Real Time Obstacle and Road Edge Detection for an Autonomous Formula SAE-Electric Race Car

By

Timothy Kelliher

21305667

Academic Supervisor:

Professor Dr. Thomas Bräunl

School of Electrical and Electronic Engineering

Submitted to The University of Western Australia in partial fulfilment for the requirements for the degree of Master of Professional Engineering.

Word Count: 6165

Abstract

The purpose of this research is to develop and implement robust LiDAR perception, including obstacle detection and road edge detection. The goals of this project are to achieve autonomous driving through a track delineated by cones and driving on the internal roads of UWA. The obstacles and road edges are passed to the path planner to achieve the project goals.

This project utilises two LiDAR systems for testing. A SICK LMS111-10100, mounted on a low horizontal plane, and an ibeo LUX 4, mounted at an angle below horizontal in order to scan the ground in the distance. The LMS111-10100 scans a single layer while the ibeo LUX 4 scans four layers and completes in-built object detection and tracking.

Obstacle detection is achieved by processing the LMS111-10100 data via a Euclidean clustering algorithm. Obstacles are classified as large or small based on the number of points in each cluster. The closest point of an obstacle is reported to path planning. The objective of autonomous driving through a track delineated by cones is met through this perception algorithm. The path ahead is mapping to achieve redrive functionalities on a similar track.

Road edge detection is achieved by processing the ibeo LUX 4 data through partial differentiation, statistical analysis and weighted averaging of the results in each layer. Each layer is used to identify a region which is likely a road based on exploitation of the features of a road such as smoothness and continuity of the edges. The road edge detection algorithm is suitable for simplified road driving scenarios but needs improvement in order to achieve application of lane keeping due to errors in the exact position of edges.

Acknowledgements

I would like to thank my academic supervisor Professor Dr. Thomas Bräunl for the experiences I have been exposed to over the year undertaking this research topic. His guidance was instrumental in the completion of this project. I would also like to thank the REV Autonomous team for their support and assistance. Finally I would like to thank my family and friends for their support, motivation and encouragement throughout the course of this project.

Table of Contents

Table of Figures

Table of Equations

Acronyms, Definitions and Abbreviations

SAE	Society of Automotive Engineers
REV	Renewable Energy Vehicle Project
UWA	The University of Western Australia
LiDAR	Light Detection and Ranging
GPS	Global Positioning System
IMU	Inertial Measurement Unit
ROS	Robotic Operating System
SLAM	Simultaneous Localisation and Mapping
DBSCAN	Density Based Clustering of Applications with Noise
FCN	Fully Convolutional Network

rviz – A 3D visualisation tool of ROS

1. Introduction

1.1 Project Background

Autonomous driving is a topic of much interest of the public in recent times [1]. With commercial autonomous cars accessible to the public expected at the beginning of the next decade [1]. The technological advances in communications, controls and embedded systems have created an environment conducive with the implementation of fully autonomous driving solutions [2]. The benefits including improved safety, convenience, fuel economy and lower emissions [2]. One of the main barriers to the adoption of autonomous cars on the roads is public trust [3]. [3] highlights the key factors influencing the driverless car adoption are performance expectancy, reliability, security and privacy. Thus, research into driverless technology should address at least one of these factors.

Figure 1: The Formula SAE-Electric Test Platform

REV Autonomous has the overarching goals to create an accessible system with two main operational milestones: smooth driving around a race track delineated by cones and smooth driving on internal UWA roads. The platform for this research project is the Autonomous SAE Race car which was converted from petrol to electric by REV [4]. It was originally designed and manufactured by UWA Motorsports for Formula SAE, an international design and performance competition [4]. The use of this system for autonomous testing affords the simplicity of managing and maintaining a smaller platform while still being able to apply the same concepts of a full-sized road car.

Drive by wire for autonomous driving is implemented through a servo actuating the brake, a DC motor actuating the steering and the two rear electric motors for driving. The system sensors currently include a ibeo LUX4 4-layer LiDAR [5], a SICK LMS111-10100 2D LiDAR [5], an XSENS MTi-G-710 IMU [6] and two FLIR blackfly cameras [7]. At the centre of this autonomous system is an NVIDIA TX1 [8]. The vehicle has been the topic of numerous master's final year projects and the system have been proven to be suitable for meeting project goals [4].

The 2018 REV Autonomous team has chosen to migrate the current system into ROS. Leading autonomous research companies such as Apollo use a modified ROS platform [9], which boasts project partners such as Ford, Bosch and several other large companies [10]. This meaning the open source system has important features for an autonomous platform. Key components that ROS offers is the message passing interface which provides inter-process communication [11]. It also offers a vast library of open source robotics packages, diagnostics, visualisation and data logging and playback [11]. These allow a modular development scheme in that each node need not be dependent on one another and if there is a failure the system will still be able to function. A key factor for this project is that ROS allows the REV Autonomous team to develop packages in isolation and test without impacting one another.

The main pillars surrounding autonomous driving are sensing, path planning, mapping and execution [12]. Sensing involves taking sensor data to make it into useful information on the surrounding environment. Mapping is the useful storage of the information on the environment. Path planning is the effective navigation though the perceived environment. Finally, execution is the control required to actuate the planned path.

1.2 Project Scope

The scope of this research paper is to achieve accurate perception based on data from an array of two LiDAR's. This includes layout for the array, obstacle detection and road edge detection. The process should be effective for use in path planning for an autonomous vehicle to meet the goals of the REV Autonomous team. A review of current algorithms to achieve similar goals is included to provide background to the research.

2. Literature Review

2.1 Obstacle Detection

The process of obstacle detection is first to extract features from a point cloud and then to classify all features. The detection of obstacles must be accurate in order to allow navigation through the environment. If the obstacle detection is inaccurate and reports false positives, where there is an object where there is not, the path planning algorithm may take drastic action. If there is an obstacle where there is not an obstacle reported there may be a collision.

A Euclidean clustering approach is detailed in [13]. The points are clustered based on the distance between the current and the next point in the series. The process follows after clustering points to assign shape classifications to objects based on the size of the cluster. The approach was concluded to enable navigation through a given test environment. [14] applies a similar approach to [13] and yielded similar results. This meaning that this approach is simple and effective for use.

[15] and [16] propose a similar approach to Euclidean clustering but uses a radial masking area for clustering. The difference lying in the method for assigning each point to clusters. This approach searches in a radial area from each point and those within this are clustered and then the next point is moved onto that has no cluster identification number. This approach is similar to that presented in [13] but the main difference is the means of stepping through the point cloud. The results of this algorithm may merit examination in comparison to [13]. Key points from [13,14] and [15] are that the approach to assigning identifications are a key consideration. The method used must minimise the number of times points are accessed and the number of processor intensive calculations required.

[17] demonstrates the application of a K-means clustering algorithm. This is an adaptive method which requires no initial parameters to be set. These are entirely generated through the data set. All methods prior have required parameters to be manually set by the user and could likely differ for different scenarios. DBSCAN is mentioned in [17] for its strength in clustering points based on this spatial density. [18] presents a method termed parallel DBSCAN to achieve accelerated processing by more than 40%. This method would be effective for large data sets if a 3D LiDAR with a large number of layers and high resolution were to be used.

