A STUDY OF SENSOR FUSION BASED VEHICLE DISTANCE ESTIMATION FOR SELF-DRIVING CARS

BY

CHINNAWAT CHINNAPUN

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF MASTER OF ENGINEERING (INFORMATION AND COMMUNICATION TECHNOLOGY FOR EMBEDDED SYSTEMS)

SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY
THAMMASAT UNIVERSITY
ACADEMIC YEAR 2021

THAMMASAT UNIVERSITY
SIRINDHORN INTERNATIONAL INSTITUTE OF TECHNOLOGY

THESIS

BY

CHINNAWAT CHINNAPUN

ENTITLED

A STUDY OF SENSOR FUSION BASED VEHICLE DISTANCE ESTIMATION FOR SELF-DRIVING CARS

was approved as partial fulfillment of the requirements for the degree of Master of Engineering (Information and Communication Technology for Embedded Systems)

on April 27, 2022

Chairperson	(Jatuwat Rajruangrabin, Ph.D.)
Member and Advisor	Toshiaki Kondo (Associate Professor Toshiaki Kondo, Ph.D.)
Member	S. Kittip (Assistant Professor Somsak Kittipiyakul, Ph.D.)
Member	Yuko Hara-Azumi (Associate Professor Yuko Hara, Ph.D.)
Director	(Professor Pruettha Nanakorn, D.Eng.)

Thesis Title	A STUDY OF SENSOR FUSION BASED VEHICLE DISTANCE ESTIMATION FOR SELF-DRIVING CARS
Author	Chinnawat Chinnapun
Degree	Master of Engineering (Information and Communication Technology for Embedded Systems)
Faculty/University	Sirindhorn International Institute of Technology/ Thammasat University
Thesis Advisor	Associate Professor Toshiaki Kondo, Ph.D.
Academic Years	2021

ABSTRACT

The self-driving car has become attractive and modern. The self-driving car consists of several systems such as navigation, localization, avoidance, and more. LiDAR, Radar, GPS, Inertial Measurement Unit (IMU), Ultrasonic, camera were sensors, which those sensors were used in the self-driving car. The method to integrate those sensors work together is Sensor Fusion. This method uses diverse types of sensors that work similarly to one sensor. This research proposes a method that uses a combination of sensors for distance estimation. Multiple Lidars and a camera contribute to the improvement of data processing for the Kalman filter. The Object detection method will locate and classify the object in the research. This research will help develop driving systems such as emergency braking, velocity calculation, and collision warning system.

Keywords: Sensor fusion, LiDAR, Object detection, Kalman filter, Vision, Distance estimation, Self-driving car

ACKNOWLEDGEMENTS

I would like to thank of gratitude to my advisor, Assoc. Prof. Dr. Toshiaki Kondo and my NSTDA adviser, Dr. Jartuwat Rajruangrabin for the comment, suggestion, great support, and opportunity.

Besides my advisor and NSTDA adviser, I would like to thank the rest of my thesis committee: Assoc. Prof. Dr. Yuko Hara and Asst. Prof. Dr. Somsak Kittipiyakul. They always provide me with suggestions for interesting ideas, which was the important part to help complete the project.

I would like to thank my friends and my family who encourage me during a hard time.

Finally, thank the staff and financially support by Thailand Advanced Institute of Science and Technology (TAIST), National Science and Technology Development Agency (NSTDA), Tokyo Institute of Technology, and Thammasat University (TU).

Chinnawat Chinnapun

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1
INTRODUCTION

1.1 Importance of research

In this decade, Self-driving cars are something that will replace the way of human-driven cars. Imagine sending a command to the vehicle and it can drive to the destination by itself, which autonomous driving technologies enable self-driving cars to make it. Figure 1.1 is shown the example of self-driving car. It’s not only sole technology but the integration of several technologies. There are many systems in self-driving cars such as drive systems, perception, decisions, and more. Perception is one of the important systems, which is used by self-driving cars to measure distance and detect any objects. Combing LiDAR sensors and camera to measure distance, classification object was combined to locate obstacle, car, and pedestrians. it was used to make real-time decisions to avoid collisions.

Figure 1.1 Tesla car using artificial intelligence.
(https://bernardmarr.com/how-tesla-is-using-artificial-intelligence-to-create-the-autonomous-cars-of-the-future)

1.2 Motivation

In the changing era, studying the self-driving car is challenging. The self-driving car can operate and sense the surrounding environment without humans. The passenger is not required to take control of the self-driving car. Perception is necessary to analyze the environment. Increasing the performance of sense is interesting. Using several sensors can break the limitations of sensors. Sensor fusion using LiDAR sensor and camera was very attractive and widely used. LiDAR and camera are used for different objectives. The fusion of data from LiDAR and camera could result in more accurate vehicle distance estimation which is very important for self-driving cars than using data from a sole sensor.

1.3 Objective

The purpose of this study are as follows

- To design a method for distance estimation from the self-driving car to objects.
- Propose method Sensor fusion of LiDAR sensor and camera.
- To increase the performance of distance estimation.

1.4 Thesis outline

The structure of this thesis starts with Chapter 2 describes the literature review of self-driving cars, combining sensors, LiDAR technology, object detection, and related work about sensor fusion. Chapter 3 presents the proposed method, designing Kalman filter, and experimental design. Chapter 4 consists of the experimental results in each factor, the performance, and the accuracy of measuring the distance from the vehicle to objects in several scenarios. Chapter 5 displays a conclusion of this research and suggestions for future works.

CHAPTER 2
REVIEW OF LITERATURE

This chapter provides background knowledge about technology in Self-driving cars, LiDAR technology, data filtering, combining sensors, LGSVL simulation, YOLO object detection, and Kalman filter. For purpose methods were in chapter 3 and chapter 4, which those chapters explain developing methods.

2.1 Technology in Self-driving cars

The driverless car is the next generation of cars, the main of this is autonomous. To control the cars, Perception is necessary to analyze the environment. So, Sensors are one of mainly of systems in self-driving cars. Figure 2.1 shows sensors were used for the autonomous car. There are lots of sensors such as LiDAR, Radar, Camera, Ultrasonic, Odometric sensor [1].

To combine all the sensors, Sensor fusion is the method of combining data from different types of sensors to increase the quality of data. Each sensor had limitations sometimes. For example, GPS cannot use when used in tunnels. Using sensor fusion can help sensors when some sensor is broken. Sensor fusion particularly plays a vital role when worst weather like rain, fog, and snow as different sensors work differently in the conditions [2,3].

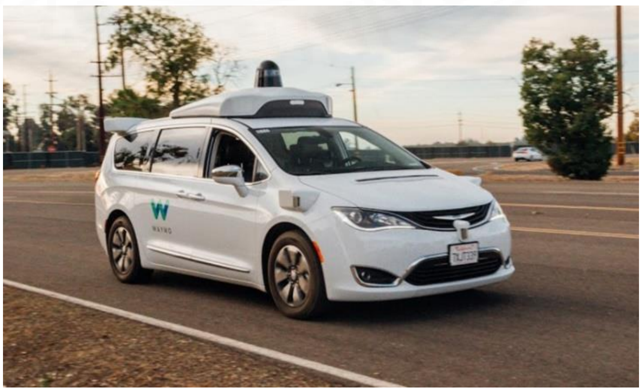


Figure 2.1 Autonomous car with sensors.
(https://europe.autonews.com/article/20180508/ANE/180509829/in-self-driving-car-race-waymo-leads-traditional-automakers)

2.2 LiDAR technology

LiDAR or Light Detection and Ranging is a remote sensing sensor [4,5]. Sometimes, it is called 3D scanning or laser scanning. It is used to measure the surface of the object. Figure 2.2 display the Velodyne Puck, which was LiDAR that use in this work. LiDAR used the laser to measure objects and use returning time to measure distance. LiDAR used principles like a SONAR sensor. It was adapted to use with self-driving cars, navigation robots and robot arms. LiDAR generate the point cloud that was the point in X-axis, Y-axis and Z-axis. The example of point cloud was demonstrated in Figure 2.3.

Figure 2.2 LiDAR sensors.
(https://autonomoustuff.com/products/velodyne-puck-hi-res)

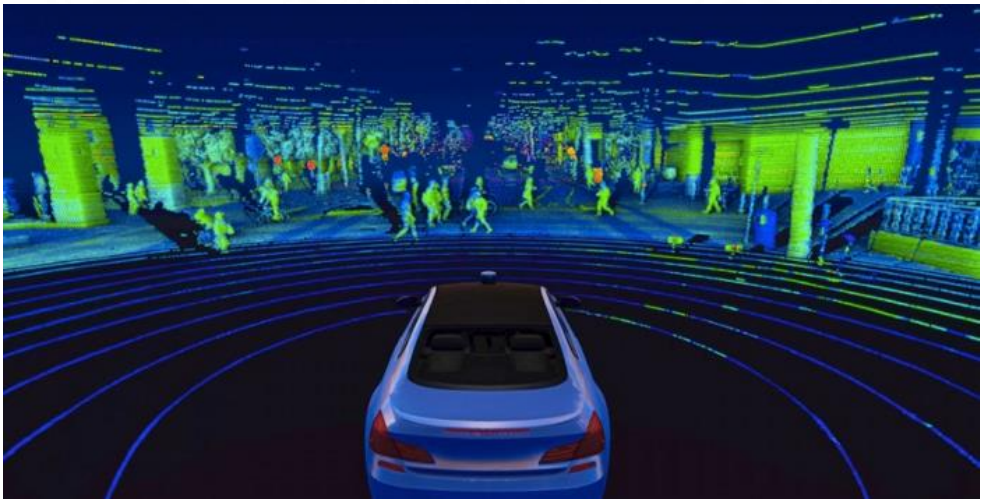


Figure 2.3 Example of Point Cloud.
(https://becominghuman.ai/three-reasons-for-the-growing-demand-of-3d-point-cloud-data-in-automatic-driving-in-2021-57d5ffae294)

2.3 Data filtering

Data from sensors may have noise for many reasons such as light, magnetic, vibrations. The problems from sensors can improve the quality of data by using the software. The filter can use Median filter and Moving average filter to software implementation. Those algorithms can apply to different types of sensors. Those algorithms depend on different factors of the sensor [6].

2.3.1 Moving average filter

The Moving average filter is a common filter in digital signal processing. This algorithm is optimal for common tasks to reduce random noise. The Moving average filter considers data in the entire window and uses an average of the data value in the window. The exam of Moving average filter is shown in Figure 2.4.

