ROAD-SIDE OBSTACLE DETECTION AND THREAT ASSESSMENT

A Thesis in
Mechanical Engineering
by
David Jeffrey Corbin

© 2014 David Jeffrey Corbin

Submitted in Partial Fulfillment
of the Requirements
for the Degree of

Master of Science

August 2014
The thesis of David Jeffrey Corbin was reviewed and approved* by the following:

Sean N. Brennan  
Professor of Mechanical Engineering  
Thesis Advisor

Christopher D. Rahn  
Professor of Mechanical Engineering

Karen Thole  
Professor of Mechanical Engineering  
Department Head of Mechanical and Nuclear Engineering

*Signatures are on file in the Graduate School.
Abstract

Considering the safety of an autonomous or semi-autonomous vehicle, detection of surrounding obstacles is mandatory. This thesis proposes three methods for static obstacle detection utilizing a downward-facing LIDAR measurement system. The obstacles observed are rollover areas, positive obstacles, and pavement edge drop-offs. Prior art explores the localization aspect of obstacle detection. However, they avoid the analysis of the threat potential of each obstacle, which is necessary for proper decision making routines. This thesis explores not only the aspect of obstacle localization but also the hazard analysis of each obstacle. Reducing the dimensionality of raw range data through pre-processing and feature extraction techniques, this thesis then performs detection algorithms on the data series and conducts rules-based classification methods to properly identify obstacles. Then, the severity of the classified obstacles is analyzed along with the obstacles’ locations relative to the lane center. Tested offline using data collected at the Thomas D. Larson Pennsylvania Transportation Institute’s (LTI) Test Track facility as well as public access roads, the results show that the methods employed can successfully locate and quantify levels of potential obstacle hazards despite noisy real-world sensor data.
# Table of Contents

List of Figures vi

List of Tables ix

Acknowledgments x

Chapter 1

Introduction 1

1.1 Motivation ........................................ 1

1.1.1 Driver Assist Systems/Active Safety Systems ........ 1

1.2 Map-Based Obstacle Detection System .................. 2

1.3 Thesis Organization ................................ 5

Chapter 2

Related Work 6

2.1 Positive Obstacle Detection .......................... 7

2.1.1 Feature Extraction Methods ........................ 8

2.1.2 Feature Classification Methods .................... 9

2.2 Negative Obstacle Detection .......................... 10

2.2.1 Pavement Edge Drop-off .......................... 12

2.3 Lane Detection ..................................... 12

2.4 Map-based Localization ................................ 14

Chapter 3

System Setup 16

3.1 Sensor Setup ....................................... 17

3.2 Power System ....................................... 18

3.3 Sensor Configuration ................................ 19

3.4 Experimental Protocol ............................... 20

3.4.1 Mapping Procedure ............................... 21
List of Figures

1.1 Point cloud generated from a downward-facing 2-D LIDAR ............ 3
1.2 Hazard identifiers for an on-road scenario .......................... 4

2.1 Drivable area detection ........................................... 10
2.2 Negative obstacle detector ........................................ 11
2.3 Pavement edge drop-off ........................................... 13
2.4 Non-ideal lane markers .......................................... 13
2.5 Negative obstacle detector ........................................ 14

3.1 LIDAR and GPS systems mounted on a vehicle ....................... 16
3.2 Figure of LIDAR close up ........................................ 17
3.3 Side mounted LIDAR system ...................................... 18
3.4 Power supply of mapping vehicle with back-up charging capabilities .. 19
3.5 Power supply discharge over time ................................ 20
3.6 Power distribution to the network of sensors ....................... 20

4.1 Road-side obstacle detection mapping workflow ..................... 23
4.2 Preprocessing workflow ........................................... 24
4.3 Point density distribution from a fixed angular resolution .......... 26
4.4 Calculating maximum detection width ................................ 26
4.5 Non-monotonically increasing LIDAR data .......................... 27
4.6 Fixed LIDAR data .................................................. 27
4.7 Preprocessing raw LIDAR data ................................... 28
4.8 Lane marker extraction workflow ................................ 29
4.9 Single lane marker convolution kernel ............................ 30
4.10 Single lane marker convolution result ............................. 30
4.11 Double lane marker convolution result ............................ 31
4.12 Rollover detection workflow ..................................... 33
4.13 Slope data presenting impending backhoe rollover ............... 34
4.14 Positive obstacle detection workflow .............................. 35
4.15 The gradual slope of a road-side embankment ..................... 38
4.16 Feature extraction using a thresholded second derivative ........ 39
A.1 Rollover detection result (ex. 1) ........................................ 64
A.2 Rollover detection result slopes (ex. 1) ................................. 65
A.3 Rollover detection result (ex. 2) ........................................ 65
A.4 Rollover detection result slopes (ex. 2) ................................. 66
A.5 Positive obstacle detection result (ex. 1): Obstacle of height 0.132m detected at 2.056m left of the lane center ....................... 66
A.6 Positive obstacle detection result (ex. 2): Obstacle of height 0.665m detected at 14.056m left of the lane center ....................... 67
A.7 Positive obstacle detection result (ex. 2): First and second derivative analysis ................................................................. 67
A.8 Pavement edge drop-off detection result (ex. 1): Drop-off length of 0.8357m and slope of 33.95° detected at 0.3805m right of the single lane marker ............................................................ 68
A.9 Pavement edge drop-off detection result (ex. 2): Drop-off length of 0.4161m and slope of 23.92° detected at 0.6579m right of the single lane marker ............................................................ 69

B.1 Lateral image of plank simulating pavement edge drop-off .......... 71
B.2 Experimental test setup ....................................................... 71
B.3 Example LIDAR scan of a wooden test plank ......................... 72
B.4 GUI allowing user to select top of pavement edge (mouse denoted by large crosshairs) ...................................................... 72
B.5 GUI allowing user to select bottom of pavement edge (mouse denoted by large crosshairs) .................................................. 72
B.6 Plank 3 in run 1 ................................................................. 73
B.7 Plank 5 in run 5 ................................................................. 73
B.8 Hand-measured slope vs. GUI slope with error bars .................. 75
B.9 Hand-measured slope vs. automated slope with error bars ........... 75
B.10 Hand-measured slope vs. automated length with error bars ......... 76
B.11 Hand-measured slope vs. automated length with error bars ......... 76
List of Tables

B.1 GUI test results of the six planks in five runs ................. 74
B.2 GUI test results compared to hand measurements ............. 74
B.3 Automated test results of the six planks in five runs ........ 74
B.4 Automated test results compared with hand measurements .... 75
Acknowledgments

I would like to thank all who have helped me during the process of this thesis. Firstly, I would like to thank my advisor, Dr. Sean Brennan. Without his knowledge and experience this thesis could not be completed. I appreciate his patience and willingness in answering my questions.

I would like to thank Dr. Alex Brown for helping me understand what a graduate student is. His guidance, support, and perspective have helped me through these semesters. I would like to specially thank Patrick Proffitt, Bobby Leary, and Paul Stankiewicz for their continual support repeatedly collecting data, and for their inspiring discussions about anything and everything. I would also like to thank Ted Dorris for lending an ear and his time for discussions about graduate life.

I would like to recognize Margalit Goldschmidt, whom without, I would not have made it this far.

Lastly, I would like to thank my family: Dr. Deb Liczewek, Dr. Dave Corbin, and Dr. Elise Corbin for their tireless support, encouragement, and love. Without you, I cannot succeed.
1

Introduction

1.1 Motivation

Despite the fact that drivers are generally able to observe their environment and ego-states reasonably well, even the most qualified drivers can experience a loss of control. From 2004-2008, single-vehicle collisions accounted for close to 60% of all fatal collisions in the United States. Tallying at 109,936 driver fatalities, single-vehicle collisions account for 41% more fatal collisions than those with two vehicles and multiple vehicles combined [1].

The National Highway Traffic Safety Administration attributes the causes of these collisions to excessive speed, driver fatigue, distraction, alcohol, and recently, cellphone usage. However, to a safe and lucid driver, a number of other factors contribute to the risk of collision, generally factors that impair visualization of the roadway. These include vehicle design, road design, weather, and road environment [1–4].

During loss of control, a driver may become panicked. Combining that panicked state with environmental factors such as snow, fog, heavy rain, and very sunny conditions, a human pilot or its robotic assistant can lose the ability to track the vehicle’s trajectory and the hazardous obstacles surrounding it. These facts have motivated advancements in automobile and roadway design.

1.1.1 Driver Assist Systems/Active Safety Systems

Numerous implementations of active safety systems (referring to technology assisting in the prevention of a crash) have arisen, including collision avoidance systems, drowsy driving warnings, and lane departure warning systems, all of which alert the driver to
take preventative measures when they stray from a safe path. Employing radar, laser rangefinders, stereo cameras, and inertial measurement units helps prevent the driver from departing his or her lane or colliding with static or dynamic obstacles.

The automotive industry has introduced new systems in the market including Honda’s Intelligent Night Vision System in 2004 which detects pedestrians and warns the driver with an auditory alarm [5, 6], Mazda’s Smart City Brake Support which uses lasers to detect obstacles in front of the vehicle and apply brakes and cut engine power [7], and Volvo’s Collision Warning with Auto Brake in 2007 which employs a radar/camera fusion to detect obstacles in front of the vehicle and warn the driver with a heads up display and applies brakes if the driver does not react [8, 9].

These solutions attempt to solve very specific problems while observing an assumption: that the operator or robotic assistant always maintains a visual on the features. Thus these solutions cannot assist the driver in situations where the road may be occluded by environmental factors. This issue calls for a system to illuminate fixed roadside obstacles’ locations on the road that may cause accidents without visually observing them.

1.2 Map-Based Obstacle Detection System

The focus of the proposed thesis is the hierarchical classification of hazards stored in a compact but sufficient map format required for safe operation within vehicles with good positioning (for instance, using GPS) but potentially poor visibility of obstacles. The remainder of this section identifies the technology and general methods used to accomplish this.