[19] proposes an algorithm for tracking multiple moving targets. Tracking is an important feature for the application of autonomous cars to improve standards of safety and achieve

prediction not possible for human drivers. [19] uses a Kalman filter to fuse tracking results from both 3D LiDAR and camera. Furthermore, [20] proposes a simpler approach for object detection and tracking by using a single plane of LiDAR points from a 4-layer LiDAR. [20] demonstrating the algorithms performance in similar scenarios to [19]. The inclusion of object tracking greatly complicating the task of object detection.

2.2 Road Edge Detection

The current REV Autonomous road edge detection algorithm was developed by Thomas Drage in 2013 [21] and improved by Thomas Churack in 2015 [22]. The Lidar data is transformed to the base of the car frame based on physical attachment and IMU readings. The summarised algorithm as in [23] is as follows:

1. Stepping out from the centre of the data set looking for a point cluster which meets the slope condition. Interleave left/right until found.
2. Perform stepping to the left and right (separately) of this cluster, increasing the size. Fit lines and record the slope and correlation coefficient at each step.
3. Road edges are at the point which maximised at each step.
4. Calculate overall fit line and check that the slope and correlation conditions are met.
5. Utilise a second order Kalman filter to identify the correct edge value based upon the correlation coefficient data.

It was concluded that this method was suitable for road edge detection of smooth curbed roads and complex scenarios featuring uneven roads with poorly defined edges [23]. This algorithm performs at an impressive standard. Although the need for research exists to improve on the current condition of the autonomous vehicle.

[24] suggests a two stage approach. Firstly, the road segment and road edge points are identified using the elevation information extracted from the range data. Secondly, the identified 3D road-edge points are further projected and validated on the 2D ground plane. This is reported to improve system robustness and reduce the computational complexity for real time applications [24]. This algorithm stands different to that in [21,22,23] with the selection of the road area determined by the output of a discrete differential filter. The paper presents the idea of temporal smoothing of detected road curbs such as the Kalman filter to reduce errors [24]. This idea having been adopted in the algorithm in [23] to much success validates this recommendation.

Another method that utilises the rate of change in the depth of the point cloud is presented in [25].

A method for road edge detection is presented in [26] based on LiDAR data histogram approach. This uses a high resolution 64-layer LiDAR to provide clarity in the point cloud. This may be achievable if the LiDAR data from a lesser resolution LiDAR is taken from a number of frames. [26] states the assumption that, in an urban environment, the ground ahead of the vehicle should be flat. This is may not be applicable if not in a specific environment as hills and the camber on roads means this is a large approximation.

The previous methods have involved filtering, statistical analysis and line fitting to detect road edges. The use of a fully convolutional neural (FCN) utilising only Lidar scan data is discussed in [27]. The proposed approach achieves high level performance on the KITTI road benchmark and can provide high accuracy road segmentations in any lighting conditions, also working real time on GPU-accelerated hardware [27].

The FCN was designed specifically for semantic segmentation and was designed to have a large receptive field to process high resolution feature maps [27]. The tailored design of this FCN could be the driving factor behind the high performance of this method. The Lidar used in this paper also has a much higher number of scan layers so the performance with a 4-layer platform may greatly differ.

[28] proposes a road detection method based on camera and LiDAR, the fusion in both the data level and feature level. The Lidar point clouds are projected into the images and obtain original height images which are sparse. Then, up sampling these height images via a joint bilateral filter so that all pixels will have their own height values [28]. The pairing of LiDAR and camera data for processing is a key point raised in this paper. [28] suggests the lack of improvement of road edge detection due to keeping each form of vision separate and not fusing until the results of each sensor are processed. This statement being potentially detrimental to the success of the project as the current structure is that camera and Lidar vision are processed separately. [29] draws a simpler approach to [28], transforming the lidar point cloud 3D plane into the camera 3D plane and then to the camera coordinate 2D point, then vertical and horizontal histograms for rough road detection are undertaken. The article concludes that this method can be applied to quickly estimate the approximate road regions [29].

A key outcome in the papers proposing methods for the fusion of camera and LiDAR processing is the clarification of the strengths of each sensor. The LiDAR is effective at detecting objects and generating an accurate hypothesis for their position [30]. While vision is used as a classifier responsible for validation or final classification of objects [30]. The result of combining LiDAR and camera data prior to processing has reported improvements in the false alarm rate in [28,29,30]. A method in which this is applied on the REV Autonomous platform may lead to conflicts in the modular structure. The processing power required is also not examined in these papers so there may be need for caution in allocation of computer resources.

3. Experimental Design

3.1 High Level LiDAR Processing Architecture

The high-level architecture is presented in Figure 2 below. The LiDAR data requires a ROS driver in order to publish the point cloud and complete any pre-processing necessary. The point cloud is then processed to determine features and objects in the point cloud. The output of this node would then be passed to a LiDAR and camera fusion node in which similar objects are fused to make up for any short comings of each sensor approach and reach a more accurate and robust solution. This complete environment is then to be passed to path planning in order to reach decisions on how to plan the vehicles navigation.

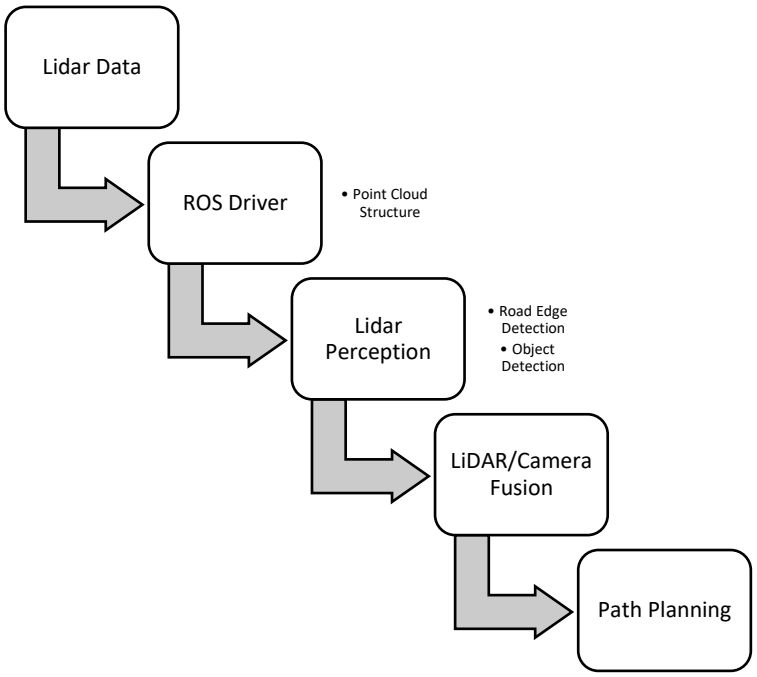

Figure 2: Flow Chart of High Level System Overview

A mapping node would exist between the LiDAR/Camera fusion in order to develop a more robust system. A key outcome of mapping is the ability to redrive with more knowledge of the environment. In the application of the REV Autonomous team this means moving through the track at a faster speed after the first lap is completed.

3.2 LiDAR Setup

The LiDAR set up design is important to ensure there are limited blind spots and the point cloud generated is optimised for use. If there is too much redundant information from two LiDAR's scanning the same area this will be a waste of information. Thus, the approach must maximise the usefulness of the point cloud while not creating any dangerous blind spots.

The LMS111-10100 is mounted on a low horizontal plane which is extremely useful for obstacle detection. The low horizontal plane greatly reduces the noise and amount of data required to filter out in order to acquire an obstacle. Figure 3 below demonstrates a simple example with two obstacles in front of the LiDAR. The resultant point cloud on the right-hand side displays the clarity with which objects are visualised. This scenario is common-place in the REV Autonomous team testing on a track delineated by cones.