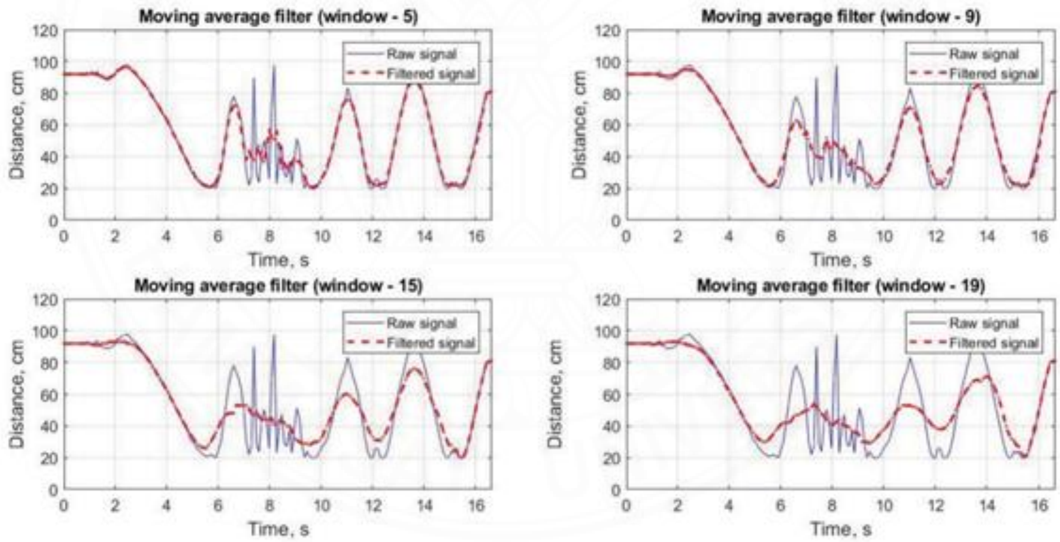


Figure 2.4 Input data and output data with the Moving average filter.

2.3.2 Median filter

The Median filter is one of the most common digital filters. It is often used to remove noise from signals or images. This algorithm can adjust the noise, which has a large amplitude and short duration. A significant delay in monotonic form does not affect the algorithm [7]. The exam of Moving average filter is shown in Figure 2.5.

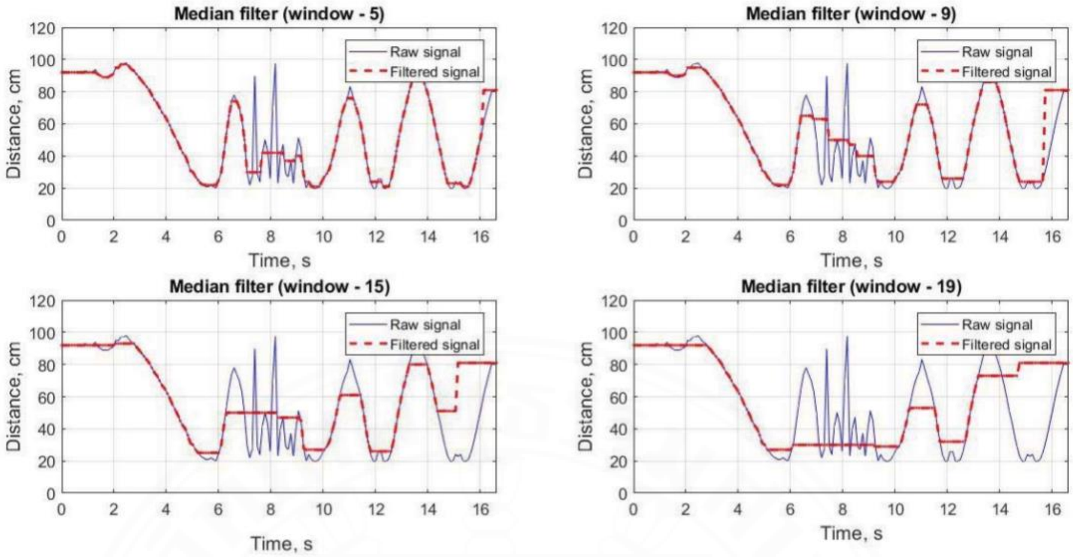


Figure 2.5 Input data and output data with the Median filter.

2.4 Measure distance of an object by combining LiDAR sensor and camera

Sensor fusion using different types of sensors had become very extensively and attractive adopted ranging from terrestrial to airborne. the integration of those attributes with an efficient fusion approach greatly benefits the reliable and consistent perception of the environment. For example, LiDAR and camera are used for different objectives. The fusion of LiDAR and camera data could result in more accurate vehicle distance estimation, which is very important for a self-driving car than using data from a single sensor. The method of sensor fusion-based approach for a self-driving car combining LiDAR and camera. The output is shown in Figure 2.6. A Kalman filter is an algorithm, which this algorithm is helpful with multiple sensor inputs. It can apply to different kinematic equations depend on the sensors type [8].

Figure 2.6 Sensor fusion of LiDAR and camera.

2.5 LGSVL Simulation

For testing algorithms or theory, the simulation was a good choice. It helps to develop and test systems fast in the virtual environment. Figure 2.7 shows LGSVL simulator is a simulator that can develop and test self-driving cars. It enables the simulation of various scenarios and different sensors. It can adjust the type of sensors and limitations of sensors. And it can integrate lots of sensors in one vehicle [9].

Figure 2.7 LGSVL simulator.
(https://www.svlsimulator.com/docs/archive/2020.06/sensor-visualizers)

2.6 YOLO Object detection

To classify objects in the image. In general, there are many different algorithms. But the best algorithm is YOLO object detection. It was an algorithm that use a neural network to provide real-time object detection. YOLO uses deep learning that can classify each object from the images. Figure 2.8 shows the result of YOLO object detection, it can classify objects such as bicycle, car, truck, and traffic light [10-12].

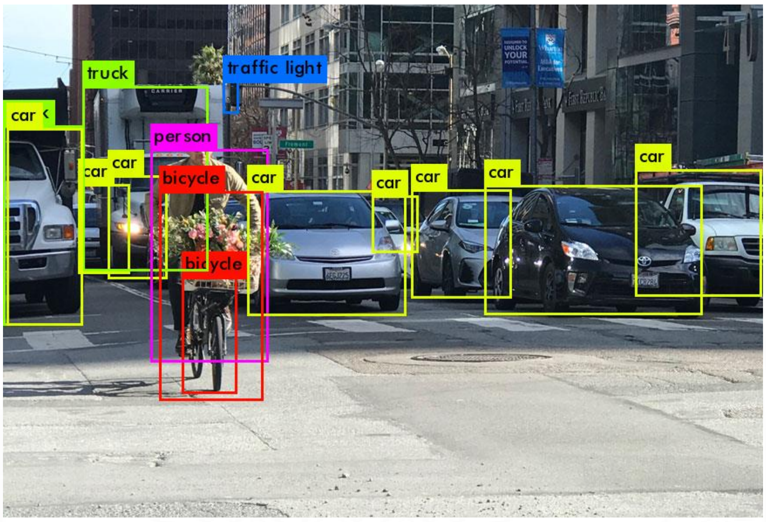


Figure 2.8 Result from YOLO.
(https://ichi.pro/th/kar-trwc-cab-watthu-baeb-rei-yl-thim-dwy-yolo-yolov2-laea-txn-ni-yolov3-44743782178952)

2.7 Kalman Filter

Most modern systems are equipped with several sensors that provide an estimation of hidden variables based on a series of measurements. Kalman Algorithm is the method to estimates some unknown variables given the measurements observed over time. Kalman filters were useful in various applications. Kalman filters were simple forms and need small computational power.

This algorithm was created by Rudolf E. Kalman in 1960. By using Kalman Algorithm in research and applications in various fields, specifically in the field of maritime. For now, this algorithm is used in control systems, navigation systems, localization systems, and more [13-15].

There are 2 steps was using on Kalman filter. 1. Prediction State 2. Update State

2.7.1 Prediction State

For prediction state, the Kalman filter uses previous knowledge of data from sensors and dynamic model to predict the uncertainty error variance in prediction according to the various process noise present in the system. The prediction state is calculated by Equation (2.1) and Equation (2.2).

$$X_k^- = AX_{k-1} + Bu_k \tag{2.1}$$

$$P_k^- = AP_{k-1}A^T + Q \tag{2.2}$$

X_{k-1} is current state
X_k^- is future state
A is State transition matrix
B is control matrix
u_k is control variable
P_k^- is state variance matrix
Q is process noise covariance matrix

2.7.2 Update State

Update state uses the data from prediction state and data from sensor reading to estimate status. The Update state is calculated by Equation (2.3), Equation (2.4), and Equation (2.5).

$$K_k = \frac{P_{k-1}^- H^T}{HP_{k-1}^- H^T + R} \tag{2.3}$$

$$\hat{X}_k = X_{k-1}^- + K_k(y_k - HX_{k-1}^-) \tag{2.4}$$

$$P_k = (I - K_k H)P_{k-1}^- \tag{2.5}$$

K_k is Kalman gain, it is used to weigh how much the new measurement data to use to update the new estimate. The number of Kalman gain is between 0 to 1. If Kalman gain is large. The error in the measurement is less.
$\hat{X}_k$ is estimation matrix
y_k is matrix containing measurement data.
I is identity matrix.
H is measurement matrix
R is measurement variance matrix

CHAPTER 3
DESIGN AND METHODOLOGY

This chapter describes the experimental method. That consists of an overview of the system, device setup, sensor calibration, object detection, distance estimation, data cleaning, and Kalman filter.

3.1 Overview of the system.

Figure 3.1 represents system architecture. Starting from the left to the right. The LiDAR and camera are sensors. Those sensors are different types of sensors, which they used for obtaining the environment. The LiDAR obtains the point cloud and the camera obtains the image. The sensor calibration method calculates a matrix that uses for combining different types of sensors to work as one sensor. The point cloud will project to the image in the sensor fusion method. Object detection uses for classifying objects in the images. The data cleaning method eliminates needless data. Kalman filter is used to predict future status from previous data. A more detailed explanation will be below.

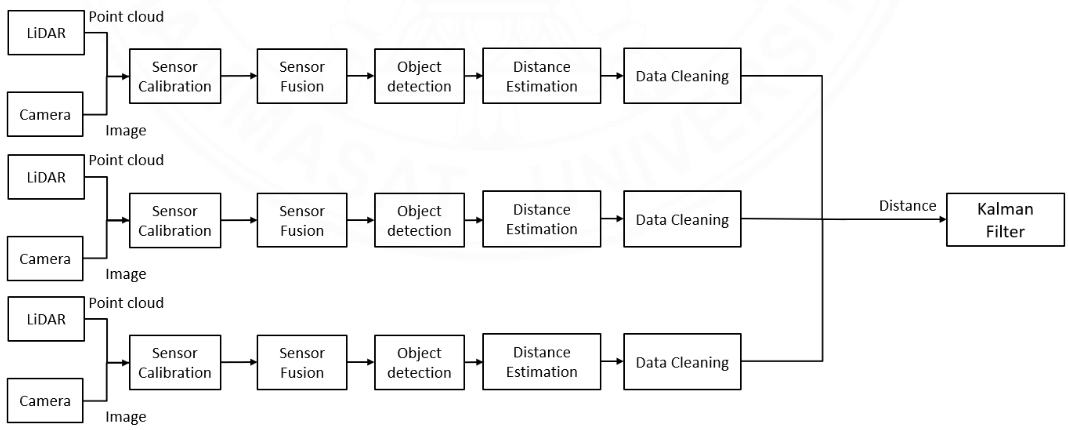


Figure 3.1 System architecture.

3.2 Device setup

Figure 3.2 shows the sensor's location on the self-driving car that uses for collecting data in this work. There are three LiDAR sensors and one camera. Two LiDAR on the front left and the front right. The last LiDAR is in the middle of the roof, and the camera is on the roof. The camera is an optical sensor, which is used to capture the images. The camera is used to understand the environment with artificial intelligence.

