Trans. Veh. Technol.*, vol. 68, no. 6, pp. 5527-5536, Jun. 2019.

[19] M. T. Emirler, K. Kahraman, M. Şentürk, B. A. Güvenç, L. Güvenç, and B. Efendioğlu, "Vehicle yaw rate estimation using a virtual sensor," *Int. J. Veh. Technol.*, vol. 2013, pp. 1-13, Apr. 2013.

[20] S. Lim and S. Lee, "Hough transform based ego-velocity estimation in automotive radar system", *Electron. Lett.*, vol. 57, no. 2, Jan. 2021.

[21] S. H. Chen and P. Newman, "Precise ego-motion estimation with millimeter-wave radar under diverse and challenging conditions", in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2018, pp. 6045-6052.

[22] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using multiple Doppler radars", in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2014, pp. 1592-1597.

[23] S. Boyd and L. Vandenberghe, *Convex Optimization*. 1st ed. Cambridge, U.K.: Cambridge Univ. Press, 2004.

[24] P. H. S. Torr and D. W. Murray, "The development and comparison of robust methods for estimating the fundamental matrix," *Int. J. of Comput. Vis.*, vol. 24, no. 3, pp. 271-300, Sep. 1997.

[25] J. Illingworth and J. Kittler, “A survey of the Hough transform,” *Comput. Vis., Grpah., Image Process.*, vol. 44, no. 1, pp. 87-116, Oct. 1988.

[26] K. G. Derpanis, “Overview of the RANSAC algorithm,” *Image Rochester NY*, vol. 4, no. 1, pp. 2–3, 2010.

[27] J. Hayakawa and B. Dariush, “Ego-motion and surrounding vehicle state estimation using a monocular camera,” in *Proc. Intell. Vehicles. Symp.*, Jun. 2019, pp. 2550-2556.

[28] J. Almeida and V. M. Santos, “Real time egomotion of a nonholonomic vehicle using LIDAR measurements,” *J. Field Robot.*, vol. 2013, no. 1, pp. 129-141, Feb. 2013.

[29] M. A. Fischler and R. C. Bolles, “Random sample consensur: A paradigm for model fitting with applications to image analysis and automated cartography,” *Commun. ACM*, vol. 24, no. 6, pp. 381-395, Jun. 1981.

[30] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. 2nd ed., Cambridge, U.K.: Cambridge Univ. Press, 2003.

[31] R. O. Duda and P. E. Hart, “Use of the Hough transformation to detect lines and curves in pictures,” *Commun. ACM*, vol. 15, no. 1, pp. 11-15, 1972.

[32] Y. Okamoto, I. Matsunami, and A. Kajiwara, “Moving vehicle discrimination using Hough transformation,” in *Proc. IEEE Radio Wireless Symp. (RWS)*, Jan. 2011, pp. 367-370.

[33] H. Song and H.-C.Shin, “Classification and spectral mapping of stationary and moving objects in road environments using FMCW radar,” *IEEE Access*, vol. 8, pp. 22955-22963, Jan. 2020.

[34] B. R. Mahafza, *Radar Systems Analysis and Design Using MATLAB*. 3rd ed., Florida, USA: CRC Press, 2013.

[35] F. Gross, *Smart Antennas for Wireless Communications with MATLAB*. 1st ed., New York, USA: McGraw-Hill, 2005.

[36] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Instantaneous ego-motion estimation using Doppler radar," in *Proc. 16th int. IEEE Conf. Intell. Transp. Syst. (ITSC)*, Oct. 2013, pp. 869-874.

[37] F. Nilsson, "Vehicle Tracking with Heading Estimation using a Mono Camera System," Master's thesis, Linköping University, Linköping, Sweden, 2018.

[38] A. S. A. Rachman, "3D-LIDAR Multi-object Tracking for Autonomous Driving: Multi-Target Detection and Tracking Under Urban Road Uncertainties," Master's thesis, Delft University of Technology, Delft, Netherlands, 2017.

[39] J. Schmidhuber, "Deep learning in neural networks: An overview," *Neural Netw.*, vol. 61, pp. 85-117, Jan. 2015.