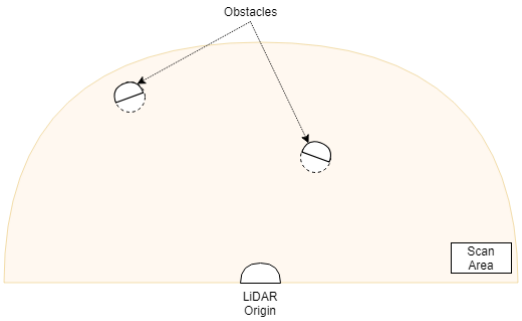

Figure 3: Horizontal Laser Scan

The processing to identify these obstacles will be point clustering. With the correct set up of the LiDAR the point processing is greatly simplified and will result in less CPU time consumed to achieve accurate obstacle detection. A predicted down side to this low horizontal plane is that obstacles may be above this plane. Thus, the ibeo LUX4 must scan the area above in order to avoid any situations where the scan area does not hit an obstacle in front of the car.

The ibeo LUX 4 is a 4-layer LiDAR which completes its scan at an adjustable angle pitched down from the horizontal, from a higher position on the car. This is greatly useful for road edge detection as it will be scanning the road at a suitable distance from the vehicle. Also, removing the blind spot above the horizontal plane scanned by the SICK LiDAR. The suitable distance to scan the road ahead of the car depends on the application plans for the car, a higher speed requiring information from further ahead of the car.

3.3 Object Detection

The ibeo LUX 4 LiDAR features on-board object detection and tracking. The resultant from this device may be used in conjunction with the robust object detection gathered from the SICK LMS111-10100 LiDAR. The will require fusion of the object positions prior being published to be fused with the camera object detection results.

3.4 Obstacle Detection

Cone detection requires feature extraction and then classification. The processing of this problem may be simplified by having a very simple classification scheme. For the application in this project the vehicle does not need to know what the object is in front of the car so to design an object detection scheme that is robust and meets the scope of this report only needs to report the location of an object. Then the car may avoid it regardless of whether it is a person or a cone. The assumption that there will be no need to track and predict motion based on the classifications like in other papers. This could be dangerous in practice if an animal or person were to move into the path of the car and a more processor hungry program could have predicted and mitigated the impact of this risk. The layout of the point cloud and a situation with two obstacles is displayed below.

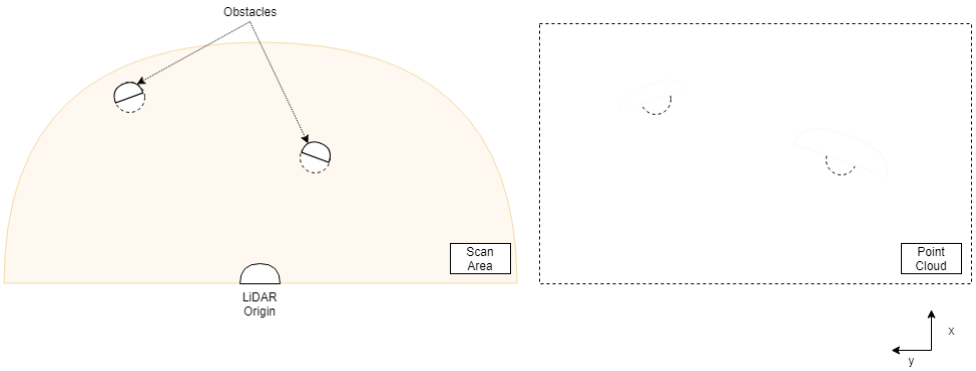

Figure 4: Horizontal Laser Scan and Resultant Point Cloud

A Euclidean clustering algorithm is applied to a horizontal point cloud. The points are examined based on their distance and angle to the next point in a series as displayed in Figure 5 below. This method should be effective at clustering a point cloud which has no incidence with the ground. If the point cloud were to include the ground it would be clustered in with objects. This would create inaccuracy in the position of objects and potentially the width. If the point cloud is going to be incident with the ground a method to remove the points within a threshold from the ground plane should be implemented.

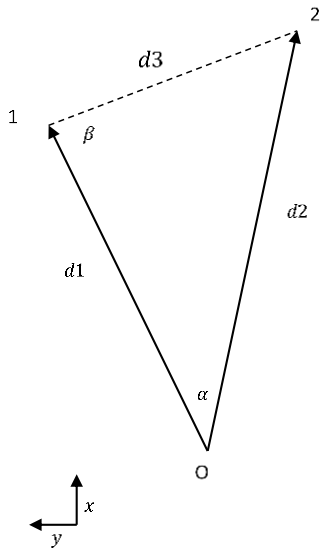

Figure 5: Point Clustering Dimensions for Two Arbitrary Points A and B

Given each point on the horizontal plane has point data $\{x_i, y_i\}$ as presented above where $i = 1,2$. Angle α is a known parameter from the LiDAR scanning angle. Thus, calculation of the distance between both points and the angle β between them is given by the following equations.

Equation 1: Distance between two consecutive points

$$d3 = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2}$$

And,

Equation 2: Angle between the origin and two consecutive points

$$\beta = arctan\left(\frac{d2\ sin\ \alpha}{d1 - d2\ cos\ \alpha}\right)$$

The following code is implemented to conduct point clustering.

If (distance d3 is less than threshold)

The two points are identified as being in the same cluster.

Else If (angle β is greater than threshold)

The two points are identified as being in the same cluster.

Else

The two points are not in the same cluster.

Start a new cluster with point i+1 as the starting point.

Figure 6: Simplified Clustering Algorithm

This code is looped through comparing each point i to point $i + 1$ until all points have been allocated to clusters. This process reduces redundant examination of points as each point is only examined a maximum of two times. Figure 7 displays as in one iteration point i and $i + 1$ are compared and the following iteration $i + 1$ and $i + 2$. This is the only two times point $i + 1$ is accessed in the comparison. This architecture selected as the simplest computation complexity for accessing points.

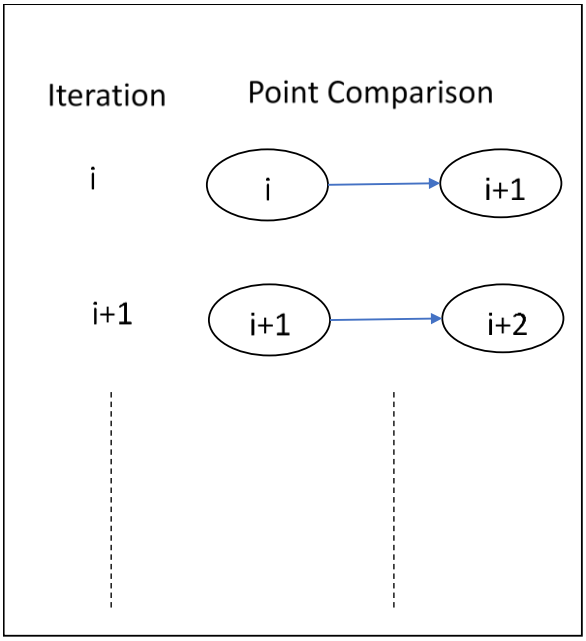

Figure 7:Point Comparison Scheme

An identification number is assigned to each point based on this test, signifying which cluster a point belongs to. All clusters are grouped together and their closest point to the car is published to the other nodes. Reporting the closest point to the car ensures that each instance of the point cloud provides robust information for path planning. Alternatively the centroid of the point cluster could be reported and the shape inferred from the number of points in the cluster.

