A VEHICLE TRACKING SYSTEM USING THERMAL AND LIDAR DATA

A Thesis

by

ANDY HWANG

Submitted to the Office of Graduate and Professional Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE

Chair of Committee,	Sivakumar Rathinam
Committee Members,	Swaminathan Gopalswamy
	Xiubin Wang
Head of Department,	Andreas A. Polycarpou

May 2020

Major Subject: Mechanical Engineering

ABSTRACT

Object detection is important for autonomous vehicles. While regular cameras can be easily affected by low light environments or high brightness objects, thermal cameras can still get sharp images in those conditions. In this project, an object detection system is developed with thermal images and LiDAR data to achieve vehicle detection and status estimation in extreme lighting conditions. A convolutional neural network that is trained for this project can detect objects in thermal images, and then a tracking algorithm developed in the project can track the same objects between images from different time frames. LiDAR data can be projected to the thermal image plane after calibrations, and once the bounding boxes of the detected objects have been made by the neural network, the LiDAR points within the bounding boxes can be associated to the objects. The system can use the bounding boxes and their associated LiDAR data to estimate the status of the objects, such as location and velocity.

ACKNOWLEDGEMENTS

I would like to thank my committee chair, Dr. Rathinam, and my committee members, Dr. Gopalswamy and Dr. Wang, for their guidance and support throughout the course of this research.

Thanks also go to George Chustz and Vamsi Krishna Vegamoor for their help with the experiments and all the support.

Finally, thanks to my mother and father for their encouragement.

CONTRIBUTORS AND FUNDING SOURCES

This work was supervised by a thesis committee consisting of Dr. Sivakumar Rathinam and Dr. Swaminathan Gopalswamy of the Department of Mechanical Engineering and Dr. Xiubin Wang of the Department of Civil Engineering.

The model is section 3.8.3.1 is provided by Dr. Rathinam.

All other work conducted for the thesis was completed by the student independently. Graduate study was supported through research and teaching assistantships from Texas A&M University.

TABLE OF CONTENTS

Page

ABSTRACT ii

ACKNOWLEDGEMENTS iii

CONTRIBUTORS AND FUNDING SOURCES iv

TABLE OF CONTENTS v

LIST OF FIGURES vii

LIST OF TABLES ix

1. INTRODUCTION 1

2. RELATED RESEARCH 5

2.1. LiDAR and camera object detection 5
2.2. Object detection with convolutional neural network 5
2.3. 3D object detection 6
2.4. Solving Perspective-n-Point and Mapping thermal images to 3D 7

3. SYSTEM DESIGN 8

3.1. System Overview 8
3.2. Thermal Camera 9
3.2.1. Mounting Solution 9
3.2.2. Data Acquiring and Publishing 10
3.2.3. Thermal Camera Calibration 10
3.3. Object Detection 14
3.3.1. Thermal Image Dataset 15
3.4. Object Tracking 15
3.4.1. Object Tracking Algorithm 16
3.4.2. Matching Algorithm 17
3.5. Camera Distance Estimate 17
3.5.1. Method 1 18
3.5.2. Method 2 20
3.5.3. Comparison Between Methods 20
3.6. LiDAR and Calibration 21

3.7. Thermal Camera and LiDAR Calibration 22
3.7.1. Calibration Process 23
3.8. Sensor fusion 26
3.8.1. Data association 26
3.8.2. LiDAR and camera distance estimate combination 27
3.8.3. Extended Kalman filter 28

4. EXPERIMENTAL RESULTS 31

4.1. Direct sunlight test 31
4.2. Experiment 1 31
4.3. Experiment 2 34

5. CONCLUSIONS AND FUTURE WORK 41

REFERENCES 42

LIST OF FIGURES

Page

Figure 3.1 System structure 8

Figure 3.2 Thermal image and image of the mesh board 11

Figure 3.3 Transparency paper with printed checkerboard 12

Figure 3.4 3D printed checkerboard on tape 13

Figure 3.5 Multiple object detection using YOLO algorithm 14

Figure 3.6 The structure of the tracking algorithm 16

Figure 3.7 Method 1 explanation 18

Figure 3.8 Results of the method 1 19

Figure 3.9 Method 2 20

Figure 3.10 Thermal camera and LiDAR calibration process 22

Figure 3.11 the process of extracting corners 23

Figure 3.12 The target board in LiDAR data. Front view (left) and top front view (right). The low confident points are marked in red. 24

Figure 3.13 filtering the edges of scan lines, and calculating the center of the circles and the corners of the board 25

Figure 3.14 the thermal image of Figure 3.13 26

Figure 3.15 The observer reference frame 28

Figure 4.1 Experiment 1 setup: A and B are the LiDARs, only B is used in the experiment. C is the thermal camera. 31

Figure 4.2 Experiment 1 target 1 global longitudinal distance 33

Figure 4.3 Experiment 1 target 2 global longitudinal distance 34

Figure 4.4 target 1 global longitudinal distance 36

Figure 4.5 target 1 global latitudinal distance 36

Figure 4.6 target 2 global longitudinal distance 37
Figure 4.7 target 2 global latitudinal distance 37
Figure 4.8 target 1 δx comparison 38
Figure 4.9 target 1 δy comparison 38
Figure 4.10 target 1 velocity comparison 39
Figure 4.11 target 2 δx comparison 39
Figure 4.12 target 2 δy comparison 40
Figure 4.13 target 2 velocity comparison 40

LIST OF TABLES

Page

Table 3.1 Flir A35 calibration result 13

1. INTRODUCTION

Autonomous vehicles, AVs, are one of the most important research topics with the potential to save lives and improve the safety of public transportation. AVs have many components, such as sensors, controllers and communication. Sensors are an important part of an AV system, and LiDARs and cameras are the most commonly used sensors for AV applications. We are interested in thermal cameras because they have better performance in some off-nominal conditions and people and because other animals are distinct from other cold objects in a thermal image.

All objects emit radiation when they are at a temperature greater than absolute zero. The wavelength and frequency of the radiation emitted from an object are associated with the temperature of the object. According to the Wien's displacement law, higher temperatures emit shorter peak wavelengths. The temperature range commonly experienced on Earth's surface emits radiation within the infrared wavelength range; therefore, infrared sensors, such as a Microbolometer, can be used as a temperature sensor. A thermal camera is an array of infrared sensors, and it outputs a temperature distribution. A temperature distribution can be visualized to an image for display or for further usage. Objects which have a different temperature from the environment, such as warm-blooded animals and moving vehicles, will have a clear contrast within a thermal image, and those objects are also the primary targets of an object detection system for moving vehicles.

In contrast to a regular camera that needs a certain amount of light to capture a clear image, a thermal camera has the advantage to still capture a clear image in low

light or even in completely dark conditions. In low light conditions, such as at night, regular camera-based vehicle detection will be unable to operate effectively. Also, when there is a special lighting condition, for instance, direct sunlight in the view or reflections from a bright light source, regular camera-based systems must rely on algorithms to reduce the effects. A thermal camera-based detection system can avoid the problems caused by different lighting conditions; even though a thermal vision system will have its obvious weak points, such as rainy conditions, this research is concentrated on the application of a thermal camera in a vehicle detection system.

A light detection and ranging (LiDAR) system uses lasers to measure the distance between the LiDAR sensor and objects by measuring the laser travel time. The LiDAR system first sends a laser pulse to an object and collects the reflection from the object. The distance can be calculated by multiplying the time elapsed between sending and receiving and the speed of light. A LiDAR system can contain more than one laser scanning module (channel) to increase the laser coverage. Another way to increase the coverage of a LiDAR system is to add mechanical devices, such as a motor, to move the scanning module, so the LiDAR system can rotate to obtain a 360-degree surround view; however, having moving mechanical parts in a system will make the system more likely to fail as time passes.

On the other hand, radio detection and ranging (radar) systems are another widely used system; the difference between the systems depends on the wavelength of electromagnetic radiation used. A radar system transmits and receives radio signals to detect objects, and a LiDAR system uses lasers. Objects smaller than the wavelength

may not produce an adequate reflection for radar detection since radio uses larger wavelengths. In order to get more details, shorter wavelengths are preferred, and that is the reason why lasers are commonly used for scanning the environment. A radar system made for cars, for instance the Delphi ESR, usually contains a one-dimension radio array to cover a certain range. Unlike the radars made for detecting large vehicles, such as airplanes, the smaller radars are often not manifested with rotational mechanical devices. Since a radar with a one-dimensional radio array only detects the object on the same level as the radar is on, the radar may not get the correct distance when part of a target is behind other objects. For instance, when a pedestrian stands next to the front of a car, a camera might capture the pedestrian while a radar system could only capture the car.

Stereo camera systems are also a common way to obtain distance information. By comparing the location of an object in two different images taken by two different cameras at the same time, the distance between the camera system and the object can be calculated. There are stereo camera systems on the market, such as Flir Bumblebee 2, and the camera usually comes with software for distance calculation. The Open Source Computer Vision Library (OpenCV) also provides functions for users to obtain depth maps from stereo images taken by any calibrated cameras. However, thermal cameras usually do not have adequate pixel resolutions, and the distance resolution of a stereo camera system highly depends on the pixel count. Assuming the detection range is 1 to 40 meters. The line from a lens to another is referred to as a baseline. If a pair of cameras with a 320-horizontal-pixel sensor and 48 degrees of view angle, like the Flir A35, needs to see an object which is one meter from the middle point of the baseline, the farthest

distance between the camera lenses can be about 0.9 meters. Choosing the farthest distance is to reserve more pixels for farther objects. With the 0.9-meter long baseline, the distance error is about 7 meters per 0.15 degrees or per pixel when this stereo system is looking at a target which is 40 meters away from the center of the baseline, and the resolution is not enough for this research.

Another method is using a camera to detect the left and right edges of a car, and then use the average or largest allowed size of cars and the pixels counted between the left and right edges to calculate the distance to the target. This method is not very accurate. If a vehicle is not in a regular size, the distance estimate can be wrong.

A LiDAR system can obtain the distance information for the objects in a thermal image since the system covers both horizontal and vertical directions, and a LiDAR system can also provide enough resolution at a far distance. Therefore, in this research, a combination of thermal camera and LiDAR is used for detection.

The related research will be presented in the next chapter, and it includes research that uses LiDAR and camera systems, object detection with neural networks and solving a PnP problem. In chapter 3, the thermal and LiDAR object detection system will be introduced in detail from the hardware calibration to the detection algorithm. The experimental results are shown in chapter 4 which includes direct sunlight in the camera view and vehicle detection at night. The conclusion is in chapter 5.