Recent technology has helped prevent accidents caused by distracted operators and obstructed vision. One particularly promising technology is LIDAR-based measurement systems, which light a surface with a narrow-beam laser, and measure the time it takes for the reflection to return to the sensor. This time-of-flight reading is converted into distance to obtain an accurate range reading from the sensor. With today’s LIDAR, these measurements are obtained quite accurately, on the order of a few millimeters of error. Additionally, the laser is spinning at a relatively fast rate – 100 times a second – and multiple pulses can be emitted during each rotation, as much as one every half degree or even finer. The resulting LIDAR return thus generates a finely measured distance cross-section measured across a “slice” of the roadway geometry. These slices, particularly when the LiDAR is mounted to obtain cross-sections of the road, have the
potential to automate many road geometry data collection procedures resulting in a 3-D representation of the road known as a point cloud (Figure 1.1). This representation is helpful in interpreting information about the road that humans and other sensors cannot and using that information to construct automated safety systems.

The thesis proposed seeks to take advantage of the LIDAR technology by performing offline detection of multiple obstacle types and the level of hazard they provoke that can be used in the construction of a compact map of drivable areas and hazards for safe vehicle operation. Most research in this field generally assumes that each vehicle is equipped with mapping capabilities and maintains its own map used for localization. By developing a map prior to navigation, we can avoid equipping navigating vehicles with costly sensors. Instead the vehicle will be equipped with the map, shrunk to only include geometric boundaries of obstacles categorized by specific hazard levels corresponding to safe vehicle operation.

To avoid having to detect every obstacle at every instant on-the-fly, vehicles operating with computer assistance or full automation must be able to localize themselves in a predefined map in order to follow prescribed paths or perform specific tasks. Some researchers have applied Simultaneous Localization and Mapping (SLAM) to navigation problems [10, 11], and some have used DGPS\(^1\) for localization in open areas [12]. But knowing a vehicle’s location is not the same as knowing where it should go. Performing routine mapping of an environment can provide a vehicle with all the information needed used for path planning or driver assistance. Accidents due to rollover, collision, or driving over a rut or drop-off can be avoided. The types of obstacles examined include:

1. Large immobile collision hazards, bumps, or other “positive obstacles” extending

\(^1\)Differential Global Positioning Systems improve the location accuracy compared to standard GPS from 15 meters to the centimeter level
Figure 1.2. Hazard identifiers for an on-road scenario

from the ground plane

2. Vehicle rollover hazards resulting from upward or downward lateral slopes

3. Pavement edge drop-offs that could cause a vehicle to become immobilized

As outlined in the following section on prior art, the first obstacle type has a highly developed body of literature supporting effective approaches and algorithms, but the measurement of the feature and its integration into a map structure small enough to be employed in passenger vehicles is not well documented.

Identifying the above hazard types using a fusion of inertial and laser sensors and quantifying their level of threat followed by a simplification of the features will result in a map that is far more compact than the type typically used in autonomous vehicles research, e.g. [13], where maps of high spatial density are required. The objective of the proposed work is not to create a map for localization, but to create one for path planning. In other words, meaningful environmental features will be maintained in real time for autonomous or driver assist systems, with localization assumed to occur using DGPS or other reliable technology. Figure 1.2 shows the type of information intended for extraction from three dimensional scans of an environment by an appropriately equipped mapping vehicle.
The theoretical landscape in Figure 1.2 shows contours of hazards shaded by their severity – each identified pothole, rollover risk area, and immobile obstacle will be represented by boundary coordinates and a severity gradient in the stored map. Then a developed heads-up display in a vehicle’s driver warning system or path planner in an autonomous system can access the information without requiring an entire point cloud or 3-D mesh of the environment. This will save on bandwidth requirements for navigation vehicles, and allow the autonomous or driver assist systems to plan paths or provide warnings for equipment in real time.

1.3 Thesis Organization

The remainder of this thesis is organized in the following format. Chapter 2 discusses the recent and current work in the field of obstacle detection and lane detection. Chapter 3 is an overview of the hardware instruments of the system setup implemented in the data collection aspect of this project. Chapter 4 covers the method and algorithms employed in this thesis. Chapter 5 shows the experimental results of define methods. Chapter 6 concludes the thesis and expresses ideas for future work.
Chapter 2

Related Work

The unmanned ground vehicle (UGV) has become a common topic in vehicle and robotic research in the last decade since the first DARPA Grand Challenge. However seemingly difficult, the process of entire vehicular automation of a ground vehicle can be divided into three tasks: perception, planning, and control [14]. The review that follows focuses on the perception task, because this thesis seeks to find and create a minimum-size representation of an on-road driving environment suitable for safe navigation through detection of multiple obstacle types. The majority of the research dedicated to obstacle detection utilizes LIDAR as a primary sensor, due to its accuracy and robustness at close ranges in outdoor environments. In fact, the majority of teams in the recent DARPA Grand and DARPA Urban Challenges used LIDAR as the primary obstacle-detection method [14–17].

In general, as a primary objective, the studies of primary focus in this literature review are those that seek to define a “drivable area” for autonomous systems. A drivable area in this thesis refers to a segment of ground flat enough for a vehicle to safely traverse. Further, it is assumed in this thesis that a drivable area has bounds, and that exiting these bounds signify that a vehicle will encounter an obstacle or un-drivable surface. The current literature explores the detection of the location of these bounds: obstacles that protrude from the ground, called positive obstacles, and voids in the ground plane, called negative obstacles.
2.1 Positive Obstacle Detection

Positive obstacles are generally collision hazards, and several recent studies have explored methods for detection of these objects from a vehicle’s perspective [18–21]. These studies typically focused on urban environments with obstacles of predictable size, such as curbs. While a curb is a suitable boundary for most urban environments, it is a limitation of true obstacle detection that it can be unclear whether or not a curb is considered a drivable area; in emergency situations, it may be, whereas in normal driving it never should be. Driving environments feature an array of different obstacles to contend with that can break or bound a drivable area including trees, concrete barriers, guard-rails, and berms. In most cases, the exclusion of positive obstacles from drivable areas is desirable, especially in urban environments where man-made obstacles should generally be avoided.

Differing from the detection process in urban environments, other research efforts in positive obstacle detection have explored the detection in a more diverse environment to account for a larger array of positive obstacles that are found in natural settings. These efforts have generally simplified the detection process to a slope analysis of LIDAR data [22–24]. Obstacles are detected when the height slope in the lateral or longitudinal directions of a detected segment exceeds a set threshold. The simplicity of this technique in obstacle detection is unrestricted to specific driving environments. Many algorithms used for positive obstacles on urban roadways can be fitted for rural and even off-road scenarios. Some of the processes that detect curbs and barriers (as those in [18–21]) can be used to detect trees and walls. This method however does not account for certain factors such as the change in slope, thus not being able to differentiate between a gradual slope (e.g. a hill) and a steep slope (e.g. a wall). Gradual slopes are very prevalent in many environments and are necessary to detect, thus a more careful analysis of how slope affects drivability is part of this study.

Obstacle detection can be performed directly from a raw input signal. This can be done by predicting the result with extracted features from using a Kalman filter [19, 25]. However, due to many assumptions made, most often positive obstacle detection is generally performed through two steps: feature extraction, consisting of line segmentation, and feature classification. The following is a more in depth look at the current techniques used for both steps.
2.1.1 Feature Extraction Methods

The dimensionality reduction of the feature extraction step is primarily concerned with partitioning the data into an accessible segments that can easily and quickly be classified. Multiple methods can be used for feature extraction, each with a common goal: to prepare the data for segment classification. Certain studies have used the Ramer-Douglas-Peucker algorithm for line segmentation [17, 26]. This algorithm, commonly known as Split-and-Merge or Iterative End-Point Fit, divides a curve recursively. By selecting split points (initially the first and last point) and fitting a line between them, the other data points between the split points are evaluated to a fixed threshold of a normal distance from the fitted line. The data point with the largest normal distance greater than the threshold becomes a new split point. The process is then repeated until no data point lies outside the normal threshold distance. The algorithm is defined in Algorithm 1 in Chapter 4 Section 4.4.1.1. By setting a height threshold, either arbitrary or based on the mean-error, the studies are able to model a set of range data points to fewer line segments.

Other studies attempt to simplify the extraction step by separating the data into equal and smaller segments. Liu et al. [18] performs feature extraction by creating a Digital Elevation Map (DEM) from 2-D sequential laser range data. The sequential data is combined and broken into multiple cells, each of which has a global $x$, $y$, and $z$ coordinate, thus forming a grid of data points. Peterson et al. [27] partitions the range data into multiple “windows” of equal size in the $x$ direction then uses the notable Haar wavelet transform for feature extraction. After the wavelet transform, each window will correspond to a local wavelet coefficient which can be thresholded to signify a curb or road.

Another popular segmentation method used is the Hough transform [16, 20]. By fitting a line representative of the road surface, the range data points can be segmented into two categories relative to a height threshold normal to the road surface [20]. Furthermore, the Hough transform can fit lines to data assumed to be obstacles as in [16]. Here, Zhang et al. [16] omit detecting the road surface using the Hough transform. Instead the input range signal is convolved with a differential filter used to detect maximal changes in slope. These maxima then represent feature bounds of a candidate road segment which is then fitted with a line biased near its center. The data between the feature bounds (the road) is omitted and lines representing the road boundaries are fitted to the remaining data using the Hough transform.
2.1.2 Feature Classification Methods

Note that most LIDAR-based sensing systems cannot differentiate between a barrier and a median containing tall grass; it is only the features and context of the collected data that allow one to classify tall grass as traversable. Similarly, classifying a small clump of weeds as a hump may be inappropriate on a highway setting, but completely appropriate on a mining road where the same profile may be a boulder. Thus, the classification of features is very context-specific and dependent on the perceived severity of misclassification.

Feature classification is typically seen as a binary decision process concerning examination of detected features. The extracted features are usually sent through a series of tests in the determination process of their state of nature. The methods used in recent studies centralize around comparing a measured slope or height to the expected slope or height.