Minor classification of objects is a resultant of this process in the examination of the number of points in each cluster. This may indicate whether an object is large or small. This size must be considered in path planning to determine whether a collision may occur with the sides of the vehicle and the sides of this object.

3.5 Road Edge Detection

The features of a road which are exploited in this method are the smoothness and continuity of a road. The smoothness is reflected by the rate of change in the depth of each layer in the point cloud as presented. These levels may be compared to a threshold for identification of the road region. However, as noise increases in the point cloud layer the peaks become less pronounced. The continuity is exploited by using the previous road edge positions in the previous layer as the proposed position of the road edges in the next layer. If the road were to move in a corner a reasonable region is proposed for the next layer.

For each point in the point cloud the partial differentiation is as follows:

Equation 3: Partial derivative function for point cloud

$$\frac{\partial x}{\partial y}(k) = \frac{x(k) - x(k+1)}{y(k) - y(k+1)}$$

Figure 8 below represents a typical road profile on the x-y plane. The curbs show two distinct changes on either side of the car if located at 0 and road edges at an arbitrary width around ± 65. There are two distinct peaks in the position of the road edges.

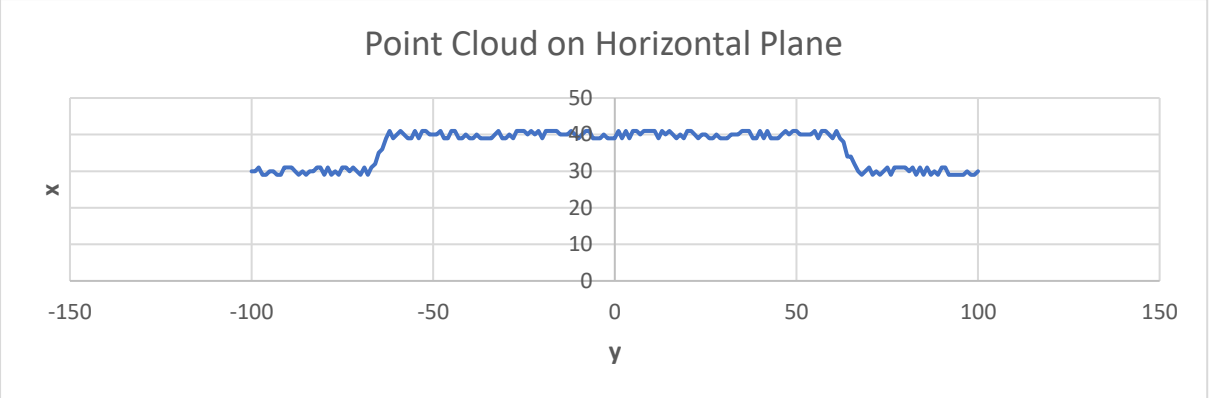

Figure 8: Plot of a Synthetic Point Cloud Typical of a Road

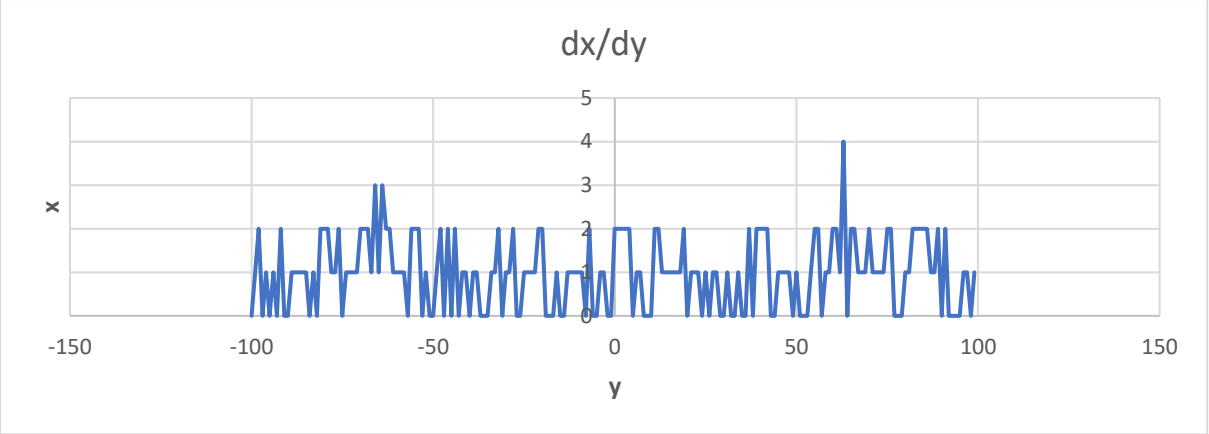

Figure 9: Plot of the Partial Derivative of a Synthetic Point Cloud Typical of a Road

Each layer of the point cloud is assigned identification numbers of which points are road area and the largest group is the most likely to be the road area. If this area is too small, no edges are reported. The groups are defined by the begin and end point and maintain their relation to individual points in the cloud. Each point in a group is under the rate of change threshold set through testing to ensure the baseline is above the level of noise.

The following step is to find the standard deviation of the largest group and use it to expand out in each direction. The search will continue until a point is greater than n standard deviations, where n is a tunable parameter. This technique will eliminate the likelihood of a group being selected due to high noise in a layer. The standard deviation is calculated with only respect to the depth information in the following equation.

Equation 4: Standard deviation calculation

$$\sigma = \frac{1}{N-1}\sqrt{sum\ of\ squares + \frac{sum^2}{N}}$$

A method of smoothing the edges reported over time is applied to maintain reasonable order in edges reported. This is to avoid situations in which the edges reported have a large jump from one position to the other. Thus, the edges from each layer is utilised to facilitate this smoothing. After the edges are found in each layer a weighted average of all are taken with inclusion of the previous iterations reported road edges. This reduces the impact a greatly incorrect road edge will result in an unlikely report. However, this approach falls short when the edges are in different positions in each layer. If one layer is a large distance to the right and the opposite layer is directly in the path of the car the closest layer is given a higher weight than the furthest. Thus, the weights must be higher for the layers that are further below horizontal. This approach may only be applied after the first iteration, the formula is shown below.

Equation 5: Weighted average formula

$$edge_L(k) = \frac{\left(a\ edge_{l1}(k) + b\ edge_{l2}(k) + c\ edge_{l3}(k) + d\ edge_{l4}(k) + e\ edge_L(k-1)\right)}{a+b+c+d+e}$$

Where we have the following,

$a, b, c, d, e - tunable\ weights$

$edge_L(k) - array\ of\ reported\ edges$

$edge_{layer=l\#}(k) - array\ of\ edges\ found\ in\ each\ layer$

$k - iteration\ number$

4. Results and discussion

The methods are implemented on the Formula SAE-Electric vehicle. The scenarios are on the internal roads at UWA for road edge detection and at the RAC Driving Training and Education facility for cone driving.

4.1 LiDAR Setup

The Sick LMS111-10100 LiDAR is mounted at a point 1.7m forward from the middle of the back tires of the car and at a height of 0.3m from the ground with no load in the car. The scanning FOV is limited to 180 degrees due to the vehicle chassis at its back on the mounting point. The pitch angle may be adjusted to account for a load in the vehicle. A small positive angle is applied while there is no person in the car such that when a driver is in the car the scan region is close to horizontal.