2. RELATED RESEARCH

2.1. LiDAR and camera object detection

LiDARs and cameras have been largely used to detect vehicles and pedestrians. In early stages, LiDARs were used to detect the region of interest rather than to detect objects directly (Szarvas, Sakai, & Ogata, 2006) (Premebida, Monteiro, Nunes, & Peixoto, 2007). LiDAR data was used to separate objects that are closer to itself than the background, and the location of those objects as a region of interest. Limiting the number of pixels which need to be interpreted by a detection algorithm can accelerate the process because the algorithms can take large amounts of computational power. Computational power requirements can be reduced by using the region of interest within corresponding images instead of further objects' regions. Lowering the computational power can lower the processing time to achieve faster or real-time object detection when the computational power is limited.

2.2. Object detection with convolutional neural network

Convolutional neural networks (CNNs) became popular due to the improvement of graphic processing unit (GPU) computing. GPUs are designed for matrix computing, and CNN requires matrix multiplications and dot products. A CNN looks for a data point and the points around it, so the network can be used for object classification. For object detection, a region-based convolutional neural network (R-CNN) was proposed by Girshick (Girshick, Donahue, Darrell, & Malik, 2014) that extracts 2000 region

proposals from an input image and then uses the proposals for classification with a convolutional neural network.

YOLO (Redmon & Farhadi, 2018) is an algorithm that takes a different approach to get bounding boxes and classify objects. YOLO algorithm simplifies the image by dividing the image with a large grid and predicts bounding boxes using the divided data. The grid size depends on the size of the inputs of the algorithm, but 13 by 13 is a common grid size for the YOLO algorithm. Each grid block should generate five bounding boxes with other blocks. Once the bounding boxes are located, the algorithm only classifies what is within the bounding box. A recent paper shows that YOLOv3 has higher accuracy and faster processing time compared to Faster R-CNN (Benjdira, Khursheed, Koubaa, Ammar, & Ouni, 2019).

2.3. 3D object detection

3D point cloud is a common way to store 3D data in the autonomous system field. The source of point clouds can be from a LiDAR or a depth camera. In order to detect objects from point clouds, a few methods have been developed, for instance, a motion-based detector (Dewan, Caselitz, Tipaldi, & Burgard, 2016) and a detector that uses 3D CNN (Maturana & Scherer, 2015). The motion-based detector groups the points having the same movement as an object. The 3D object detector, VoxNet, puts points in 3D occupancy grids, and then VoxNet uses a 3D CNN to recognize objects. The datasets used in 3D object recognition usually contain a large amount of points; however, the LiDAR used in this research, Velodyne VLP-16, does not provide enough channels for

object recognition at distance. For a vehicle that is 10 meters away from the LiDAR, the vehicle is usually covered by three channels of scan lines.

2.4. Solving Perspective-n-Point and Mapping thermal images to 3D

A Perspective-n-Point (PnP) problem is to solve a camera's location and rotation when given n sets of coordinates in 3D and their corresponding 2D coordinates within an image from a camera. In order to find the camera's location and rotation in respect of the LiDAR's frame, the PnP problem needs to be solved, and since 3 sets of points are used, it is a P3P problem. The P3P problem has been solved in the paper, Complete solution classification for the perspective-three-point problem (Gao, Hou, Tang, & Cheng, Volume: 25 , Issue: 8 , Aug. 2003). With the camera matrix and the P3P solution, images can be mapped in 3D space when there is a 3D point that corresponds to the pixel on the image.

3. SYSTEM DESIGN

3.1. System Overview

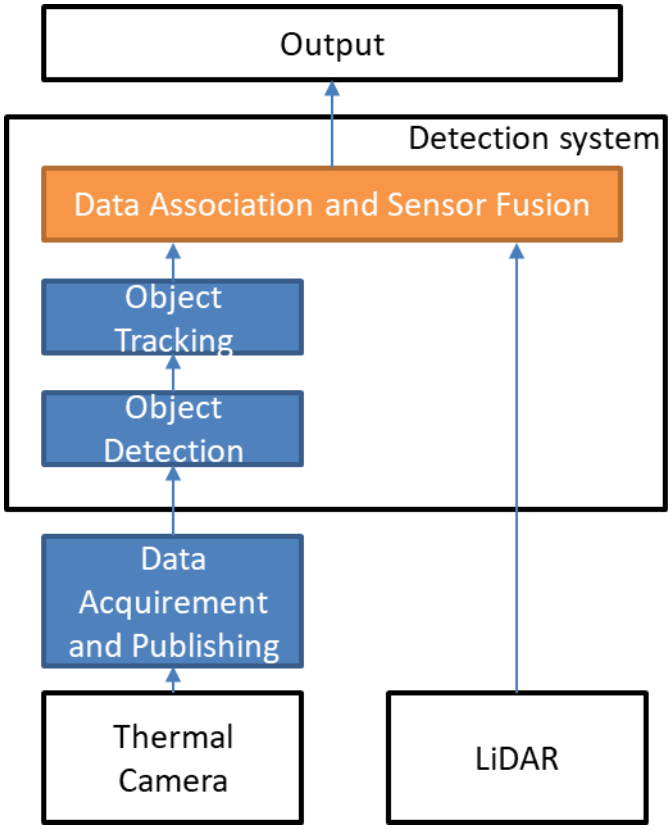

Figure 3.1 System structure

The system contains two sensors, a thermal camera and a LiDAR. The LiDAR driver uses the Robot Operating System, ROS, as its platform on Linux systems, and the thermal camera uses the GigE vision protocol, a communication protocol made for network security camera systems. The ROS is an open platform that is widely used in robotics and autonomous systems. The GigE vision protocol is not an open protocol, so

it requires a license to use. The ROS is the platform used in this project because it is designed to easily sync and record data on the platform.

The thermal data processing, shown as the blue blocks in Figure 3.1, includes acquiring and publishing images to the ROS, object detection and object tracking. The LiDAR data is used in the data association and fusion section where the LiDAR points can be associated with the detected objects, and then the data will be used to estimate the status of the objects, including the relative location and speed.

3.2. Thermal Camera

The thermal camera used in this research is FLIR A35. The FLIR A35 can measure objects with a temperature range between -25°C to 135°C, and the accuracy is ±5°C. FLIR A35 outputs a 16-bit (14-bit resolution), 320x256 pixels image at 60 Hz, and its lens provides a 48-degree field of view. The FLIR A35 relies on Power over Ethernet (PoE) for the power, so a Power over Ethernet power injector is required to power the camera. The PoE injector used in this research is TL-PoE150S from TP-LINK, and the injector provides 1Gb Ethernet connection and 15.4 watts of DC power.

3.2.1. Mounting Solution

An infrared thermal camera is unable to see through typical glass, it sees the temperature of the glass or the reflection from the glass when a thermal camera is pointed at glass; therefore, the FLIR A35 cannot be placed in the testing vehicle like other cameras. Also, the FLIR A35 is not water resistant, it might get damaged by rain if the camera was mounted permanently on top of the testing vehicle. The mounting solution for the thermal camera is using a strong suction cup mount and making an

adapter to connect the four-screw panel on the bottom side of the thermal camera and the suction cup mount. The suction cup can be easily attached to most of the surface on the testing vehicle and is designed to withstand the wind when the vehicle is moving at a speed of 150mph according to the product information. The adapter is 3D printed in PLA.

3.2.2. Data Acquiring and Publishing

The thermal camera, FLIR A35, uses GigE vision protocol. Since the manufacturer, FLIR, does not provide a camera driver that works with the ROS, a GigE-to-ROS software interface or ROS driver is needed for this project. The eBus SDK from Pleora Technologies is a library that allows developers to implement the GigE vision protocol in their code, and the eBus SDK is used in this project to communicate with the thermal camera.

A ROS thermal camera driver has been written for this project. The driver uses eBus SDK to access and acquire data from the FLIR A35 camera, and then the ROS driver publishes the image data to the ROS for further processing. The images captured by the thermal camera are in the 16-bit raw format which includes the 14-bit temperature information.

3.2.3. Thermal Camera Calibration

Thermal cameras cannot see a checkerboard calibration image printed on paper as a regular camera can; therefore, several methods of making calibration checkerboards were tested during the camera calibration process. The first method is using a 3D printed mesh board. The camera could capture the mesh board if the board has a different

temperature to the background, shown in Figure 3.2. The problem that occurs when using this method is that the calibration programs, such as MATLAB and OpenCV, could not see the mesh board as a target for calibration by default. Modifying the calibration program or manually selecting all the corners will be needed with the mesh board.

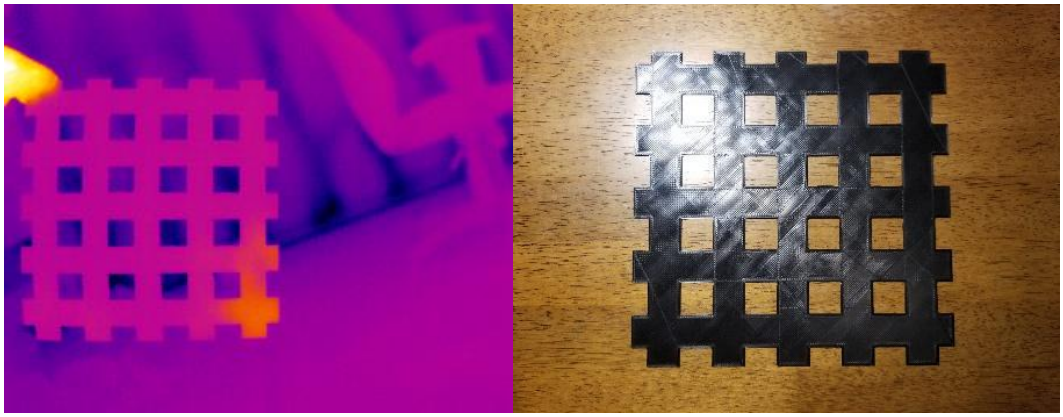

Figure 3.2 Thermal image and image of the mesh board

The second method is to print the checkerboard on transparency paper, shown in Figure 3.3. Thinner plastic products, such as plastic bags and curtains, are transparent to the thermal camera. The assumption made when using transparency paper is that the toner from the printer will be opaque to the thermal camera while the transparent portion will be transparent to the thermal camera as it is to regular cameras; however, the transparency paper used in testing was completely opaque to the thermal camera.

