Assessment of Road Infrastructure Design for Autonomous Vehicles using LiDAR Data

by

Maged Kamal Gouda Abdelkader

A thesis submitted in partial fulfillment of the requirements for

Doctor of Philosophy

in

Transportation Engineering
Department of Civil and Environmental Engineering
University of Alberta

ABSTRACT

Despite recent advances in automotive research, a fully autonomous system operating in open, unconstrained road environments has not yet been realized. Recently, several types of autonomous vehicle (AV) failures were reported, such as run-of-the-road collisions and failure to stop for occluded road users. Such failures raised concerns about AV compatibility with existing infrastructure. As such, offloading expensive and infeasible computational workloads to smart infrastructure is an active area of research. Representatives from the infrastructure owners/operators (IOOs), the automotive industry, and academia have advocated for new ways to prepare roadways for the deployment of AVs. These recommendations aim to assess and enhance road infrastructure design for AVs by IOOs. These recommendations are threefold. First, improving vehicle-to-infrastructure connectivity (V2I), known as "talking to the road." Second, enhancing the vehicle's ability to monitor the driving environment, or "seeing the road." Finally, "simplifying the road" through modifying roadway and roadside geometric design to support navigation by AVs. Following these recommendations, this thesis explores the gap between existing road infrastructure and AV technologies, specifically i) occlusion due to physical infrastructure, ii) road geometric design, iii) roadside design, iv) pavement markings, and v) traffic signs. In this dissertation, novel simulation-based and quantitative approaches were developed to segment light detection and ranging (LiDAR) data and assess road design performance for AVs using ultra-dense point clouds. The methods utilize point clouds to digitize existing roads and simulate a set of AV systems within this environment. An infrastructure surface model, efficient octree/voxel data structures, and semantic segmentation variants of deep learning methods and heuristic segmentation approaches are used to map design measures and extract locations with substandard conditions for AVs. A framework for improvement solutions is also presented. This

work helps infrastructure operators and those in the AV sector make data-driven decisions regarding smart physical and digital infrastructure investments. Methods developed in this research are the first to systemically and quantitatively assess road design for AVs on a large scale.

PREFACE

The work presented in this thesis is either published or is under review for publication in various peer-reviewed journals.

PUBLISHED PAPERS

1. **Gouda et al.** (2023). "*Using Convex Hulls with Octree/Voxel Representations of Point Clouds to Assess Road/Roadside Design for Automated Vehicles*." Automation in Construction. In press.
2. **Gouda et al.** (2022). "*Traffic Sign Extraction using Deep Hierarchical Feature Learning and Mobile LiDAR Data on Rural Highways*." Journal of Intelligent Transportation Systems: Technology, Planning, and Operations.
3. **Gouda et al.** (2021). "*An Automated Assessment of Infrastructure Preparedness for Autonomous Vehicles*." Automation in Construction. 129: 103820.
4. **Gouda et al.** (2021). "*Automated Safety Assessment of Roadside Clear Zones using LiDAR Data*." Transportation Research Record. 2675.12: 432-448.
5. **Gouda et al.** (2021). "*Octree-Based Point Cloud Simulation to Assess the Readiness of Highway Infrastructure for Autonomous Vehicles*." Journal of Computer-Aided Civil and Infrastructure Engineering.; 1–19.

CONFERENCE PRESENTATIONS/PROCEEDINGS

1. **Gouda et al.** (2024). "*Evaluation of Traffic Signs Placement Readiness for Autonomous Vehicles using Point Cloud Data*." Transportation Research Board, Washington D.C., USA, 2024.
2. **Gouda et al.** (2024). " *Lane Marking Geometric Design Assessment for Autonomous Vehicles using Light Detection and Ranging (LiDAR) Data*." Transportation Research Board, Washington D.C., USA, 2024.
3. **Gouda et al.** (2022). "*Safety Performance Assessment of Existing Roads Under Autonomous Vehicle Driving Scenarios in Digital Twins*." Road Safety on Five Continents, RS5C, Grapevine, Texas, USA, 2022.

4. **Gouda et al.** (2022). "*Leveraging Big Digital Infrastructure Data to Improve Road and Roadside Design for Automated Vehicles*." Transportation Research Board, Washington D.C., USA, 2022.
5. **Gouda et al.** (2022). "*Road and Roadside Geometric Design Assessment for Autonomous Vehicles in Virtual Reality*." Transportation Research Board, Washington D.C., USA, 2022.