The ibeo LUX4 LiDAR is mounted at a height of 1.2m at a position in line with the back wheels of the car. The LiDAR is pitched downwards at an angle of 4º below horizontal. The mount was constructed with adjustment possible for the angle to be changed at the requirements for the testing.

The roll, pitch and yaw of the whole vehicle due to suspension leads to impacts on the data from each LiDAR. ROS offers a transform library which allows declaration of parent and child coordinate frames and transforms between them. The transform for each LiDAR is linked to the base frame of the vehicle that is declared as the middle of the back wheels. Thus, the base link transform is updated for roll, pitch and yaw relative to the global frame and the effect is applied into all other child frames. This allows transformation from the LiDAR coordinate system into the global frame.

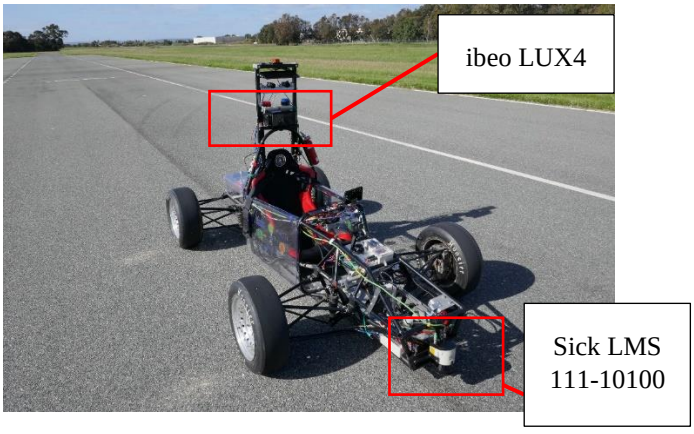

Figure 10: Formula SAE Vehicle Sensor Arrangement

The refresh rate of each LiDAR is 12.5 Hz thus a requirement of each algorithm is that is functions in real time and is finished processing the current point cloud before the next is updated.

4.2 Cone Detection

The cone detection clustering algorithm was implemented on data retrieved from the RAC DTEC facility. The cones were laid an equal distance apart with suitable width of the vehicle to be able to navigate corners. The car was run autonomously with only object detection from the LiDAR and was able to successfully navigate the delineated track. Path planning was implemented by a team member to take in cones within a given range from this algorithm and navigate through them. The cone detection algorithm ran successfully and robustly as the only issues were on path planning.

In this application the angle criteria is not implemented due to it being unnecessary in most tested applications. This would likely have a greater impact on larger objects such as cars in the frame. The results display clarity in the cones reported. This proved to not hinder the algorithm for the test application. In both scenarios the obstacle detection algorithm meets the refresh rate of the LiDAR.

Figure 10 shows the Formula SAE car inside a corner while Figure 11 displays the corresponding object readout on the x-y plane. This shows what the performance under conditions of movement while the suspension is affected by the forces of turning. Figure 12 displays the car approaching the corner, driving straight while Figure 13 displays the corresponding object readout on the x-y plane. This shows the performance under motion. The wheel base is represented by the red and green axis.

Figure 11: Image from On-Board the Formula SAE Vehicle Turning in Cone Track

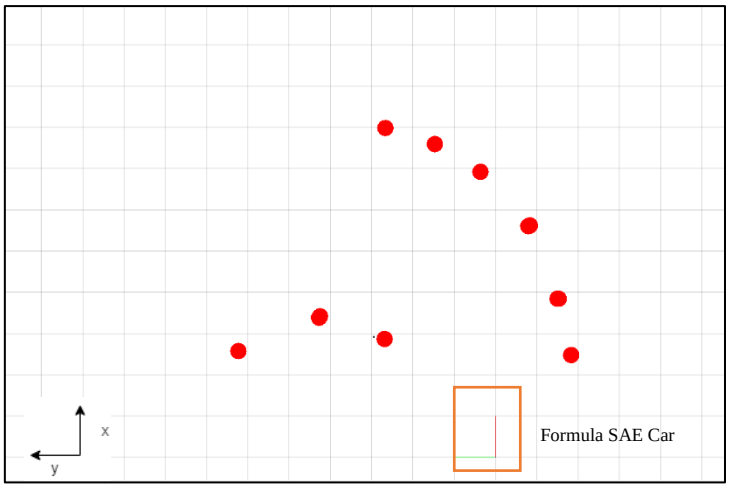

Figure 12: Cones Reported on the x-y Plane in While Turning in Cone Track

The approach to a corner lead to six cones being reported on the right-hand side and all three of the cones on the left side reported. The remaining four cones on the right side of the track have not been picked up by the LiDAR due to the scan angle being strongly affected by suspension of the car. As this is a 2D scanner the inherent limitation is such that it is strongly affected by changes in roll, pitch and yaw. A way to fix this problem would be to use a sensor with more layers. As one layer moves above a certain level, use the layer below in order to not lose any information on the environment surrounding the car.

However, the cones that have been reported are at a range of 8 metres from the car. This is suitable for driving at a slow speed but if it were to be increased the vehicle path planner may not have time to respond. As this is a corner, sufficient information is maintained due to the required slower cornering speed.

Figure 13: Image from On-Board the Formula SAE Vehicle Approaching Corner in Cone Track

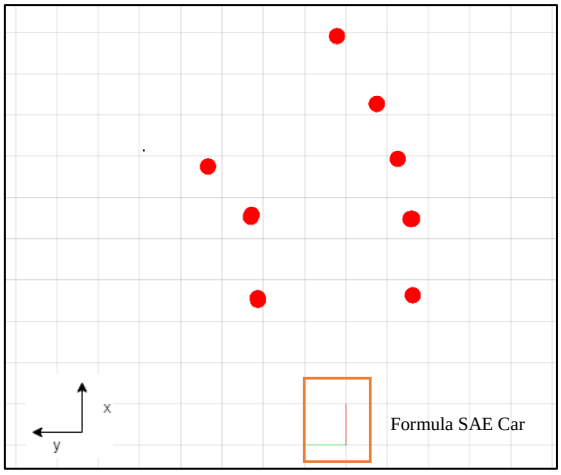

Figure 14: Cones Reported on the x-y Plane in the Scenario While Approaching Corner in Cone Track

While moving relatively straight the clustering algorithm reports objects at a distance of 10 metres from the vehicle. This data is similarly affected by the roll, pitch and yaw of the LiDAR as in case the first scenario. Five cones are observed on the right-hand side and three cones are observed on the left-hand side. Figure 13 shows that the number of cones on the right-hand side should be higher and the left-hand side shows reasonable results as the fourth cone in front of the car is eclipsed by closer ones.

The LiDAR data is greatly impacted by movement of the car chassis. This is not ideal for situations where the vehicle is traversing speed bumps of has the potential of hitting pot holes. The test platform suspension is quite rigid also, meaning on a full-size vehicle the impact while cornering will be increased. The pitch angle should be moved downwards in order to effectively function in this scenario. However, this will lead to the opposite effect,

while under braking the LiDAR will be incident with the ground, requiring a filtering algorithm to avoid any false positives.