Assuming a uniform obstacle height, as in the height of a curb, Kang et al. [20] uses the representative line fit of the road surface and probabilistically determines positive obstacles by comparing the raw data location to its expected location (i.e. the best-fit line). Thus, any obstacle detected that does not lay on the calculated plane of the ground is classified as a positive obstacle. Similarly, [17] calculates the expected range of the road plane based mainly on sensor geometry and segment orientation, and deduces whether a segment is from a road or an obstacle based on its similarity to that expected range. Further, from their DEM, Liu et al. [18] performs a rules-based classification along with a 1-D Gaussian Process Regression to detect curb points. By comparing the \( z \) value of each cell to the adjacent cells, a decision can be made about the cell’s state of nature.

Differing from the common method of comparing measured data to expected values, Zhang et al. [16] classifies extracted bounds as positive obstacles if the adjacent line segments of the bounds (extracted using a Hough transform) are perpendicular (greater than 75°) to the road surface, and surpass a threshold based on the weighted standard deviation of the segment and the total number of data points in that section. If the segment adheres to the classification rules and passes a minimum road-width test (defined as a distance between local adjacent maxima and minima points greater than 4m by [16], and 2m by [17]), then the segment is positively classified as a road segment.

Notably, all methods reviewed here have not rated the severity of each detected obstacle, but simply classified obstacles and determined their locations.
2.2 Negative Obstacle Detection

Identification of drivable areas based exclusively on positive obstacle detection assumes an ideal flat-road surface. That is that anything that is not a positive obstacle is assumed to be a drivable surface (see Figure 2.1 where the red represents positive obstacles, and all other data is considered drivable). However, in certain environments such as roads with declining embankments or common on-road construction sites, positive obstacles represent only a portion of the road surface that could be considered hazardous. Negative obstacles (ditches, potholes, and other depressions) can immobilize a vehicle if driven over, and the degree of this risk is a function of vehicle design and current vehicle motion. Even the detection of flatness can indicate a potential negative obstacle, as is the case with a very large depression that is filled with water, or even a depression that is so deep that sensors are unable to see within. Compared to positive obstacles, negative obstacle detection is far less common in the literature, and yet is generally a more difficult problem.

Sinha et al. [29] review various directions for negative obstacle detection. These include inference of negative obstacles through ray-tracing of individual pixels and comparing expected range to actual range using a stereo camera [30]; combining range data from stereo cameras with thermal sensors implying that cavities generally remain warmer at night [31]; and using multiple stereo cameras, laser rangefinders, an omnidirectional camera, and a monochrome digital camera where object detection and terrain analysis is based on geometric features and multi-spectral image-based features. The drivable area is then detected through ray-tracing from LIDAR, while negative obstacles are found by the absence of laser hits in the direction perpendicular to the drivable surface [32]. These methods, however, can only detect obstacles in real-time if the vehicle is equipped with costly sensors and sufficiently fast processors to perform these computationally intensive algorithms.
Figure 2.2. Negative obstacle detector [33]

The majority of research regarding drivable areas containing both positive and negative obstacles focuses on robots in off-road environments (search and rescue, bomb disposal, etc.) [28, 29, 33]. While this literature is illuminating, the scope of this project is on-road passenger vehicles. Berglund et al. [34] have explored safe path-planning and obstacle avoidance in road environments, passenger vehicles; however, they assume an a priori map is provided. Magnusson et al. [12] discuss scan registration algorithms, but do not consider obstacle detection. Similar to the way [20] detects positive obstacles, Larson et al. [33] performs probabilistic localization by comparing the raw data location to its expected location. Further, [33] calculated terrain traversability based on the detection of positive and negative obstacles as well as slope steepness using a single 3-D forward-facing LIDAR. Using slow-speed robots, their goal was to detect upcoming negative obstacles and stop the vehicle before navigating into these obstacles. This method classifies gaps in data as potential negative obstacles using a rules-based procedure seen in Figure 2.2. This is essentially accomplished by detecting gaps in data and comparing the measurements to thresholds determined by where the expected data should be assuming a flat surface. Then, if the detected data exceeds the threshold, the location is classified as a negative obstacle. This method is useful but is difficult to implement in real time. Further, in real environments, this approach results in a high false detection rate, which is problematic especially for fast moving vehicles.
2.2.1 Pavement Edge Drop-off

The literature demonstrates the difficulty to detect or classify negative obstacles, and so this study focuses on only one particular negative obstacle: pavement edge drop-offs. Pavement edge drop-offs, as seen in Figure 2.3, according to one study [35], have been estimated to be the cause of 30% of fatal collisions on two-lane rural roads. These collisions include the phenomena of tire scrubbing which is friction between the pavement and tire when the tire side wall becomes forced into the side of the edge drop-off [36]. Tire scrubbing requires an overcompensation of steering to return to the road surface; however, due to driver reaction time and vehicle speed, this overcompensation often forces the vehicle to encroach into the opposing traffic lane [37].

Angled pavement edges were studied in [38], who used computer-based simulations to investigate a number of vehicle types and their ability to return to the lane after dropping the front right tire off the pavement edge. These simulations revealed that the tractor-trailer was most sensitive to pavement edge drop-offs and the pickup truck was the least sensitive. They deduced the following design requirement for allowable pavement drop-off heights:

\[ D = \frac{H}{\alpha^3} \]  

(2.1)

In this equation, \( H \) is the edge drop-off height in inches, and \( \alpha \) is the drop-off slope in radians. The study determined that \( D \) should not exceed 3.5 in. rad\(^{-3}\).

The Department of Transportation considers pavement edges with a vertical differential of 3” or greater to be unsafe [39].

2.3 Lane Detection

The goal of this study is on obstacle detection, analysis, and positioning for a safety-based augmented reality; thus, localizing the detected obstacles relative to the lane's position is crucial. This requires lane detection techniques. The lane detection problem is most commonly a focus of study for Lane Keeping algorithms and Lane Departure Warning systems. The process commonly is performed through the use of cameras [40–42] and through LIDAR [43, 44]. Although the problem can seem easy in the simplest scenario (i.e. straight road, solid lane markers, fresh paint), often road surfaces are not ideal (see Figure 2.4). With faded paint lines, lines repainted slightly off, repaved roads, shadows cast on the road, and other lane markers within the lane, the robust detection
Hillel et al. [46] suggest that, regardless of the sensors that are used, the solution to the lane detection problem generally consists of the following constituent modules: image (or data) pre-processing, feature extraction, road/lane model fitting, temporal integration, and image (or data) to world correspondence (Figure 2.5).

The objective of image pre-processing is to remove irrelevant or misleading data. This can be accomplished by searching for illumination-related effects in the image and pruning affected parts of the image. A simplified pruning technique would be to define a region of interest (ROI) suitable for feature extraction based only on trusted sensor regions. In an experiment using a downward-facing LIDAR, the lower-half of the data may be defined as a ROI [47] since the upper half of the scan arc would correspond to detections above the road surface (such as trees) that are not of interest in lane detection.

There are a number of features that can be used to discriminate the lane marker from
the road, and the nature of each feature can help to define the detection algorithms used. The color of the lane marker can be detected using a monocular camera. The reflectivity or brightness of the lane marker can be measured using LIDAR, which in turn allows lane detection through reflectance thresholding [48]. Both camera and LIDAR sensors enable adaptive thresholding, wherein the past several time sequences of data are averaged to calculate an appropriate threshold; this method was used in [49]. The Hough transform is a common lane detection algorithm as well as an obstacle detection algorithm [50–52]. Another effective method for lane detection is the convolution method using kernels that can be used for images and 2-D LIDAR scans. Convolving kernels with the data can be used with a wide array of features. McCall et al. [45] simultaneously filter video data for circular reflector markings, solid lane markings, and segmented line markings.

2.4 Map-based Localization

One can use map-based features as a means of generating prior expectations on sensor inputs, and thereby enabling real-time obstacle detection and processing using relatively simple algorithms. The formation of feature maps involves collecting raw data, processing it, and storing the data in a server on-board or accessible from the automated vehicle. Certain studies have stored entire point clouds in a map [22, 53]. This method minimizes any errors that may arise from creating and storing features in maps; however, this approach stores far more data than what is necessary. This reveals a fundamental trade-off in using maps: the accuracy of the map, versus their size. Creating maps of only relevant data minimizes the size of the feature maps, but assumes that very robust algorithms exist to extract “relevant” data from a data stream. For example, Levinson
et al. [54] proposed storing only planar road data and removing all data that does not lay on the road surface. This method however does not store relevant information to obstacle avoidance; it is used only for localization of a vehicle relative to the map.
Chapter 3

System Setup

The process of road-side obstacle detection and threat assessment begins with data collection within a static environment using a sensor-equipped mapping vehicle. A Volkswagen Passat was instrumented with a 2-D SICK LIDAR\(^1\) and an integrated GPS-IMU\(^2\) system (Figure 3.1); all relevant data necessary for lane and obstacle detection was saved during data collection to an onboard data-acquisition computer.

\(^1\)Derived from the combination of the words “light” and “radar” [55].
\(^2\)Global Positioning System, and Inertial Measurement Unit.

Figure 3.1. LIDAR and GPS systems mounted on a vehicle
3.1 Sensor Setup

A description of each of the sensors that comprise of the vehicle-borne mapping system is provided below.

**LIDAR sensor:** The primary component of the vehicle-borne LIDAR system is a SICK LMS 511-10100 PRO mounted on the rear of a vehicle, as shown in Figure 3.2. Returning range and intensity, the LIDAR system has a detection range of 80m, a measurement variance of ±6mm, and a user-configurable sampling speed from 25 to 100 Hz, and angular resolution from 0.1667° to 1°. For these experiments, the sensor was set to an angular resolution of 0.1667°, and a scan speed of 25 Hz revolution rate.

The LIDAR could also be reconfigured to hang off the side of the vehicle as illustrated in Figure 3.3. This configuration allowed for detection of hazards farther to the right of the drivable road including pavement edge drop-offs. This positioning was necessary to capture steep drop-offs that would be occluded if measured from the center rear of the roof area, which is the normal scanning position for lane-marker mapping.

**GPS-IMU sensor:** The vehicle is also equipped with an integrated GPS-IMU Novatel SPAN system to collect the position and the orientation information of the vehicle.
Figure 3.3. Side mounted LIDAR system

This is a defense-grade system whose position errors in the latitude and longitude data, with full satellite visibility, are about 2 meters ($\sigma$) and the errors in the orientation angles are $0.017^\circ$, $0.02^\circ$, and $0.042^\circ$ ($\sigma$) for the roll, pitch, and yaw angles, respectively.