[40] J. M. García, D. Zoeke, and M. Vossiek, "MIMO-FMCW radar-based parking monitoring application with a modified convolutional neural network with spatial priors," *IEEE Access*, vol. 6, no. 1, pp. 41391-41398, Jul. 2018.

[41] Z. Zhang, Z. Tian, and M. Zhou, "Latern: Dynamic continuous hand gesture recognition using FMCW radar sensor," *IEEE Sensors J.*, vol. 18, no. 8, pp. 3278-3289, Apr. 2018.

[42] Texas Instruments, "AWR1642 Single-Chip 77- and 79-GHz FMCW Radar sensor," May 2017. https://www.ti.com/product/AWR1642/ (accessed June 26, 2020).

[43] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, vol. 34, no. 3, pp. 276-280, Mar. 1986.

[44] D. Kellner, M. Barjenbruch, J. Klappstein, J. Dickmann, and K. Dietmayer, "Wheel extraction based on micro Doppler distribution using high-resolution

radar," in *Proc. IEEE MTT-S Int. Conf. Microw. Intell. Mobility (ICMIM)*, pp. 1-4, Apr. 2015.

[45] T. Visentin, J. Hasch, and T. Zwick, "Polarimetric RCS measurements of selected two-wheeled vehicles for automotive radar," in *Proc. Eur. Radar Conf.*, pp. 53-56, Oct. 2017.

[46] Mark A. Richards, *Fundamentals of Radar Signal Processing*. New York, USA: McGraw-Hill, 2005.

[47] R. F. W. Coates, G. J. Janacek, and K. V. Lever, "Monte Carlo simulation and random number generation," *IEEE J. Sel. Areas. Commun.*, vol. 6, no. 1, pp. 58-66, Jan. 1988.

[48] L. van de Maaten and G. Hinton, "Visualizing data using t-SNE," *J. Mach. Learn. Res.*, vol. 9, pp. 2579-2605, Nov. 2008.

[49] F. Folster and H. Rohling, "Lateral velocity estimation based on automotive radar sensors," *in Proc. CIE Int. Conf. Radar*, Oct. 2006, pp. 1-4.

[50] X. Zhu, Z. Luo, P. Fu, and X. Ji, "VOC-ReID: Vehicle re-identification based on vehicle-orientation-camera," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR) Workshops*, Jun. 2020, pp. 2566-2573.

[51] P. E. Rybski, D. huber, D. D. Morris, and R. Hoffman, "Visual classification of coarse vehicle orientation using Histogram of Oriented Gradients features," in *Proc. IEEE Intell. Vehicles Symp.*, Jun. 2010, pp. 877-884.

[52] Y. Zeng, Y. Hu, S. Liu, J. Ye, Y. Han, X. Li, and N. Sun, "RT3D: Real-time 3-D vehicle detection in LiDAR point cloud for autonomous driving," *IEEE Robot. Autom. Lett.*, vol. 3, no. 4, pp. 3434-3440, Oct. 2018.

[53] Z. Peng and C. li, "A portable K-band 3-D MIMO radar with nonuniformly spaced array for short-range localization," *IEEE Trans. Microw. Theory Techn.*, vol. 66, no. 11, pp. 5075-5086, Nov. 2018.

[54] Y. Kim, I. Alnujaim, and D. Oh, "Human activity classification based on point clouds measured by millimeter wave MIMO radar with deep recurrent neural networks," *IEEE Sensors J.*, vol. 21, no. 12, Jun. 2021.

[55] M. Andres, P. Feil, W. Menzel, H.-L. Bloecher, and J. Dickmann, "3D detection of automobile scattering centers using UWB radar sensors at 24/77 GHz," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 28, no. 3, pp. 20–25, Mar. 2013.

[56] D. Kellner, M. Barjenbruch, K. Dietmayer, J. Klappstein, and J. Dickmann, "Instantaneous lateral velocity estimation of a vehicle using Doppler radar," in *Proc. 16th Int. Conf. Inf. Fusion*, Jul. 2013, pp. 877-884,.

[57] S. Lim, J. Jung, B.-h. Lee, S.-C. Kim, and S. Lee, "CNN-based estimation of heading direction of vehicle using automotive radar sensor," *IET Radar, Sonar Navigat.*, vol. 15, no. 6, pp. 618-626, Jun. 2021.