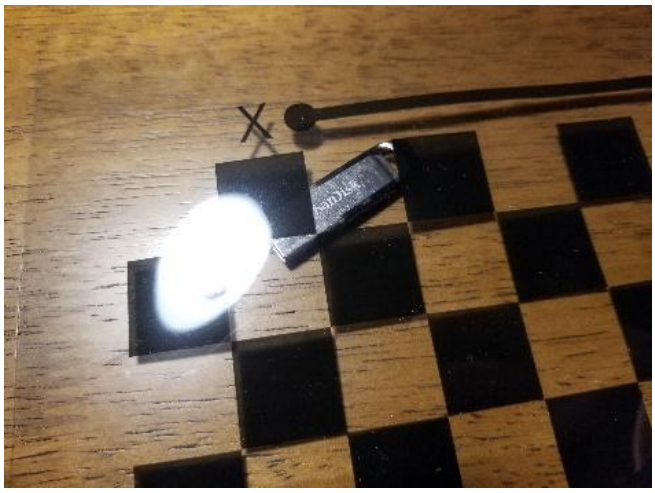

Figure 3.3 Transparency paper with printed checkerboard

The third method is a checkerboard 3D printed on a known transparent film to regular and thermal cameras, shown in Figure 3.4. It is difficult to 3D print a perfect checkerboard without a bottom layer because each block on the board only contacts the surrounding blocks at a point. 3D printing a checkerboard on a thin film that is nearly transparent to thermal cameras can provide support for the blocks. Furthermore, each block has been printed in a shape of trapezium to limit the visibility of the sides of the blocks and to have enough strength to prevent bending because of the weight. As a result, the film-supported checkerboard can be detected by the MATLAB camera calibration tool.

Table 3.1 shows the result of calibration of a FLIR A35 thermal camera with a default F=9mm lens. Where f_x and f_y are the focal length expressed in pixel units, and $c_x \; and \; c_y$ are the location of the center pixel.

Table 3.1 Flir A35 calibration result

Intrinsic Matrix $\begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$	$\begin{bmatrix} 356.1022 & 0 & 166.2797 \\ 0 & 358.7729 & 145.4332 \\ 0 & 0 & 1 \end{bmatrix}$
Radial Distortion	$[-0.4469\ 0.3313\ -0.6365]$
Tangential Distortion	$[-0.0076\ -3.0241e-05]$

The mean reprojection error of the calibration is 0.28 pixels. The result of the camera calibration is used in removing image distortion and calibrating with the LiDAR. OpenCV is used to remove the image distortion in the camera software.

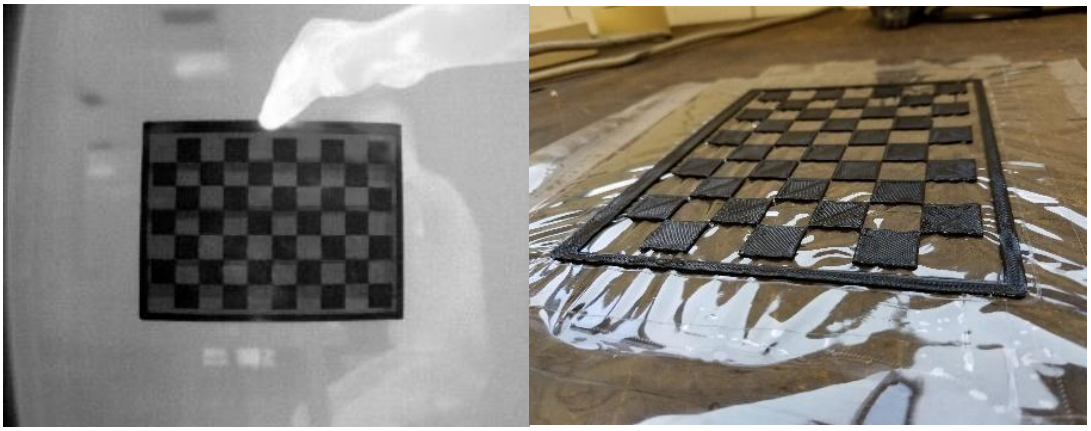

Figure 3.4 3D printed checkerboard on tape

3.3. Object Detection

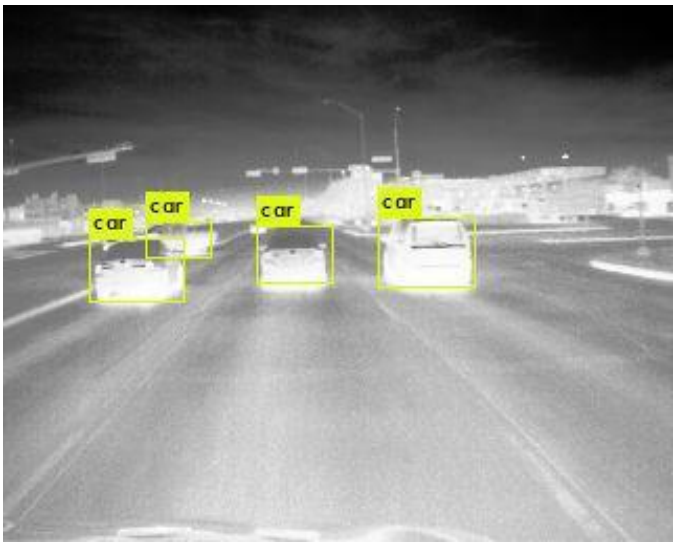

Figure 3.5 Multiple object detection using YOLO algorithm

YOLO is an image object detection and classification algorithm that is based on a convolutional neural network, CNN, and can run in real-time on a high-end configuration computer. YOLO is originally written in C++ using the Darknet machine learning library, and it is completely open source. YOLO is implemented to the object detector in this project. The object detector takes thermal images through the ROS, and then it outputs a bounding box and a class for each detected object.

Since the object detector is based on a CNN, the network needs to be trained before it can detect objects. The training dataset will be introduced in section 0, and after training with the thermal dataset, the object detector can detect objects in a thermal image, shown in Figure 3.5.

3.3.1. Thermal Image Dataset

The training dataset in this work is based on the FLIR thermal dataset. The FLIR dataset includes a large amount of 16-bit grayscale images. Each 16-bit file only contains 14 bits of data from the bits 3 to 16, and the lowest two bits are always zero. Since YOLO is optimized with 8-bit images, the dataset in this work has been downscaled to 8-bit. The maximum and the minimum values in the whole FLIR training dataset are 3500 and 1500, respectively. The way that the images are downscaled is to shift the minimum value of the whole dataset to zero and scale the maximum value down to the limit of 8 bits, 255.

Normalizing each image using its own maximum and minimum value has also been experimented in this research. The performance of the network trained with normalized images is reliable with a 99.8% detecting rate. However, when the sun or a reflection of the sun enters the camera view, the rest of the image will become too dim, and the objects in the image will not be detected by the network without filtering out the sun.

3.4. Object Tracking

The object tracking system continuously takes the outputs from the object detector, including the bounding box and the class of detected objects, and finds the same objects between different time frames, and then the tracking system assigns a consistent ID number to the same object in all the time frames. The same object will be assigned to the same ID as it was in the last frame. When an object is covered by other objects or temporarily leaving the camera view for a certain amount of time, the tracking

system can reassign the same ID to the object as what the object had before. The output of the tracking system is the input, bounding boxes and classes, with the assigned ID number.

3.4.1. Object Tracking Algorithm

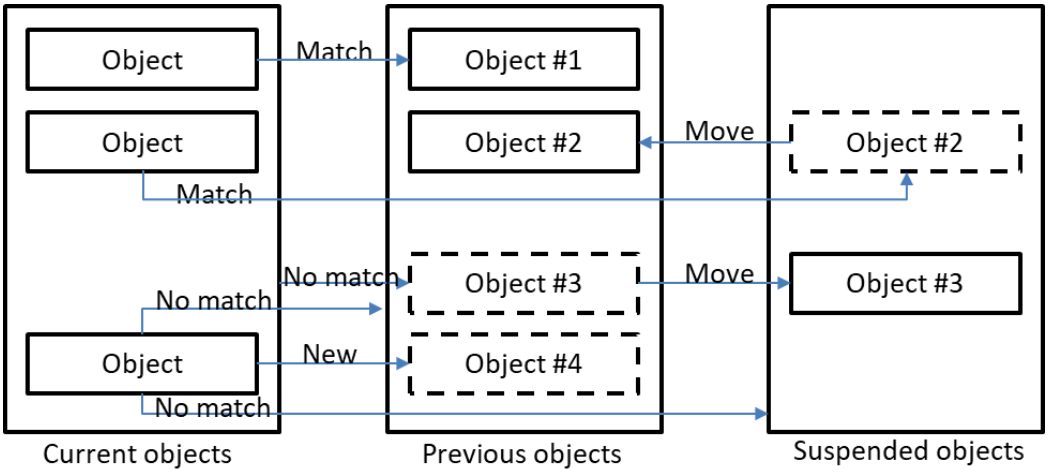

Figure 3.6 The structure of the tracking algorithm

The bounding boxes from the neural network are the input of the object tracking algorithm. The algorithm has three lists, current, previous and suspended, of objects saved, shown in Figure 3.6. When the system starts, there is no object in the previous object list, so all the input objects, the current objects, from the neural network will be assigned a new ID and then moved to the previous object list. When new current objects come in the next time frame, the algorithm checks the previous object list for matches. If a current object matches the previous object, the current object will be assigned the same ID as its match. The method of match determination is described in section 3.4.2. If a current object cannot match any of the previous or suspended objects, it will be assigned to a new ID. When a previous object cannot be paired with any of the current objects, the

previous object will be suspended. A current object can only be paired with one previous or suspended object.

The suspended list contains timers to monitor all the suspended objects. Once an object's timer reaches a predefined time, the suspended object will be removed from the list. The ID of a removed object will be released and can be reused when needed. However, before the timer reaches the predefined time, an object can be paired with a current object if they match. Once a suspended object paired with a current object, the current object will get the suspended ID and then move to the previous object list in the next time frame.

3.4.2. Matching Algorithm

The matching algorithm uses the previous size, location and differential of the previous sizes and locations of the bounding boxes of an object to predict its current status. The nearest neighbor algorithm is used to pair the current detections and the prediction with weights and predefined limits. At the same speed, a farther object has a smaller angle change than a closer object; therefore, a smaller bounding box can only have a smaller location differentiation between frames. The time interval between frames also affects the location differentiation tolerance. A large time interval between two continuous frames increases the location differentiation tolerance.

3.5. Camera Distance Estimate

The camera distance estimate uses the bounding boxes from the object detector to estimate the distance between the camera and an object that the system is tracking. There are two different methods used for the distance estimate in this project.

