Vehicle Detection and Classification from a LIDAR equipped probe vehicle

Thesis

Presented in Partial Fulfillment of the Requirements for the Degree Master of Science in the Graduate School of The Ohio State University

By

Rong Yang, M.S.

Graduate Program in Electrical and Computer Engineering

The Ohio State University

2009

Thesis Committee:

Benjamin Coifman, Advisor

Charles Toth
ABSTRACT

Vehicle detection and classification is important in traffic analysis and management. Various sensing techniques can be used in this field, while most preceding work relies on sensors mounted along the roadway, this study develops a mobile platform using a LIDAR equipped probe vehicle to collect ambient traffic data while it drives. A vehicle detection method is developed to extract on-road vehicles from the background. The system employs two LIDAR sensors to measure the speed of the detected vehicles and then their length. A vehicle classification scheme is developed using length and height to sort the vehicles into six pre-defined categories. Ground truth data were generated from a developed GUI interface. Both the vehicle detection algorithm and and the vehicle classification algorithm are evaluated by comparing the LIDAR measurement with the ground truth data, with good result.
ACKNOWLEDGMENTS

I would like to thank my adviser, Dr. Benjamin Coifman, for his patient guidance and consistent support in my graduate study and research.

I also would like to thank Keith Redmill, for his contribution to Chapt. 2 in this thesis; Horst Kloeden, for his contribution to Sec. 3.1 in this thesis; Daniel Brandesky, for his help to generate the ground truth data.

Finally, I would like to thank my parents and boyfriend, Qing Xie, for their everlasting support to my pursing in research.
VITA

November 28, 1985 ...................... Born - Maanshan, China

2007 .............................................. B.S. Electronics Engineering,
Tsinghua University, China

2007-2009 ............................. Graduate Research Associate,
The Ohio State University.

FIELDS OF STUDY

Major Field: Electrical and Computer Engineering
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapters:</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. Introduction</td>
<td>1</td>
</tr>
<tr>
<td>2. Data Collection System</td>
<td>3</td>
</tr>
<tr>
<td>2.1 Data Collection System</td>
<td>3</td>
</tr>
<tr>
<td>2.1.1 LIDAR</td>
<td>3</td>
</tr>
<tr>
<td>2.1.2 DGPS</td>
<td>4</td>
</tr>
<tr>
<td>2.1.3 OBD</td>
<td>5</td>
</tr>
<tr>
<td>2.1.4 Camera</td>
<td>5</td>
</tr>
<tr>
<td>2.2 Campus Loop Route</td>
<td>7</td>
</tr>
<tr>
<td>3. Vehicle Detection in the LIDAR Image</td>
<td>8</td>
</tr>
<tr>
<td>3.1 Data Processing</td>
<td>9</td>
</tr>
<tr>
<td>3.1.1 3-Dimensional LIDAR Image</td>
<td>9</td>
</tr>
<tr>
<td>3.1.2 LIDAR image in Time Series</td>
<td>11</td>
</tr>
<tr>
<td>3.2 Region Segmentation</td>
<td>13</td>
</tr>
<tr>
<td>3.2.1 Region Segmentation on the T and Y dimensions</td>
<td>13</td>
</tr>
</tbody>
</table>
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1 Date of Runs</td>
<td>7</td>
</tr>
<tr>
<td>3.1 Vehicle Detection Error Statistics</td>
<td>31</td>
</tr>
<tr>
<td>4.1 Statistics of classification performance</td>
<td>47</td>
</tr>
<tr>
<td>4.2 Statistic of classification performance, row # is associated with the assigned class #, column # indicates the true class #, the last row is the total number of vehicles in each class</td>
<td>47</td>
</tr>
<tr>
<td>B.1 Link Information</td>
<td>58</td>
</tr>
</tbody>
</table>
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>The probe vehicle.</td>
<td>4</td>
</tr>
<tr>
<td>2.2</td>
<td>LIDAR scan Vertically</td>
<td>5</td>
</tr>
<tr>
<td>2.3</td>
<td>Concurrent Image Sample, (a) rear view ; (b) left side view; (c) front view</td>
<td>6</td>
</tr>
<tr>
<td>2.4</td>
<td>Campus Loop Route</td>
<td>6</td>
</tr>
<tr>
<td>3.1</td>
<td>Vehicle Detection Architecture</td>
<td>9</td>
</tr>
<tr>
<td>3.2</td>
<td>LIDAR Image: (a) an example of single LIDAR frame; (b) 3D LIDAR image, at $y \approx 5m$, one can see, from left to right, a box truck, sedan, city bus, SUV and sedan. Pedestrians are visible a few meters further back, then tress and finally buildings (points have been shaded to highlight the vehicles, as per Sec.3.4.1).</td>
<td>10</td>
</tr>
<tr>
<td>3.3</td>
<td>The same LIDAR Image as seen (a) spatially and (b) as a time series. Both view are looking straight down. As per Sec. 3.4.1, the vehicles have been highlighted with darker points for illustrative purpose.</td>
<td>12</td>
</tr>
<tr>
<td>3.4</td>
<td>Region Segmenting Architecture</td>
<td>13</td>
</tr>
<tr>
<td>3.5</td>
<td>Region Segmentation on the Y-T plane: (a) the projection of LIDAR points on the first two dimensions, (b) final segmenting result on Y-T plane, LIDAR points inside the regions are shown in the figure, (c) the converted discrete image from (a), (d) after excluding regions containing less than 3 pixels or that are too wide to be vehicles (wider than 5 meters)</td>
<td>15</td>
</tr>
</tbody>
</table>
3.6 Region Segmentation of the Z dimension: (a) the projection of one vehicle without anything above it, (b) the discretized image of (a), (c) the projection of one vehicle going under an overpass, and (d) the discretized image of (c) ............................... 19

3.7 Idealized Road Curb Model .................................................. 21

3.8 Line Segmenting of LIDAR scanning points ............................. 22

3.9 Example of Line Segmenting: (a) the original LIDAR points in 2D Cartesian coordinate system (the actual curb is circled for reference); (b) the line segmenting result of the LIDAR points; (c) the zoomed-in view of the circled area from (a), the points of potential curbs are marked with circles; (d) the zoomed-in view from (b) of the same area from (c), the potential curb line is plotted in solid. ....................... 23

3.10 Idealized Road Median Model .............................................. 24

3.11 Detect Road Median .............................................................. 24

3.12 Post-processing of road curb detection: (a) is the architecture of the post-processing algorithm; (b) one example of the detected road curb before post-processing; (c) the same link as (b) after post-processing . 28

3.13 Vehicle Detection Result (a) is the 2D region for the detected vehicles; (b) is the 3D LIDAR image, detected vehicles have been highlighted . 29

3.14 An example of Pedestrians crossing the road ............................. 30

3.15 Interface of GUI for ground truth ............................................ 31

3.16 Example of Detection Error: (a) The two circled vehicles are partially occluded and are only detected in one LIDAR, as a result of the occlusion they are not detected by the algorithm; (b) the circled non-vehicles are erroneously detected due to miss-detected road boundary . . . . . . . . 32

4.1 Vehicle Speed Measurement .................................................... 35
4.2 Measuring Travel Time from two LIDAR images. (a) scanning samples from both LIDARs of one vehicle, dashed lines are from side LIDAR #1, solid lines are from side LIDAR #2; (b) 2D image of the vehicle in side LIDAR #1; (c) 2D image of the vehicle in side LIDAR #2; (d) the optimal fit between the two LIDAR images. 37

4.3 LIDAR image of a detected vehicle: (a) the 2D LIDAR projection of side LIDAR #1, the rectangular shows the converted region; (b) the converted binary image of (a); (c) the 2D LIDAR projection of side LIDAR #2; (d) the converted binary image of (c). 37

4.4 Image Difference Curve: curve of the vehicle in Fig. 4.3; 38

4.5 Error estimation for Travel Time Measurement: $I_{m_f}(\cdot)$ and $I_{m_r}(\cdot)$ are defined in Eqn. 4.2, $T$ is the time interval between two adjacent LIDAR frames. 39

4.6 Selected link for speed measuring validation: (a) an aerial photo of part of the link (Google Maps); (b) GPS route of the selected link, the probe vehicle is moving North Bound. 41