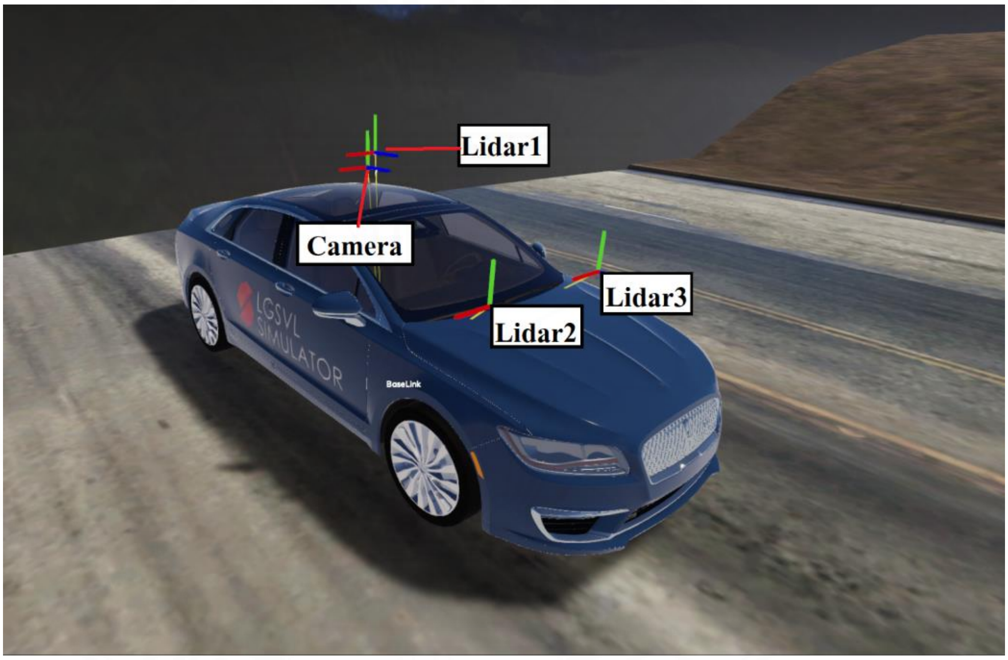


Figure 3.2 Location of sensors.

3.3 Sensor calibration

LiDAR and camera have different coordinate systems. Sensor calibration is used to convert sensor data to the same coordinate system. The transformation matrix of the coordinate system uses rotation and translation matrix to calculate. Equation (3.1) and Equation (3.2) are used for calculating the transformation matrix.

$$M_{projective} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_x \\ r_{21} & r_{22} & r_{23} & t_y \\ r_{31} & r_{32} & r_{33} & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.1}$$

$$\begin{bmatrix} U \\ V \\ 1 \end{bmatrix} = M_{projective} \begin{bmatrix} X \\ Y \\ Z \\ 1 \end{bmatrix} \tag{3.2}$$

$M_{projective}$ is the projective matrix, and this matrix helps project point cloud to the image. X , Y and Z are point cloud in each axis on the point cloud coordinate system. r and t are rotation and translation vectors of installation location LiDAR and camera.

U and V are x-axis and y-axis distance on the image coordinate system. Each LiDAR has a different installation location. Also, they have different transformation matrices.

By using the transformation matrix with the point cloud matrix, we can plot the point cloud to the image. And data will be the same coordinate system. Figure 3.3 shows the green points are point cloud which it was detected by LiDAR. In this method, the system knows the depth of the point by reading the data of each point.

Figure 3.3 Point cloud was projected to image.

3.4 Object detection

For selection point cloud from the images. YOLO object detection was used for locating object in the image. YOLOv3-608 was model to use in this work, which this model has mAP equal 57.9 and 20 fps. This algorithm can locate interesting object that you want. Figure 3.4 demonstrate classification object from image, there are car, bicycle, person, and traffic light. The bounce box and type of objects were generated from this algorithm.

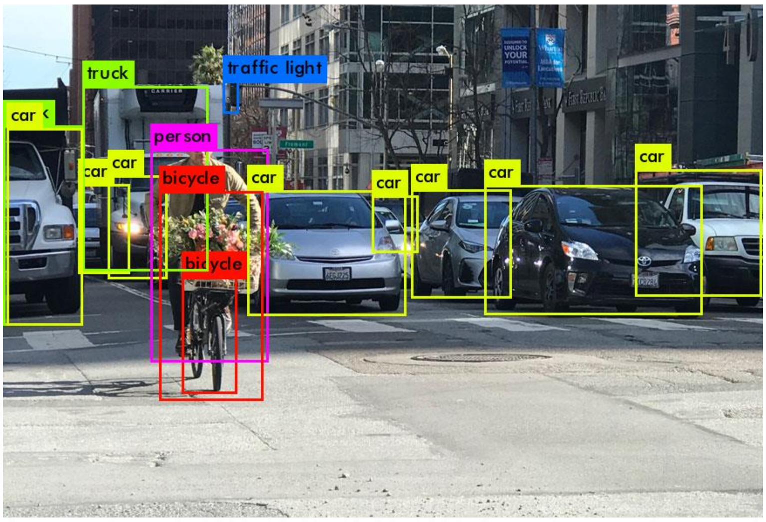


Figure 3.4 Example from YOLO.

3.5 Distance estimation

The objective of sensor fusion is to measure the distance from the sensor to the object. The distance measured from the point cloud is shown in Figure 3.5. The red point in the bounding box is the point cloud that was projected to the image. The interest object is in the bounding box, and the bounding box is created by object detection. The data are unusable because of many point cloud from the environment and objects in the bounding box. It is hard to select an objective point cloud. To select point cloud, the system uses Median filter for selecting the point cloud in the bounding box to get the distance from the sensor to the object.

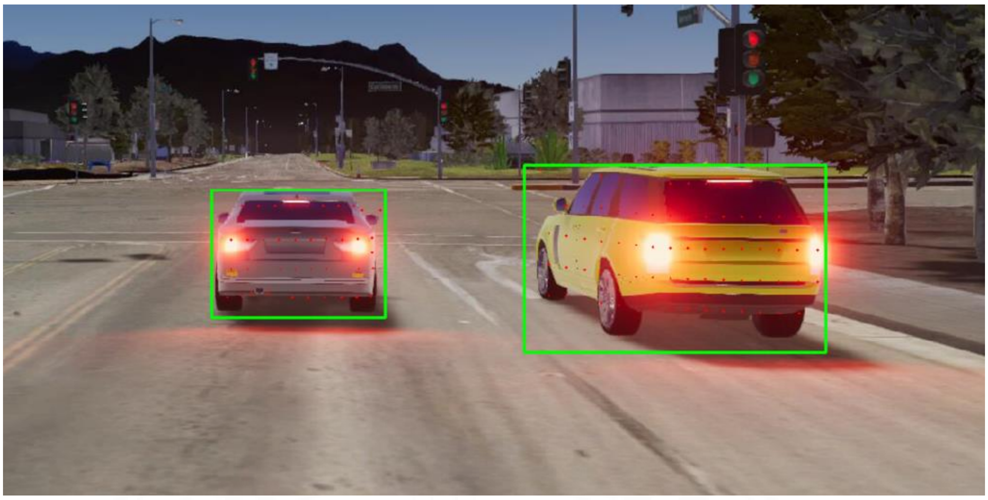


Figure 3.5 Selection Point cloud from the image.

3.6 Data cleaning

Data cleaning, also known as a data preprocessing, is a method to edit or eliminate needless data from the database. This method is created because inconsistent data affect the outcome. The insufficient data may make data recording errors. The data management makes the result better than the original data. In this research, Data cleaning method uses the Median filter and the moving average filter. The amount of data in the data cleaning method is 21. The amount of data affects the delay of output and data performance.

3.6.1 The Median filter

The Median filter is a non-linear digital filter used in image processing and signal processing. The advantage of the Median filter is used to remove impulse noise and to smooth the signal. In robotic systems, the signals from sensors always have noise from the surroundings and sensor dropout.

3.6.2 The Moving average filter

The Moving average filter is a linear filter unlike the Median filter and it uses average for entire windows. This filter uses the process of computer and time less than the Median filter because it doesn't need time to sort.

3.7 Kalman filter

Designing Kalman filter depending upon the different Kinematic equations that want to use and different types of sensor reading which depend on integrating into the system. The kinematic equation is shown in Equation (3.3).

$$\begin{aligned} x &= x + \dot{x}\Delta t \\ y &= y + \dot{y}\Delta t \\ \dot{x} &= \dot{x} \\ \dot{y} &= \dot{y} \end{aligned} \tag{3.3}$$

The changing time is defined as Δt, which this variable can track distance and velocity with the equation. The variables $x, y, \dot{x}, \dot{y}$ represent to distance from the sensor to objects and the velocity of the object in the x-axis and the y-axis respectively.

3.7.1 Predict state

The state model is shown in Equation (3.4) where $\hat{X}_k$ shows the current state and $\hat{X}_{k+1}$ predicts the next state. The transition functions are $x + \dot{x}\mathit{\Delta} t$ and $y + \dot{y}\mathit{\Delta} t$ in the model that estimates the distance of the object to the sensor on x-axis and y-axis.

$$\hat{X}_k = \begin{bmatrix} x \\ y \\ \dot{x} \\ \dot{y} \end{bmatrix}, \hat{X}_{k+1} = \begin{bmatrix} x + x\dot{\mathit{\Delta}}t \\ y + \dot{y}\mathit{\Delta} t \\ \dot{x} \\ \dot{y} \end{bmatrix} \tag{3.4}$$

Equation (3.5) represents A, which A was matrix, which this matrix is used to calculate the previous state to the current state. And Matrix B is null because the vehicle does not have input to the system. Δt is sampling time for each frame.

$$A = \begin{bmatrix} 1 & 0 & \Delta t & 0 \\ 0 & 1 & 0 & \Delta t \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3.5}$$

Q represents process noise covariance matrix, that matrix is calculated with new prediction covariance in the estimation. The process noise covariance matrix is calculated in Equation (3.6). The variable σ_α^2 define as acceleration.

$$Q = \begin{bmatrix} \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} & 0 \\ 0 & \frac{\Delta t^4}{4} & 0 & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & 0 & \Delta t^2 & 0 \\ 0 & \frac{\Delta t^3}{2} & 0 & \Delta t^2 \end{bmatrix} \alpha_\alpha^2 \tag{3.6}$$

3.7.2 Update state

The update state will calculate when receiving a sensor reading for the distance of the tracking objects. The update state use timestamp difference between two reading times and input data from measurement.