3.5.1. Method 1

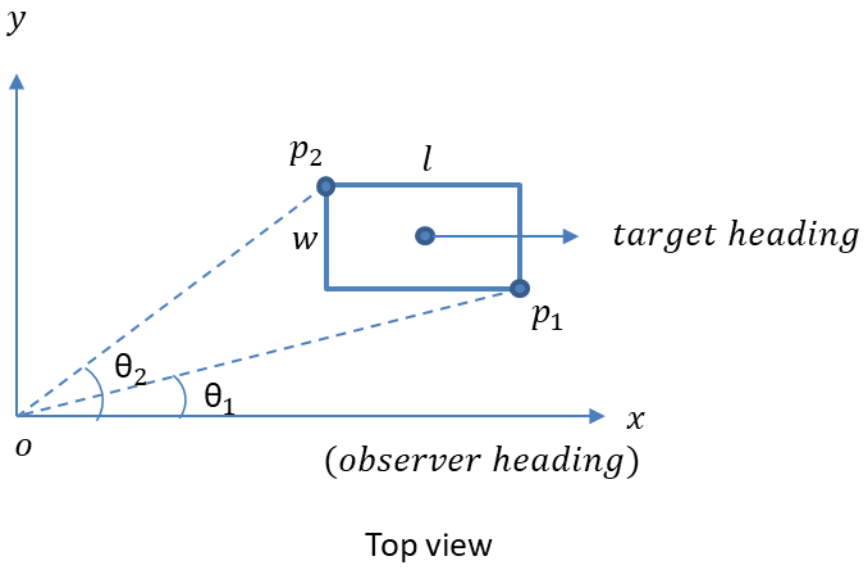

Figure 3.7 Method 1 explanation

The method 1 assumes that: 1. the length and width of a target vehicle are known and 2. the heading of the target vehicle is the same as the observer. A regular size vehicle, such as Honda Civic or Toyota Corolla, is about 4.6 meters long and 1.8 meters wide, and when using this method, all vehicles are assumed to be in the same regular size. Point p2(x2, y2) and point p1(x1, y1) in Figure 3.7 can be denoted as:

Equation 3.1

$$x_2 = \frac{\tan(\theta_1) * l + w}{\tan(\theta_2) - \tan(\theta_1)}$$

$$y_2 = x_2 \tan(\theta_2)$$

$$x_1 = x_2 + l$$

$$y_1 = y_2 - \mathrm{w}$$

Where the θ_1 and θ_2 can be obtained by multiplying the left and right pixel value of the bounding box to the field of view per pixel.

When there is no relative movement between a target and the moving observer, the target and the observer must head toward the same direction. When the bounding box of a target stays still for a period of time while the observer is moving, the ratio of the height and the compensated width of the bounding box will be registered as the standard ratio of the object. The compensated width is the original width subtracted the estimated width of the part of the bounding box that contains the side of the target vehicle. When a new compensated bounding box ratio is close to the standard value, the distance estimate has a high confidence, and when the current ratio is far from the standard, the confidence is low. As shown in Figure 3.8, the ratio is away from the registered value between 22 and 25 second, and the error of the estimated distance is large in the same period.

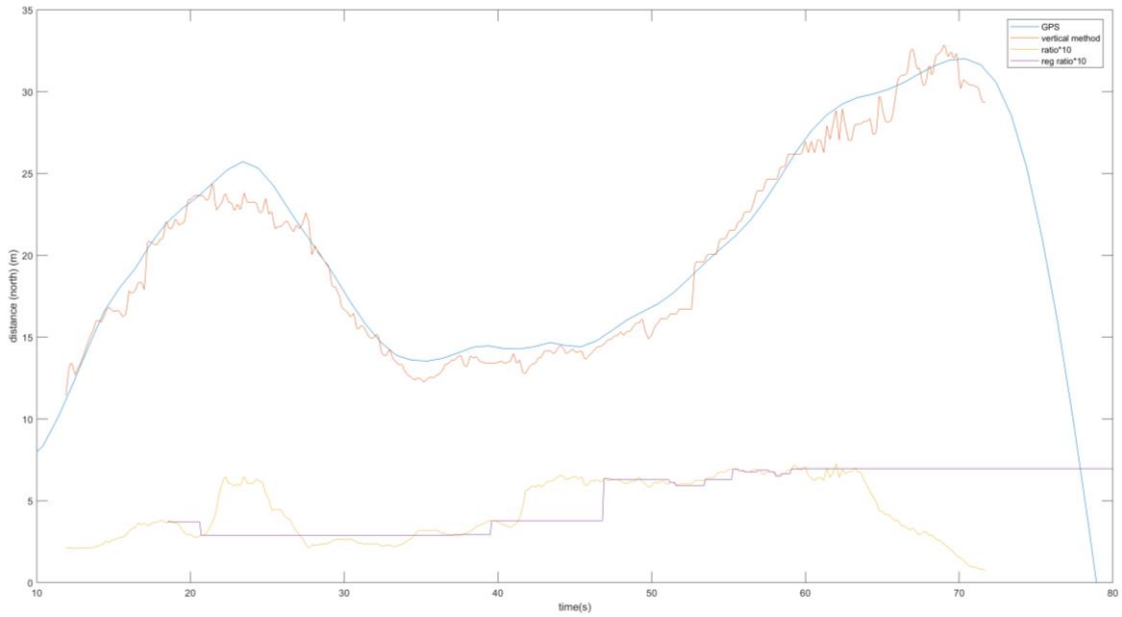

Figure 3.8 Results of the method 1

3.5.2. Method 2

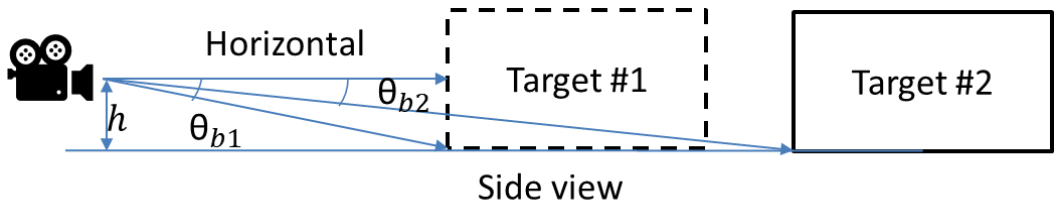

Figure 3.9 Method 2

When the field of view, the horizon and the height of the camera (h) are known, the longitudinal distance of between the target #1, shown in Figure 3.9, and the observer can be denoted as:

Equation 3.2

$$d_1 = h/\tan(\theta_{b1})$$

Where the θ_{b1} can be converted from the bottom edge of the bounding box, and the horizon can be obtained by calibration with a lidar. The calibration will be introduced in the section 0 and 0.

3.5.3. Comparison Between Methods

The two methods are based on the object detector to provide reliable object detections. In method 1, the actual vehicle size will affect the distance estimate, while the method 2 will not be affected. The method 1 cannot be used when a target is turning, and the confidence value of the method can indicate when a turning is happening. The method 2 works when a target is turning, but the method does not have a confidence estimate. Both methods require the information from the camera matrix to work, and the method 2 needs extra calibration for the horizon.

3.6. LiDAR and Calibration

The LiDAR used in this project is Velodyne VLP-16. Velodyne VLP-16 has 16 channel laser scanners, 100 meters of scanning range, 360 horizontal FOV and ± 15° vertical FOV, and it provides 300,000 points per second with an accuracy of ±3 cm. Velodyne provides a ROS driver for the LiDAR, so the LiDAR is ready after the driver installation. The communication interface on the VLP-16 is Ethernet.

In order to get the correct coordinates of objects, the LiDAR needs to be leveled or calibrated to the ground. Finding the ground plane by selecting three different LiDAR points which locate on the ground can provide all the information needed for the ground calibration. The ground calibration is also needed for the camera distance estimation method 2.

The LiDAR's x axis needs to be calibrated with the observing vehicle's wheel axis or heading, so the detection can be integrated with the GPS data for the certification.

3.7. Thermal Camera and LiDAR Calibration

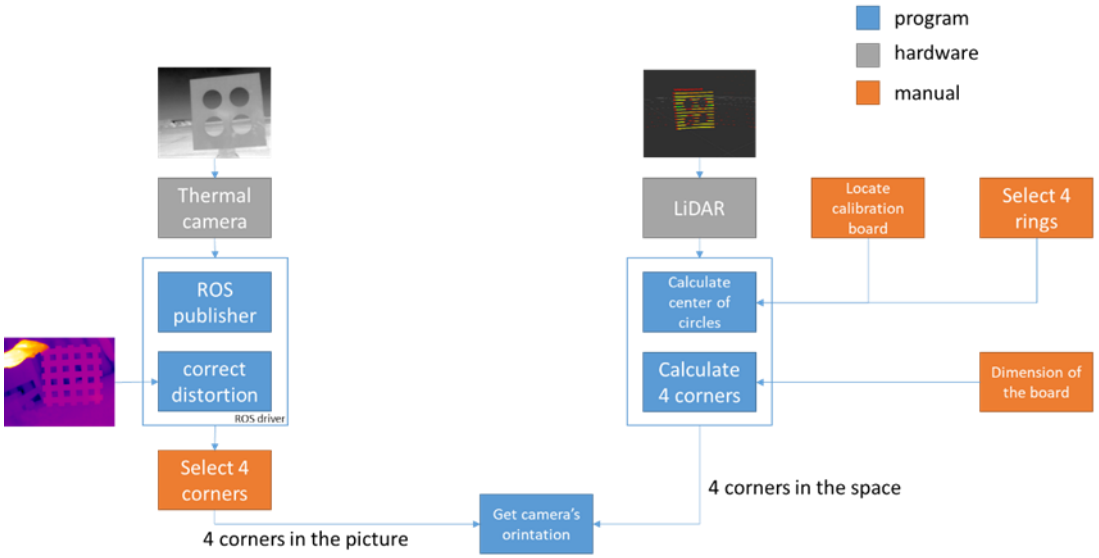

Figure 3.10 Thermal camera and LiDAR calibration process

The purpose of calibrating the thermal camera and the LiDAR is to obtain the relative location and orientation of the camera from the perspective of LiDAR. As the size of the target is known and the camera matrix has been obtained, obtaining the relative location and relative orientation can be found using the points that can be observed or calculated from both the camera and LiDAR data. The process is shown in Figure 3.10. The calibration process in this research uses the four corners of the target. The pixel location of the target corners from the camera can be easily observed and manually selected. The location of the target corners in the LiDAR data usually must be interpreted from other points due to the limitation of using only 16 channels with a total of 30 degrees of view.