4.7 Speed Error of parked vehicles, there is a total of 283 vehicles in the plot: (a) observed speed error versus theoretical speed error, (b) theoretical speed error versus the speed of probe vehicle (the relationship is also represented by Eqn. 4.6), (c) histogram of $V_{obj}-V_{err}$ 42

4.8 Two groups of LIDAR frames: Group 1 includes frames collected between the front bumper and roughly the center of the vehicle; Group 2 includes frames collected between roughly the center of the vehicle and the rear bumper of the vehicle. 44

4.9 Height versus Length distribution of 927 vehicles. 45

4.10 $H_1$ versus $H_2$ distribution of vehicles in group 2 (889 vehicles), the markers are the same as Fig. 4.9 48

4.11 an example of misclassified Pickup with full bed: (a) a picture from the side camera; (b) 3D LIDAR image. 49

A.1 LIDAR resolution in the Cartesian coordinate system (a) is the resolution on the Y-dimension, (b) is the resolution on the Z-dimension 54
CHAPTER 1

INTRODUCTION

Vehicle detection and classification are critical for traffic management. Various sensing techniques are used for vehicle detection, including loop detectors and video detectors. Almost all of the vehicle detection and classification tools in conventional practice are based on fixed sensors mounted in or along the right of way ( [1], [2] ). This research investigates the feasibility of using vehicles mounted sensors instead of the fixed detectors. Thereby enabling alternative monitoring techniques such as the moving observer ( [3], [4] ) that were previously too labor intensive to be viable.

In this research, LIDAR (LIght Detection and Ranging) is used to monitor the ambient traffic around a probe vehicle. The LIDAR does not distinguish between vehicles and non-vehicles, as such, much of this work is focused on segmenting the vehicles from the background. successfully detected by our vehicle detection algorithm. A vehicle classification method based on the length and height of the detected vehicles is developed to classify the vehicles into six categories. The classification accuracy is over 90% for the ground truth data set.

The rest of this thesis is organized as follows. First, the data collection system will first be introduced in Chpt. 2. Then in Chpt. 3, the vehicle detection algorithm
will be presented. Chpt. 4 will describe our method to classify the detected vehicles. Finally Chpt. 5 presents conclusions and future extensions of this study.
CHAPTER 2

DATA COLLECTION SYSTEM

2.1 Data Collection System

This research employs a van equipped with multiple sensors and a data acquisition system (Fig. 2.1). The resulting data are used to detect and classify other vehicles on the roadways. The system records six different sensor data streams, the first five are used for automated detection and positioning: LIDAR (Light Detection and Ranging), DGPS (Differential Global Positioning System), RADAR (Radio Detection and Ranging), Gyroscope, and OBD (On-Board Diagnostics) Data Bus. The sixth, used for validation, consists of three cameras mounted on the probe vehicle to catch the front, left side, and rear views. The various data streams arrive asynchronously, so all of the data are time stamped to one common clock. The RADAR and Gyroscope are not used in this research, each of the remaining four sensor system will be discussed blow.

2.1.1 LIDAR

There are two LIDAR sensors mounted on the left side of the Van (the driver’s side). These two LIDAR sensors are at a height of about 2.05 meters relative to the
ground and are 1.405 meters apart from each other. They scan vertically, sweeping 180 degrees, from straight up to straight down at 0.5 degree increments (Fig. 2.2). The maximum range of the two driver side LIDAR sensors is 80 meters, with a resolution of 0.25 centimeters. The scanning frequency of the two side LIDAR sensors is about 37 Hz.

2.1.2 DGPS

The DGPS receiver mounted on the van is a Trimble AG312 GPS receiver with Omnistar VBS correction. It reports the fix of the probe vehicle at a frequency of 5 Hz. When VBS mode is active, the positioning error should be less than 1 meters 95% of the time. The information provided in each report contains: time stamp (seconds since midnight), GPS time (seconds since the beginning of the week), longitude and latitude (degrees), altitude (meters), speed (meters/sec), heading direction (radians) and status of the receiver. Information that is used in this research from the DGPS sensor includes: time stamp, longitude and latitude.
2.1.3 OBD

The OBD captures information from the probe vehicle’s Engine Control Unit. The feature most relevant to the work is the measurement of the probe vehicle’s speed at a rate of about 2.5 Hz. The resolution of the measurement is about 1 kilometer/hour.

2.1.4 Camera

For the purpose of validation, three cameras are mounted on the Van. The images collected by these three cameras provide a multi-view of the environment around the probe vehicle. The cameras capture images at a rate of 10 Hz. The size of each image is 320 x 240 pixels. Fig. 2.3 is a sample of the images captured concurrently from the three cameras.
Figure 2.3: Concurrent Image Sample, (a) rear view; (b) left side view; (c) front view

Figure 2.4: Campus Loop Route
<table>
<thead>
<tr>
<th>Run #</th>
<th>Date</th>
<th>Start time</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dec.4, 2008</td>
<td>7:57</td>
</tr>
<tr>
<td>2</td>
<td>Dec.4, 2008</td>
<td>14:01</td>
</tr>
<tr>
<td>3</td>
<td>Dec.8, 2008</td>
<td>13:57</td>
</tr>
<tr>
<td>4</td>
<td>Jan.22, 2009</td>
<td>14:00</td>
</tr>
<tr>
<td>5</td>
<td>Feb.26, 2009</td>
<td>14:02</td>
</tr>
<tr>
<td>6</td>
<td>Mar.5, 2009</td>
<td>14:02</td>
</tr>
<tr>
<td>7</td>
<td>Mar.12, 2009</td>
<td>13:55</td>
</tr>
</tbody>
</table>

Table 2.1: Date of Runs

2.2 Campus Loop Route

The driver of the probe vehicle is instructed to drive the vehicle along a pre-designed route, shown in Fig. 2.4. The route covers local roads on and around the OSU campus with an approximate length of 23 miles. Half of the route follows a campus bus route (most of the tour from Woody Hayes Ave and South). The Data collected for this research consists of seven runs collected between Dec. 2008 and Mar. 2009, as listed in Table 2.1.
CHAPTER 3

VEHICLE DETECTION IN THE LIDAR IMAGE

The LIDAR sensors provide a rich view of the world around the probe vehicle. This raw view, however, does not make any distinction between vehicles and non-vehicles data points. So this chapter presents our methodology for segmenting the vehicles from the background and from each other. Fig. 3.1 shows the architecture of the vehicle detection system. Data from multiple sensors is first processed and integrated to generate a 3D LIDAR image. Then, region segmenting extracts all the objects that are potentially vehicles. At the same time the road boundary is detected and extracted from the LIDAR image if it is within the field of view. Based on the road boundary detection result, the objects that are within the on-road area are extracted for further analysis. The details of vehicle detection algorithm will be discussed in three sections. In Section 3.1, important properties of the 3D LIDAR image will be discussed. Then in Section 3.2, region segmentation based on a region growing algorithm will be introduced and applied to the 3D LIDAR image. In Section 3.3, an algorithm will be developed to detect the road boundary, so that the regions on the road can be used to differentiate vehicles from the background. Finally, the performance of the developed vehicle detection algorithm will be presented in Section 3.4.
3.1 Data Processing

3.1.1 3-Dimensional LIDAR Image

As described in Section 2.1.1, each of the two side LIDAR sensors effectively serves as a vertical planar scanner. Because the two LIDAR sensors are 1.405 meters apart, any moving target will appear differently in the two views, thereby allowing for speed measurement. The sensor readings deliver distance of 361 points in a 2-Dimensional Polar coordinate system. One LIDAR frame is a set of these 361 2D points\(^1\). Knowing the range and direction of each point, the LIDAR frame can be transformed into a 2D Cartesian coordinate system, Fig. 3.2(a) shows an example.

\(^1\)For the purposes of this work, the 2D Frame is treated as if it came from a single instant. In reality, the LIDAR sweeps through the scan over time while the probe vehicle is moving, thereby skewing the points from the assumed vertical plane. The impact of this distortion, however, is negligible.
frame. The Y-axis is the horizontal distance of the points from the LIDAR and the Z-axis is the height of the points off the ground. The two dashed lines in Fig. 3.2(a) intersect at the position of the LIDAR in the Cartesian coordinate system, which is $(0, 2.05)$.

Figure 3.2: LIDAR Image: (a) an example of single LIDAR frame; (b) 3D LIDAR image, at $y \approx 5m$, one can see, from left to right, a box truck, sedan, city bus, SUV and sedan. Pedestrians are visible a few meters further back, then tress and finally buildings (points have been shaded to highlight the vehicles, as per Sec.3.4.1).