R is a covariance measurement matrix, which σ_i, σ_j , and σ_k are variances of each sensors. The matrix size of R is 6x6 because of there are three input data from sensors was used in self-driving car. The covariance measurement matrix is shown in Equation (3.7).

$$R = \begin{bmatrix} \alpha_i & 0 & 0 & 0 & 0 & 0 \\ 0 & \alpha_i & 0 & 0 & 0 & 0 \\ 0 & 0 & \alpha_j & 0 & 0 & 0 \\ 0 & 0 & 0 & \alpha_j & 0 & 0 \\ 0 & 0 & 0 & 0 & \alpha_k & 0 \\ 0 & 0 & 0 & 0 & 0 & \alpha_k \end{bmatrix} \tag{3.7}$$

H is the extraction matrix. The matrix size of H is 6x4, which is used to converts $\hat{X}_k$ matrix to equal y_k matrix size. The extraction matrix is shown in Equation (3.8).

$$H = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \end{bmatrix} \tag{3.8}$$

y_k is the matrix of measurement data, which the size of this matrix is 6x1. The matrix of measurement data is shown in Equation (3.9). It contains variables x_1, y_1, x_2, y_2, x_3, y_3, which x_1 is the distance from the object to LiDAR1 on the x-axis and y_1 is the distance from the object to LiDAR1 on the y-axis. x_2 is the distance from the object to LiDAR2 on the x-axis and y_2 is the distance from the object to LiDAR2 on the y-axis. x_3 is the distance from the object to LiDAR3 on the x-axis and y_3 is the distance from the object to LiDAR3 on the y-axis. There are three LiDAR sensors are used in the self-driving car.

$$y_k = \begin{bmatrix} x_1 \\ y_1 \\ x_2 \\ y_2 \\ x_3 \\ y_3 \end{bmatrix} \tag{3.9}$$

CHAPTER 4
EXPERIMENTAL RESULT AND DISCUSSION

This character explains the results of the experiments according to the steps in character 3. The experiments separate into Static and dynamic testing.

4.1 Static Testing

For static testing, the self-driving car does not move while the other environment will move. There are four experiments for testing, non-obstacle measurement, obstacle measurement, and Pedestrian.

4.1.1 Experiment 1: Detection Car when there are non-obstacles

In the first experiment, the self-driving car was stopping at the intersection. Two cars stop in front of a self-driving car. Then those cars move, the yellow car will turn right, and the white car turns left. The simulated image was shown in Figure 4.1.

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

Figure 4.1 Situation of experiment 1.

(a) Scene 1, (b) Scene 2, (c) Scene 3, (d) Scene 4.

The signal in Figure 4.2 represents the input signal and output signal. The Y-axis is the distance value in meters and the X-axis is frames. The upper graphs are input from three LiDAR sensors representing different colors. The lower graphs are output from each method, consisting of Kalman filter without data cleaning, Kalman filter with moving average filter, and Kalman filter with median filter and ground truth. In the case of a white car, the input signal is pretty good. There is no impulse noise. The result comparison of Kalman filter without data cleaning, Kalman filter with Moving average filter and Kalman filter with Median filter are like ground truth.

In the case of a yellow car, the input signal and the output signal are shown in Figure 4.3. This case measures the distance to a yellow car. The data from each sensor was a strong signal. There were few noises on frame 45 because object detection cannot detect on this frame, so the input value equal to zero. However, the output results from each method are pretty good because the input signals are lost signal in a short time. The output from all method approach to the ground truth.

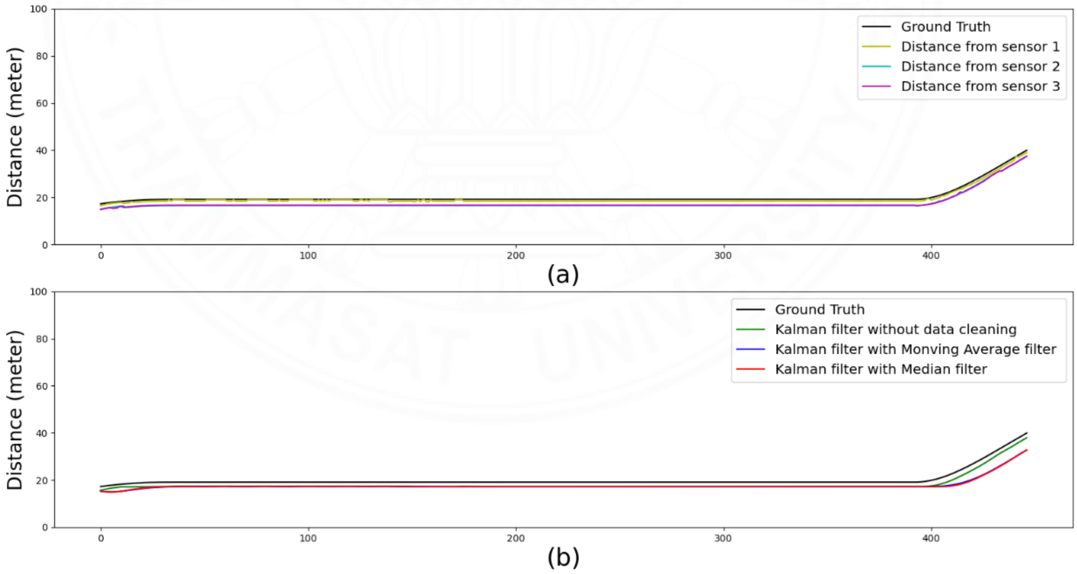


Figure 4.2 Graph of experiment 1 reports measuring the distance to a white car.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

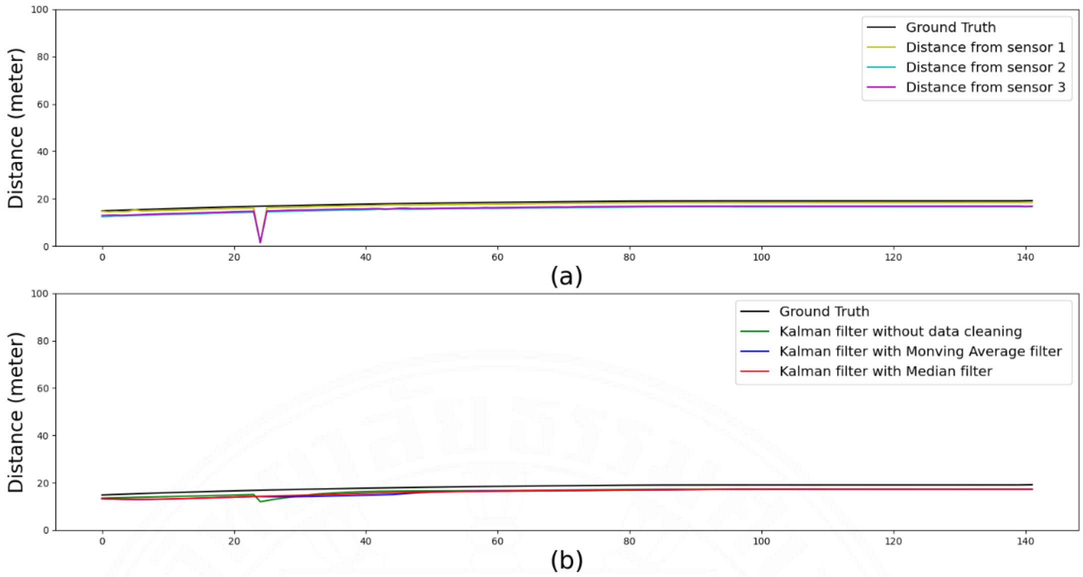


Figure 4.3 Graph of experiment 1 reports measuring the distance to a yellow car.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

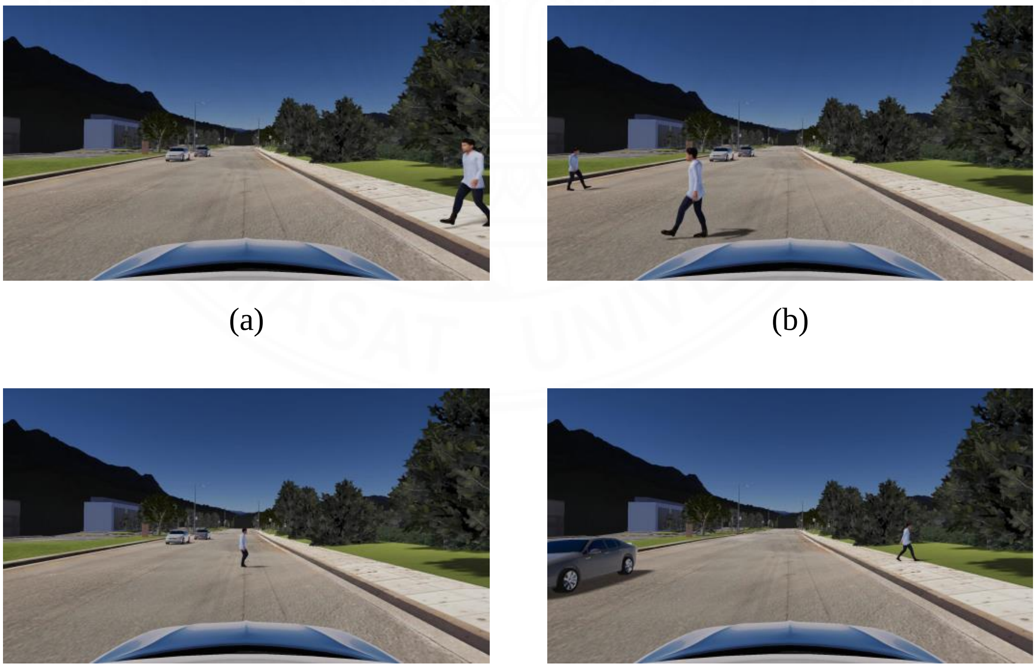
The experimental result is reported in Table 4.1, in which the first column is object. The object 1 is a white car, and the object 2 is a yellow car. Column 2 to 4 are the result in terms of Root Mean Square Error (RMSE). There are Kalman filter without data cleaning, Kalman filter with Median filter, and Kalman filter with Moving average filter, respectively. The unit of RMSE is in meters. The result of object 1 by using the Kalman filter without data cleaning is 2.06 meters, the result of Kalman filter with median filter is 2.84 meters, and the result of Kalman filter with moving average filter is 2.81 meters. The result of object 2 by using the Kalman filter without data cleaning is 1.99 meters, the result of Kalman filter with median filter is 2.15 meters, and the result of Kalman filter with moving average filter is 2.26 meters.

Table. 4.1 Evaluation comparison of experiment 1 between Kalman filter, Kalman filter with Median Filter, and Kalman filter with Moving Average Filter.

Object	Root Mean Square Error (Meters)		
	Kalman Filter Without Data Cleaning	Kalman Filter With Median Filter	Kalman Filter With Moving Average Filter
1	2.06	2.84	2.81
2	1.99	2.15	2.26

4.1.2 Experiment 2: Detection Car when there are obstacles

In the experiment 2, the self-driving car was stopping, while two cars were driving in the opposite way. The obstacles have been added by adding two pedestrians walking across the road in front of the self-driving car. The experimental is shown in Figure 4.4.

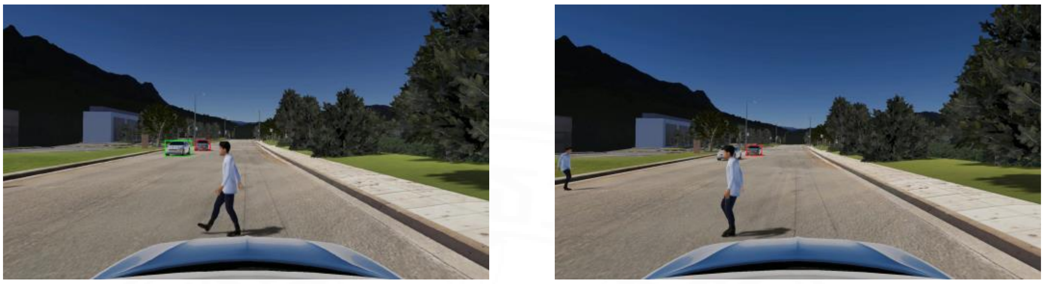


(a) (b) (c) (d)

Figure 4.4 Situation of experiment 2.