3.7.1. Calibration Process

Velodyne provides the ROS driver for the VLP-16 LiDAR. The LiDAR data can be obtained by subscribing to the LIDAR ROS topic, and the data is in Point Cloud format. Since the data is in Point Cloud format, the Point Cloud Library, a library made to handle point cloud format, is used to assist part of the data processing in this program. The purpose of this calibration is to use the LiDAR point cloud to locate the four corners of the target board, and then the four corners are used to solve the location and orientation of the thermal camera. The corners extraction process is shown in Figure 3.11.

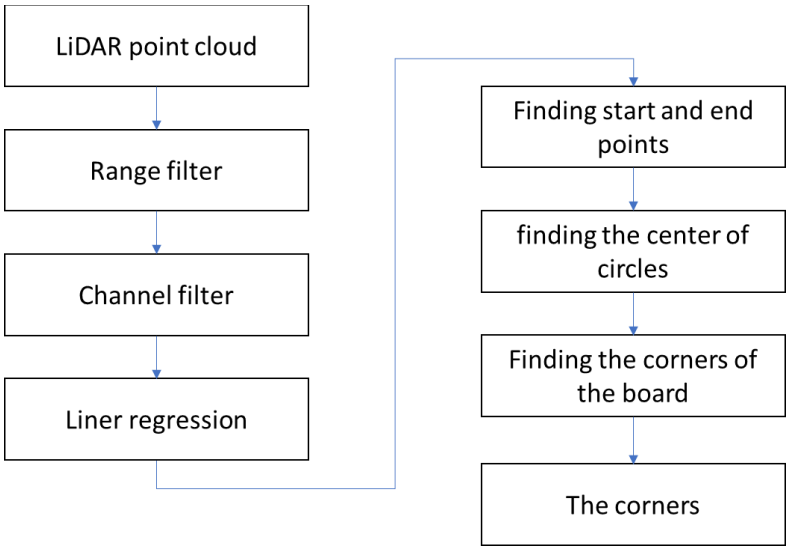

Figure 3.11 the process of extracting corners

The target board, shown in Figure 3.12 and Figure 3.14, used for the calibration has four circular areas removed. As a circle's width continuously changes when looking at different heights, the width value observed by a LiDAR can be used to estimate where

the LiDAR scan line is projected on the target board. When two scan lines go through two of the four circular empty areas, two of the center of the circles and the tilt angle of the target board can be obtained. This calibration program uses all the circles on the target board and requires at least two scan lines on each circle. As shown in the Figure 3.12 left, each continuous scan line is captured by the same laser scanner, and the points which are captured by a specific scanner will be in a specific channel.

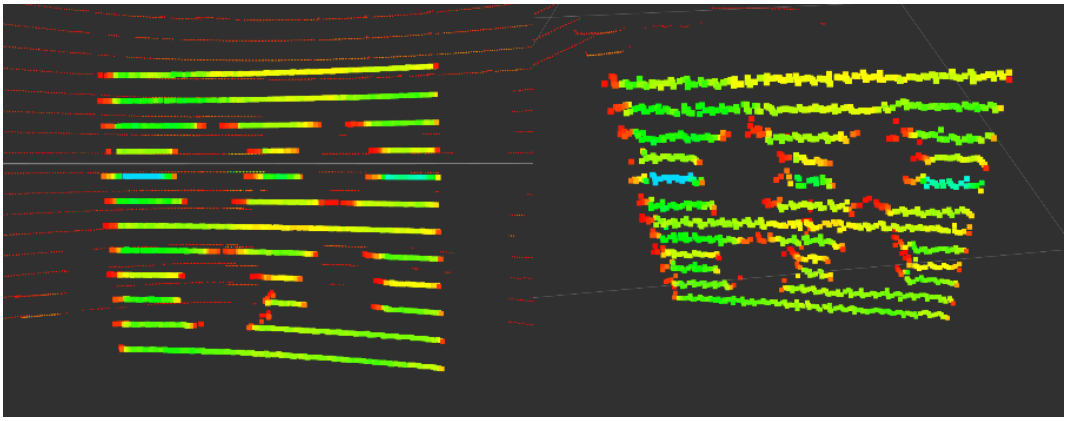

Figure 3.12 The target board in LiDAR data. Front view (left) and top front view (right). The low confident points are marked in red.

The LiDAR data can be noisy when the confidence of a region is low, and low confident points usually happen near edges and fall behind the actual surface, shown in Figure 3.12; however, filtering out all of the low confident points in a point cloud will cause a large amount of data loss. In most of the cases, the LiDAR observed circle sizes will be increased and exceed the actual size when the low confident points are removed. A filter has been applied to the LiDAR data to reduce the data loss while correcting the issue of points falling behind the surface. First, the filter finds a line that has the closest distance to all the confident points on the same channel by solving its linear regression

using the reweighted least squares method. Second, the filter projects all the points on the channel to the closest distance line and uses the new points for further processing.

Since there are two scan lines on a circle, there are enough edge points, shown as the pink points in Figure 3.13, can be extracted and used for the calculation of the center of a circle. To calculate the center of the circles in space, three points from the edge of one of the circles will be picked, and then the three points are used to extract two vectors to form a plane. With three points transformed to the two-dimensional plane, the center of the circle is calculated by solving the general equation of a circle, and then the center coordinate is transformed back to the 3D point cloud.

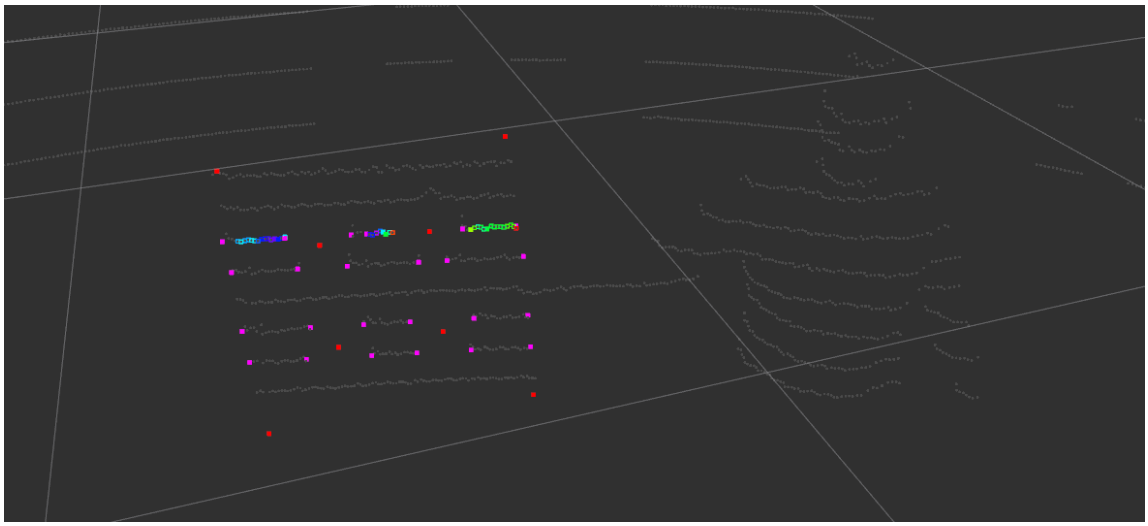

Figure 3.13 filtering the edges of scan lines, and calculating the center of the circles and the corners of the board

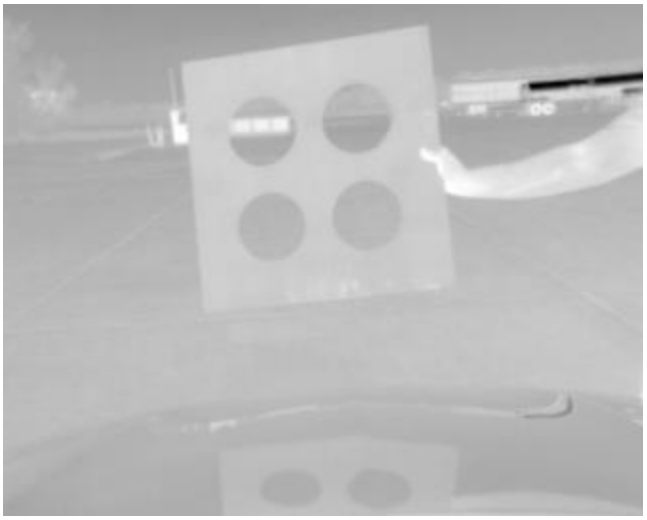

Figure 3.14 the thermal image of Figure 3.13

The dimensions of the target circle board are known. By obtaining the center of circles, the corners of the board can also be obtained. The corners in the thermal image which is matched to the corresponding points in the point cloud need to be manually selected. With three points in space and their 2D counterparts in the thermal image, the relative location and orientation of the thermal camera are obtained by solving the P3P problem.

3.8. Sensor fusion

3.8.1. Data association

The data association process reads the output data from the neural network in the form of bounding boxes, and then the system converts the bounding boxes to areas in the space by using the camera matrix, relative position, and orientation information. Due to the limitation of having only 16 channels in the LiDAR, often there are not enough

points within the area to rebuild the model of the object. Therefore, the points will be used only to provide the object's distance from the LiDAR.

The amount of points that is in a bounding box projection space is varied. An object that is close to the LiDAR can be covered by hundreds of LiDAR points, but when an object is far away from the LiDAR, the LiDAR may have just a few points or, in some cases, no LiDAR points within the bounding box. The points remaining in the bounding box projection space are used to estimate the location of the object.

With the LiDAR points in the bounding box space, the average point, closest longitudinal distance and the standard deviation of the points will be calculated. The average point is usually a better estimate when there is no point that falls to another object's surface in the background. Then there is a point that falls to a distance surface which does not belong to the target object, the standard deviation will increase. The standard deviation can be used as the confidence value for the accuracy of the average point estimate. When the standard deviation is low, the confidence is high. When the confidence of the average point estimate is low, the closest longitudinal distance is a more reliable estimate.

3.8.2. LiDAR and camera distance estimate combination

The LiDAR distance estimate switches between the average location and the closest longitudinal distance depending on the average estimate confidence. The closest longitudinal distance can be associated with the center of the bounding box to output a location estimate. The LiDAR estimate is more accurate than the camera estimate, so the camera estimate is only used when the LiDAR data is unavailable.

The camera distance estimate is the average of both methods when the confidence is high. When the confidence is low, only the bottom edge method will be used. When the LiDAR data becomes unavailable, the last error between the LiDAR estimate and the camera estimate will be used as the offset to correct the camera distance estimate.

Equation 3.3

$$output = cam_{current} - cam_{lastlidar} + lidar_{last}$$

3.8.3. Extended Kalman filter

A target's velocity can be obtained by subtracting its location between different time frames and then dividing the result by the time, but the velocity generated by this method can be very noisy. The extended Kalman filter from "Identifying Cut-In Vehicles by Fusing Radar and Vision Data for Truck Platooning Safety" is for making a better velocity estimate.