If we define the heading direction of the probe vehicle as the X-axis, a 3D LIDAR image can be constructed from the successive 2D frames. Given the speed of the probe vehicle, the distance between two adjacent frames can be calculated via Eqt. 3.1

$$X_n = \sum_{i=1}^{n-1} \Delta T \cdot V_i$$  \hspace{1cm} (3.1)

Where, $V_i$ is the speed of the probe vehicle between the $i^{th}$ frame and $(i-1)^{th}$ frame, $\Delta T$ is the time gap between two adjacent frames, $(X)_n$ is the traveling distance of the probe vehicle at the $n^{th}$ frame. Provided that the probe vehicle does not make any
significant turns, Eq. 3.1 yields an adequate X-coordinate of the 3D LIDAR image for a road section. Fig. 3.2(b) shows an example of the 3D LIDAR image for one road link on Woodruff Ave.

3.1.2 LIDAR image in Time Series

In the 3D space described in the previous section (Fig. 3.2(b)), the information in the X-dimension is extracted from the speed of the probe vehicle and the other dimensions come from the 2D planar scanning frames. In an urban area, the speed of the probe vehicle can change rapidly, which will distort the vehicles seen in the LIDAR. The distance between LIDAR frames is determined by the speed of the probe vehicle. If the probe vehicle changes speed very frequently, the distance between LIDAR frames will also vary quickly. Therefore, using traveling distance as the third dimension of the 3D LIDAR image will lead to the following two consequences:

- inconstant frame density over the X-dimension
- When the probe vehicle fully stops, the apparent length of moving objects will shrink to zero.

As will be addressed in the next section, a threshold distance is used to decide whether two points belong to one object. The inconstant frame density over the X-dimension makes it difficult to choose a global threshold for the segmenting process. In the extreme case when the probe vehicle is fully stopped, the distance between frames will be zero. The moving objects seen during this period fall at one location. These vehicles would largely disappear from a plot like Fig. 3.2(b) with the X-dimension based on a physical distance.
Instead, if one plots the time series with the X-dimension a function of time, the frame density will not be affected by the motion of the probe vehicle. In the case when the probe vehicle is fully stopped, the moving objects will still be manifest as 3D objects in the time series of LIDAR frames. Therefore, the above two problems will be avoided. Consider Fig. 3.3(a), with a distance based X-dimension, the probe vehicle stops at 31 m and 16 vehicles appear as a single vertical line, as circled in the figure. Changing the X-dimension to time, Fig. 3.3(b) shows the same scene. The probe vehicle was stopped from 7 to 54 seconds, and the 16 passing vehicles are now distinct from one another. In either case, the appearance of the target vehicle’s points will be a function of its speed and the probe vehicle’s speed. As will be discussed below, in most cases, these distortion can be accounted for. In the following sections, the 3D image is processed in time series.
3.2 Region Segmentation

The objective of our region segmentation is to extract the boundary of every single object that may be a vehicle from the 3D LIDAR image. Two main segmenting steps are enumerated in Fig. 3.4. The first step of segmenting will provide the boundary of the T-dimensional (Time stamp of LIDAR frame) and the Y-dimensional (lateral distance to the probe vehicle). Then, in the second step of segmenting, information from the Z-dimensional (Height) will be analyzed to differentiate between vehicles and non-vehicles. The boundary of the Z-dimensional will be decided for each possible vehicle region extracted in the first segmenting step. The following region growing algorithm is used in both steps. The segmenting result for each potential vehicle is in the form of 3D boundary box.

3.2.1 Region Segmentation on the T and Y dimensions

The key assumption of region segmentation is that points belonging to the same object should be close to each other. In the LIDAR Image (e.g., Fig. 3.2(b)), almost
all of the objects that rise above the ground are still connected by the ground points. Therefore, in order to separate the objects, it is necessary to exclude the points on the ground. A threshold height is used to differentiate the ground points from non-ground points. For this work, it is assumed that vehicles are on the ground and that their height should be larger than 1 meters. Hence, there must exist points of each vehicle between the height of 0.5 meters and 3 meters. For the purpose of finding all the potential vehicles in the 2D projection, only the LIDAR points between the two height bounds will be projected onto the Y-T plane. After projecting the 3D points onto the Y-T plane, a set of clouds of the LIDAR points should be evident, e.g., Fig. 3.5(a). In most cases, points belonging to a given object should be closer to each other than points from different objects. In the projected image each individual object appears as one cloud of LIDAR points.

As can be seen in Fig. 3.5(a), there are several clouds of points, which represent different objects in the 3D LIDAR image. After obtaining the 2D projection image, the next step is to find the boundary of each cloud. Since the number of objects in one image is unknown and it may vary from image to image, a region growing algorithm is used to segment regions. In conventional image processing the basic idea of region growing algorithm is to partition the image into several regions such that the pixels in the same region will have the similar color. Unlike a conventional digital image whose information is presented by the color of the pixels, the 2D LIDAR projection (Fig. 3.5(a)) express the information through the distribution of LIDAR points. In order to apply the technique, the 2D LIDAR projection is first discretized (Fig. 3.5(b)), where each pixel is assigned a value based on whether there is one or more LIDAR

2Given the fact that the height of ground is not exactly 0 meters, in practice, 0.5 meters is chosen as the lower bound to make sure most of the ground points are excluded.
Figure 3.5: Region Segmentation on the Y-T plane: (a) the projection of LIDAR points on the first two dimensions, (b) final segmenting result on Y-T plane, LIDAR points inside the regions are shown in the figure, (c) the converted discrete image from (a), (d) after excluding regions containing less than 3 pixels or that are too wide to be vehicles (wider than 5 meters)
points in the associated area in the 2D LIDAR projection, as follows. The Y-T plane is divided into several small rectangular areas, each of which will be associated with one pixel in the discretized image. Equation 3.2 shows the relationship between the Y-T plane and the discretized image.

\[(i, j) \leftarrow \{(t, y) | Y_{max} - j \cdot \Delta y \leq y \leq Y_{max} - (j - 1) \cdot \Delta y, t = t_i\}\] (3.2)

where, \((i, j)\) represents one pixel in the digital image; the set on the right side of the arrow is the corresponding area in the Y-T space; \(Y_{max}\) is the converting range of the Y-dimension, which is set to 50 m; \(\Delta y\) is the width of the rectangular area, \(t_i\) is the time stamp of the corresponding LIDAR frame. Then, the value of each pixel is assigned to 1 if there are LIDAR points in the corresponding area in Y-T space; 0 otherwise. The most important part of the discretization process is how to select \(\Delta y\), the step size of the Y-dimension. Ultimately it is a trade off between two competing objectives

- points belonging to one cluster should be converted to connected pixels
- points from different objects (different clouds of LIDAR points) should be converted to disconnected pixels in the binary image

The distance between the LIDAR points belonging to the same objects is mainly decided by the scanning resolution of the LIDAR. A detailed discussion of LIDAR scanning resolution is presented in App. A. Based on the discussion, in order to the satisfy the first converting criterion, \(\Delta y \geq 0.4 m\). On the other hand, vehicles in adjacent lanes could be very close to each other. The \(\Delta y\) should be as small as possible to keep separate vehicles apart. So in our algorithm, 0.4 m is chosen as the converting step size. Fig. 3.5(c) shows the discretized image.
The region growing algorithm is applied to the discretized image. In this process, pixels with the value of 1 (1-pixel) are segmented into several regions. The process starts with a randomly selected 1-pixel, which is called seed pixel. The seed pixel is initialized as the first pixel of one region. Then, the following step will be iterated until all the connected pixels have been visited:

1. Find all the boundary pixels that are on the edge of the region;

2. For each boundary pixel, find the neighbor pixels that have not been visited. If the pixel is a 1-pixel, add it to the region.

The iteration stops when there is no more 1-pixel added to the region. Then, an unvisited 1-pixel will be selected as the new seed pixel. The above steps will be repeated. The 'region growing' will be continued until all the pixels have been visited.

After applying region growing algorithm, all the regions containing LIDAR points will be extracted from the binary image. The discretized image is then cleaned by discarding very small regions (smaller than 3 pixels) and regions that are wider (Y-dimension) than 5 meters (it is assumed that vehicles are less than 5 meters wide).