(a) Scene 1, (b) Scene 2, (c) Scene 3, (d) Scene 4.

Figure 4.5 demonstrates a scenario that there is a pedestrian crossing the road. The object detection was disturbed by pedestrians, which cannot detect cars. Measuring can't use at this time, then the measuring distance is none.



(a) (b)

Figure 4.5 The pedestrians are walking across the road.
(a) Pedestrian crossing the road. (b) Pedestrian blocking vision.

The input signal and the output signal of a black car are shown in Figure 4.6. It was the measuring distance to a black car. The object detection was disturbed by pedestrian, which created noisy signal that at the frame 90 to 105 and 240 to 252. The object detection cannot locate the car. So, the distance data is none. The Bottom graph is the output signals of this case that were compared on each method. The output signal of Kalman filter without data cleaning was sensitive to noise. The method of Kalman filter with Moving average filter uses the average of 21 data input, which the output data from this method changes a bit but over a long duration. The method of Kalman filter with Median filter uses input 21 windows. The output of the Kalman filter with the Median filter was approximate to ground truth. This Kalman filter with the Median filter can adjust the spike noise in a short duration.

Figure 4.7 reports the input signal and the output signal of a white car. For input signal, the object detection was disturbed by pedestrians likewise a black car. It created noisy signals during frames 105 to 115 and 255 to 260. Object detection cannot use at this time. The bottom graph is the output signal in the case of a white car. The output of the Kalman filter without data cleaning was sensitive to noise. The output signal of the Kalman filter with Moving sensitive a bit from noise. The output of the Kalman filter with Median filter does not affect from the noisy signal.

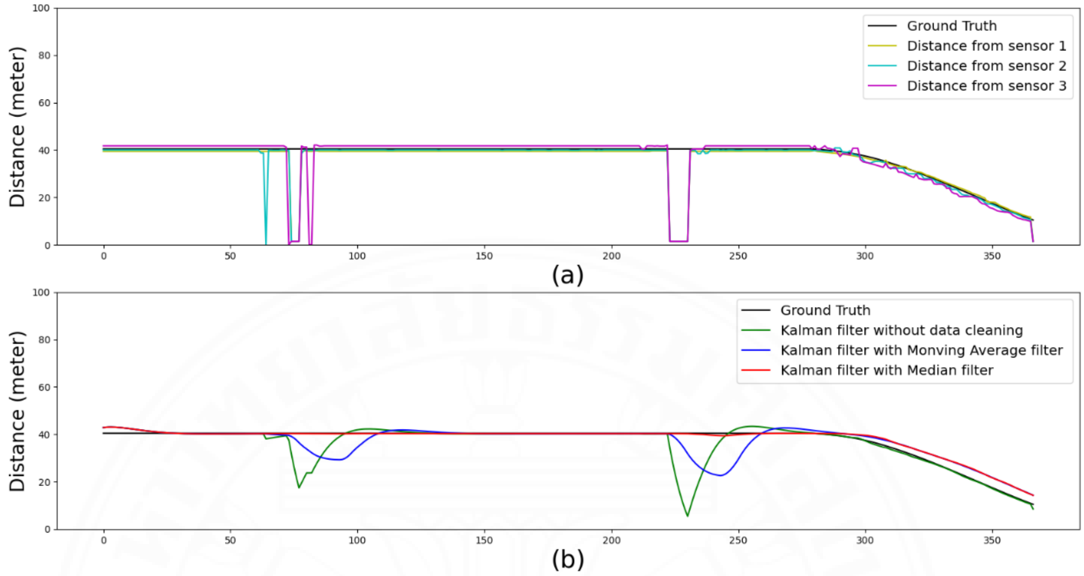


Figure 4.6 Graph of experiment 2 reports measuring the distance to a black car.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

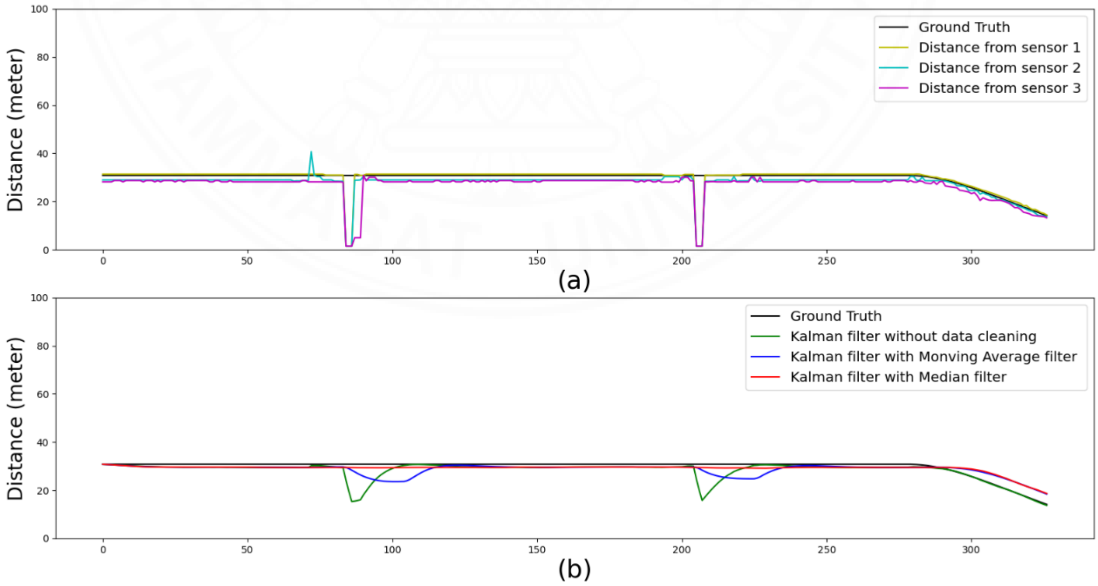


Figure 4.7 Graph of experiment 2 reports measuring the distance to a white car.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

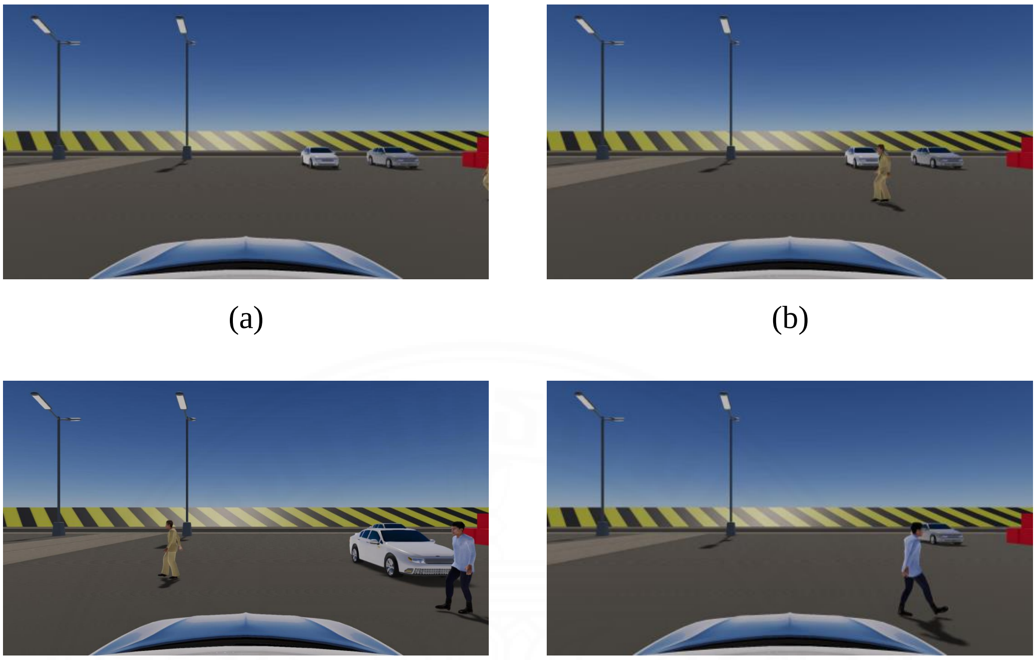
The experimental result is reported in Table 4.2. object 1 is a black car, and object 2 is a white car. The result of object 1 by using the Kalman filter without data cleaning is 6.32 meters, the result of Kalman filter with median filter is 3.19 meters, and the result of Kalman filter with moving average filter is 5.36 meters. The result of object 2 by using the Kalman filter without data cleaning is 3.41 meters, the result of Kalman filter with median filter is 2.33 meters, and the result of Kalman filter with moving average filter is 3.06 meters.

Table. 4.2 Evaluation comparison of experiment 2 between Kalman filter, Kalman filter with Median Filter, and Kalman filter with Moving Average Filter.

Object	Root Mean Square Error (Meter)		
	Kalman Filter Without Data Cleaning	Kalman Filter With Median Filter	Kalman Filter With Moving Average Filter
1	6.32	3.19	5.36
2	3.41	2.33	3.06

4.1.3 Experiment 3: Detection car when there are obstacles

In experiment 3, the self-driving car was stopping. A white car on the left drove out from parking in the opposite direction and a gray car on the right was still parking. The obstacles have been added. The pedestrians walk in front of the self-driving car from the right to the left. The simulated image are shown in Figure 4.8.



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

Figure 4.8 Situation of experiment 3.

(a) Scene 1, (b) Scene 2, (c) Scene 3, (d) Scene 4.

In experiment 3, two pedestrians are walking in the parking zone. The result of a white car is shown in Figure 4.11. During frames 90 to 110, the pedestrian was blocking the car in Figure 4.9. At this moment, Object detection can detect the car. Because the pedestrian was far from the self-driving car. The input data from each LiDAR sensor were quite stable. The output signal from the Kalman filter without data cleaning has little impact. The output signals of the Kalman filter with Median filter and the Kalman filter with Moving Average filter can adjust this situation.

The result signal of gray car is shown in Figure 4.12. A white car was driving forward, and it blocked the vision of the camera. This situation was shown in Figure 4.10. There are two cases in frames 170 to 210. When object detection can detect the car. It can detect only the roof which the point cloud was not from the car. It measures point cloud from the wall, which made a spike noise and measured distance to 60 meters. When object detection cannot be detected, the data may be lost. In the graph at frames 170 to 210, there are lots of invalid data. The input data was 60 meters and 0 meters. The output from the Kalman filter without data cleaning is sensitive to this situation. The output signal during this period is not available. The output from the Kalman filter with Median filter and the Kalman filter with Moving Average filter cannot adjust this situation. The output signals are not available, because the input data is not useable for a long time. Then data cleaning doesn't suitable for this situation.