**Additional Equipment:** Apart from the SICK LIDAR and Novatel system, additional electronic equipment consisting of a back-up power distribution system and embedded computer are stored inside the vehicle. The data from the LIDAR system and the GPS-IMU unit are collected onto the computer. After returning to the laboratory, the data are post-processed to extract the features that define the environment.

### 3.2 Power System

A mapping trip may consist of an entire day of driving along hundreds of miles of roadway; thus, it is necessary to create a reliable power system to ensure that the sensors stay powered and perform correctly without faults for long durations.

Powering a network of sensors requires a reliable power source. The vehicle’s alternator is generally insufficient: it has severe voltage fluctuations as a result of engine speed and load, and must be shut off during refueling which, if there were no backup power, would require a lengthy shut-down and start-up process of the mapping system.
Figure 3.4. Power supply of mapping vehicle with back-up charging capabilities

An auxiliary 12V battery was designed into the power system to supply a steady voltage and serve as the primary power source to the mapping equipment. Keeping this battery charged for the duration of a mapping trip requires constant back-up charging which can be attained from the vehicle’s alternator or a 110VAC power supply when available. The configuration of the power supply with back-up charging can be seen in Figure 3.4.

Further, a power monitoring system was implemented within the system to allow for direct observation of the status of the power supply. The aforementioned power supply configuration allows for several hours of operation until the supply voltage drops below 11.8V (a voltage threshold representing a dead battery and that could allow for ‘brown outs’). Tests of the power system revealed that the data collection system could withstand approximately three hours of data collection until discharged to this threshold from a fully-charged battery (Figure 3.5).

3.3 Sensor Configuration

The supply voltage is distributed to each sensor through a 12 VDC power controller and distribution system. This creates a network of powered sensors all from the same supply and with each connected to the data-acquisition computer through either a high-speed gigabit network switch or USB. In Figure 3.6, the power originates from the left side and branches to each component through the red connections. The grey connections represent data transfer. One can observe that the data transfer through the USB connection from the Novatel Span unit to the computer create a loop, one that is susceptible to ground faults. This is unique to the USB port as USB transfers power through a pin whereas Ethernet does not. This potentially harmful ground fault loop was avoided through the
3.4 Experimental Protocol

The resulting system of sensors and power distribution equipment is contained in 2 completely assembled units: a roof rack housing the LIDAR, Novatel GPS/IMU system, power distribution and monitoring components; and a bin housing the auxiliary battery, installation of a serial RS-232 optical isolator.
battery charger, and embedded computer. To set up this system, the experimental procedures for setup are relatively simple and include:

3.4.1 Mapping Procedure
The procedure for mapping an environment for perception involves the following steps:

- Connect to data-acquisition computer
- Sensor calibration
  - Calibrate the LIDAR with the IMU
- Verify data stream for each component
- Record data streams

The embedded computer that manages data collection is not easily accessible from the front of the vehicle. To interface this computer while driving, the embedded computer is controlled through a secure shell from a connected laptop.

Prior to each mapping run, the sensors are calibrated. It is important for the LIDAR and IMU to be in time sync when processing the data. In order to ensure that the time origin of both components are aligned, the vehicle is rocked back and forth in roll and pitch while parked above a stationary surface. In post-processing, this rocking gesture illustrates any temporal offset between the two components as an artificial motion of the static ground plane. The timing between the LIDAR and IMU systems can be adjusted accordingly to eliminate this effect.

The Robot Operating System (ROS) was used as the core software to launch the data streams of each sensor and to collect data. ROS maintains the same master clock on all incoming data streams to ensure a synchronized time stamp. To verify healthy data streams of each sensor, the operator can echo each sensor’s stream and run a built-in GUI visualizing tool to verify the streams are within appropriate bounds.

Next, the mapping vehicle is driven through an environment to collect data. The data stream produces a profile of the environment visible to the downward-facing LIDAR, with a swath of range points measured at each revolution of the laser. These profile scans, coupled with the corresponding vehicle pose, generate a 3-D representation of the roadway environment. At 25 Hz, the mapping vehicle can collect approximately 16 MB of raw LIDAR data per minute.

As the final step, this data is processed offline to detect obstacles and accordingly, drivable areas.
Methodology

The goal of this thesis is to detect multiple types of obstacles as mentioned in the introduction, quantify their level of hazard, and prepare their relevant data for storage in compact feature maps. The goal of compactness requires conversion of an entire 3-D point cloud to only a handful of points representative of certain obstacles that are relevant to driving decisions. This compactness saves on processing time, data storage size, and cost of the implemented vehicle.

The compact feature map is used to indicate detected obstacles for the purpose of a warning system. This usage, however, falsely suggests that the environment is unchanging. Roadways can experience variation over time usually caused by construction, vegetation changes, collisions, parked vehicles, etc. Thus, there is a need to routine update maps of the environment and the drivable areas.

Routine updating of obstacles’ locations requires dependable classification techniques for each type of obstacle. As stated earlier, the obstacle classes considered in this study are: rollover, positive obstacles, and pavement edge drop-offs. The following procedure summarizes the entire process of offline detection and analysis of obstacles that produces a warning system:

1. Data collection with mapping vehicle
2. Offline obstacle detection and boundary creation
3. Map generation of road obstacle boundaries and hazard levels
4. Warning system implemented on non-mapping vehicle
The entire pipeline of the road-side obstacle detection and mapping process (Figure 4.1) can be broken into the following categories, with each category performing a specific function within the process:

1. Preprocess Input Signal
2. Lane Marker Extraction
3. Rollover Detection
4. Collision Detection
5. Edge drop-off Detection
6. Non-Drivable Region Map Generation
7. Heads-Up Display

Each of these functions is explained in further detail in the sections that follow.

### 4.1 Preprocessing

Preprocessing raw LIDAR data is an important step in removing sensor errors and isolating relevant information about roads and road obstacles. The SICK LMS-511, like any sensor, is susceptible to noise and outliers caused by missed reflections. The data stream from the SICK LMS-511 LIDAR returns 1142 floating points and 1142 integers at 25
Hz. When mapping a large distance of roadway, the amount of LIDAR data quickly escalates (e.g. 1000 miles at 40 mph yields 2,250,000 scans, thus 5,139,000,000 data points total). With such a high data rate, it becomes difficult to remove all outliers in order to ensure minimum-error data that is necessary for the succeeding functions. Therefore, it becomes imperative to preprocess the data to verify that subsequent functions do not give erroneous data in the presence of range errors. The preprocessing module is broken into the steps summarized in Figure 4.2.

To mitigate noise and isolate relevant data, first the outliers in range measurements were removed, and the data scan was cropped to include only the region of interest. In this process, single outliers of both range and intensity data are replaced by the mean of their adjacent data points.

Next, raw range data is transformed into a Cartesian coordinate frame. The LIDAR’s 1142 range data points are spanned at equal increments from -5° to 185°. Thus the product of each data point with the cosine of it’s corresponding angular position will give that point’s relative $x$ position. Conversely, the negative of the product of each data point with the sine of it’s corresponding angular position will give that point’s relative $y$ position. By creating a diagonal matrix where the diagonal elements incrementally range by $0.1665^\circ$ (i.e. $\cos -5^\circ$, $\cos -4.8335^\circ$, $\ldots$, $\cos 184.8335^\circ$, $\cos 185^\circ$) and all off-diagonal elements are zero. These calculations can be expressed as the transformations seen in Equations 4.1 and 4.2.

$^1$Negative because the LIDAR is downward-facing.
When a LIDAR sensor is scanning a flat surface such as a road, there is uneven spatial measurements of the road due to the fixed angular resolution of the LIDAR. Objects detected directly under the sensor will have a larger point density than those detected closer to the horizons, as illustrated in Figure 4.3. With an uneven density of data points across the road surface, the detection of obstacles and road geometry using a single algorithm becomes quite difficult. By selecting a region of interest almost directly under the LIDAR, we remove areas of low density. This region of interest is found by looking at each LIDAR scan individually, searching for large “jumps” in position data from point to point. Starting at the middle index of each scan (which usually represents a point on the roadway directly under the LIDAR sensor), the algorithm steps outward looking for jumps in position greater than 1m.

With an angular resolution of 0.1667°, on an ideal flat surface with the LIDAR as shown in Figure 4.4, the maximum width of road surface detection can be calculated as follows:

\[
\begin{bmatrix}
  x_{1,1} & \cdots & x_{1,1142} \\
  \vdots & \ddots & \vdots \\
  x_{N,1} & \cdots & x_{N,1142}
\end{bmatrix}
= \begin{bmatrix}
  d_{1,1} & \cdots & d_{1,1142} \\
  \vdots & \ddots & \vdots \\
  d_{N,1} & \cdots & d_{N,1142}
\end{bmatrix}
\times
\begin{bmatrix}
  \cos 5^\circ & \cdots & 0 \\
  \vdots & \ddots & \vdots \\
  0 & \cdots & \cos 185^\circ
\end{bmatrix}^2
\]

\[
\begin{bmatrix}
  y_{1,1} & \cdots & y_{1,1142} \\
  \vdots & \ddots & \vdots \\
  y_{N,1} & \cdots & y_{N,1142}
\end{bmatrix}
= -\begin{bmatrix}
  d_{1,1} & \cdots & d_{1,1142} \\
  \vdots & \ddots & \vdots \\
  d_{N,1} & \cdots & d_{N,1142}
\end{bmatrix}
\times
\begin{bmatrix}
  \sin 5^\circ & \cdots & 0 \\
  \vdots & \ddots & \vdots \\
  0 & \cdots & \sin 185^\circ
\end{bmatrix}
\]

With the height of the LIDAR at 1.35m above ground, this allows for a maximum range of 42.5m, which corresponds to a scan distance of 21.25m on either side of the LIDAR sensor.

\[2h \tan \theta = x\]  

[Note this is a diagonal matrix]
Once the indices are found that mark discontinuous jumps, the data outside of these indices is cropped for that particular scan. This, however, results in LIDAR profiles of different widths. To prevent error in further functions, the data scan is disregarded if the total length of the scan is less than 0.45m, as a data scan of this length will not result in enough data points to viably classify any features.