Fig. 3.5(d) shows the result with the boundaries of each region for the ongoing example in the discretized plane. Finally, the regions are projected back to the original Y-T space, e.g., Fig. 3.5(b).

### 3.2.2 Region Segmentation on the Z dimension

The Y-T segmenting made no attempt to explicitly find the top of the given target. It just assumed that all vehicles of interest would have points between 0.5 meters and 3 meters high. Now to find the top of the vehicle, for each region found in the Y-T
plane, the points are now projected into the Z-T plane with a higher threshold, i.e., between 0.5 meters and 6 meters. In most cases, there will be only one object in each region (e.g., Fig. 3.6(a)). However, it is still possible that there are multiple objects within one region because in the first segmenting step information of the height is ignored. Objects overhanging other objects would be projected onto the same regions as that of the objects underneath them. For example, when the vehicle is going under an overpass, both the vehicle and the overpass above it will be seen in the projection (e.g., Fig. 3.6(c))

Once more, the 2D projection will be discretized. The converting step size is decided based on Eqn. A.6, (App. A provides a detailed derivation).

\[
\Delta z = 0.0087 \cdot y_n \cdot (1 + \frac{1.44}{y_n^2})
\]  

(3.3)

Then, region growing algorithm is applied to the discretized image, following the same process of the first segmenting step. Since the vehicles should be on the ground, regions that do not intersect the bottom of the discretized image will be discarded. Everything else is a possible vehicle and will be kept for further analysis. At the same time, the region boundary of Z-dimension or height for these objects are provided by the segmenting result.

### 3.3 Road Boundary Detection

The ultimate objective of our region segmenting is to find all of the on-road vehicles. The most effective method to distinguish the on-road area from off-road area is to know the location of the road boundary in each LIDAR image. In this section, a method is developed to detect the road boundary in LIDAR image. First, when
Figure 3.6: Region Segmentation of the Z dimension: (a) the projection of one vehicle without anything above it, (b) the discretized image of (a), (c) the projection of one vehicle going under an overpass, and (d) the discretized image of (c)
visible, the road curb is detected in each LIDAR frame. Then a post-processing step is conducted to investigate the relationship among the detected curb location in successive LIDAR frames. The performance of the method on different road geometries is then be presented.

3.3.1 Curb Detection in each LIDAR frame

Fig. 3.7 is an idealized model of a road curb in one LIDAR frame. The 2D LIDAR scanning points can be simplified as three lines. As shown in Fig. 3.7, those three lines represent road, road curb and sidewalk respectively. Several features of the road curb line are assumed for this research, as listed below.

- It is nearly vertical slope;
- The line segments before and after it should be roughly horizontal slope;
- It should be close to $0$ m off the ground;
- The height should not be too small, i.e. a typical height would be 4 inches (10 centimeter)

The road curb detection algorithm is based on the above properties. Three steps are involved in the detection algorithm.

1. The LIDAR scanning points are segmented into line segments of constant slope.
2. All line segments that are potentially road curb will be extracted
3. Curbs of road median will be differentiated from real road
The purpose of line segmenting is to divide the LIDAR scanning points into several line segments. In the 2D LIDAR frame, a line could be represented by the radius of its tangent circle and the direction of its tangent points, \((R_{tg}, -90^\circ + \alpha_{tg})\) (Fig. 3.8(a)). At the same time, in the Polar coordinate system, a point is represented by its distance to the LIDAR and its direction, \((r_{pt}, \theta_{pt})\). The sufficient condition for a point to be on a line is given by Eqn. 3.4.

\[
r_{pt} \cdot \sin(\alpha_{tg} - \theta_{pt}) = R_{tg}
\]

(3.4)

In each LIDAR frame, the line segmenting process starts from the point at the scanning direction of about \(-80^\circ\). Since objects should not be too close to the probe vehicle, the first several points detected by the LIDAR should be ground points. In the line segmenting process, the first 10 points (between \(-80^\circ\) and \(-75^\circ\)) will be initialized as the first line. The line segmenting process then continues ‘counterclockwise’ (relative to Fig. 3.8(b)) to check each point with Eqn. 3.4. If Eqn. 3.4 is satisfied, most of the time, LIDAR has no detecting return in the range between \(-90^\circ\) and \(-80^\circ\). Thus, this area is skipped in the line segmenting process.
a point will be added into the current line segment and the parameters for the current line segment will be re-calculated. Otherwise, a new line will be started. The line segmenting process will be continued until all the points under the height of 1 meter have been checked.

After segmenting the points into lines, the next step is to find all the potential curbs based on the four necessary features of road curb described above. The explicit criteria applied to extract the potential road curb lines are listed blow. Here, \( L_n \) is used to represent the current line; \( L_{n-1} \) and \( L_{n+1} \) are, respectively, the line segments right before and after the current line segment (neighbor line segment).

- \( \| \text{slope}(L_n) \| > 0.2 \), the minimum slope required for the potential curb line
- \( \| \text{slope}(L_{n-1,n+1}) \| < 0.12 \), the maximum slope allowed for the two neighbor lines.
Figure 3.9: Example of Line Segmenting: (a) the original LIDAR points in 2D Cartesian coordinate system (the actual curb is circled for reference); (b) the line segmenting result of the LIDAR points; (c) the zoomed-in view of the circled area from (a), the points of potential curbs are marked with circles; (d) the zoomed-in view from (b) of the same area from (c), the potential curb line is plotted in solid.

- $\text{median}\{z_{pi}, pi \in L_n\} - \text{min}\{z_i\} < 0.2$ (meters), $\text{min}\{z_i\}$ represents the height of the ground in the current frame, 0.2 is the maximum height allowed for the curb line segment

- $\text{max}\{z_{pi}, pi \in L_n\} - \text{min}\{z_p, pi \in L_n\} > 0.05$ (meters), 0.05 is the minimum height required for a road curb line segment

Line segments satisfying all four of the above conditions will be extracted as the potential road curb. In Fig. 3.9, only one line satisfies all the four conditions. The extracted potential road curb is highlighted in Fig. 3.9(c) and Fig. 3.9(d). Sometimes there will be more than one potential curb extracted from the segmented lines, mainly due to the existence of a road median. The idealized model of road median is shown in Fig. 3.10.
Figure 3.10: Idealized Road Median Model

Figure 3.11: Detect Road Median
The feature we use to distinguish a road median curb from a road boundary curb is the shadow behind the road median curb (see Fig. 3.10). In order to detect the shadow, LIDAR points from multiple frames are used. First, all the ground points \( z_i < 0.5 \text{ meters} \) in the neighboring frames\(^4\) are projected onto the Y-T plane. Fig. 3.11(a)-(b) repeat the curb detection from Fig. 3.9. As can be seen in Fig. 3.11(c), across many successive frames there is a gap at the location of \( y \simeq 9 \) meters, which is caused by the shadow of the road median. In order to detect the gap mathematically, the density of ground points along Y-dimension is analyzed. If there is no road median, the density of LIDAR points on the ground should follow Eqt. 3.5 (derived from Eqt. A.3, where \( z_n \simeq 0. \))

\[
Den(y) = \frac{1}{\Delta y} = \frac{1}{0.0178 + 0.0042 \cdot y^2} 
\]

(3.5)

where, \( Den(y) \) is a function of \( y \), if the surface were flat. As \( y \) increases, the density of the points on the ground will decrease due to larger and larger scanning angles. In contrast to this ideal density function, given the distribution of all the observed ground points in the neighboring frames (see Fig. 3.11(c)), the empirical density function can be calculated. Both the ideal density curve and the empirical density curve are shown in Fig. 3.11(d). Note that both curves have been normalized so that they start at the value of 1. The difference between the empirical density and the ideal density provides sufficient information about the height change of the surface. At the location of the first road curb \( (y \simeq 5 \text{ meters}) \), there is a spike in the empirical density curve. The spike is caused by the the abrupt height increase of the road curb.

\(^4\)in this study, we made no attempt to optimize the number of frames used in this step. Empirically, frames within the previous 15 and next 15 frames are chosen as the neighboring frames, which yields satisfactory results in this study.
Then, at the location of gap \((9 \text{ meters} < y < 10 \text{ meters})\), the observed density drops almost to zero, causing a local minimum. This local minimum is associated with the shadow of the road median (see Fig. 3.10).

In order to differentiate the road median curb from the road boundary curb, the following steps are applied to all the extracted road curbs.