[58] F. Roos, D. Kellner, J. Dickmann, and C. Waldschmidt, "Reliable orientation estimation of vehicles in high-resolution radar images," *IEEE Trans. Microw. Theory Techn.*, vol. 64, no. 9, pp. 921–928, Sep. 2016.

[59] R. Nitzberg, "Clutter map CFAR analysis," *IEEE Trans. Aerosp. Electron. Syst.*, vol. AES-22, no. 4, pp. 419-421, Jul. 1986.

[60] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining*, Aug. 1996 pp. 226-231.

[61] J. Shlens, "A tutorial on principal component analysis," 2014, *arXiv:1404.1100*.

[62] T. Zhang and B. Yang, "Big data dimension reduction using PCA," in *Proc. IEEE Int. Conf. Smart Cloud (SamrtCloud)*, Nov. 2016, pp. 152-157.

[63] A. de Cheveigné and J. Z. Simon, "Denoising based on time-shift PCA," *J. Neurosci. Methods*, vol. 165, no. 2, pp. 297-305, Sep. 2007.

[64] H. Huang, X. Liu, T. Zhang, and B. Yang, "Regression PCA for moving objects separation," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2016, pp. 1-6.

[65] L. Breiman, J. H. Friedman, R. A. Olshen, and C. J. Stone, *Classification and Eegression Trees*. Boca Raton, USA: CRC Press, 1984.

[66] W. Yang, K. Wang, and W. Zuo, "Neighborhood component feature selection for high-dimensional data," *J. Comput*, vol. 7, no.1, pp. 161-168, Jan. 2012.

[67] G. Toussaint, "Solving geometric problems with the rotating calipers," in *Proc. Medit. Electrotech. Conf.*, May 1983, pp. 1-8.

[68] V. Aiello, P. N. Borazjani, E. Battista, and M. Albanese, "Next-generation technologies for preventing accidental death of children trapped in parked vehicles," in *Proc. IEEE 15th Int. Conf. Inf. Reuse Integr.*, Aug. 2014, pp. 508-513.

[69] J. N. Booth 3rd, G. G. Davis, J. Waterbor, and G. McGwin Jr, "Hyperthermia deaths among children in parked vehicles: an analysis of 231 fatalities in the United States, 1999–2007," *Forensic Sci., Med., Pathol.*, vol. 6, no. 2, pp. 99-105, Jun. 2010.

[70] M. F. Abulkhair, A. Aldahiri, H. Alkhatabi, H. Alonezi, L. Mulla, and S. Razzaq, "Sensor based hyperthermia alert car application," *Int. J. Appl. Inf. Syst.*, vol. 5, no. 2, pp. 44-55, May 2016.

[71] J. P. S. Barrera, G. M. Sandoval, G. C. Ortiz, R. N. González, and E. R. Aguilar, "A multi-agent system to avoid heatstroke in young children left in baby car seats

inside vehicles," in *Proc. Int. Conf. Comput. Sci. Comput. Intell.*, Mar. 2014, pp. 245-248.

[72] S. Abdulatif, Q. Wei, F. Aziz, B. Kleiner, and U. Schneider, "Micro-Doppler based human-robot classification using ensemble and deep learning approaches," in *Proc. IEEE Radar Conf. (RadarConf)*, Apr. 2018, pp. 1043-1048.

[73] D. W. Paglieroni, D. H. Chambers, J. E. Mast, S. W. Bond, and N. R. Beer, "Imaging modes for ground penetrating radar and their relation to detection performance," *IEEE J. Sel. Topics Appl. Earth Observ. Reomete Sens.*, vol. 8, no. 3, pp. 1132-1144, Mar. 2015.

[74] J. Li, Z. Zeng, J. Sun, and F. Liu, "Through-wall detection of human being's movement by UWB radar," *IEEE Geosci. Remote Sens. Lett.*, vol. 9, no. 6, pp. 1079-1083, Nov. 2012.

[75] M. G. Amin, *Radar for Indoor Monitoring: Detection, Classification, and Assessment*. Boca Raton, FL, USA: CRC Press, 2017.