Next the data is interpolated over the $x$ direction to a resolution of 3mm equal spacing in the $x$ direction to remove uneven spatial sampling caused by geometrical positioning of
the LIDAR. In order to perform interpolation, the data must be monotonically increasing in the $x$ direction. By marching through each index, any data that shares an $x$ value and has the smaller $y$ value is removed (Figures 4.5 and 4.6). Interpolation also prepares for future spatial convolution techniques performed on the position and intensity data.

Finally a filtered set of position data is created using a 2nd order Butterworth filter with cutoff frequency $\frac{0.2}{10}$. The entire preprocessing procedure largely removes all common
error sources, as seen by comparing the raw LIDAR data in Figure 4.7 (top) to the processed data (same figure, bottom plots).

### 4.2 Lane Marker Extraction

In order to designate the position of an object relative to the lane center, a local path-fixed reference point is needed that indicates the origin of measurements of lane features. Most conveniently this reference point is a lane marker or better yet, the lane center. However, these reference points require a successful detection process of the lane markers’ location. To find a lane stripe within LIDAR data, a mask representing the expected intensity variation is convoluted across the LIDAR scan’s intensity data. The goal is to identify either a double or single lane marker (see Figure 4.8). Considering that lane marker extraction is usually conducted in this research as a post-processing map-generation technique, we assume the data collection vehicle is in the lane thus signifying that lane markers could be seen in both the positive and negative realms of the LIDAR data, relative to vertical.
4.2.1 Convolution

Lane marker features are extracted from the LIDAR’s intensity using a convolution method. Assuming the data collection vehicle is within the lane, a binary kernel for a single lane marker, as seen in Figure 4.9 is convolved across the positive section of the intensity data. Conversely, a binary double lane marker mask is convolved across the positive intensity data. Training data of intensity data determined the width of the single lane stripe mask to be $0.1 \pm 0.05m$ and the width of the double lane stripes to be $0.1 \pm 0.05m$ with a $0.15 \pm 0.05m$ space between the stripes.

Once the masks are convolved, the peaks from the convolution (seen in Figure 4.10) are examined and the maximum peak is determined resulting in a match for the lane marker center. This process is repeated on the positive intensity data with the double lane marker mask.

Positive matches of both lane markers provide the information needed to calculate the lane center. The arithmetic mean of the position of each lane marker center is assumed to be the lane center.

4.2.2 Error Mitigation

Due to the nature of convolution, peaks can arise in areas that do not have the lane marker features. Most notably, in the double lane marker case, as the mask begins to

---

3The single lane marker appears to the right of the vehicle due to the orientation of the downward-facing LIDAR.
pass over the double marker, the right stripe from the mask will align with the left stripe of the intensity data. And as it continues to pass, the mask will align with both stripes, then the left stripe from the mask will align with the right stripe of the intensity data. This phenomena culminates in a result similar to Figure 4.11. The figure shows the lane marker center (the space between the lane stripes) as the max peak near 1.5m. As seen, the pseudo alignment results high peaks roughly half the size of the max peak,
thus resulting in a lower second-peak ratio and possibly a higher signal-to-noise ratio. Unfortunately, these peaks falsely represent noise. This observation leads to a desire to mitigate these occurrences.

Variance analysis of the convolution result was next used to find the left and right lane marker edges. This variance analysis assumes knowledge of the lane marker width and surrounding pavement; in essence, it is assumed that the pavement reflectivity near the marker provides a good estimate of background reflectivity from which the lane markers can be found. Specifically, the variance in reflectivity is calculated from data sampled between 0.6m and 0.25m before the lane marker, and then is used to calculate a threshold that is four standard deviations different than the mean, \(4\sigma\) above the background. This threshold is used to define the transition from unmarked pavement to a painted lane marker. To use this threshold, points are evaluated marching toward the lane marker center from 0.6m, and the first index to exceed the set threshold is denoted as one edge of the lane marker. This process is repeated on both sides of each lane marker (single and double) resulting in a location of the lane marker edges and lane marker width since the convolution result alone can only detect the lane marker’s center location.

Further, the lane edge information is useful to eliminate false-positives that may be detected as peaks in the correlation filter. If it is known that peaks in correlation appear to represent a lane marker in variance also, then these lane markers outside the current

\[X \times 10^4\]

\[\text{Arbitrary Units}\]

\[\text{X (meters)}\]

\[0.9\]

\[1\]

\[1.1\]

\[1.2\]

\[1.3\]

\[1.4\]

\[1.5\]
lane can be avoided as being classified within the current lane.

Even with both of these methods, false detection of potential lane markers does occur. In this work, if the correlation peak of the false positive is higher than 76.9% of the main peak\textsuperscript{4}, then the main peak cannot be accurately classified as a lane marker center. Further, if the variance analysis indicates that there is a signal-to-noise ratio less than 1, then the lane marker scan is tagged as having no clear lane marker detection.

4.3 Rollover Detection

Once the lane center position has been found for use as a reference point in feature localization, then obstacle detection begins. The first of the road obstacles examined for mapping is vehicle rollover. This obstacle is vehicle specific as it depends on the size and weight distribution of the intended vehicle. Rollover collisions can be categorized as untripped or tripped. The disturbances that generally cause vehicle roll are either tire forces, inertial effects, and gravity, caused by cornering maneuvers (untripped), or vehicle stance on irregular road surfaces (tripped) \cite{56}. Because this thesis covers offline detection of a static environment for warning systems, the vehicle’s dynamics are assumed to be unknown and it is assumed that the inertial forces have no effect; thus, the primary rollover collision metrics are based principally off static vehicle configuration in its environment. The rollover detection module can be broken down into the steps summarized in Figure 4.12.

In order for a vehicle to avoid rollover, the center of mass of the vehicle must lie within the vehicle’s track; otherwise, the vehicle is considered likely to rollover. The word “likely” is used here because one cannot completely discount all vehicle dynamics. Roll motion can be excited or mitigated by inertial forces, and thus rollover can occur at slopes less than the calculated static rollover angle, or may not occur at the static rollover angle. To be conservative in this study, a rollover margin of 20° is added to any static roll threshold to account for potential inertial forces that may also tip over the vehicle. Chen et al. \cite{57} developed models for roll thresholds indicating axle liftoff in tractor trailers. Their threshold was a roll angle of 3.41°. This thesis inflates that threshold to 20° to account for more stable vehicles. A study measuring the Tilt Table Ratio (TTR) of vehicles with Static Stability Factors (SSF) ranging between 1.06 and 1.44 measured the average TTR to be 1.04 (45.86°) \cite{58}. This average TTR from that study is used as the nominal static roll threshold in this thesis.

\textsuperscript{4}This implies that the second-peak-ratio is less than 1.3.
To determine all possible rollover slopes in the areas surrounding the roadway, the vehicle stability is evaluated at each lateral position within a particular profile. To do this, each LIDAR scan is evaluated at each lateral point by sliding the vehicle’s width across the Earth-fixed representation of the geometric profile and determining the contact points of the right and left tires within that profile. The approximate static roll angle of the vehicle is then obtained from the slope of the line connecting the two contact points, assuming the vehicle’s heading is normal to the plane of the LIDAR scan and that the suspension deflection is negligible. Figure 4.13 shows how the rollover slopes correspond to the vehicle’s configuration in the LIDAR scan. The plot in red represents the cross-section of the environment, whereas the blue plot represents the slope of the specific vehicle when its center lies at that particular lateral position.

The calculation of the vehicle roll stability at each position in the road enables the calculation of a boundary for safe paths free from rollover. This calculation requires only locations on a scan that would cause rollover for that specific vehicle. Referring again to Figure 4.13, the calculated rollover slope for a specific vehicle may be between 0 and 0.4 (where the peaks are). But, assuming that the rollover threshold of the vehicle is 0.27, one can represent a rollover boundary by only storing the offsets from the center of the lane that first encounter slopes greater or equal to 0.27. This reduces the size of the
map collected from thousands of data points per scan, to only two or less lateral offsets, thus reducing the size of the feature map substantially.

As reference, for a mile of roadway (with a LIDAR scan every 0.3m), the size of a 3-D point cloud is approximately 92MB. The size of a point cloud of only road surface data similar to [54], considering the LIDAR mounted 1.5m above the road surface, would be approximately 66MB. Assuming that every scan produces unique rollover thresholds, this worst-case rollover boundary map would only be 2MB.

### 4.4 Positive Obstacle Detection

To avoid collisions with fixed obstacles near the road’s edge, positive obstacles are detected and mapped. Before detection can be performed, a specific definition of a positive obstacle must be defined. For the purposes of this thesis, a positive obstacle is classified by fulfilling the following criteria:
1. Greater than $70^\circ$ from the surface of the road. This is an angle at which tire scrubbing is probable to occur and at which a vehicle’s bumper would collide with the obstacle [16, 59].

2. The obstacle transitions from the flat road to a $70^\circ$ slope within 0.1m (e.g. the width of the tire patch). Otherwise, the vehicle could drive up the obstacle.

3. The height of the object is greater than 0.2m, roughly the clearance distance of most vehicle axles. And greater than the maximum height at which tires can climb in scrubbing (0.127m) [59].

In this thesis, positive obstacles are detected through an analysis of the range data’s change in slope in the lateral direction, $\frac{\partial^2 y}{\partial x^2}$. This analysis involves two approaches to the feature extraction problem, followed by a combination of the extraction results then used to classify detected objects as positive obstacles. The positive obstacle detection module is broken into the steps summarized in Figure 4.14.