- find all the ground points \((z_i < 0.5 \text{ meters})\) in the neighboring frames.
- calculate the density function along Y-dimension, based on the empirical LIDAR points distribution.
- compare the empirical density function with the ideal density function (Eqn. 3.5): if there exists local minimum after the road curb, the curb is considered to be road median curb; otherwise, it is considered the road boundary curb.

### 3.3.2 Post-process of Road Boundary Detection

Section 3.3.1 described the details of road curb detection in each LIDAR frame. The algorithm is vulnerable to the abrupt change of road features due to the following three reasons:

- Occlusion of the road median can preclude the detection of the gap behind the rising curb of the road median. Hence the rising curb of the road median will be detected as road boundary curb.

- When road median exists, the road is comparatively wider. In such case, the road boundary curb often becomes hard to detect due to low resolution far away from the LIDAR.
• Some road boundary curbs have similar shape of the road median, i.e., a drop-off after the curb. Such road boundary curbs may be mis-detected as road median.

To mitigate these problems, we post-process the detected curbs in each frame individually to investigate the relationship between the curbs across frames. The post-processing algorithm is based on the following assumptions.

1. Most of the time, both the road boundary and the road median will gradually change in terms of lateral distance to the probe vehicle. In other words, there should not be any abrupt jump of the lateral distance.

2. When a road median curb is detected in one LIDAR frame, a road boundary curb should exist in the same frame.

These two assumptions provide the theoretical foundation for the post-process algorithm. The architecture of the algorithm is shown in Fig. 3.12(a). First, the detected curbs are distributed into groups based on the distance in the Y dimension and proximately in the X-dimension. The segmenting is conducted in a 2D space (see Fig. 3.12(b)), instead of using LIDAR frame time stamp, the distance along the route is chosen to be one dimension of the 2D space to better segment the curb points (since curbs could remain static over time). Then, the curb type of each group will be checked. Based on the first assumption, curb type should not change within one group. The inconsistent curb type must be caused by curb-type classification error. For those groups with two curb types (i.e., road boundary and road median), the portion of each curb type will be calculated. The curb type of the group is decided by the majority type. In other words, if more than 50% of the curb points are road
median, then all the curbs in the same group will be classified as road median; and vice versa. Third, gaps shorter than 1 meter within each group will be interpolated. This step is applied to road boundary and road median alike. In the end, all small curb groups (less than 2 meters) with abrupt lateral distance change from the neighboring curbs will be discarded.

### 3.4 Vehicle Detection Performance

#### 3.4.1 Find on-road vehicles

In Sec. 3.2, each single object that is possibly vehicle has been extracted from the 3D LIDAR image. Then in Sec. 3.3, the on-road area is distinguished from off-road area based on the detected road boundary. Taking the union of the two sections,
objects on the road area are extracted as potential vehicles. Fig. 3.13 shows the vehicle detection result on the same link as Fig. 3.2(b). The two plots in Fig. 3.13 show the same Y-T region: (Fig. 3.13(a)) 2D region of the detected and (Fig. 3.13(b)) the 3D LIDAR image.

Figure 3.13: Vehicle Detection Result (a) is the 2D region for the detected vehicles; (b) is the 3D LIDAR image, detected vehicles have been highlighted

On the other hand, there might be non-vehicles objects on the road area, e.g, pedestrians going cross the road or road construction cones (Fig. 3.14 shows an example). Just by using the road boundary, these objects can not be filtered out. However, by conducting a second pass based on the size of the objects, most of these non-vehicles can be excluded. The second pass investigates the length of the objects (the method to measure the length of the object is explained in Sec. 4.2). Vehicles are assumed to be longer than 1 meter. Hence, any objects with length smaller than 1 meter will be filtered out. After the above two passes of filtering, on-road vehicles are extracted from the 3D LIDAR image.
3.4.2 Validation for Vehicle Detection

A GUI program is developed to display the LIDAR detection result as well as to generate ground truth data. Fig. 3.15 shows the interface of the GUI program. The developed vehicle detection algorithm has been tested on six pre-defined links. The detection result on these six links is provided in Table 3.1.

The total miss-detection rate over all of six links is 3.3% and non-vehicle detection rate is 3.6%. The false detection of non-vehicles is mainly caused by erroneously including non-road areas within the detected road boundaries. The miss-detection of vehicles is mainly caused by partial occlusion. Fig. 3.16 shows an example of both false detected non-vehicle and miss-detected vehicles.

Detailed information of these six links is included in App. B
Figure 3.15: Interface of GUI for ground truth

<table>
<thead>
<tr>
<th>Link #</th>
<th>Total # of vehicles from ground truth data</th>
<th># of correctly detected vehicles</th>
<th># of mis-detected vehicles</th>
<th># of non-vehicles erroneously labeled vehicles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>343</td>
<td>335</td>
<td>8</td>
<td>14</td>
</tr>
<tr>
<td>2</td>
<td>235</td>
<td>230</td>
<td>5</td>
<td>0</td>
</tr>
<tr>
<td>3</td>
<td>879</td>
<td>842</td>
<td>37</td>
<td>20</td>
</tr>
<tr>
<td>4</td>
<td>439</td>
<td>417</td>
<td>22</td>
<td>40</td>
</tr>
<tr>
<td>5</td>
<td>144</td>
<td>138</td>
<td>6</td>
<td>3</td>
</tr>
<tr>
<td>6</td>
<td>375</td>
<td>374</td>
<td>1</td>
<td>10</td>
</tr>
<tr>
<td>total #</td>
<td>2415</td>
<td>2336</td>
<td>79</td>
<td>87</td>
</tr>
</tbody>
</table>

Table 3.1: Vehicle Detection Error Statistics
Figure 3.16: Example of Detection Error: (a) The two circled vehicles are partially occluded and are only detected in one LIDAR, as a result of the occlusion they are not detected by the algorithm; (b) the circled non-vehicles are erroneously detected due to miss-detected road boundary
In Chap. 3, target vehicles that are within the on-road area have been detected and extracted in each side LIDAR sensor respectively. With the two side LIDAR sensors, it is possible to measure the speed of the detected target vehicles relative to the probe vehicle. Furthermore, based on the measured speed, the length of the target vehicles could be calculated. In addition, side LIDAR sensor provides rich view of target vehicles' shape, from which height could be extracted for the target vehicles. A length-height based vehicle classification rule is then developed to divide vehicles into six categories. Ground truth data is generated to validate the effectiveness of the developed vehicle classification algorithm.

The rest of this chapter consists of two sections. In Sec. 4.1, an algorithm is developed to measure the speed of the detected target vehicles and the corresponding measuring error is analyzed as well. In Sec. 4.2, a vehicle classification rule is designed based on the length and height of the target vehicles. Then the performance of the classification rule is presented.
4.1 Vehicle Speed Measurement

Since there are two LIDAR sensors mounted on the probe vehicle, it is possible to measure the speed of the detected vehicles based on the comparison of images between the two LIDAR sensors. As shown in Fig. 4.1(a), the distance between the two LIDAR sensors is 1.405 meters. Let $\Delta t$ represents the moving time of the object from LIDAR #2 to LIDAR #1, then the relative speed of the object, $V_r$ and the absolute speed, $V_{obj}$ are given by Eqt. 4.1.

\[
\begin{align*}
V_r &= \frac{1.405}{\Delta t} (m/s) \\
V_{obj} &= V_r + V_{pb} (m/s)
\end{align*}
\]  

(4.1)

Where, $V_{pb}$ is the speed of the probe vehicle, which can be obtained from OBD. Observed vehicles could be moving in the same direction as the probe vehicle or the opposing direction. In Eqt. 4.1, the direction is represented by the sign of the calculated speed, if $V_{obj} > 0$, the object is moving in the same direction as the probe vehicle. Otherwise, it is moving in the opposing direction. This research employs two steps in the process of measuring the speed of the vehicle, as will be discussed in detail below.

4.1.1 Match regions from two LIDARs

The first step for seed measurement is to match observations of a given target between the two LIDAR sensors. The detected vehicles from both LIDARs need to be matched in pairs, which include one region from each LIDARs respectively. The matching process consists of two main steps:
Figure 4.1: Vehicle Speed Measurement

- First, project the detected vehicles onto Y-T space (i.e., 2D), so that each vehicle is represented by a rectangular region (see Fig. 4.1(b)).

- Then, match the regions that have the largest overlapping area (the shaded areas in Fig. 4.1(b)).