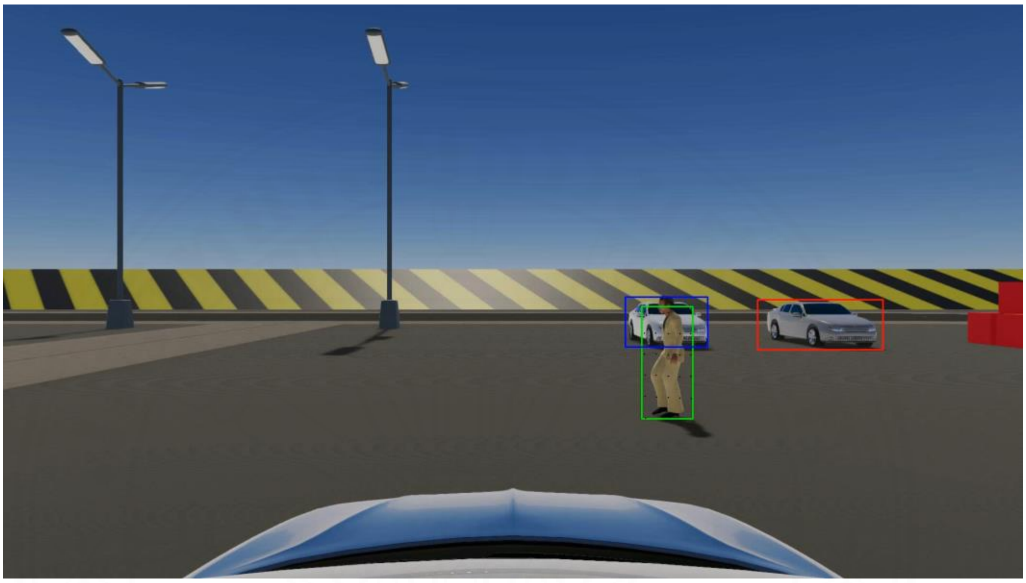


Figure 4.9 The pedestrian is walking when blocks a white car.

Figure 4.10 The white car drives when the pedestrians are walking.

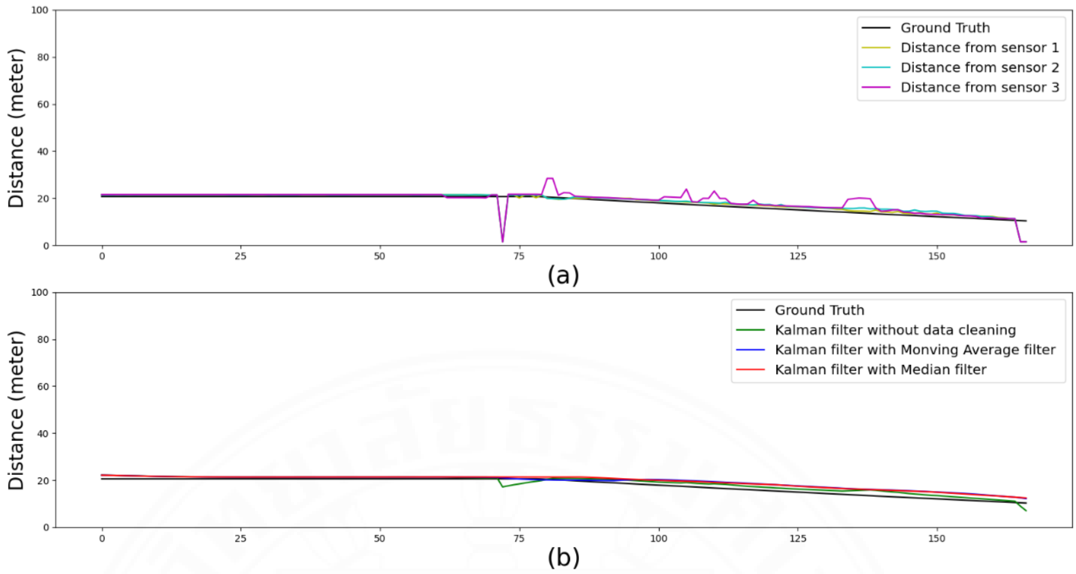


Figure 4.11 Graph of experiment 3 reports measuring the distance to a white car.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

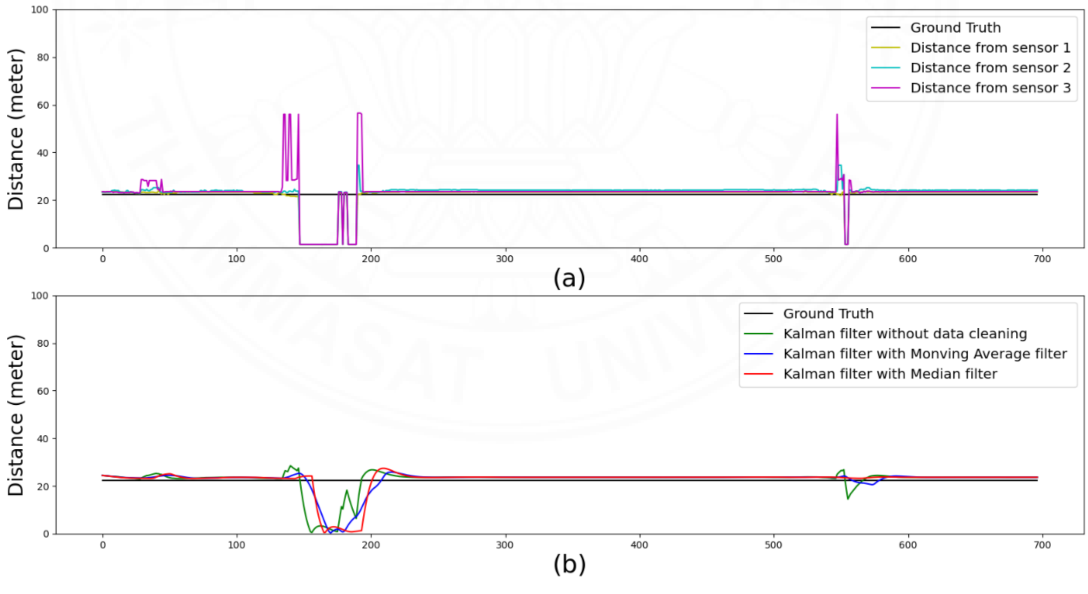


Figure 4.12 Graph of experiment 3 reports measuring the distance to a gray car.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

Table 4.3 illustrates experimental result is reported in object 1 is a white car, and object 2 is a gray car. The result of object 1 by using the Kalman filter without data cleaning is 1.67 meters, the result of Kalman filter with median filter is 2.05 meters, and the result of Kalman filter with moving average filter is 2.07 meters. The result of object 2 by using the Kalman filter without data cleaning is 4.49 meters, the result of Kalman filter with median filter is 4.82 meters, and the result of Kalman filter with moving average filter is 4.32 meters.

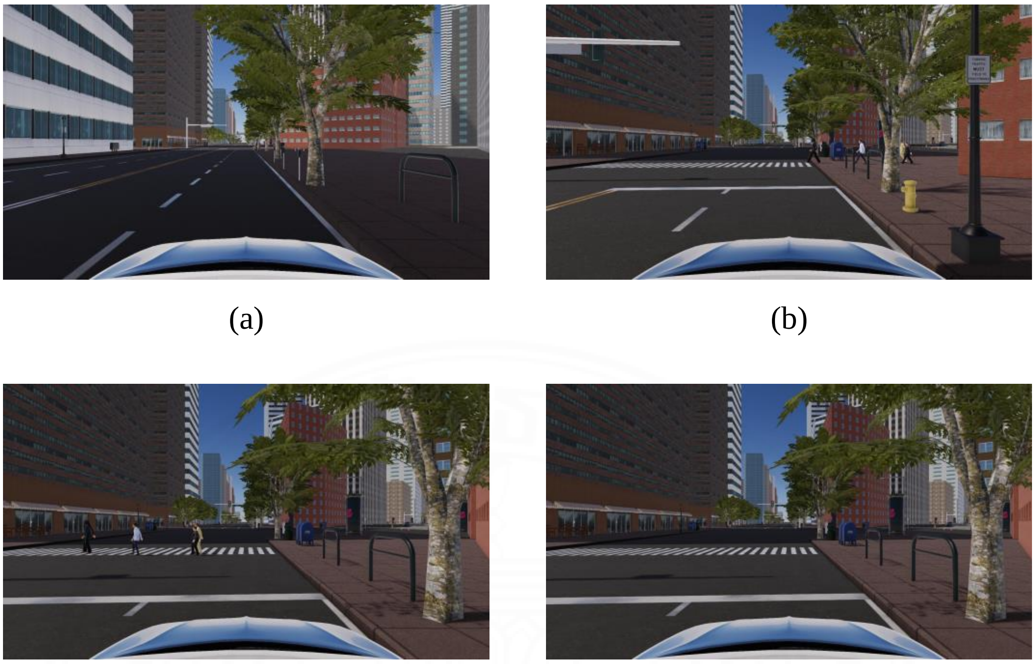
Table. 4.3 Evaluation comparison of experiment 3 between Kalman filter, Kalman filter with Median Filter, and Kalman filter with Moving Average Filter.

Object	Root Mean Square Error (Meters)		
	Kalman Filter Without Data Cleaning	Kalman Filter With Median Filter	Kalman Filter With Moving Average Filter
1	1.67	2.05	2.07
2	4.49	4.82	4.32

4.1.4 Experiment 4: Detection pedestrians

Experiment 4, The self-driving car is driving then stops at crosswalks. There are four groups of pedestrians crossing the road. The simulated image was shown in Figure 4.13.

The object detection algorithm can detect pedestrians and divide them into 3 groups. Because two pedestrians walking together in the last group.



(a) (b)

(c) (d)

Figure 4.13 Situation of experiment 4.

(a) Scene 1, (b) Scene 2, (c) Scene 3, (d) Scene 4.

The results of each pedestrian group were shown in Figures 4.15, 4.16, and 4.17. The input signals from each graph have a lot of noisy and uncertain signals. Because the point cloud was from pedestrians and was not from pedestrians. This situation is shown in Figure 4.14. When selecting point-cloud sometimes it is from other. The outputs of each method don't approximate to ground truth. The output of the Kalman filter without data cleaning was very sensitive when compared with other methods. The output of the Kalman filter with Moving Average filter and Median filter were similar. Both algorithms cannot adjust this experiment, but those algorithms make the output signal smoother than only using the Kalman filter.