---

**Figure 4.14.** Positive obstacle detection workflow
4.4.1 Feature Extraction

4.4.1.1 Ramer-Douglas-Peucker

The first feature extraction method utilizes the Ramer-Douglas-Peucker (RDP) algorithm. As outlined in Algorithm 1, this algorithm divides a curve recursively. By selecting vertices (initially the first and last data points of the sample) and fitting a line between them, the other data points between the adjacent vertices are evaluated with respect to a distance tolerance, $\epsilon$. If a given point between two adjacent vertices is closer than $\epsilon$ to the fitted line, that data point is removed. The data point with the largest normal distance greater than $\epsilon$ becomes a new vertex. The process is then repeated until no extra data points lie outside the normal threshold distance and only vertices remain. With regards to a LIDAR scan, this reduces the dimensionality of the data sample by reducing the number of data points (vertices) within the scan thus creating line segments of varying lengths that connect the data points (vertices). The vertices generated signify locations of a large change in slope, which can signify a potential positive obstacle. The corresponding line segments between adjacent vertices represent the slope of said potential obstacle.

Set first and last index of data as initial vertices
Fit a line segment between the vertices
Detect point $P$ with maximum normal distance $d_P$ to the line

while data point $P$ lies outside threshold do
   Fit a line segment between each adjacent vertex;
   Between each set of adjacent vertices, find $d_P$;
   if $d_P$ is larger than tolerance $\epsilon$ then
      set $d_P$ as a new vertex;
   end
end

Merge all segments

Algorithm 1: Ramer-Douglas-Peucker algorithm

The tolerance, $\epsilon$, is used to determine the resolution of the resulting data segmentation. This implies that changing the tolerance changes the number of vertices. Increasing $\epsilon$ decreases the number of vertices, and decreasing $\epsilon$ increases the number of vertices. Because vertices signify potential positive obstacles, in an ideal driving scenario (no obstacles within the lane), it is desirable to have no vertices on the road surface. To limit the number of vertices on the road surface, $\epsilon$ should be related to the standard deviation of the road surface. If the road surface is analyzed between the detected lane markers, then one can perform linear regression of the road surface to obtain the standard devia-
tion, \( \sigma \), of each data point to the best fit road surface. The resulting \( 4\sigma \) from the linear model of the road acts as the tolerance in the RDP algorithm.

### 4.4.1.2 Second Derivative Analysis

The RDP algorithm’s effectiveness of line segmentation can stand alone as a single feature extraction method, however there are scenarios where it will lead to a false identification of positive obstacles. These scenarios are gradually increasing slopes analogous to a roadside embankment. The RDP algorithm will identify several points on the gradually increasing slope as vertices, and any line segment on that slope that exceeds an angle \( \phi \) (70° in this thesis), corresponding to the vehicle’s ability to “climb” the object, and a height of 0.2m is classified as a positive obstacle. An analysis of the data sample’s change of slope (second derivative) would disregard a gradually increasing slope as a positive obstacle, for it is more likely to account as a rollover rather than a collision. Thus the data is run through an additional feature extraction method to observe the second criteria of a positive obstacle, searching for data points that represent abrupt changes in slope. Thus the data is run through an additional feature extraction method to observe the second criteria of a positive obstacle that the obstacle must transition from the flat road to a slope of \( \phi \) within a lateral distance \( dx \) (0.1m in this thesis).

The second feature extraction method selects data points from the original data sample that exceed a second derivative threshold. This threshold is defined based on an object with geometry that would not allow a vehicle’s tires to climb at high speeds, resulting in damage to the vehicle’s body or supporting framework. This feature extraction technique avoids the false positive detection of gradually increasing slopes as seen in Figure 4.15. When \( \frac{\partial^2 y}{\partial x^2} \) exceeds a threshold representing an abrupt change in slope and \( \frac{dy}{dx} \) exceeds 70°, then a potential positive obstacle has been detected. The threshold relates to the definition of a positive obstacle, that is, the obstacle must be greater than 70° from the road surface, and the slope must increase to its greatest slope within 0.1m. The calculation of the second derivative threshold, \( \varepsilon \), can be expressed as:

\[
\varepsilon = \left| \frac{\tan (\theta - \phi)}{dx} \right|
\]

(4.5)

Where \( \theta \) is the angle of the road surface, \( \phi \) is the angle threshold used to define the first criterion of a positive obstacle (70° in this thesis), and \( dx \) is the lateral distance that makes up the second criterion of a positive obstacle (0.1m in this thesis).

This second derivative feature extraction technique extracts the data indices that
mark an abrupt change in slope. Figure 4.16 illustrates the process. The third subplot, representing the second derivative, shows the second derivative threshold boundary indicated by the red lines. The green highlighted indices represent the indices that exceed the second derivative threshold. The intersection of this extracted set of features with the RDP extracted set filters out extracted points that are not a good representation of abrupt changes in slope. Thus all points the exceed the second derivative threshold are extracted rather than the first data point that exceeds it. This creates clusters of extracted feature points. Each cluster of data points that exceeds the second derivative threshold represents an object with an abrupt change in slope. Take note of the green highlighted indices between the threshold bounds. This ensures that each cluster of data corresponds to a single object. This ensures that there is only one cluster of extracted indices for every object, which is a necessary requirement for the feature classification step.

4.4.1.3 Combination

Once both sets of extracted data are determined, the results of the feature extraction techniques are narrowed even further to a set of vertices that are more specific, more accurate, and a better representation of abrupt changes in slope. Potential outliers from the individual extraction techniques are removed by finding the intersection of both sets of extracted data sets. The intersection of the extracted data sets from the RDP
algorithm and the second derivative analysis result in a more specific set of data points corresponding to potential positive obstacle corners (see Figure 4.17. In this figure, the first plot shows both sets of extracted data sets. The second plot shows the intersection of those two sets resulting in a better representation of the obstacle’s corners.
4.4.2 Feature Classification

The extracted set of data points represents potential positive obstacles. To classify those points as true positive obstacles, they are required to meet a minimum height. The analysis of the obstacles’ height fulfills the third and final positive obstacle criterion. Simply, the first and last extracted point within each cluster of exceeded second derivative data are used to measure the height. The difference in $y$ represents the height. If this value is greater than 0.2m (roughly the clearance distance of most vehicle axles), then those points are classified as a positive obstacle, and the point closest to the road surface represents the boundary edge of the obstacle. Further, this height analysis is also used to analyze the obstacle’s severity and traversability.

4.5 Negative Obstacle Detection

Negative obstacles are detected in the same fashion as positive obstacles i.e. analysis of second-derivative position data in the lateral direction governed by slope and height thresholds.

An additional aspect of negative obstacle detection encompasses the detection of obstructed regions of data. When dealing with negative obstacles, unless the scanning device is directly above the depression, a corner of the obstacle will be occluded (see Figure 4.18). This phenomenon can be a hindrance in the classification and severity analysis of the detected obstacle. Data representing the sidewalls of the depression would be non-existent, thus extrapolation techniques would be necessary to estimate the depression’s geometry (depth and slope). Although, without an accurate measurement of the depth and slope of the negative obstacle’s depression, proper classification is impractical.

Studies have addressed the problem of detecting occluded negative obstacles by checking for gaps in the data and extrapolating depths of negative obstacles using a rule-based algorithm similar to [33]. However, due to the many assumptions made regarding occluded negative obstacles, a side-mounted LIDAR was used in this thesis to collect unobstructed data of a particular kind of negative obstacle, pavement edge drop-offs (pictured in Figure 4.19). In this thesis, a pavement edge drop-off is defined as a vertical elevation difference of 0.1 meters between two adjacent roadway surfaces either paved or unpaved [59]. Further, a pavement edge is only classified as a drop-off if the slope of the edge is greater than 50°.
4.5.1 Pavement Edge Drop-off Detection

During post-processing, the slope and length of each road-edge profile is measured. The process used is similar to the RDP algorithm employed in the positive obstacle detection algorithm. First, a best-fit line of the road surface is produced from data points near the vehicle. Then, iteratively marching toward the road edge from the lane marker, the normal distance from the best-fit line to the data is calculated. Once a threshold distance has been surpassed (8mm was found to be a robust value), that threshold data point signifies the end of the pavement top and the beginning of the road edge drop-off.
Figure 4.20. Detection process – finding best-fit line of road surface

A similar process is then repeated for detecting the drop-off. A line is drawn through the threshold data point, parallel to the road surface. This line is then rotated toward the outside of the pavement, pivoting on the threshold data point until the density of data points within a fixed distance from the line (8mm) is at a maximum. That angle of this “most enveloping” line represents the angle of the drop-off. Next, the algorithm marches along this best fit line for the drop-off, calculating the perpendicular distance from this second best-fit line to the measured road-side data. Again, once a threshold distance is surpassed (14mm), that second threshold data point signifies the end of the drop-off. The distance between the two threshold data points is considered the length of the drop-off, and the angle from horizontal between each best-fit line is considered the slope of the drop-off. The entire process of pavement edge drop-off detection is visualized in Figures 4.20 through 4.23. This success of this process on simulated uniform pavement edges can be seen in Appendix B.

4.5.2 Non-Uniform Edge Drop-off Detection

The above detection process is simple; however, it uses arbitrary thresholds for detection and also assumes that pavement edge drop-off maintains a uniform slope along its length (Figure 4.24). In many common roadways, pavement edges are damaged or eroded, and indeed within the research group there is an active project to measure this damage increase over time for selected roadways. Figures 4.25 and 4.26 show an example of an
eroded pavement edge.

These areas of pavement edges are of most interest to this study due to their highly non-uniform geometry. Their significance is that they may pose more of a danger to passenger vehicles because the original pavement edge slope has been eroded, resulting in a more extreme slope. Further, their geometries present a complication in the definition of a drop-off; with such an uncertain shape, determining representative characteristics
of the edge (slope and length) is not easy.

The geometries also pose a problem to the data processing algorithm explained above. Because there is not a single continuous slope transitioning from the pavement to the shoulder, the algorithm returns results that depend strongly on the location, size, and amount of debris and vegetation around these eroded areas. Because such features are quite irregular and change quickly in the direction of travel, highly variable drop-off
measurements (as a function of travel) are obtained as different areas are tagged as the “best fit” slope. Further, this illustrates how an implementation of the Ramer-Douglas-Peucker algorithm would fail in depicting an accurate slope, and further fail in properly categorizing the hazard level.