Fig. 4.1(b) shows the matching results for two different hypothetical vehicles. The right hand vehicle is first observed in LIDAR #2 (dashed-line rectangle is to the left of the solid-line rectangle), which means the target vehicle is overtaking the probe vehicle and $\Delta t > 0$. The left hand vehicle is first observed in LIDAR 1# (solid-line rectangular is to the left of the dashed-line rectangular), which means the target vehicle is overtaken by the probe vehicle and $\Delta t < 0$. The target vehicle is either moving in the opposite direction of the probe vehicle or the target vehicle is moving in the same direction and is being overtaken by the probe vehicles.
4.1.2 Measure the travel time of the object vehicle between two LIDARs

After matching the detected regions, the next step is to calculate $\Delta t$ for each pair. Formalizing the definition, $\Delta t$ is the time difference between the two instants when the same 'feature' (e.g., the front bumper) of the vehicle is seen in LIDAR #1 and LIDAR #2. One might be tempted to use the front bumper by taking the first LIDAR frame in which the vehicle is seen. However, as shown in Fig. 4.2(a), the LIDAR collects discrete scanning samples of the vehicle. It is very likely that the scanning samples taken by one LIDAR sensor is at different positions on the vehicle from those taken by the other LIDAR sensor. Therefore, a more robust method to measure $\Delta t$ is to fit of the entire sequence of LIDAR images from the vehicle in one LIDAR sensor to the corresponding sequence from the other LIDAR sensor. Assuming the frames can only be shifted by integer quantities relative to one another, Fig. 4.2(d) shows the final result (note some error remains due to sub integer offset).

In order to find the optimal fit of the two LIDAR images, the 3D LIDAR points of the vehicle are first projected onto the Z-Y plane. Then, the 2D LIDAR projection will be descritized into pixel image. Both the projecting and converting steps are similar to those in Section. 3.2.2. Fig. 4.3 shows the 2D projection and the converted discrete image of a detected vehicle in both LIDAR sensors.

The process of finding the optimal fit between the two LIDAR images is based on the discrete images. First, the image difference curve ($ImDiff(shiftN)$) is calculated.

---

$^5$In the projecting process, a lower bound on the height is used to exclude the points near the ground. In addition, in order to identify the pixel number in each column of the binary images from both LIDARs, an upper bound is also applied to the height of the LIDAR points. Both bounds can be chosen arbitrarily as along as most of the vehicles' points will be included and ground points will be excluded.
Figure 4.2: Measuring Travel Time from two LIDAR images. (a) scanning samples from both LIDARs of one vehicle, dashed lines are from side LIDAR #1, solid lines are from side LIDAR #2; (b) 2D image of the vehicle in side LIDAR #1; (c) 2D image of the vehicle in side LIDAR #2; (d) the optimal fit between the two LIDAR images.

Figure 4.3: LIDAR image of a detected vehicle: (a) the 2D LIDAR projection of side LIDAR #1, the rectangular shows the converted region; (b) the converted binary image of (a); (c) the 2D LIDAR projection of side LIDAR #2; (d) the converted binary image of (c)
based on Eqt. 4.2.

\[
ImDiff(shiftN) = \frac{1}{PixNum} \cdot \sum_{i=\min(1-shiftN,1)}^{\max(M-shiftN,N)} \| I_m2(i + shiftN) - I_m1(i) \| \quad (4.2)
\]

where, \( I_m1(n) \) represents the \( n \) – th column of the descritized image from LIDAR #1, \( I_m2(m) \) represents the \( m \) – \( th \) column of the descritized image from LIDAR #2. \( PixNum \) is the total number of pixels in both images. \( N \) is the total number of columns in \( I_m1 \) and \( M \) is the total number of columns in \( I_m2 \). \( \| I_m2(n) - I_m1(m) \| \) is the summation of the difference between the two columns\(^6\). When \( shiftN = 0 \), the first column of \( I_m1 \) will be aligned to the first column of \( I_m2 \). If \( shiftN > 0 \), \( I_m1 \) is shifted by \( shiftN \) columns to the right; otherwise, \( I_m1 \) is shifted by \(-shiftN\) columns to the left. Sweeping over a range of \( shiftN \), Fig. 4.4 shows an example of the curve.

Figure 4.4: Image Difference Curve: curve of the vehicle in Fig. 4.3;

\(^6\)Assuming \( I_m1 \) has \( N \) columns, \( I_m1(n) \) will be a zero column vector if \( n < 1 \) or \( n > N \); these ‘zeros-filled’ columns facilitate the process of calculating \( ImDiff(shiftN) \) with different number of \( shiftN \).
Figure 4.5: Error estimation for Travel Time Measurement: $Im_f(\cdot)$ and $Im_r(\cdot)$ are defined in Eqn. 4.2, $T$ is the time interval between two adjacent LIDAR frames.

Once the image difference curve is obtained, the local minimum on the curve, $(shiftN^*, ImDiff^*)$, will be extracted. Then, the travel time, $\Delta t$, can be calculated from Eqn. 4.3.

$$\Delta t = T_1(1) - T_2(1) - shiftN^* \cdot T$$ \hspace{1cm} (4.3)

Where, $T_1(1)$ and $T_2(a)$ are the time stamps of the first frame from side LIDAR #1 and side LIDAR #2 respectively; $T$ is the LIDAR sampling period.

With the optimal integer fit between the two images, image frames from one LIDAR will be aligned to the 'closest' frames in the other LIDAR. However, since the LIDARs collect discrete frame samples from the vehicles, there might still be a time offset ($t_{off}$) between the aligned frames. As illustrated in Fig. 4.5, the worst case scenario is that scan frames collected by one LIDAR are right in the middle of those collected by the other LIDAR. Under such case, $t_{off} = \frac{T}{2}$. Other than this case, $Im_f(1)$ will be aligned to either $Im_r(1)$ or $Im_r(2)$ whichever is closer to $Im_f(1)$.
Therefore,
\[ |t_{off}| \leq \frac{T}{2} = \frac{1}{2 \cdot Frq} = \frac{1}{74} \text{(sec)} \]  
(4.4)

Where, \( Frq = 37 \text{Hz} \) is the frequency of the LIDAR.

Eqn. 4.4 provides an important foundation to estimate the error of the measured speed. Combining Eqn. 4.4 and Eqn. 4.1,

\[
(V_r)_{err} = \left| \frac{1.405}{\Delta t} - \frac{1.405}{\Delta t + t_{off}} \right| = \left| \frac{V_r^2}{1.405/t_{off} + V_r} \right|
\]
\[
\Rightarrow (V_r)_{err} \leq \frac{V_r^2}{103.97 + |V_r|} \text{(m/s)} \]  
(4.5)

\[
\Rightarrow (V_{obj})_{err} = (V_r)_{err} \leq \frac{V_r^2}{103.97 + |V_r|} \text{(m/s)}
\]

Where, \( V_r \) and \( V_{obj} \) are defined in Eqn. 4.1, \( (V_r)_{err} \) is the absolute error of \( V_r \) and \( (V_{obj})_{err} \) is the absolute error of \( V_{obj} \).

### 4.1.3 Validation for Speed Measurement

In order to validate the performance of the speed measuring algorithm, vehicles parallel parked along a selected Link are extracted manually from the 7 runs (Table 2.1). Fig. 4.6 shows the location of the link, which is on Neil Ave. between Lane Ave. and Dodridge St. There are two lanes in both direction on this link. Typically, vehicles are parked in the second lane from road center in both directions. A total of 283 parked vehicles were seen on these 7 runs.

For parked vehicles, \( V_{obj} = 0 \text{m/s} \), the target speed relative to the probe vehicle is simply the negative of the probe speed, i.e., \( V_r = -V_{pb} \). The probe speed is available from the OBD data. Hence, the absolute error of \( V_{obj} \) could be derived from Eqn. 4.5.

\[
V_{err} \leq \frac{V_r^2}{103.97 + |V_r|} = \frac{V_{pb}^2}{103.97 + V_{pb}} \]  
(4.6)

40
Figure 4.6: Selected link for speed measuring validation: (a) an aerial photo of part of the link (Google Maps); (b) GPS route of the selected link, the probe vehicle is moving North Bound.