4.3 Road Edge Detection

The road edge detection algorithm has been implemented on a data set from driving the internal roads of UWA. The road area is clear, and curbs are well defined and paved. The path shown in Figure 15 was chosen as the test area. The road width is consistently 4.12m for the path driven. The scenario can be broken into two distinct tests, road edge detection while driving straight path and road edge detection while cornering.

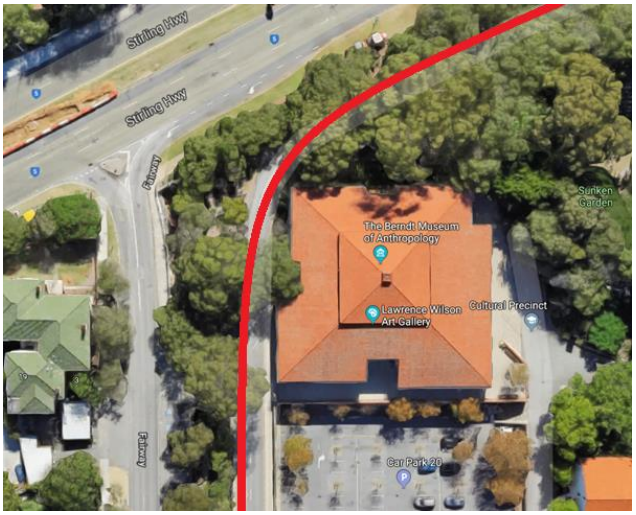

Figure 15: Google Maps Image of Test Route

Each layer of the ibeo LUX 4 has an increasing level of noise as it is reporting further distances from the vehicle. It is observed that the algorithm for road edge detection is not effective at discerning the edges in the noisiest layer. This is improved by using unique rate of change threshold values for each layer. The difference observed after this change was implemented proved to be positive and the accuracy of road edge detection was improved. In both scenarios the road edge detection algorithm meets the refresh rate of the LiDAR.

While driving straight the road edge detection algorithm should work the best as all four layers of the scan are on the road ahead and thus the curbs are distinct and in similar positions in each layer. The cornering scenario will incur issues as curbs will be in different positions in each layer. Also, suspension while cornering will affect the amount of noise in each layer of the scan.

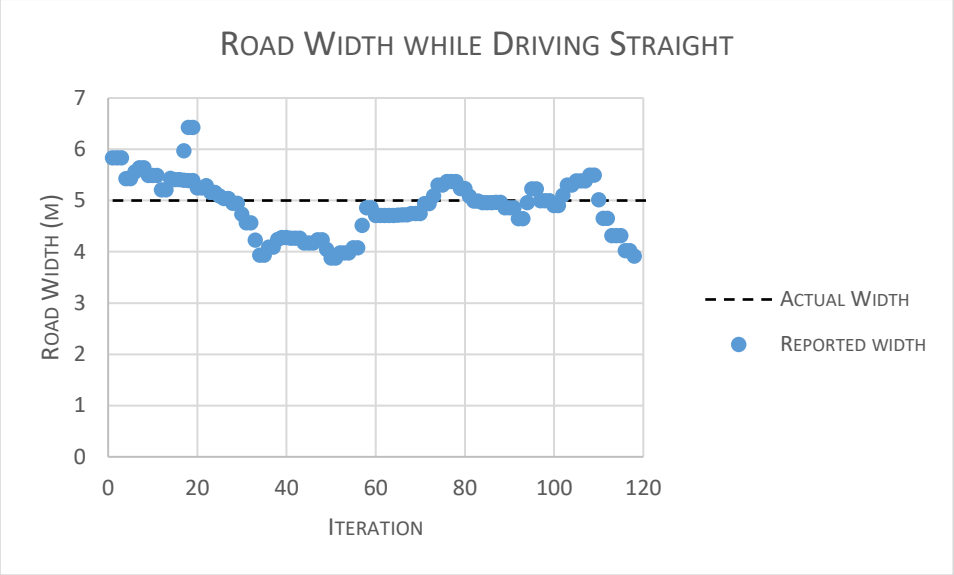

Figure 16: Plot of Road Width While Driving Straight vs. Road Edge Detection Iteration

The road width presented in Figure 16 shows a relatively poor performance at the beginning and as the process continues the values became more stable. The average error in this test run while driving straight was 13.91% with a maximum error of 22.48% and minimum error of 0.12%.

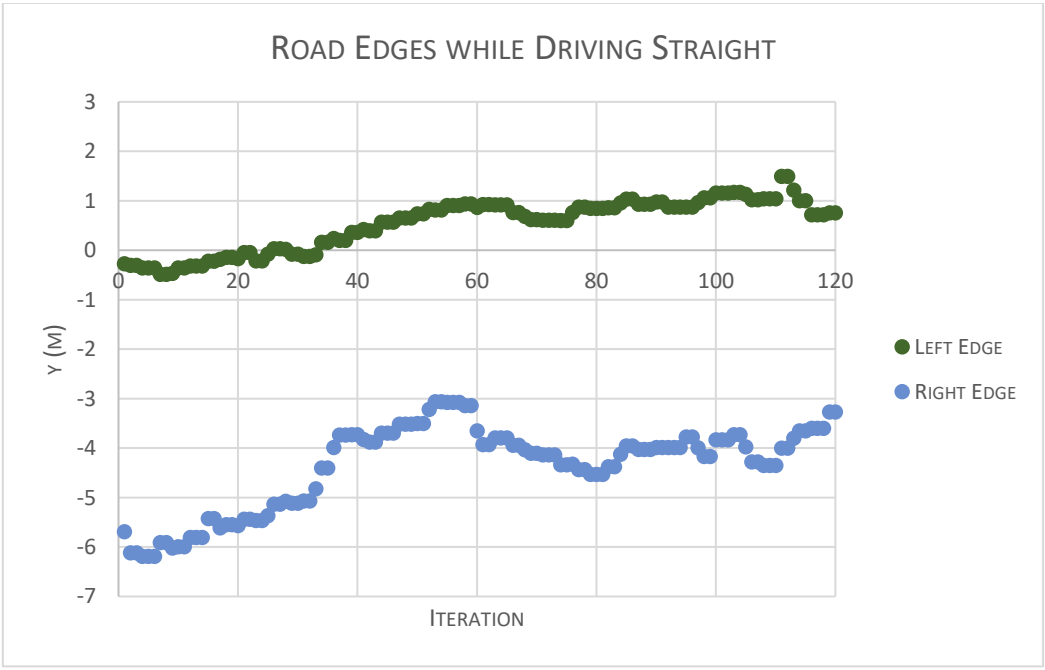

Figure 17: Plot of Road Edges While Driving Straight vs. Road Edge Detection Iteration

The road edges are reported with error in their width but the positionings are matching that of a road while small adjustments are made to the steering of the vehicle whilst driving straight. The reported road edges have a low amount of largely reported false positives of where the road edges are. This consistency means that the car will not make any large path planning decisions and should be able to follow the road given the two edges. The lane keeping in the half of the road designated to traffic moving each direction may not be achievable without improvements to the algorithm.