To illustrate the difficulty of processing the irregular pavement edges, a scan of the non-uniform pavement edge is shown in Figure 4.25), which corresponds to the drop-off
Figure 4.27. Sample profile scan of non-uniform pavement edge

seen in Figure 4.27. As evident in the figure, the position of the start and end of the pavement edge is uncertain. Even in the photo, the difference between pavement and shoulder is not entirely clear, as the grey area underneath the black asphalt is simply an older pavement that has deteriorated but still maintains some angle to the soil surface. Further, the soil surface is intentionally banked at nearly the same drop-off profile as the pavement. Thus, there is extreme ambiguity where the pavement is considered to end and the shoulder to begin.

To illustrate the effects of this uncertainty, the algorithm returns the results shown in Figure 4.28. The plot shows the best fit of the road marked in red, and shows the algorithm tagging the pavement edge as occurring between the two dark crosses; this is indeed the correct pavement slope that the original pavement was built with, prior to damage. However, a manual inspection might perceive that the pavement edge in fact does not end at the second cross, and might classify the eroded edge as the true slope. Further, with such damage present, it may be a more appropriate to locate the damaged region as the start of the pavement edge drop-off.

Because of the extreme variability shown in these specific scans, a new method was implemented to prevent the algorithm from measuring ambiguous pavement edge drop-offs falsely. To detect non-uniform pavement edges, the raw data is filtered and an inflection point of that filtered data is found to the right of the end of the road (see Figure 4.29. A tangent line is drawn through the inflection point (Figure 4.30) and
the deviation of the distance from the raw data to the tangent line is calculated and shown in (Figure 4.31) as the distance between the green dashed lines. The higher the deviation, the more non-uniform the slope is. Analysis of severely eroded edges has shown that, if the deviation of the slope surpasses 10 times the deviation of the road surface, then the slope is likely to be non-uniform and severely degraded (e.g. washed-out or damaged). Figure 4.32 shows the complete process of detecting and measuring non-uniform pavement edge drop-offs. For purposes of vehicle safety, it is generally sufficient to note the presence of severe pavement damage nearby the road edge, rather than quantify exactly the “correct” slope.

The above procedure is able to measure and identify both uniform and non-uniform pavement edges, validated by testing the process on a uniform edge. The results, shown in Figure 4.33, show the capability of the defined process on a uniform pavement edge.

### 4.6 Path Planning/Warning System

The above methods produce a classification of potential threats to vehicle motion nearby the current lane. These are readily stored in a database and even presented to a driver as a warning. One possible warning system is a “heat map” of threats shown through a heads-up display. This warning concept entails displaying every feature in one map with each feature labeled as individual obstacles. When the driver approaches the location of an obstacle, an auditory warning system can direct the driver’s attention to the obstacle.
that he/she is approaching. Figure 4.34 illustrates the heads-up display “heat map” concept by projecting obstacles’ locations onto the windshield for the driver to see. This example figure only displays one hazard marked in red.

Accordingly, each obstacle detected is weighted by severity. Rollover takes a high severity weight due to the extreme severity of such accidents; similarly, non-rollover angles are weighted slightly less proportional to an obstacle’s proximity to the rollover
threshold. Positive obstacles are weighted based on the height of the obstacle. Pavement edge drop-offs are weighted based on their slope and length. These severities become useful in certain operating situations where split-second decision making is critical. For instance, in a scenario where the driver needs to depart the drivable area for some reason and his/her options are between driving over an $20^\circ$ embankment or into a 1m tall barrier, the operator could be advised to travel over the embankment instead of

**Figure 4.31.** Detection process – calculating standard deviation of slope data

**Figure 4.32.** Non-uniform detection method
Figure 4.33. Uniform pavement edge tested with detection algorithm

Figure 4.34. Example of a heads-up display presenting a “heat map” of surrounding hazards through a possibly immobilizing positive obstacle.
Chapter 5

Experimental Results

This chapter explains the implementation and outcomes of the aforementioned obstacle detection processes, specifically examining the success and failure for each process. Data was collected from the Thomas D. Larson Pennsylvania Transportation Institute’s (LTI) Test Track facility (Figure 5.1), and this data source was used in the sections that follow for rollover detection and positive obstacle detection. Data was also collected with the side-mounted LIDAR, and this was specifically used to test pavement edge drop-off detection. This data was collected from a 6 mile stretch of road near West Pine Grove Road, State College, PA, pictured in Figure 5.2.

Figure 5.1. Thomas D. Larson Pennsylvania Transportation Institute’s Test Track facility
5.1 Preliminary Results

5.1.1 Rollover Detection

An example of the result of rollover detection applied to a single scan of the LIDAR cross-section can be seen in Figure 5.3. The corresponding plot of the severity of the rollover slopes is seen in Figure 5.4. Figure 5.4 shows that there is a noticeable peak in roll slope to the right of the lane center. This signifies that if a vehicle’s center were to reach approximately 3.3 meters right of the lane center, then the vehicle could experience a roll slope of approximately 0.34 (34°). Depending on the vehicle, this is large enough that it may induce rollover. This data is translated to the profile of the road in Figure 5.3 marked with “x”s.

Additional analyses of slope calculation for rollover detection can be seen in the Appendix A, with each representing the roll slopes of the vehicle at each lateral position in a particular LIDAR scan.

Comparable to Figure 5.4, Figure 5.6 shows approximately 20% grade slopes near ±0.811m left of the lane center. (Note that this figure depicts a single lane bridge bounded by a Jersey barrier.)
Figure 5.3. Rollover detection

Figure 5.4. Rollover detection slopes
Figure 5.5. Rollover detection result

Figure 5.6. Rollover detection result slopes
5.1.2 Positive Obstacle Detection

Similar to rollover detection, positive obstacle detection results in labeled locations near the lane that appear to have obstacles present. The height of the obstacle is used hereafter as a measurement of the positive obstacle’s hazard level. Figure 5.7 illustrates the resulting detection of a properly classified obstacle. In this particular profile of the road at location (40.86278, -77.83360), a positive obstacle lays approximately 9.25m to the right of the lane center. The height analysis reveals this obstacle to have a height of 0.74m. Figures 5.8 through 5.10 illustrate the feature extraction process.

An important aspect of positive obstacle detection is that it provides the ability to detect multiple obstacles in one profile in order to determine which near-road obstacle is most threatening to the driver. This thesis does not model a decision making process; however, it does measure multiple lateral offsets and heights simultaneously and thereby provides a significant amount of data suitable to assist in decision-making. Figure 5.11 illustrates detection of multiple obstacles in one profile. Additional analyses of positive obstacle detection can be seen in the Appendix A.
Figure 5.8. RDP resulting corners (ex. 1)

Figure 5.9. Close up of RDP corners (ex. 1)
Figure 5.10. Second derivative analysis of data (ex. 1)

Figure 5.11. Detection of multiple positive obstacles: obstacle of height 0.803m detected at 2.282m left of the lane center, obstacle of height 0.809m detected at 2.598m left of the lane center
5.1.3 Pavement Edge Drop-off Detection

Figure 5.12 shows the measurement of a pavement edge drop-off (Figure 5.13) using the method described in Chapter 4. In this figure the road surface is to the left of the drop-off. The drop-off end marked with the cyan circle at (-0.34, -0.46) represents the edge of the pavement and the second cyan circle represents the end of the drop-off. This particular edge drop-off is 0.3872m long and 23.27° and it lays 1.1781m to the right of the single lane marker. Additional analyses of pavement edge drop-off detection can be seen in the Appendix A.

5.1.3.1 Failure Analysis of Positive Obstacle Detections

The methods of classifying near-roadway hazards are not perfect, and there are certain scenarios that will regularly cause the algorithms to fail. For example, the Ramer-Douglas-Peucker algorithm may not always segment a surface to a high enough resolution to resolve obstacles, nor does it always select appropriate data points as vertices. Another factor is that the calculation of the second derivative of a data series is a noise-sensitive process, and because this second derivative is used, the results are sometimes in error. At certain locations there may be a critical vertex, but the second derivative analysis
Figure 5.13. Pavement edge drop-off

fails to highlight the vertex as data above the second derivative threshold. Figure 5.14 shows the RDP vertices and the data exceeding the second derivative threshold. As seen, the vertex at (-2.07, -1.16) does not intersect with the second derivative selection. This indicates that there is no “top” of the obstacle detected, and therefore a height cannot be calculated. Thus this positive obstacle is not correctly classified.

5.1.3.2 Failure Analysis of Pavement Edge Drop-off Measurements

The pavement edge drop-off measurement is strongly dependent on LIDAR data, but because this data is sometimes noisy, the algorithm too experiences inaccurate measurements with certain profiles. Figure 5.15 shows the measurement of the profile seen in Figure 5.16. This drop-off is measured to be 0.5243m and 41.32°, whereas a more realistic measurement would be taken from the corners as seen in Figure 5.17, measuring 0.0855m and 76.2724°. This failure is due to the ambiguity and highly variable edge drop-off.
Figure 5.14. Positive obstacle detection failure

Figure 5.15. Pavement edge drop-off detection
Figure 5.16. Pavement edge drop-off

Figure 5.17. Pavement edge drop-off
Chapter 6

Conclusion

There has been an ongoing pursuit towards the development of a safe autonomous vehicle due to the many factors that can impair or distract the modern driver. Certain research studies have focused on the aspect of obstacle detection with hopes to properly identify obstructions that potentially interfere with vehicle trajectory and driver response. Beyond simple obstacle detection and localization a critical problem lies within the analysis of the threat potential of each obstacle, which is necessary for proper decision making routines.

This thesis demonstrates the possibility of applying vehicle-based infrastructure measurements and analysis to standard obstacle detection algorithms in order to produce roadway hazard maps. The algorithms were tested using offline processing of data collected both from a controlled setting (test track) as well as on limited public roads. These tests demonstrated the feasibility of using these measurements to quantify specific potential hazards to vehicle operation, including an ability to quantify the level of potential hazard relative to the location of the vehicle, for multiple hazard types.

6.1 Future work

This system was designed to become a foundation upon which multiple safety systems can be implemented and tested. The knowledge of obstacle locations within an integrated feature map is very useful; this allows the vehicle to alert drivers of their location relative to surrounding obstacles through a heads-up display. Additionally, by comparing mapping results to each other over time, very robust estimates of hazard areas can be obtained. Through integration with GPS, these same algorithms can significantly
decrease the false positive detection rate and more robustly classify obstacles. And with GPS, one can present obstacle warnings to conventional vehicle operators without the need to equip their vehicles with high-cost LIDAR mapping sensors.