[76] N. Hafner, I. Mostafanezhad, V. M. Lubecke, O. Boric-Lubecke, and A. Host-Madsen, "Non-contact cardiopulmonary sensing with a baby monitor," in *Proc. 29th Annu. Int. Conf. IEEE Eng. Med. Biol. Soc.*, Aug. 2007, pp. 2300-2302.

[77] A. R. Diewald, J. Landwehr, D. Tatarinov, P. D. M. Cola, C. Watgen, C. Mica, M. Lu-Dac, P. Larsen, O. Gomez, and T. Goniva, "RF-based child occupation detection in the vehicle interior," in *Proc. 17th Int. Radar Symp. (IRS)*, May 2016, pp. 1-4.

[78] *2019 Hyundai Santa Fe Rear Occupant Alert Aims to Protect Kids From Hot Cars.* Accessed: Oct. 27, 2019. [Online]. Available: https://www.consumerreports.org/car-safety/hyundai-santa-fe-rear-occupant-alert-aims-to-protect-kids-from-hot-cars/

[79] D. Lowd and J. Davis, “Improving Markov network structure learning using decision trees,” *J. Mach. Learn. Res.*, vol. 15, pp. 501-532, Feb. 2014.

[80] J. Su and H. Zhang, “A fast decision tree learning algorithm,” in *Proc. 21st Nat. Conf. on Artif. Intell.*, Jul. 2006, pp. 500-505.

[81] J. M. Lee, J. W. Choi, and S. H. Cho, “Movement analysis during sleep using an IR-UWB radar sensor,” in *Proc. IEEE Int. Conf. Netw. Infrastruct. Digi. Content*, Sep. 2016, pp. 486-490.

[82] S. Lee, B.-H. Lee, J.-E. Lee, and S.-C. Kim, “Statistical characteristic-based road structure recognition in automotive radar systems,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 7, pp. 2418-2429, Jul. 2019.

[83] J. M. Keller, M. R. Gray, and J. A. Givens, “A fuzzy K-nearest neighbor algorithm,” *IEEE Trans. Syst., Man, Cybern.*, vol. SMC-15, no. 4, pp. 580-585, Jul./Aug. 1985.

[84] C.-C. Chang and C.-J. Lin “LIBSVM: A library for support vector machine,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, pp. 1-27, Apr. 2011.

[85] C. Zhang and Y. Ma, *Ensemble Machine Learning: Methods and Applications*. New York, NY, USA: Springer, 2012.

[86] Y. Freund, R. E. Schapire, and N. Abe, “A short introduction to boosting,” *J. Jpn. Soc. Artif. Intell.*, vol. 14, no. 5, pp. 771-780, Sep. 1999.

[87] L. Breiman, “Bagging predictors,” *Mach. Learn.*, vol. 24, no. 2, pp. 123-140, Aug. 1996.

[88] G. Martínez-Muñoz, D. Hernández-Lobato, and A. Suárez, “An analysis of ensemble pruning techniques based on ordered aggregation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 2, pp. 245-259, February 1996.

[89] I. H. Whitten and E. Frank, *Data Mining: Practical Machine Learning Tools and Techniques with JAVA Implementations*. San Mateo, CA, USA: Morgan Kaufmann, 2000.

[90] K. Huang, W. Nie, and N. Luo, "Fully polarized SAR imagery classification based on deep reinforcement learning method using multiple polarimetric features," *IEEE J. Sel. Topics Appl. Earth Observ. Remote Sens.*, vol. 12, no. 10, pp. 3719-3730, Oct. 2019.

[91] M. S. Seyfioğlu, A. M. Özbayoğlu, and S. Z. Gürbüz, "Deep convolutional autoencoder for radar-based classification of similar aided and unaided human activities," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 54, no. 4, pp. 1709-1723, Aug. 2018.

[92] S. Chen, H. Wang, F. Xu, and Y.-Q. Jin, "Target classification using the deep convolutional networks for SAR images," *IEEE Trans. Geosci. Remote Sens.*, vol. 54, no. 8, pp. 4806-4817, Aug. 2016.

[93] S. Lim, S. Lee, J. Jung, and S.-C. Kim, "Detection and localization of people inside vehicle using impulse radio ultra-wideband radar sensor," *IEEE Sensors J.*, vol. 20, no. 7, pp. 3892-3901, Apr. 2020.