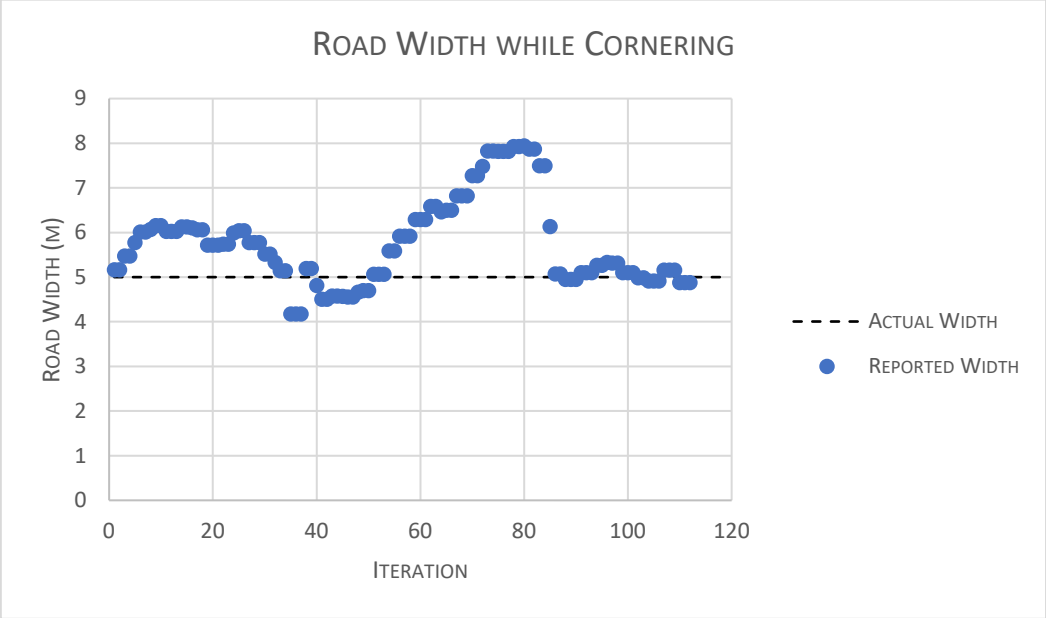

Figure 18: Plot of Road Width While Cornering vs. Road Edge Detection Iteration

The road width presented in Figure 18 shows a large amount of error when the car begins going straight again and then stabilises. The average error in this test run while cornering was 17.90% with a maximum error of 58.85% and minimum error of 0.30%. The maximum error was experienced as the car was in the beginning to stop turning.

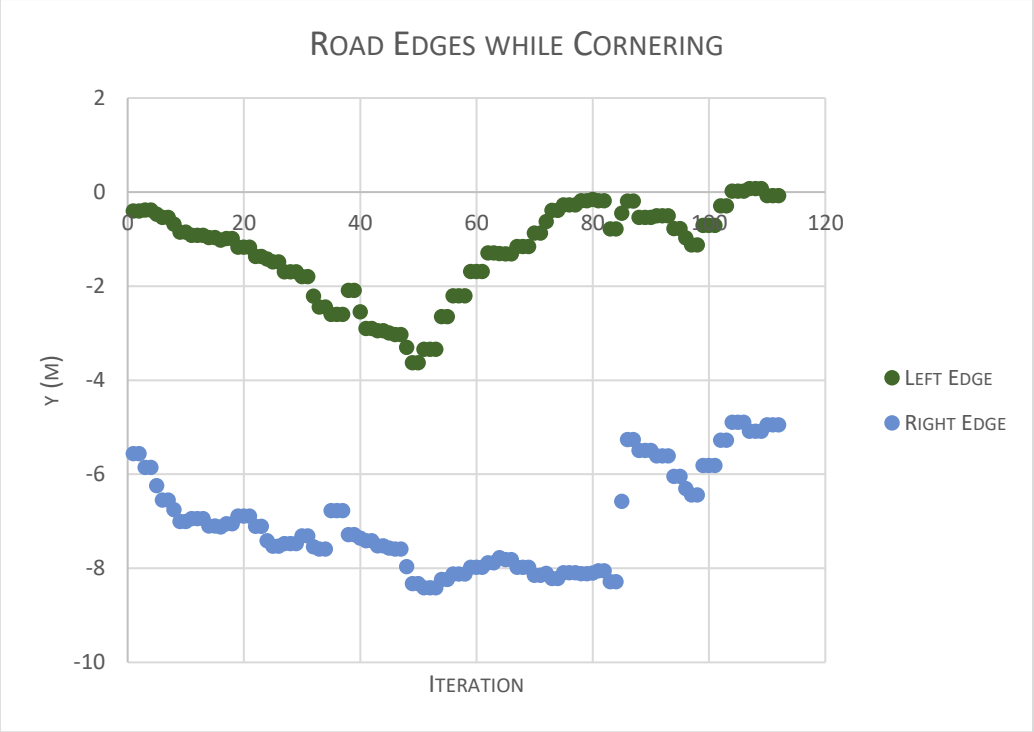

Figure 19:Plot of Road Edges While Cornering vs. Road Edge Detection Iteration

While cornering the road edges reported shift to the right as the corner begins and shifts back to similar positions as driving straight as the corner is completed. A key feature is that the edges are smoothly shifted to the right in the beginning. This means the path planner would be given information necessary to begin cornering. Once again there are no false negatives in front of the vehicle meaning there will be no dangerous false positives even though the error as mentioned in the road width is large as the corner finishes.

The road edges reported in both scenarios are completed without any pre-processing of the point cloud before being searched for road edges. A method may be applied to reduce the levels of noise in the point cloud. This would make the features more discernable without impacting the algorithm. One down side would be the processing required.

The current state of road edge detection would likely serve for autonomous driving on the roads if the lane keeping is not required to be observed. Error in the edges could lead to issues when the car is required to follow closely. The best approach would be to follow waypoints set in the center of the edges reported.

4.4 Discussion of Results

The obstacle detection algorithm applied to the collected data set performs effectively to enable navigation through an environment. The applications of this to a commercial vehicle could be possible as a simple solution. A feature this is lacking compared to some of the forerunners in the field is object tracking. The differences in a system that includes tracking could be the prediction and prevention of a hazardous event. For the scope of this project the tracking is not required due to the successful completion of the cone driving scenario.

The object detection provided from on-board the ibeo LUX4 LiDAR is was not used as a supplementary source of objects for the cone driving scenario. The absence of processing required in the main system computer means this is a low-intensity approach for improving the object detection. If the objects from each LiDAR were fused the environment may be clearer and more robust for path planning.

The road edge detection approach applied detected the road edges in both scenarios with a large amount of error. The ways to improve this method could be to change thresholds implemented or to attempt a different approach. The literature review in this paper details several different approached with the machine learning approach being the likely best choice due to the way autonomous systems are progressing.

The road edge detection algorithm will not cope if there is a car or obstacle in front of the vehicle on the road. The algorithm may be improved if fused with object detection and due to the knowledge of where objects exist these points may be filtered out and any road edges behind the objects be found. This technique would lead to less accuracy since the road edge detection will require a number of points to function correctly but could prove useful for operation on actual roads.

Once road edge and object detection is completed the mapping of the positions in the global frame needs to be completed. This has not yet been implemented due to the lack of accurate odometry of the car which is required for transformation from the LiDAR’s coordinate system into the global frame. Once this is implemented the vehicle will be able to map its environment by storing cone positions and road edges. Then the re-driving features will be able to be implements.

5. Conclusions and Topics for Further Investigation

The algorithms applied here for successful cone driving and prospective road driving meet the scope of the project. The obstacle detection algorithm is sufficient for autonomous driving on a track delineated by cones. Driving on internal roads may only be met if traffic conditions are not to be observed. Thus, the future research should focus on the development of a more robust road detection scheme. The results of cone detection is comparable with that presented in current literature although is lacking object tracking. The results of object detection is lagging behind that in current literature but may be adapted to be comparable over time. Each of the algorithms meets the requirements of the real-time processing showing the effectiveness and robustness of the code implemented.