Fig. 4.7 shows the comparison between observed speed error ($V_{obj}$) and theoretically calculated error bound on $V_{err}$ from Eqt. 4.6. At the same time, the histogram of $V_{obj} - V_{err}$ is also shown in Fig. 4.7(b). Over 55% of the observed speed error is smaller than the theoretical speed error bound. $V_{obj} - V_{err}$ is no larger than 0.92 m/s, which might be caused by measuring error of OBD or other systematical error sources. Although we were unable to obtain the speed error for moving vehicles with the current experiment condition, the result shown in Fig. 4.7 provides evidence for the performance of the designed speed measuring algorithm.

### 4.2 Vehicle Classification

In this section a rule-based vehicle classification method will be discussed. The features used in the classification are length and height of the observed vehicles. First,
Figure 4.7: Speed Error of parked vehicles, there is a total of 283 vehicles in the plot: (a) observed speed error versus theoretical speed error, (b) theoretical speed error versus the speed of probe vehicle (the relationship is also represented by Eqt. 4.6), (c) histogram of $V_{obj} - V_{err}$
the length and height are extracted for each vehicle. Then, the developed classification algorithm separates the vehicles into one of the six categories: Motorcycle/bicycle (class 1), Passenger Car (class 2), SUV/small Van (class 3), Pickup truck (class 4), small bus/large van/single unit Truck, e.g., UPS truck (class 5), and Bus/Large Truck (class 6).

### 4.2.1 Vehicle Length and Height Extraction

Based on the measured the speed, the length of the observed vehicle can be extracted. As noted in Sec. 2.1.1, the time interval between each LIDAR frame is $\frac{1}{37}$ sec, given the relative speed between the target vehicle and the probe vehicle, the length can be calculated from Eqn. 4.7.

$$L = \text{Num}_{fr} \cdot \frac{1}{37} \cdot |V_r| (m) \quad (4.7)$$

Where, $\text{Num}_{fr} = \max((\text{Num}_{fr})_1, (\text{Num}_{fr})_2)$, the number of frames included in whichever LIDAR image that has the most frame. It is assumed that both the probe vehicle and the target vehicle maintain roughly constant speed during the time when the target vehicle is observed. Hence, $V_{pb}$ is the average speed of the probe vehicle during this period.

The extraction of vehicle height consists of three steps: first, the height of each frame sample is measured via Eqn. 4.8;

$$h_i = \max\{z_j, j \in \text{Im}(i)\} - \min\{z_j, j \in \text{Im}(i)\} \quad (4.8)$$

Where, $\text{Im}(i)$ includes all the LIDAR points in the $i$th frame of of LIDAR image $\text{Im}$, including any adjacent ground points (note that $h_i$ is indexed from the front bumper of the vehicle to the rear bumper of the vehicle), $z_j$ is the coordinate of the
Figure 4.8: Two groups of LIDAR frames: Group 1 includes frames collected between the front bumper and roughly the center of the vehicle; Group 2 includes frames collected between roughly the center of the vehicle and the rear bumper of the vehicle.

LIDAR points on the Z-axis. Then the frame samples are divided into two groups based on Eqt. 4.9,

$$F_{r_i} \in \begin{cases} 
\text{Group 1}, & i < \frac{Num_{fr}}{2} \\
\text{Group 2}, & i \geq \frac{Num_{fr}}{2}
\end{cases} \quad (4.9)$$

finally, as shown in Fig. 4.8, the height of each group will be extracted by using the median height of the frames included in the group.

$$H_i = \text{median}\{h_j, j \in \text{Group } i\} \quad (4.10)$$

Therefore, three features have been extracted for each vehicle: L (length of the vehicle), H1 (height of group 1) and H2 (height of group 2). A vehicle classification algorithm based on these three features will be discussed in the next section.
4.2.2 Length-Height based Vehicle Classification

The developed vehicle classification rules conduct two levels of classifying. The first level is based on the length of a given vehicle, which separates vehicles into three groups:

- **Group 1:** small vehicles, including motorcycles and bicycles. Vehicles will be classified into this group if $L < 2.3\, \text{m}$;

- **Group 2:** large vehicles, including city buses and Large trucks. Vehicles will be classified into this group if $L > 7.4\, \text{m}$;

- **Group 3:** medium sized vehicles, including passenger cars, SUV, vans, pickup and small single unit trucks (e.g., UPS truck). Vehicles will be classified into this group if $2.3\, \text{m} \leq L \leq 7.4\, \text{m}$

Figure 4.9: Height versus Length distribution of 927 vehicles
Since the probe vehicle is running along the campus route, which covers urban street, most vehicles observed are in group 2. Ground truth data of vehicles’ type are generated manually using the GUI tool (Fig. 3.15) \(^7\) from the six pre-defined links from App. B. Fig. 4.9 shows the Length and Height distribution of the 927 observed vehicles (note that the vertical axis is calculated by \(H = 0.5 \cdot (H_1 + H_2)\)). The first level of classification provides clear boundaries for class 1 and class 6, i.e., vehicles in group 1 are classified into class 1 and vehicles in group 2 are classified into class 6. On the other hand, vehicles of the other four classes are mainly distributed in group 2. The boundaries between these four classes are not clear in Fig. 4.9. Hence, a second level of classifying is conducted on vehicles in group 2 based on the extracted features: \(H_1\) and \(H_2\). The classification rules are listed below.

1. vehicles will be classified as class 2 (passenger cars) if \(H_1 \leq 1.5 m\) and \(H_2 \leq 1.4 m\);

2. vehicles will be classified as class 3 (SUV/small Van) if \(H_1 \leq H_2 + 0.1 m\) and \(1.4 m < H_2 \leq 1.95 m\);

3. vehicles will be classified as class 4 (pickup truck) if \(H_1 \geq 1.5 m\) and \(H_1 > H_2 + 0.1 m\);

4. vehicles will be classified as class 5 (small bus/large van/small single unit truck) if \(H_1 \leq H_2 + 0.1\) and \(H_2 > 1.95 m\);

Fig. 4.10 shows the distribution of all of the vehicles in group 2 as well as the boundary classification boundary for each class. Table 4.1 and Table 4.2 show the results of the classification performance for each class.

\(^7\)the same GUI tool used for generating vehicle detection ground truth data, shown in Fig. 3.15
### Table 4.1: Statistics of classification performance

<table>
<thead>
<tr>
<th>class #</th>
<th># of correctly classified vehicles</th>
<th>total # of vehicles</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>11 (100%)</td>
<td>11</td>
</tr>
<tr>
<td>2</td>
<td>548 (94.16%)</td>
<td>582</td>
</tr>
<tr>
<td>3</td>
<td>190 (89.62%)</td>
<td>213</td>
</tr>
<tr>
<td>4</td>
<td>42 (77.78%)</td>
<td>54</td>
</tr>
<tr>
<td>5</td>
<td>36 (100%)</td>
<td>36</td>
</tr>
<tr>
<td>6</td>
<td>31 (100%)</td>
<td>31</td>
</tr>
</tbody>
</table>

### Table 4.2: Statistic of classification performance, row # is associated with the assigned class #, column # indicates the true class #, the last row is the total number of vehicles in each class

<table>
<thead>
<tr>
<th>class #</th>
<th>true class 1</th>
<th>true class 2</th>
<th>true class 3</th>
<th>true class 4</th>
<th>true class 5</th>
<th>true class 6</th>
<th>total # assigned to the class</th>
</tr>
</thead>
<tbody>
<tr>
<td>assigned class 1</td>
<td>11</td>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>assigned class 2</td>
<td>0</td>
<td>548</td>
<td>11</td>
<td>9</td>
<td>0</td>
<td>0</td>
<td>568</td>
</tr>
<tr>
<td>assigned class 3</td>
<td>0</td>
<td>28</td>
<td>190</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>218</td>
</tr>
<tr>
<td>assigned class 4</td>
<td>0</td>
<td>1</td>
<td>11</td>
<td>42</td>
<td>0</td>
<td>0</td>
<td>54</td>
</tr>
<tr>
<td>assigned class 5</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>3</td>
<td>36</td>
<td>0</td>
<td>39</td>
</tr>
<tr>
<td>assigned class 6</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>31</td>
<td>31</td>
</tr>
<tr>
<td>total # of the class</td>
<td>11</td>
<td>582</td>
<td>213</td>
<td>54</td>
<td>36</td>
<td>31</td>
<td></td>
</tr>
</tbody>
</table>

47
Figure 4.10: $H_1$ versus $H_2$ distribution of vehicles in group 2 (889 vehicles), the markers are the same as Fig. 4.9

Besides the size of the vehicles, the classification also relies on the shape of the vehicles. For example, pickup trucks usually have a lower tail, which leads to the result of $H_2 < H_1$. By dividing the vehicles into two parts and measuring the height in each part, pickup trucks are usually separated from other vehicles. However, exceptions to the rule occur when the pickup is equipped with a rack or otherwise has a full bed. Fig. 4.11 shows an example where a pickup is misclassified because it has a full bed and thus, looks more like a van to our algorithm. In our database (927 vehicles), 3 out of 56 pickups are misclassified for this reason.