3.8.3.1. Model

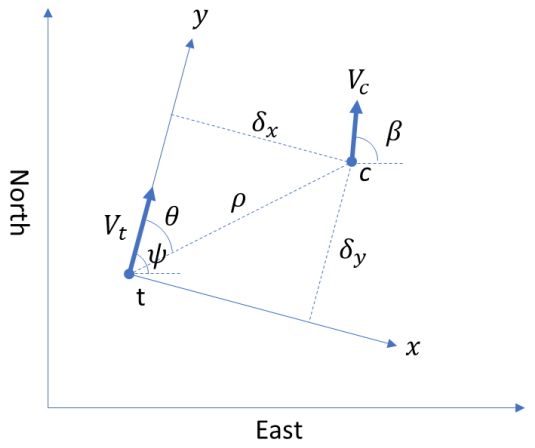

Figure 3.15 The observer reference frame

Point $t(x_t, y_t)$ represents the observer, and point $c(x_c, y_c)$ represents the target vehicle, shown in Figure 3.15. Assuming the accelerations $\dot{V}_c$ and angular speed $\dot{\beta}$ of a target are 0. The state transition and observation models can be denoted as:

Equation 3.4

$$\begin{bmatrix} \dot{\delta}_x \\ \dot{\delta}_y \\ \dot{V}_c \\ \dot{\beta} \end{bmatrix} = \begin{bmatrix} 0 & \dot{\psi} & \sin(\psi - \beta) & 0 \\ -\dot{\psi} & 0 & 0 & \cos(\psi - \beta) \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix} \begin{bmatrix} \delta_x \\ \delta_y \\ V_c \\ \beta \end{bmatrix} + \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \end{bmatrix} V_t$$

$$y = (\delta_x, \rho, \dot{\rho})$$

Where $\begin{bmatrix} \delta_x \\ \delta_y \end{bmatrix} = \begin{bmatrix} \sin(\psi) & -\cos(\psi) \\ \cos(\psi) & \sin(\psi) \end{bmatrix} \begin{bmatrix} x_c - x_t \\ y_c - y_t \end{bmatrix}$, ψ and β are the heading of the observer and the target vehicle. V_t represent the velocity of the observer.

The equation can be rewritten in discrete time:

Equation 3.5

$$\bar{x}_k = \bar{f}(\bar{x}_{k-1}, u_{k-1})$$

$$y_k = h(\bar{x}_k, u_k)$$

3.8.3.2. Filtering

Since the model is a nonlinear system, the nonlinear version of Kalman filter, extended Kalman filter, can be applied to the system. The extended Kalman filter uses the Jacobean matrices of the state transition and observation equations in the Kalman filter equations in order to linearize the nonlinear model. The Jacobean matrices of the state transition and observation equations are:

Equation 3.6

$$F_k = \left. \frac{\partial \bar{f}}{\partial \bar{x}} \right|_{\hat{x}_{k-1}}$$

$$H_k = \frac{\partial h}{\partial \bar{x}}\Big|_{\hat{x}_{k|k-1}}$$

The predicted covariance matrix is given by

Equation 3.7

$$P_{k|k-1} = F_{k-1} P_{k-1} F_{k-1}^T + Q_{k-1}$$

Where Q_k is the process noise covariance matrix. The Kalman gain matrix is

Equation 3.8

$$K_k = P_{k|k-1} H_k^T (H_k P_{k|k-1} H_k^T + R_k)^{-1}$$

where R is the measurement noise covariance matrix. Update of the state covariance matrix:

Equation 3.9

$$P_k = (I - K_k H_k) P_{k|k-1}$$

The predicted stated is given by

Equation 3.10

$$\hat{x}_k = \hat{x}_{k|k-1} + K_k (y_k - H_k \hat{x}_{k|k-1})$$

4. EXPERIMENTAL RESULTS

4.1. Direct sunlight test

In a single-vehicle direct sunlight detection test, the network missed one frame in 74 seconds when the system was running at 7 frames per second on a laptop computer. The sun, as a high-temperature object which can affect the normalized network, has been filtered out by removing a specific temperature range.

4.2. Experiment 1

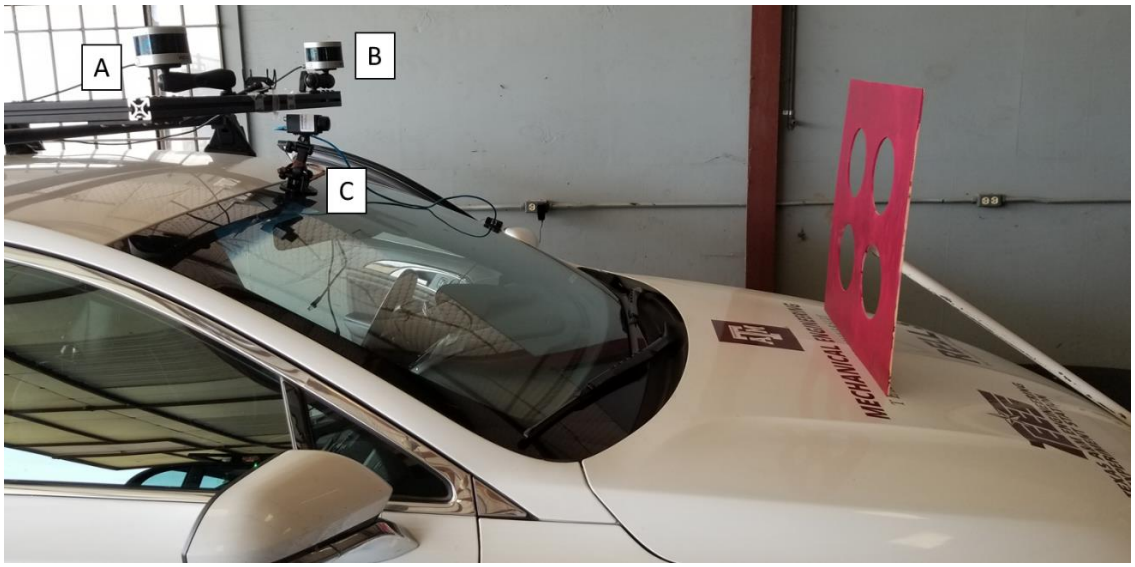

Figure 4.1 Experiment 1 setup: A and B are the LiDARs, only B is used in the experiment. C is the thermal camera.

The experiment started about an hour after sunset with clear weather conditions. The testing vehicle, shown in Figure 4.1, was driven behind two other target vehicles. The distance between the testing vehicle and the target vehicles were between 10 and 50 meters, and all the vehicles were driven at a speed between 15 to 30 kilometers per hour.

GPS units are set up on all the vehicles for tracking algorithm verification purposes, and the GPS units received positioning signals from 15 to 22 satellites during the experiment. The target vehicle 1 is a Toyota RAV4, and the target vehicle 2 is a Honda Civic.

The outputs from the detection system are distances between the observation vehicle and target vehicles in path coordinates. To compare detection results with GPS logs, both the detection results and the GPS logs are converted to Cartesian coordinates in meters. The heading data for the conversion is extracted from the GPS logs by comparing two continuous points in the GPS logs. Therefore, before the test vehicle starts moving, the heading data can be noisy due to GPS signal drifting.

The detection system usually needs horizontal 10 pixels to recognize a vehicle. The width of a Civic is about 1.8 meters. With the 48-degree lens on the thermal camera and the 10 pixels wide requirement for vehicle detection, the valid detecting distance is about 65 meters for a Civic. Since the testing distance range was between 10 and 50 meters, the target vehicles were within the detecting range during the experiment except when the target vehicles left the 48-degree camera view while turning.

Figure 4.2 and Figure 4.3 show the GPS distance and the output LiDAR distances from the system. When a target is 40 meters ahead of the testing vehicle, a normal size car with a height of 1.4 meters, such as target vehicle 2, should still be covered by LiDAR's 2-degree vertical angular resolution. However, as shown in Figure 4.3, the target 2 has a lot of missing data points when the distance is less than 40 meters. On the other hand, while using the same calibration data, the system can locate target 1

for most of the time because of the target's size. In order to avoid the coverage problem in Figure 4.3, the amount of LiDAR channels needs to be increased or the targets need to be larger.

The LiDAR distance estimator uses the average point in this experiment, and that causes the spikes which can be found on the LiDAR line in Figure 4.2. The GPS lines are not smooth in both figures due to the fact that the GPS base station was set to a wrong frequency. In the next experiment, the problems are fixed by lowering the LiDAR, using larger target vehicles, using the closest point for distance estimator and reconfiguring the GPS modules.