In the future the road edge detection algorithm could be moved to machine learning. This is the main perception-based method for the future of autonomous driving. The method with machine learning must be of a similar speed and increased accuracy for it to be implemented on the vehicle. Alternatively, an approach where the LiDAR and camera data is fused prior to processing could lead to positive results and merits investigation based on current literature. The inclusion of mapping is vital in future research. It is the next step once accurate obstacles and road edges are reported. Future work should consider the functionality of mapping the environment in real time with an effective storage strategy. Investigation into a SLAM algorithm using the point cloud from both LiDAR's could meet this.

6. References

[1] K. Bimbraw, “Autonomous cars: Past, present and future a review of the developments in the last century, the present scenario and the expected future of autonomous vehicle technology”, Informatics in Control, Automation and Robotics, 2015. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/xpls/icp.jsp?arnumber=7350466

[2] I. Pettersson, I.C. M. Karlsson, “Setting the stage for autonomous cars: a pilot study of future autonomous driving experiences”, IET Intelligent Transport Systems, 2014. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/stamp/stamp.jsp?tp=&arnumber=7243385

[3] K. Kaur, G. Rampersad, “Trust in driverless cars: Investigating key factors influencing the adoption of driverless cars”, Journal of Engineering and Technology Management, vol. 48, pp. 87-96, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0923474817304253

[4] The REV Project, “Self Driving Formula SAE Race Car” , The REV Project, 2018. [Online]. Available: http://therevproject.com/vehicles/sae2010.php

[5] AutonomouStuff, “LiDAR Comparison Chart”, AutonomouStuff, 2018. [Online]. Available: https://autonomoustuff.com/lidar-chart/

[6] XSENS, “MTi-G-710 Product Overview”, XSENS, 2018.

[7] FLIR, “FLIR BLACKFLY GIGE VISION”, FLIR, 2018. [Online]. Available: https://www.ptgrey.com/support/downloads/10131

[8] NVIDIA, “NVIDIA Jetson TX1 System-on-Module”, NVIDIA Corporation, 2016. [Online]. Available: http://images.nvidia.com/content/tegra/embedded-systems/pdf/JTX1-Module-Product-sheet.pdf

[9] Apollo, “Apollo GitHub: An Autonomous Driving Platform”, Baidu Apollo, 2018. [Online]. Available: https://github.com/ApolloAuto/apollo

[10] Apollo, “Apollo Partners”, Baidu Apollo, 2018. [Online]. Available: http://apollo.auto/index.html

[11] ROS, “Core Components”, ROS, 2018. [Online]. Available: http://www.ros.org/core-components/

[12] D. Rose, “The Four Pillars of Self-Driving Cars”, Strategy Wise, 2018. [Online]. Available: https://strategywise.com/the-four-pillars-of-self-driving-cars/

[13] Y. Peng, D. Qu, Y. Zhong, S. Xie, J. Luo, “The Obstacle Detection and Obstacle Avoidance Algorithm Based on 2-D LiDAR”, IEEE International Conference on Information and Automation, 2015. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/stamp/stamp.jsp?tp=&arnumber=7279550

[14] A. Catapang, M. Ramos, “Obstacle Detection using a 2D LiDAR system for an Autonomous Vehicle”, IEEE International Conference on Control System, Computing and Engineering, 2016. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/document/7893614

[15] S. Lee, Y. Moon, N. Ko, H. Choi, H. Lee, “A Method for Object Detection Using Point Cloud Measurement in the Sea Environment”, IEEE Underwater Technology, 2017. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/stamp/stamp.jsp?tp=&arnumber=7890290

[16] H. Najdataei, Y. Nikolakopoulos, V. Gulisane, M. Papatriantafilou, “Continuous Parallel LiDAR Point-cloud Clustering”, IEEE International Conference on Distributed Computing Systems, 2018. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/stamp/stamp.jsp?tp=&arnumber=8416334

[17] M. Wei, L. Chao, X. Liu, “A New Clustering Algorithm with Adaptive Attractor for LiDAR Points”, IEEE International Conference on Progress in Informatics and Computing, 2014. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/stamp/stamp.jsp?tp=&arnumber=6972288

[18] N. Chen, X. Gao, H. Li, “Parallel DBSCAN with Priority R-tree”, IEEE International Conference on Information Management and Engineering, 2010. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/stamp/stamp.jsp?tp=&arnumber=5477926

[19] W. Mei, G. Xiong, J. Gong, Z. Yong, H. Chen, H. Di, “Multiple moving target tracking with hypothesis trajectory model for autonomous vehicles”, IEEE International Conference on Intelligent Transportation Systems, 2017. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/document/8317845

[20] D. Wang, I. Posner, P. Newman, “Model-free detection and tracking of dynamic objects with 2D LiDAR”, The International Journal of Robotics Research, 2015. [Online]. Available: http://journals.sagepub.com.ezproxy.library.uwa.edu.au/doi/abs/10.1177/0278364914562237

[21] T. Drage, “Development of a Navigation Control System for an Autonomous Formula SAE-Electric Race Car”, Final Year Thesis, School of Electrical and Electronic Engineering, University of Western Australia, Perth, 2013.

[22] T. Churack, “Improved Road-Edge Detection and Path-Planning for an Autonomous SAE Electric Race Car”, Final Year Thesis, School of Electrical and Electronic Engineering, University of Western Australia, Perth, 2015.

[23] T. Drage, T. Churack, T. Braunl, “LIDAR Road Edge Detection by Heuristic Evaluation of Many Linear Regressions”, Intelligent Transportation Systems, 2015. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/xpls/icp.jsp?arnumber=7313489

[24] W. Zhang, “LIDAR-based road and road edge detection”, Intelligent Vehicles Symposium, 2010. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/document/5548134/?part=1

[25] B. Kang, S. Choi, “Pothole detection system using 2D LiDAR and camera”, Ninth International Conference on Ubiquitous and Future Networks, 2017. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/document/7993890

[26] L. Chen, J. Yang, H. Kong, “LiDAR-Histogram for fast road and obstacle detection”, IEEE International Conference on Robotics and Automation, 2017. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/document/7989159

[27] L. Caltagirone, S. Scheidegger, L. Svensson, M. Wahde, “Fast LIDAR-based road detection using fully convolutional neural networks”, Intelligent Vehicles Symposium, 2017. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/document/7995848/

[28] X. Han, H. Wang, J. Lu, C. Zhao, “Road detection based on the fusion of Lidar and image data”, International Journal of Advanced Robotic Systems, 2017. [Online]. Available: http://journals.sagepub.com.ezproxy.library.uwa.edu.au/doi/full/10.1177/1729881417738102

[29] S. Gu, Y. Zhang, J. Yang, H. Kong, “Lidar-based urban road detection by histograms of normalized inverse depths and line scanning”, European Conference on Mobile Robots, 2017.

[Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/document/8098682/

[30] F. Zhang, D. Clarke, A. Knoll, “Vehicle detection based on LiDAR and camera fusion”, Intelligent Transportation Systems, 2014. [Online]. Available: https://ieeexplore-ieee-org.ezproxy.library.uwa.edu.au/document/6957925/