The classification performance statistics are shown in both Table 4.1 and Table 4.2. The average correct classification rate over all the six classes is 92.56%. Table 4.2 provides detailed information about the classification result for each classes. Based
on our database, classification errors occur among class #2, class #3 and class #4. Vehicles, all of these three classes have similar size. Furthermore, the measurement of vehicles’ length relies on the measurement of $V_r$. As discussed in Sec. 4.1.2, the measuring error of $V_r$ increases as $V_r$ increases, which is one important factor for misclassifying vehicles, which are not in class #1, as class #2. In addition, our definition of vehicle classes are not very specific. Vehicles with the size that are near the classification boundaries might be misclassified. One might develop much more specific classification rules employing prototypes of different types of vehicles. Unfortunately, with the current frequency of our LIDAR scans, it is difficult to measure a more accurate shape of the detected vehicles.
CHAPTER 5

CONCLUSIONS

This research has demonstrated that a probe vehicle mounted system for vehicle
detection and classification can be implemented with LIDAR sensors. In this study,
the LIDAR is mounted on a probe vehicle with an OBD and a DGPS providing
the motion and position of the probe vehicle. First, objects that are potentially
vehicles are extracted from the 3D LIDAR image using a combination of the region
segmentation and road boundary detection. After finding all possible vehicles, the
approach dynamically finds the curb to differentiate between on-road vehicles and
other targets. The developed vehicle detection method can effectively extract vehicles
from the background environment with a miss-detection rate of 3.3% and a non-vehicle
detection rate of 3.6%. Both the relative and absolute speed of the detected vehicles
can then be measured by matching vehicle features between the two LIDAR sensors
and extracting the time the target takes to traverse the gap between the sensors.
After measuring a target’s speed relative to the probe, the target’s length can be
recovered. Then, a vehicle classification rule is developed based on the height and
length of the detected vehicles. The vehicle classification system sorted vehicles into
the pre-defined classes with 92.6% accuracy.
There are several extensions to this research that could improve the performance of
the vehicle detection. For example, most of the non-vehicle objects that are ultimately
detected as vehicles are due to road boundary detection errors. By incorporating the
observed vehicles’ motion, the road boundary detection performance could be further
improved; both over short time scales by projecting trajectories of observed targets,
and over long time scales by amassing observations over many runs. Hence, the
vehicle detection error rate could be further decreased.

Likewise, there are several extensions that could improve the performance of the
vehicle classification. Obviously reducing non-vehicle detections would help. But
beyond that, the present vehicle classification rules sort vehicles into six categories
using only three features: Length, $H_1$ and $H_2$ (Sec. 4.2). One can develop much more
precise classification categories by employing prototypes of different types of vehicles.
BIBLIOGRAPHY


APPENDIX A

LIDAR SCANNING RESOLUTION

The scanning resolution of the side LIDAR is 0.5 degrees in the Polar coordinate system. In the Cartesian coordinate system, the resolution is decided by both the height (Z dimension) and the distance from the LIDAR (Y dimension).

A.1 resolution on Y-dimension

In Fig. A.1(a), the resolution of LIDAR scanning at one point \((y_n, z_n)\) is represented by \(\Delta y\), which is the distance between scanning point \((y_n, z_n)\) and the next scanning point \((y_{n+1}, z_{n+1})\). Since the points are projected onto the first two dimensions, only distance on the Y dimension will be considered. Given the resolution of the scanning angle \((\Delta \theta = 0.5^\circ)\). The relationship between \(\Delta y\) and \((y_n, z_n)\) could be derived as the following.

\[
\tan \theta_n = \frac{2.05 - z_n}{y_n} \quad (A.1)
\]

\[
\tan \theta_{n+1} = \tan(\theta_n - \Delta \theta) = \frac{2.05 - z_n}{y_n + \Delta y} \quad (A.2)
\]

Combining Equation. A.1 and Eqt. A.2, the relationship between \(\Delta y\) and \((y_n, z_n)\) is shown in Eqt. A.3. Here, \(\tan \Delta \theta \simeq 0.0087\).
\[ \Delta y = 0.0087 \cdot (2.05 - z_n) + \frac{0.0087}{2.05 - z_n} \cdot y_n^2 \]  

(A.3)

Figure A.1: LIDAR resolution in the Cartesian coordinate system (a) is the resolution on the Y-dimension, (b) is the resolution on the Z-dimension.

Eqn. A.3 assumes that the points are at the same height. When two points are at different height, the distance on Y-dimension will be smaller than that derived from Eqn. A.3 (\( \Delta y' < \Delta y \), see Fig. A.1(a)). For a vehicle 7 meters away from the probe vehicle (two lanes away), the distance between the points could be estimated as the following (let \( z \approx 1 \))

\[ \Delta y \approx 0.0087 \cdot (2.05 - 1) + \frac{0.0087}{2.05 - 1} \cdot 7^2 \approx 0.41 \]  

(A.4)

A.2 resolution on Z-dimension

Fig. A.1(b) demonstrates how the lateral distance of the region effect the LIDAR resolution on the Z-dimension. Based on the similar derivation as that for Eqn. A.3, the following equation could be obtained.
\[ \Delta z = y_n \cdot \tan \Delta \theta \cdot \left(1 + \frac{(2.05 - z_n)^2}{y_n^2}\right) \]  \hspace{1cm} (A.5)

In Eqt. A.5, \(0.5 \leq z_n \leq 6\) (converting boundary of height). Assuming \(z_n \approx 3.25\), Eqt. A.5 could be further simplified.

\[ \Delta z = 0.0087 \cdot y_n \cdot \left(1 + \frac{1.44}{y_n^2}\right) \]  \hspace{1cm} (A.6)
APPENDIX B

PRE-DEFINED LINKS FOR VEHICLE DETECTION VALIDATION

In Fig. B.1, the links have been highlighted to illustrate their locations. Then in Table B.1, the detailed information of each link is provided. Links #1, #2, #3 and #5 are all predominantly four lanes (two each direction), no median, and the probe vehicle typically drove in the right-most lane. Link #4 is similar, with the exception that it has a raised median for roughly half of its distance. Finally link #6 does not have a median, but it only has two lanes (one each direction).
Figure B.1: Route of the Links
<table>
<thead>
<tr>
<th>Link #</th>
<th>Road Section</th>
<th># of passes (of each run)</th>
<th>Length (km)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>On Kenny Road, from the intersection at Ackerman Rd to the intersection at Woody Hayes</td>
<td>1 of Southbound</td>
<td>1.686</td>
</tr>
<tr>
<td>2</td>
<td>On W Lane Ave, from the intersection at N High St to the intersection at Neil Ave</td>
<td>1 of Eastbound; 2 of Westbound</td>
<td>0.366</td>
</tr>
<tr>
<td>3</td>
<td>On W Lane Ave, from the intersection at Kenny Rd to the intersection at Neil Ave</td>
<td>1 of Eastbound; 1 of Westbound</td>
<td>1.649</td>
</tr>
<tr>
<td>4</td>
<td>On Woody Hayes Dr, from the intersection at Kenny Rd to the intersection at Cannon Dr</td>
<td>3 of Eastbound; 2 of Westbound</td>
<td>1.221</td>
</tr>
<tr>
<td>5</td>
<td>On Ackerman Rd &amp; Dodridge St, from the intersection at Neil Ave to the intersection at Kenny Rd</td>
<td>1 of Westbound</td>
<td>2.079</td>
</tr>
<tr>
<td>6</td>
<td>On Woodruff Ave, from the intersection at Tuttle Park PI (the section between Tuttle PI and Neil Ave is not included in the research) to the intersection at College Rd</td>
<td>1 of Eastbound; 2 of Westbound</td>
<td>0.413</td>
</tr>
</tbody>
</table>

Table B.1: Link Information