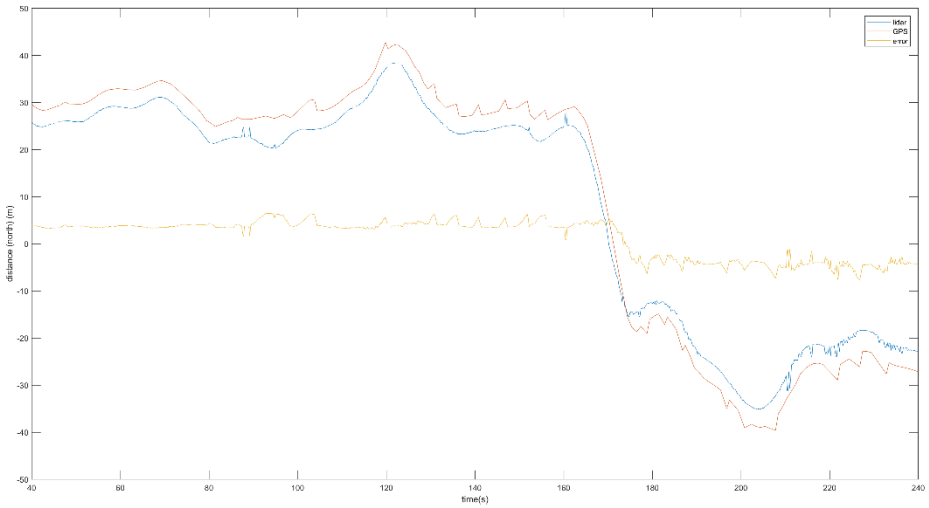

Figure 4.2 Experiment 1 target 1 global longitudinal distance

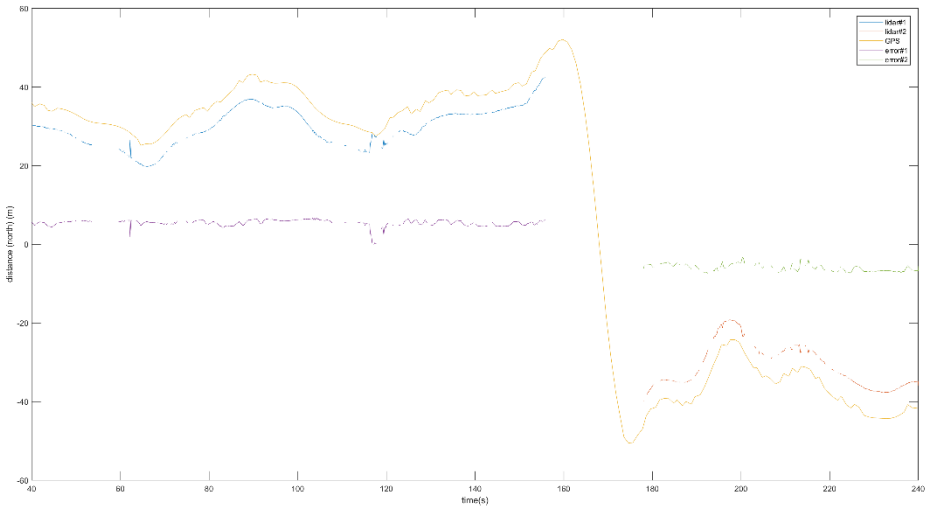

Figure 4.3 Experiment 1 target 2 global longitudinal distance

4.3. Experiment 2

The experiment was done during daytime with clear weather conditions. This experiment includes two target vehicles, and both targets are SUVs (Toyota RAV4) which is larger than the dimensions setting in the camera distance estimator, see section 3.5.1. The LiDAR and the thermal camera are located on the hood which is about 1.2 meters above the ground and are both tilted down for about 5 degrees. All the vehicles have GPS modules installed for verification.

The GPS modules used in this experiment are ublox ZED-F9Ps. The ublox ZED-F9P provides 0.01m position accuracy on both horizontal and vertical in real-time kinematic (RTK) positioning mode. For the RTK mode to work during the experiment, a GPS base station is placed nearby.

The distance between the observer and targets are in a range between 10 to 40 meters. All the vehicles are driven at a speed between 20 to 40 km/h. The target #1 has been in the camera view during the experiment, and the target #2 is not in the view between 75 to 84 seconds after the experiment started.

Figure 4.4, Figure 4.5, Figure 4.6 and Figure 4.7 are the results of the object tracking and distance estimate. The data is noisy before the observer starts moving because the heading of the vehicle is calculated by the GPS velocity. Since the GPS units are not placed at the rear end of the target vehicles, the error between GPS and LiDAR based distance estimate is expected. The longitudinal error for target #1 and #2 are about 2 and 2.5 meters respectively. The blue spikes on the LiDAR line is the camera distance estimate, and the estimate only appears when the LiDAR data is unavailable.

Figure 4.8, Figure 4.9, Figure 4.10, Figure 4.11, Figure 4.12 and Figure 4.13 show the result of the extended Kalman filter. The filter starts at 15 seconds. In the δ_x and δ_y plots for both targets, the filtered values follow the data from distance estimators well. The filtered velocity plot for both targets show the filtered data is more reliable than the data from the estimators.