[94] T. Hastie, R. Tibshirani, and J. Friedman,*The Elements of Statistical Learning: Data Mining, Inference, and Prediction*. New York, NY, USA: Springer, 2009.

[95] R. D. Reed and R. J. Marks, *Neural Smithing: Supervised Learning in Feedforward Artificial Neural Networks*. Cambridge, MA, USA: MIT Press, 1998.

[96] C. Cortes and V. Vapnik, "Support-vector networks," *Mach. Learn.*, vol. 25, no. 3, pp. 273-297, Sep. 1995.

[97] T. G. Dietterich, “An experimental comparison of three methods for constructing ensembles of decision trees: bagging, boosting, and randomization,” *Mach. Learn.*, vol. 40, no. 2, pp. 139–157, Aug. 2000.

[98] S. Lim, J. Jung, and S.-C. Kim, and S. Lee, “Deep neural network-based in-vehicle people localization using ultra-wideband radar,” *IEEE Access*, vol. 8, pp. 3892-3901, pp. 96606-96612, May 2020.

[99] H. Song and H.-C. Shin, “Single-channel FMCW-radar-based multi-passenger occupancy detection inside vehicle,” *Entropy*, vol. 23, no. 11, pp. 1472, Nov. 2021.

[100] A. Lazaro, M. Lazaro, R. Villarino, and D. Girbau, “Seat-occupancy detection system and breathing rate monitoring based on a low-cost mm-wave radar at 60 GHz,” *IEEE Access*, vol. 9, pp. 115403-115414, Aug. 2021.

[101] H. Abedi, S. Luo, V. Mazumdar, M. M. Y. R. Riad, and G. Shaker “AI-powered in-vehicle passenger monitoring using low-cost mm-wave radar,” *IEEE Access*, vol. 20, pp. 18998-19012, December 2021.

[102] N. Munte, A. Lazaro, R. Villarino, and D. Girbau, “Vehicle occupancy detector based on FMCW mm-wave radar at 77 GHz,” *IEEE Sen. J.*, vol. 22, no. 24, pp. 24504-24515, Dec. 2022.

[103] S. Lim, S. Lee, J.-H. Choi, J. Yoon, and S.-C. Kim, “Mutual interference suppression and signal restoration in automotive FMCW radar system,” *IEICE Trans. Commun.*, vol. E102-B, no. 6, pp. 1198-1208, Jun. 2019.

[104] Federal Communications Communications, Washington, D.C., United States, *FCC Seeks to Enable State-of-the-Art Radar Sensors in 60 GHz Band*, Accessed: December 20, 2022. [Online]. Available: https://www.fcc.gov/document/fcc-seeks-enable-state-art-radar-sensors-60-ghz-band-0.

[105] V. V. Viikari, T. Varpula, and M. Kantanen "Road-condition recognition using 24-GHz automotive radar," *IEEE Trans. Intell. Transp. Syst.*, vol. 10, no. 4, pp. 639-648, Dec. 2009.

[106] J. Hasch, E. Topak, R. Schnabel, T. Zwick, R. Weigel, and C. Waldschmidt, "Millimeter-wave technology for automotive radar sensors in the 77 GHz frequency band," *IEEE Trans. Microw. Theory Techn.*, vol. 60, no. 3, pp. 845-860, Mar. 2012.

[107] N. Guo, R. C. Qiu, S. S. Mo, and K. Takahashi, "60-GHz millimeter-wave radio: Principle, technology, and new results," *EURASIP J. Wireless Commun. Netw.*, vol. 2007, no. 1, pp. 1-8, 2007.

[108] S. Neemat, F. Uysal, O. Krasnov, and A. Yarovoy, "Reconfigurable range-Doppler processing and range resolution improvement for FMCW radar," *IEEE Sens. J.*, vol. 19, no. 20, pp. 9294-9303, Oct. 2019.

[109] S. Rao. *Introduction to mmWave Sensing: FMCW Radars*. Dallas, TX, USA: Texas Instruments, 2017. [Online]. Available: https://training.ti.com/intro-mmwave-sensing-fmcw-radars-module-5-angle-estimation.