Figure 4.14 Pedestrian is walking across the road

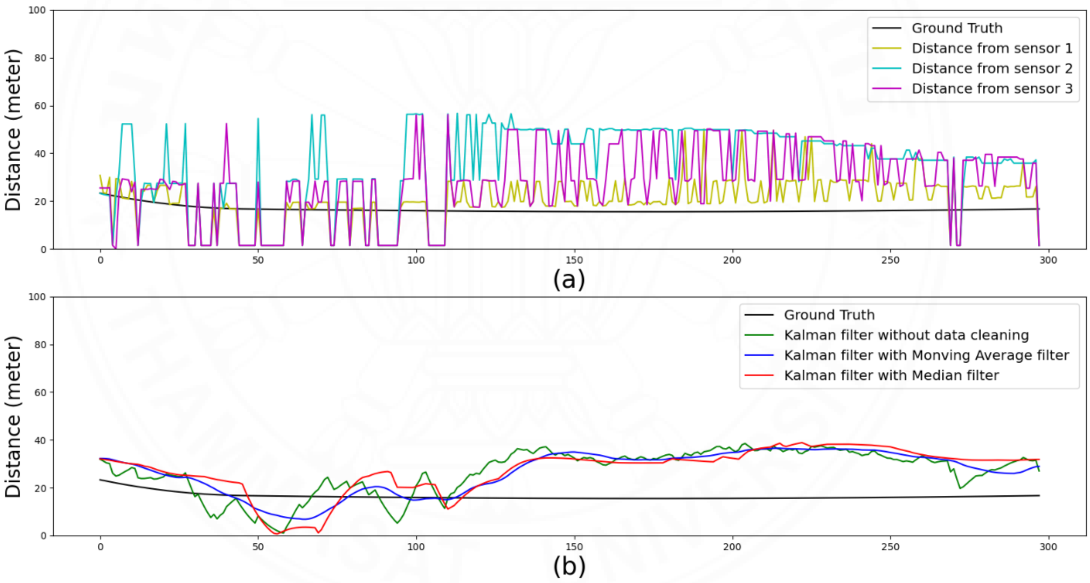


Figure 4.15 Graph of experiment 4 reports measuring the distance to pedestrian in blue frame.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

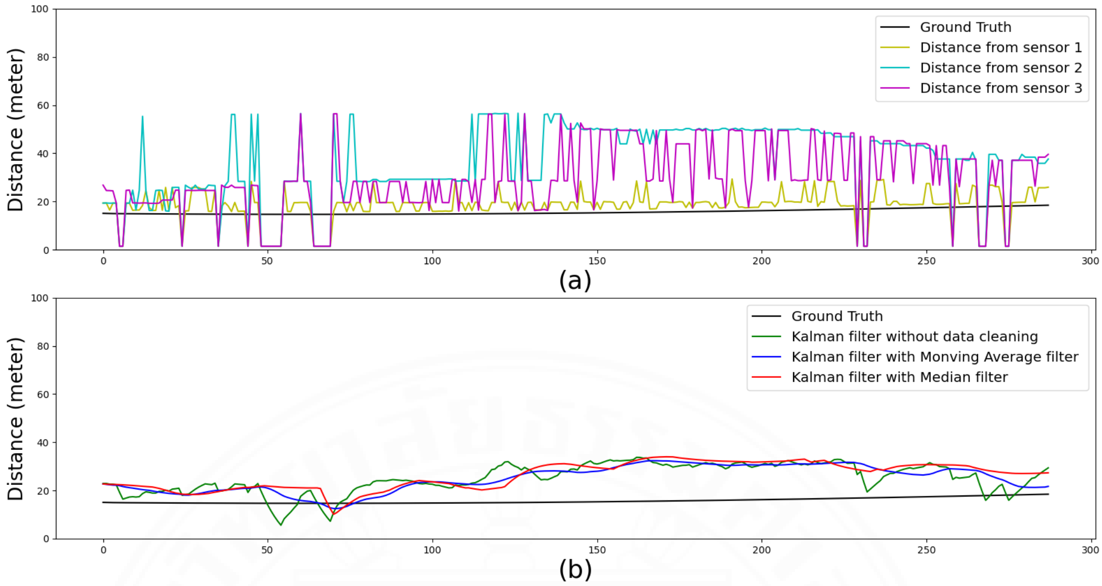


Figure 4.16 Graph of experiment 4 reports measuring the distance to pedestrian in a green frame.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

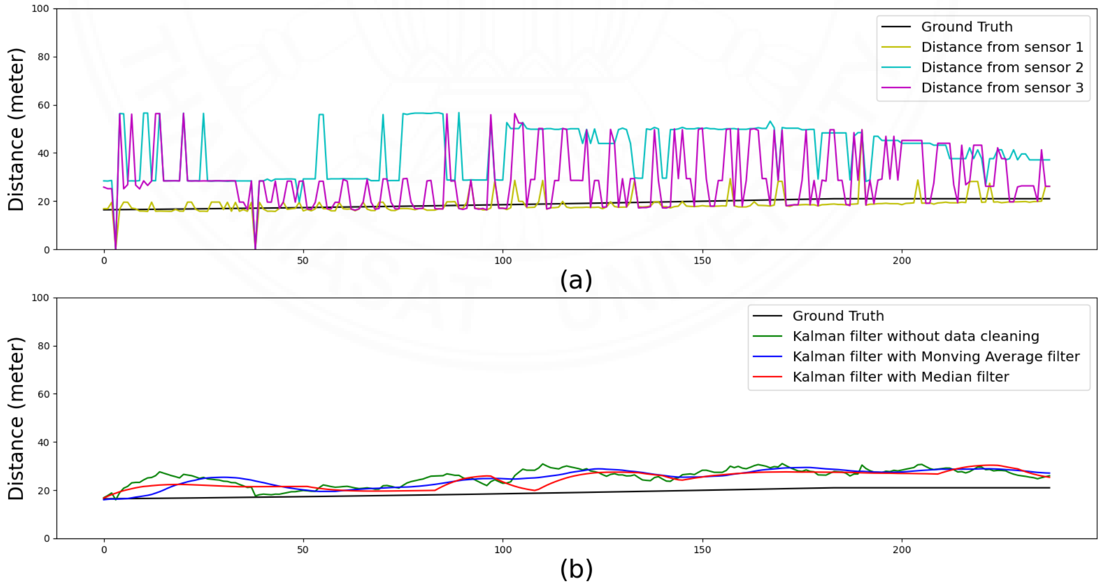


Figure 4.17 Graph of experiment 4 reports measuring the distance to pedestrian in a yellow frame.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

Table 4.4 reports experimental result is reported in object 1 is pedestrian in a blue frame, object 2 is pedestrian in a green frame., and object 3 is pedestrians in a yellow frame. The result of object 1 by using the Kalman filter without data cleaning is 13.85 meters, the result of Kalman filter with median filter is 14.23 meters, and the result of Kalman filter with moving average filter is 13.34 meters. The result of object 2 by using the Kalman filter without data cleaning is 10.65 meters, the result of Kalman filter with median filter is 11.23 meters, and the result of Kalman filter with moving average filter is 10.17 meters. The result of object 3 by using the Kalman filter without data cleaning is 6.77 meters, the result of Kalman filter with median filter is 5.62 meters, and the result of Kalman filter with moving average filter is 6.28 meters.

Table. 4.4 Evaluation comparison of experiment 4 between Kalman filter, Kalman filter with Median Filter, and Kalman filter with Moving Average Filter.

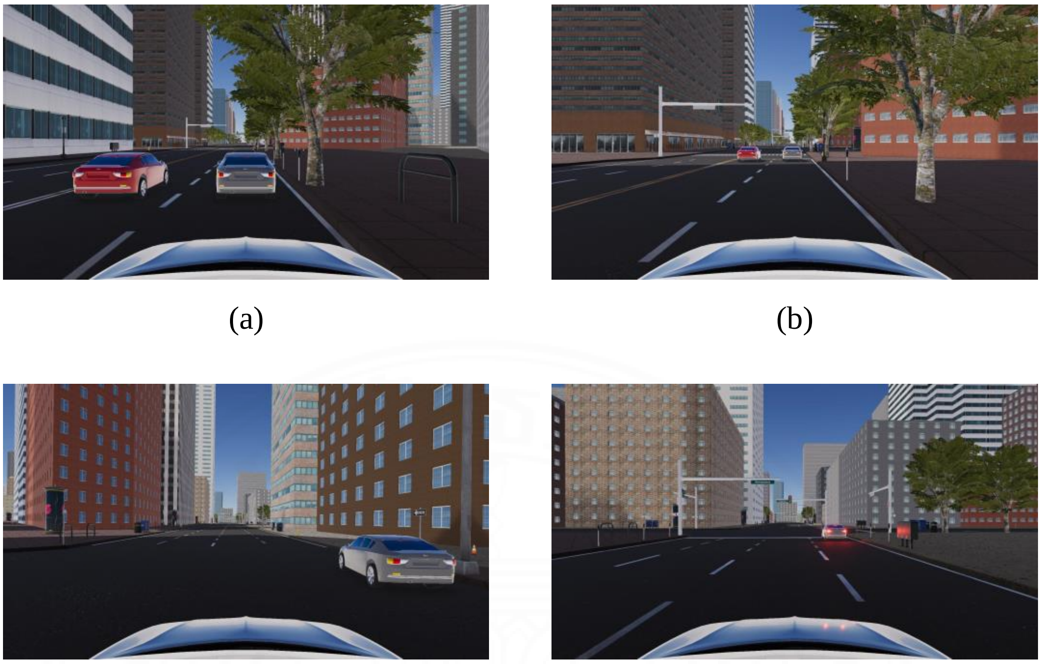
Object	Root Mean Square Error (Meters)		
	Kalman Filter Without Data Cleaning	Kalman Filter With Median Filter	Kalman Filter With Moving Average Filter
1	13.85	14.23	13.34
2	10.65	11.23	10.17
3	6.77	5.62	6.28

4.2 Dynamic Testing

Dynamic testing is the measurement of the distance from a self-driving car to another cars. The self-driving car moves while the other environment is moving.

4.2.1 Experiment 5: Detection Car when self-driving car moving

In this experiment, the self-driving car moves forward on the road. There are a red car and a gray car on the front. The self-driving car turns right at the intersection. The red car drives straight, and the gray car turns right. After that, the self-driving car follows the gray car, then stops at the next intersection. The simulated image was shown in Figure 4.18.



(a) (b)

(c) (d)

Figure 4.18 Situation of experiment 5.
(a) Scene 1, (b) Scene 2, (c) Scene 3, (d) Scene 4.

For the red car, the input signal and the output signal are shown in Figure 4.19. There is a little noise from the sensors in a short duration. The outputs from each method are almost similar. For the gray car, the input signal and the output signal are shown in Figure 4.20. There are interesting input signals during frames 185 to 210. This signal was created when the self-driving car was turning right. The LiDAR from positions 2 and 3 got invalid point cloud because the position of the sensor cannot measure the object at this time. The output from the Kalman filter without data cleaning was a bit sensitive to this noise. The output signal from the Kalman filter with the Median filter and Moving average filter are similar. Both algorithms can adjust this spike noise.