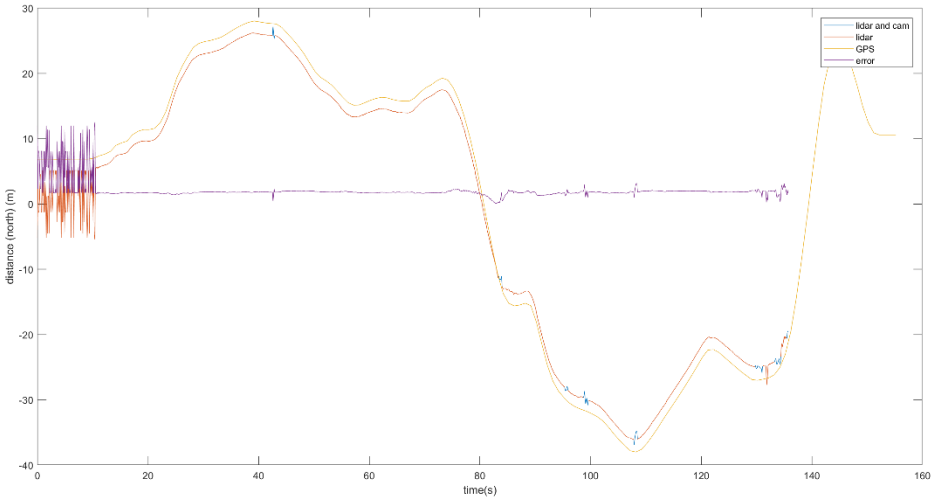

Figure 4.4 target 1 global longitudinal distance

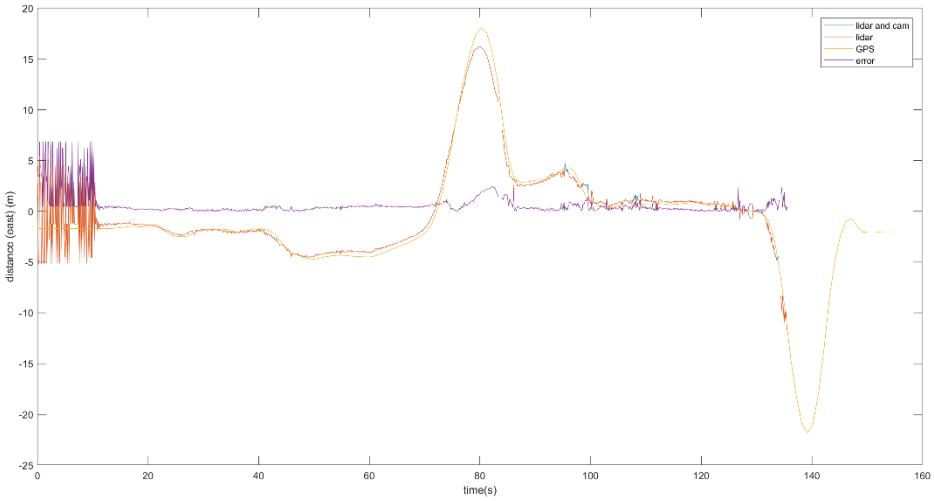

Figure 4.5 target 1 global latitudinal distance

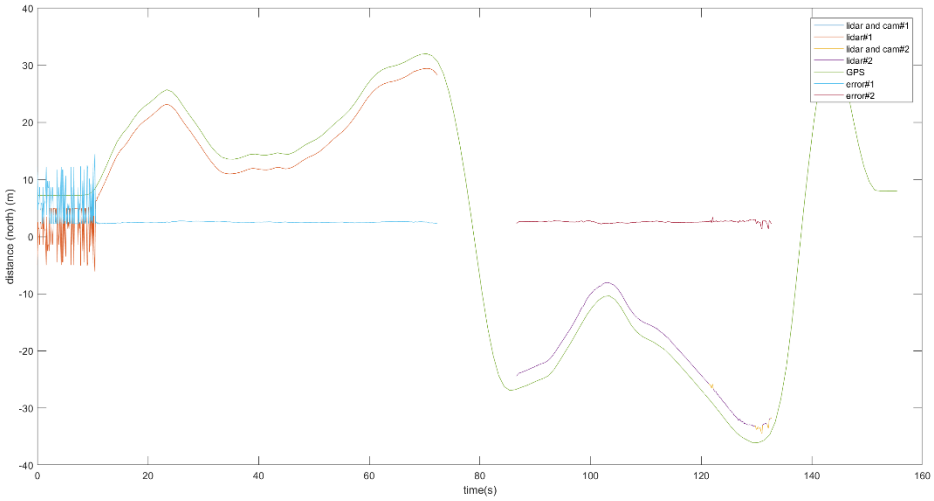

Figure 4.6 target 2 global longitudinal distance

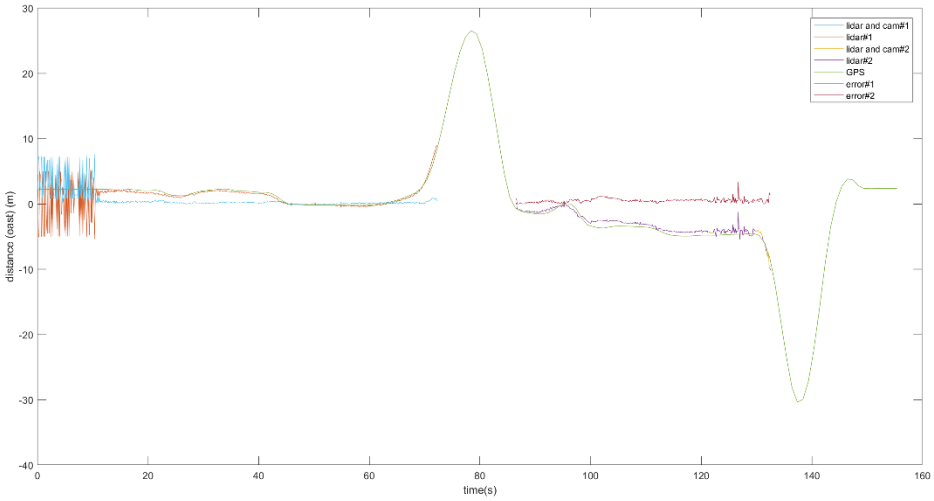

Figure 4.7 target 2 global latitudinal distance

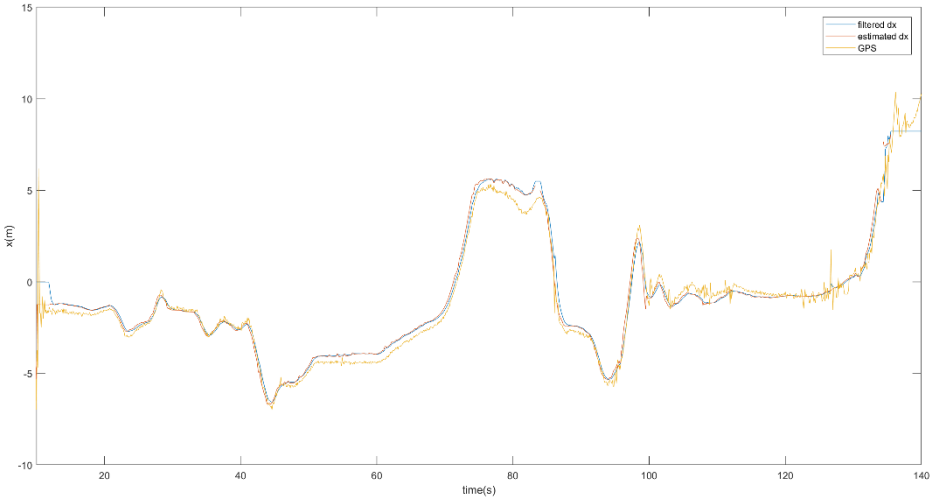

Figure 4.8 target 1 δx comparison

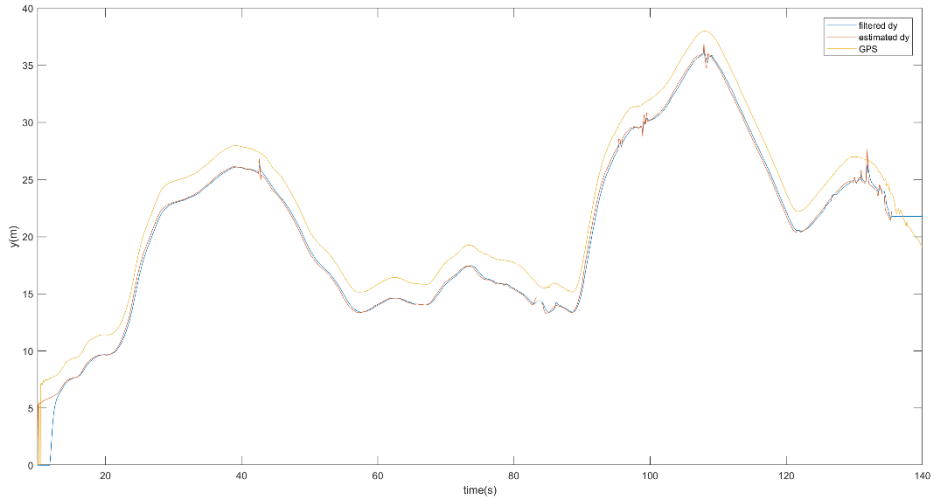

Figure 4.9 target 1 δy comparison

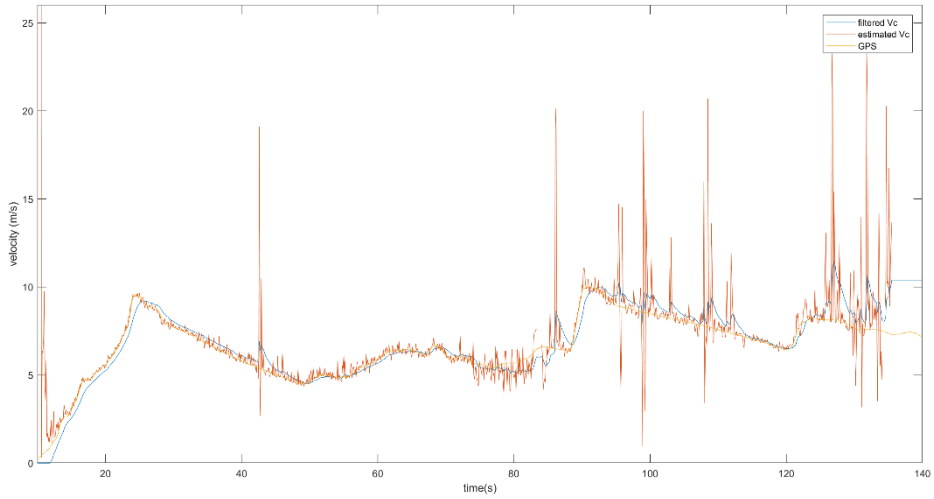

Figure 4.10 target 1 velocity comparison

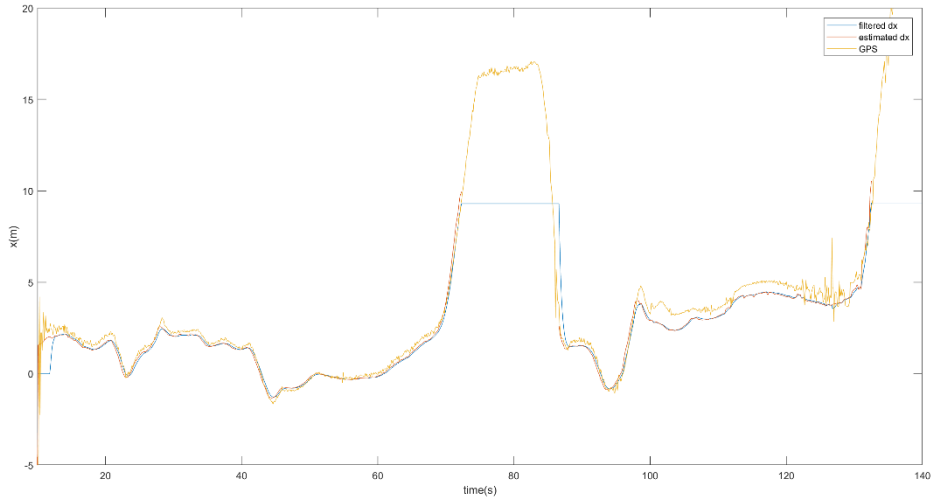

Figure 4.11 target 2 δx comparison

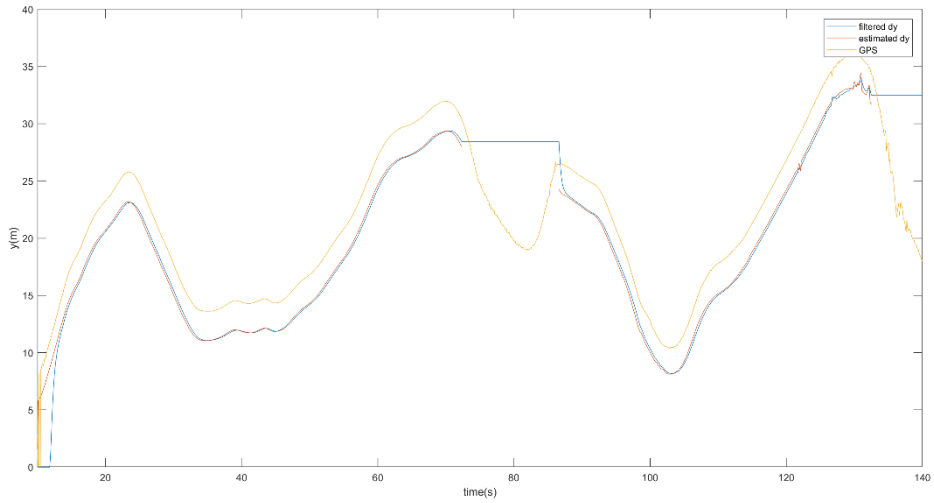

Figure 4.12 target 2 δy comparison

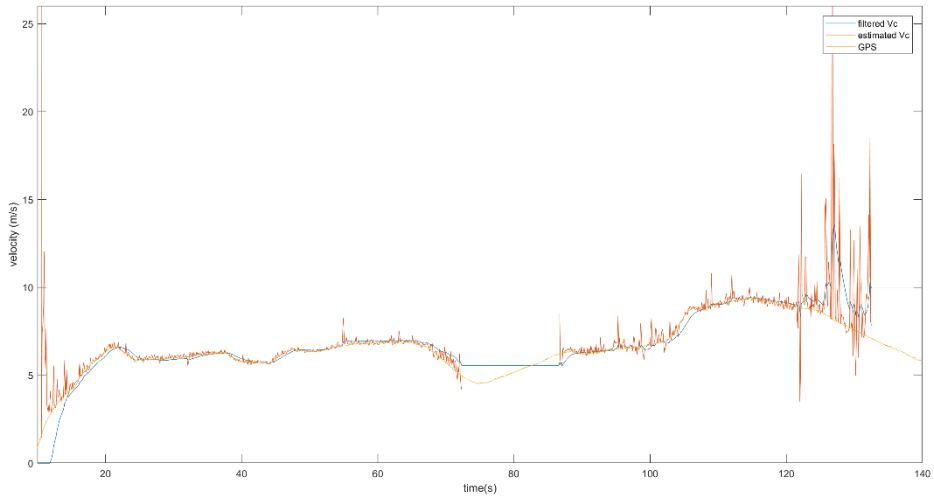

Figure 4.13 target 2 velocity comparison

5. CONCLUSIONS AND FUTURE WORK

A real-time vehicle tracking system that uses thermal images and LiDAR data was implemented in this thesis. The tracking system includes a ROS driver for the thermal camera used in the project, a calibration process to calibrate a LiDAR and a thermal camera, a tracking algorithm, a distance estimator using camera data and a sensor fusion algorithm to estimate the status of other vehicles. The tracking system does not require visible light because of the usage of a thermal camera; therefore, this system can help improve an AV's sensing system especially at nighttime or when the sun is in the camera view. The object detector works with only vehicles. Pedestrian and animal detection, which is not included in the neural network training dataset, should be added to the system for future work.

REFERENCES

Benjdira, B., Khursheed, T., Koubaa, A., Ammar, A., & Ouni, K. (2019). Car Detection using Unmanned Aerial Vehicles: Comparison between Faster R-CNN and YOLOv3. *2019 1st International Conference on Unmanned Vehicle Systems-Oman (UVS).* IEEE.

Dewan, A., Caselitz, T., Tipaldi, G. D., & Burgard, W. (2016). Motion-based detection and tracking in 3D LiDAR scans. *2016 IEEE International Conference on Robotics and Automation (ICRA).* Stockholm, Sweden: IEEE.

Gao, X.-S., Hou, X.-R., Tang, J., & Cheng, H.-F. (Volume: 25 , Issue: 8 , Aug. 2003). Complete solution classification for the perspective-three-point problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 930 - 943.

Girshick, R., Donahue, J., Darrell, T., & Malik, J. (2014). Rich feature hierarchies for accurate object detection and semantic segmentation. *The IEEE Conference on Computer Vision and Pattern Recognition* (pp. 580-587). IEEE.

Maturana, D., & Scherer, S. (2015). VoxNet: A 3D Convolutional Neural Network for real-time object recognition. *2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS).* Hamburg, Germany: IEEE.

Premebida, C., Monteiro, G., Nunes, U., & Peixoto, P. (2007). A Lidar and Vision-based Approach for Pedestrian and Vehicle Detection and Tracking. *Intelligent Transportation Systems Conference.* Seattle, WA, USA: IEEE.

Redmon, J., & Farhadi, A. (2018). YOLOv3: An Incremental Improvement.

Szarvas, M., Sakai, U., & Ogata, J. (2006). Real-time Pedestrian Detection Using.

Intelligent Vehicles Symposium. Tokyo, Japan: IEEE.