6.1.1 Positive Obstacle Classification

This thesis assumed that the hazard level of positive obstacles rely on their height; but there is a big difference between collision with a bush and collision with a light pole. It is possible to classify severity of positive obstacles based not only on their geometry but also on their material. With the detection of positive obstacle locations, the obstacles can be classified into multiple categories consisting of curbs, concrete barriers, guard rails, trees etc. By fusing LIDAR and camera data, the supplemental material estimates makes more accurate classification of positive obstacles possible.

For man-made positive obstacles such as very steep slopes, these may then have severity differing from natural edges produced by the top profile of vegetation. One can further process this data to obtain severity metrics not only based on the material but also on the slope and length of the surface beneath the obstacle, for example large bushes that may be situated above steep slopes, whose profile may be occluding the implicit rollover hazard that could occur upon impact with the bush.

6.1.2 Occupancy Grid

Another potentially hazardous obstacle of common roadways especially urban areas is pedestrian movement. In certain driving conditions, drivers may not be completely conscious of their surroundings. An awareness of typical pedestrian traffic around roadways can represent a new type of obstacle. One can address pedestrian traffic without complex sensing systems to track pedestrians simply by designating common pedestrian traffic locations. These high-traffic regions can be marked as a special type of “no-go” location similar to a static positive obstacle. Such designations of high-traffic areas can be used to create a bound to which the vehicle should also adhere during all maneuvers.
This appendix contains additional examples of detection for each obstacle covered in this thesis.

A.1 Rollover Detection

Figure A.1. Rollover detection result (ex. 1)
Figure A.2. Rollover detection result slopes (ex. 1)

Figure A.3. Rollover detection result (ex. 2)
A.2 Positive Obstacle Detection

Figure A.4. Rollover detection result slopes (ex. 2)

Figure A.5. Positive obstacle detection result (ex. 1): Obstacle of height $0.132$ m detected at $2.056$ m left of the lane center
Figure A.6. Positive obstacle detection result (ex. 2): Obstacle of height 0.665m detected at 14.056m left of the lane center

Figure A.7. Positive obstacle detection result (ex. 2): First and second derivative analysis
A.3 Edge Drop-off Detection

Figure A.8. Pavement edge drop-off detection result (ex. 1): Drop-off length of 0.8357m and slope of 33.95° detected at 0.3805m right of the single lane marker
Figure A.9. Pavement edge drop-off detection result (ex. 2): Drop-off length of 0.4161m and slope of 23.92° detected at 0.6579m right of the single lane marker.
Control Surface Experiment

Figure B.1 shows an example of a control surface, or wooden “planks”, that were manufactured for the purposes of this experiment. A total of six test surfaces with different drop-off lengths and angles were used in the experiment. These test surfaces were laid out on a road as shown in Figure B.2, and the test vehicle was driven past these surfaces at a slow speed (\(~10\) mph). Figure B.3 shows an example of the raw LIDAR scan of one of the test surfaces. The LIDAR scan was able to clearly capture the profile of the test surface. These raw data were processed using two different methods (GUI and automated) and the corresponding results are shown in Sections B.1 and B.2, respectively, of this appendix.

B.1 GUI Data Processing

In this method, the LIDAR data were processed by building a GUI that assists the user to identify the length and the slope of the test surfaces. This approach allows the user to visually identify the exact points that constitute the slope by clicking on two points on the line, subsequently using the points to make the measurements. An example of this process is shown in Figures and . The results of this GUI were used to guide the development of an algorithm to similarly find the slope and elevation, which is necessary for processing the large amounts of data expected in this project.

Example scans for the test surfaces used in the GUI LIDAR evaluation are shown in Figures B.6 and B.7. The results for all six wooden test surfaces are shown in Table B.1, and the aggregate results from each wooden test surface are shown in Table B.2. The results clearly show that, in all cases, the average measured angle from the LIDAR
scan was within ±3° (average error: 1.61°) of the hand-measured angle of the control surface. Furthermore, an average error of 0.165” (4.191mm) was observed across the length measurements, an error well below the required level to be feasible relative to hand measurements (0.22” (5.588mm)). A visual representation of the accuracy of the GUI data processing, including an error of 2 standard deviations in the LIDAR data, is shown in Figures B.8 and B.9. These figures illustrate that the automated LIDAR
Figure B.3. Example LIDAR scan of a wooden test plank

Figure B.4. GUI allowing user to select top of pavement edge (mouse denoted by large crosshairs)

Figure B.5. GUI allowing user to select bottom of pavement edge (mouse denoted by large crosshairs)
processing algorithm agrees closely with the hand measurements as well as with the human-guided GUI-results.

B.2 Automated Data Processing

Because of the large amount of data that are obtained from LIDAR scans, we also developed an algorithm to automatically identify the length and angle of the pavement edge drop-off. We are currently able to extract angle data to within $\pm 4^\circ$ (average error: $2.48^\circ$) of the hand-measured angle and the average error in the length is 0.13" (3.302mm). The results from the automated algorithm can be erroneous in some situations if the algorithm improperly identifies the slope. The results from the automated algorithm are shown in Tables B.3 and B.4.

A visual representation of the accuracy of the automated data processing including
Table B.1. GUI test results of the six planks in five runs

<table>
<thead>
<tr>
<th>Test Surface</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle (°)</td>
<td>Length (°)</td>
<td>Length (°)</td>
<td>Length (°)</td>
<td>Length (°)</td>
<td>Length (°)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>42.3</td>
<td>2.80</td>
<td>45.9</td>
<td>2.08</td>
<td>43.1</td>
</tr>
<tr>
<td>2</td>
<td>46.7</td>
<td>3.20</td>
<td>50.3</td>
<td>3.43</td>
<td>46.9</td>
</tr>
<tr>
<td>3</td>
<td>46.7</td>
<td>5.46</td>
<td>46.7</td>
<td>5.97</td>
<td>47.9</td>
</tr>
<tr>
<td>4</td>
<td>63.1</td>
<td>2.05</td>
<td>77.2</td>
<td>1.85</td>
<td>65.3</td>
</tr>
<tr>
<td>5</td>
<td>67.1</td>
<td>2.78</td>
<td>70.9</td>
<td>2.22</td>
<td>66.7</td>
</tr>
<tr>
<td>6</td>
<td>67.7</td>
<td>4.88</td>
<td>67.7</td>
<td>4.37</td>
<td>68.6</td>
</tr>
</tbody>
</table>

Table B.2. GUI test results compared to hand measurements

<table>
<thead>
<tr>
<th>Location Site</th>
<th>Hand Angle (°)</th>
<th>LIDAR Angle (°)</th>
<th>Hand Length (in)</th>
<th>LIDAR Length (in)</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>45±1</td>
<td>45.9±8.4</td>
<td>2.32±0.25</td>
<td>2.44±0.64</td>
</tr>
<tr>
<td>2</td>
<td>45±1</td>
<td>47.5±8.4</td>
<td>3.20±0.25</td>
<td>3.21±0.64</td>
</tr>
<tr>
<td>3</td>
<td>45±1</td>
<td>47.8±8.4</td>
<td>5.40±0.25</td>
<td>5.82±0.64</td>
</tr>
<tr>
<td>4</td>
<td>45±1</td>
<td>69.0±8.4</td>
<td>1.83±0.25</td>
<td>1.91±0.64</td>
</tr>
<tr>
<td>5</td>
<td>45±1</td>
<td>67.7±8.4</td>
<td>2.50±0.25</td>
<td>2.61±0.64</td>
</tr>
<tr>
<td>6</td>
<td>45±1</td>
<td>68.7±8.4</td>
<td>4.75±0.25</td>
<td>4.84±0.64</td>
</tr>
</tbody>
</table>

an error of 2 standard deviations in the LIDAR data can be seen in Figures B.10 and B.11. An error of 1° and 0.25" (6.35mm) was used for the hand measurements of slope and length data, respectively. These error values represent the decimation accuracy of the tools used to measure the data, and are used under the assumption that the hand measurements recorded were accurate and precise. Without this assumption, the size of the error would be larger, particularly in the field measurements, due to the uncertainty and variability of the hand-measured length and slope of the pavement edge drop-off. This variability in hand measurements was measured and is tabulated in later sections.

Table B.3. Automated test results of the six planks in five runs

<table>
<thead>
<tr>
<th>Test Surface</th>
<th>Run 1</th>
<th>Run 2</th>
<th>Run 3</th>
<th>Run 4</th>
<th>Run 5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Angle (°)</td>
<td>Length (°)</td>
<td>Length (°)</td>
<td>Length (°)</td>
<td>Length (°)</td>
<td>Length (°)</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1</td>
<td>48.5</td>
<td>2.63</td>
<td>45.9</td>
<td>2.19</td>
<td>50.2</td>
</tr>
<tr>
<td>2</td>
<td>47.2</td>
<td>3.22</td>
<td>50.3</td>
<td>2.56</td>
<td>45.5</td>
</tr>
<tr>
<td>3</td>
<td>45.4</td>
<td>5.53</td>
<td>46.0</td>
<td>5.47</td>
<td>47.9</td>
</tr>
<tr>
<td>4</td>
<td>63.2</td>
<td>1.76</td>
<td>82.4</td>
<td>1.19</td>
<td>68.5</td>
</tr>
<tr>
<td>5</td>
<td>68.9</td>
<td>2.53</td>
<td>70.8</td>
<td>2.08</td>
<td>69.3</td>
</tr>
<tr>
<td>6</td>
<td>68.9</td>
<td>4.64</td>
<td>69.2</td>
<td>4.21</td>
<td>67.6</td>
</tr>
</tbody>
</table>
Figure B.8. Hand-measured slope vs. GUI slope with error bars

Figure B.9. Hand-measured slope vs. GUI length with error bars

Table B.4. Automated test results compared with hand measurements
Figure B.10. Hand-measured slope vs. automated slope with error bars

Figure B.11. Hand-measured slope vs. automated length with error bars
Bibliography