[110] E. Hyun, Y.-S. Jin, and J.-H. Lee, "A pedestrian detection scheme using a coherent phase difference method based on 2D range-Doppler FMCW radar," *Sensors*, vol. 16, no. 1, p. 124, January 2016.

[111] A. De Groote, M. Wantier, G. Cheron, M. Estenne, and M. Paiva, "Chest wall motion during tidal breathing," *J. Appl. Physiol.*, vol. 83, pp. 1531–1537, November 1997.

[112] K. Han and S. Hong, "Detection and localization of multiple humans based on curve length of I/Q signal trajectory using MIMO FMCW radar," *IEEE Microw. Wireless Compon. Lett.*, vol. 31, no. 4, pp. 413-416, Apr. 2021.

[113] E. Debie and K. Shafi, “Implications of the curse of dimensionality for supervised learning classifier systems: theoretical and empirical analyses,” *Pattern Anal. Appl.*, vol. 22, no. 2, pp. 519-536, May 2019.

[114] C. Chourpiliadis and A. Bhardwaj, “Physiology, respiratory rate,” Tech. Rep., 2019.

초 록

최근 들어 자율주행에 대한 사람들들의 관심이 증가하면서 자율주행용 센서들과 신호 처리 기법에 대한 연구가 활발히 이루어지고 있다. 자율주행용 플랫폼으로는 자율주행 로봇, 드론, 차량 등이 개발되고 있으며, 자율주행용 센서로는 라이다, 카메라, 레이더, 초음파 센서 등이 사용되고 있다. 이들 중 레이더 센서는 빛이 없는 환경이나 악천후에도 안정적인 성능을 발휘하며 타 센서들에 비해 넓은 감지 거리를 가져 필수적인 센서로 여겨지고 있다. 레이더 센서는 로봇, 차량, 드론에 장착되어 주변 환경을 인지하거나 탐지된 타깃들의 상대속도, 거리, 각도 정보를 얻을 수 있다.

본 학위 논문에서는 레이더 센서 기반의 실내외 자율주행을 위한 향상된 신호 처리 기법들을 제안한다. 자율주행에서 동시적 위치 추정 및 지도작성 기술의 중요성이 높아지고 있다. 자율주행 기기가 동시적 위치 추정 및 지도작성을 위해 자신의 위치를 정확하게 추정하려면 움직이는 플랫폼의 회전 각도나 속도와 같은 자차의 움직임에 대한 정보가 필수적으로 고려되어야 한다. 본 학위 논문에서는 추가적인 기기 설치 없이 레이더 센서만을 이용하여 레이더가 장착된 플랫폼의 자차 움직임을 추정하는 연구를 제안한다. 또한, 고속도로나 도심 도로와 같은 복잡한 환경에서 주행 시 다양한 타깃이 존재하며 예측 불가능한 상황이 발생할 수 있다. 만약 주변 차량의 경로를 미리 예측할 수 있다면 사전에 잠재적인 위험을 예방할 수 있을 뿐만 아니라 차선을 변경하거나 자차의 속도를 조절하는데 활용할 수 있다. 따라서 본 학위 논문은 레이더 센서를 이용하여 주변 차량의 진행 방향을 추정하는 연구를 제안한다. 마지막으로, 차량 내 유아나 동물을 두고 내려 사망에 이르는 사건이 빈번하게 발생함에 따라 자율주행 차량에 실내 탑승객 감지 시스템이 의무적으로

장착되도록 요구되고 있다. 탑승객의 위치를 모니터링함으로써 승객의 안전을 보장할 수 있으며, 차량 내 에너지를 효율적으로 관리할 수 있다. 또한 레이더 센서를 이용함으로써 비접촉으로 다중 사람을 감지할 수 있으며, 사람이 움직이지 않더라도 위치를 정확하게 추정할 수 있다. 따라서 본 학위 논문은 차량 내 레이더 센서를 사용하여 탑승객의 수와 위치를 탐지하는 기법을 제안한다.

주요어: 임펄스 무선 광대역 레이더, 자율주행, 자차 움직임 추정, 주파수 변조 연속파 레이더, 차량 내 승객 탐지, 차량 진행 방향 추정
학번: 2017-25081