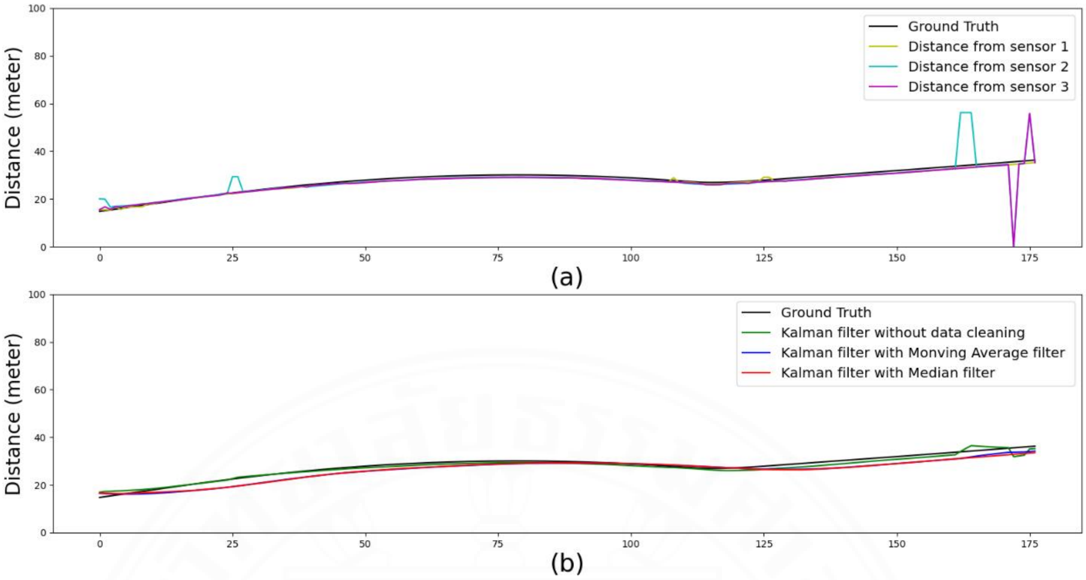


Figure 4.19 Graph of experiment 5 reports measuring the distance to a red car.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

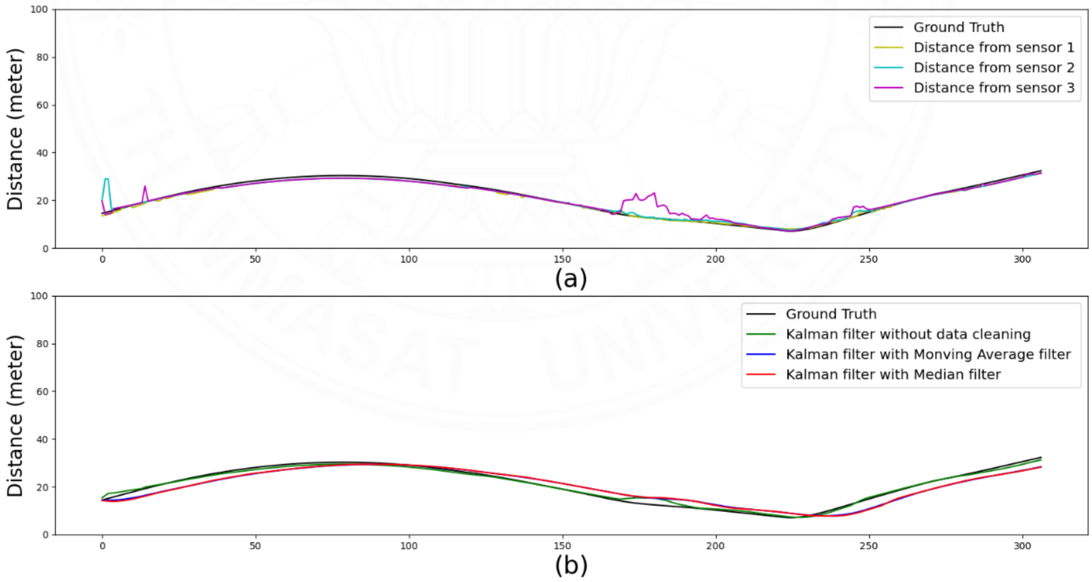


Figure 4.20 Graph of experiment 5 reports measuring the distance to a gray car.

(a) Ground truth and input from each sensor

(b) Comparative distance estimation between each method

Table 4.5 shows experimental result is reported in object 1 is a red car, and object 2 is a gray car. The result of object 1 by using the Kalman filter without data cleaning is 1.18 meters, the result of Kalman filter with median filter is 2.1 meters, and the result of Kalman filter with moving average filter is 2.05 meters. The result of object 2 by using the Kalman filter without data cleaning is 1.11 meters, the result of Kalman filter with median filter is 2.55 meters, and the result of Kalman filter with moving average filter is 2.52 meters.

Table. 4.5 Evaluation comparison of experiment 5 between Kalman filter, Kalman filter with Median Filter, and Kalman filter with Moving Average Filter.

Object	Root Mean Square Error (Meters)		
	Kalman Filter Without Data Cleaning	Kalman Filter With Median Filter	Kalman Filter With Moving Average Filter
1	1.18	2.1	2.05
2	1.11	2.55	2.52

CHAPTER 5
CONCLUSION AND SUGGESTION

This thesis proposes a distance estimation for the self-driving car. The proposed method uses sensor fusion, which uses a different type of sensor. There are LiDAR sensors and a camera, those sensors used for sensing the surrounding environment. The LiDAR sensor receives the point cloud, and the camera receives the images. The proposed method makes the integration of the sensors work as one sensor for measuring the distance from the self-driving car to the objects.

The research was divided into three forms consist of Kalman filter without data cleaning, Kalman filter with Median filter, and Kalman filter with Moving Average filter. The Kalman filter without data cleaning were sensitive to the noisy signal when the input signal is lost or impulse noise. The Median filter and the Moving Average filter were used with the data cleaning step. The Kalman filter with Median filter adjusts impulse noise. As a result, the results of this method are more performance. When the invalid input data is more than half of the number of data cleaning steps. The Median filter does not remove invalid input data in this situation. The Kalman filter with Moving Average filter makes the smooth result. The average input data effect to slowly. The impulse noise affects the result a bit but in a long time. The Kalman filter with Median filter and Kalman filter with the Moving Average has the disadvantage of not having a delay in data cleaning step. The delay time depends on the number of data cleaning step.

The measuring distance from pedestrians has limitations due to the collecting data from LiDAR sensors. The pedestrians are small when compared to the bounce box. Mostly point cloud isn't point cloud from the pedestrians. The proposed method can't adjust the data close to the ground truth. The measuring distance from cars was good due to the size of cars. There is lots of point cloud project on cars. As a result, the distance estimation was easier. Object detection has lots of effects on this work. Most invalid data occur when object detection cannot detect the object. The improvement of object detection will increase the performance of distance estimation.

However, this research needs to validate the result from the real situation. The experiment is tested in simulation. The experiment cases don't cover all situations. We will make it in a real environment in the future. There are more factors about obtaining data from the real environment such as Translucency, black object, and more. We expect this research to be useful for those interested and to be part of the future.

REFERENCES

Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., & Ouni, K. (2019, February). Car detection using unmanned aerial vehicles: Comparison between faster r-cnn and yolov3. In *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS)* (pp. 1-6). IEEE. doi: https://doi.org/10.1109/UVS.2019.8658300

Campbell, S., O'Mahony, N., Krpalcova, L., Riordan, D., Walsh, J., Murphy, A., & Ryan, C. (2018, June). Sensor technology in autonomous vehicles: A review. In *2018 29th Irish Signals and Systems Conference (ISSC)* (pp. 1-4). IEEE. doi: https://doi.org/10.1109/ISSC.2018.8585340

Chirkov, D. G., & Stotckaia, A. D. (2020, May). Methods of Software Data Filtering for Working with Sensors in the Field of Robotics. In *2020 XXIII International Conference on Soft Computing and Measurements (SCM)* (pp. 121-124). IEEE. doi: https://doi.org/10.1109/SCM50615.2020.9198779

Ćorović, A., Ilić, V., Đurić, S., Marijan, M., & Pavković, B. (2018, November). The real-time detection of traffic participants using YOLO algorithm. In *2018 26th Telecommunications Forum (TELFOR)* (pp. 1-4). IEEE. doi: https://doi.org/10.1109/TELFOR.2018.8611986

Deilamsalehy, H., & Havens, T. C. (2016, October). Sensor fused three-dimensional localization using IMU, camera and LiDAR. In *2016 IEEE SENSORS* (pp. 1-3). IEEE. doi: https://doi.org/10.1109/ICSENS.2016.7808523

Keatmanee, C., Baber, J., & Bakhtyar, M. (2013). Simple example of applying extended kalman filter. In *1st International Electrical Engineering Congress, Chiangmai city, Thailand.*

Kocić, J., Jovičić, N., & Drndarević, V. (2018, November). Sensors and sensor fusion in autonomous vehicles. In *2018 26th Telecommunications Forum (TELFOR)* (pp. 420-425). IEEE. doi: https://doi.org/10.1109/TELFOR.2018.8612054

Kumar, G. A., Lee, J. H., Hwang, J., Park, J., Youn, S. H., & Kwon, S. (2020). LiDAR and camera fusion approach for object distance estimation in self-driving vehicles. *Symmetry*, 12(2), 324. doi: https://doi.org/10.3390/sym12020324

Li, Y., & Ibanez-Guzman, J. (2020). Lidar for autonomous driving: The principles, challenges, and trends for automotive lidar and perception systems. *IEEE Signal Processing Magazine*, 37(4), 50-61. doi: https://doi.org/10.1109/MSP.2020.2973615

Mutauranwa, L., & Nleya, M. (2013, September). An efficient median filter in a robot sensor soft ip-core. In *2013 Africon* (pp. 1-5). IEEE. doi: https://doi.org/10.1109/AFRCON.2013.6757651

Redmon, J., Divvala, S., Girshick, R., & Farhadi, A. (2016). You only look once: Unified, real-time object detection. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 779-788).

Rong, G., Shin, B. H., Tabatabaee, H., Lu, Q., Lemke, S., Možeiko, M., ... & Kim, S. (2020, September). Lgsvl simulator: A high fidelity simulator for autonomous driving. In *2020 IEEE 23rd International conference on intelligent transportation systems (ITSC)* (pp. 1-6). IEEE. doi: https://doi.org/10.1109/ITSC45102.2020.9294422

Sharma, K. R., Honc, D., & Dušek, F. (2014, September). Sensor fusion for prediction of orientation and position from obstacle using multiple IR sensors an approach based on Kalman filter. In *2014 International Conference on Applied Electronics* (pp. 263-266). IEEE. doi: https://doi.org/10.1109/AE.2014.7011716

Van Brummelen, J., O'Brien, M., Gruyer, D., & Najjaran, H. (2018). Autonomous vehicle perception: The technology of today and tomorrow. *Transportation research part C: emerging technologies,* 89, 384-406. doi: https://doi.org/10.1016/j.trc.2018.02.012

Warren, M. E. (2019, June). Automotive LIDAR technology. In *2019 Symposium on VLSI Circuits* (pp. C254-C255). IEEE. doi: https://doi.org/10.23919/VLSIC.2019.8777993

APPENDIX

Laser Scanner Specification

The specification of LiDAR sensor in this research is Velodyne Puck, which this sensor is shown in Figure A.1. The Specification of Velodyne Puck is describe as follows:

- Range up to 100 m
- Field of view (vertical): 30° (+15° to -15°)
- Field of view (horizontal/azimuth): 360°
- Rotation rate: 5 - 20 Hz

Figure A.1 Velodyne Puck

BIOGRAPHY

Name	Chinnawat Chinnapun
Education	2017: Bachelor of Engineering (Robotics and Automation Engineering) Institute of Field Robotics King Mongkut's University of Technology Thonburi.

Publication

Chinnapun, C., Rajruangrabin, J., & Kondo, T. (2021, September). A self-driving car based on a combination of Median and Kalman filters. In 2021 60th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE) (pp. 848-852). IEEE.