ACKNOWLEDGMENTS

I would like to thank my supervisor, Dr. Karim El-Basyouny, for his guidance, efforts, and support during my studies at the University of Alberta.

My sincere appreciation goes to my defense committee members for their editorial recommendations and advice.

I would like to express my sincere gratitude to my professors.

I thank my family, parents, brothers, and friends for their spiritual support throughout my graduate studies in Canada.

I also thank Dr. Sharon Harper for her editorial recommendations, advice, and continuous support.

I thank Alberta Transportation (AT) for providing the data used in this thesis.

This research was supported by the Natural Sciences and Engineering Research Council of Canada. I acknowledge the support of the Natural Sciences and Engineering Research Council of Canada (NSERC).

Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada (CRSNG).

I would like to acknowledge the support and funding I have been awarded and extend my gratitude to all of the award and funding sponsors. This includes the Izaak Walton Killam Memorial Award, the University of Alberta President's Doctoral Prize of Distinction, ITS Canada awards, Transport Canada, etc.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

- AADT: Average Annual Daily Traffic
- AASHTO: American Association of State Highway and Transportation Officials
- ADAS: Advanced Driver Assistance Systems
- AI: Artificial Intelligence
- ASD: Available Sight Distance
- AT: Alberta Transportation
- AUT: Autonomous Technology
- AV: Autonomous Vehicles
- AVs: Autonomous Vehicles
- BoVPs: Bag-of-Visual-Phrases
- CAS: Collision Avoidance Systems
- CAV: Connected Autonomous Vehicle
- CAVs: Connected Autonomous Vehicles
- CC: Cloud Compare
- CNN: Convolutional Neural Networks
- CoC: The City of Calgary
- CoE: The City of Edmonton
- CORDIS: The Community Research and Development Information Service
- CRSNG: Cette recherche a été financée par le Conseil de recherches en sciences naturelles et en génie du Canada
- DBSCAN: Density-Based Spatial Clustering of Application with Noise
- DMV: Department of Motor Vehicles
- DSD: Decision Sight Distance
- DSM: Digital Surface Model
- DTM: Digital Terrain Model
- DZG: Z-Gradient Norm
- FHWA: Federal Highway Administration
- FN: false Negatives
- FOV: Field of Vision

- FP: false Positives
- FTVG: Future Transport Visions Group
- GIS: Geographic Information System
- GNSS: Global Navigation Satellite System
- GPS: Global Positioning System
- GPU: Graphical Processing Unit
- HD: High-Definition
- HOG: Histogram of Oriented Gradient
- HRMTD: High-Resolution Micro Traffic Data
- I2V: Infrastructure-to-Vehicle
- IIHS: Insurance Institute for Highway Safety
- IOOs: Infrastructure Owners/Operators
- ITS: Intelligent Transportation Systems
- LAD: Look-Ahead Distance
- LiDAR: Light Detection and Ranging
- LOS: Line-of-Sight
- MLS: Mobile Laser Scan
- MUTCD: Manual on Uniform Traffic Control Devices
- NACTO: National Association of City Transportation Officials
- NCHRP: National Cooperative Highway Research Program
- NSERC: Natural Sciences and Engineering Research Council of Canada
- ODD: Operational Design Domain
- PCA: Principal Component Analysis
- PRT: perception reaction time
- PSD: Passing Sight Distance
- RANSAC: Random Sample Consensus
- ROI: Regions of Interest
- ROR: Run-off-Road
- SAE: Society of Automotive Engineers
- SD: Sight Distance

- SLAM: Simultaneous Localization and Mapping
- SSD: Stopping Sight Distance
- SURF: Speeded Up Robust Features
- SVMs: Support Vector Machines
- TAM: Transportation Asset Management
- TIN: Triangular Irregular Network
- TP: True Positives
- TRB: Transportation Research Board
- TSI: Traffic Sign Inventories
- TTO: Time to Takeover
- US DOT: United States Department of Transportation
- USA: The United States of America
- V2I: Vehicle-to-Infrastructure
- V2N: Vehicle-to-Network
- V2P: Vehicle-to-Pedestrian
- V2V: Vehicle-to-Vehicle
- V2X: Vehicle-to-Everything Communications
- WCPP: Wildlife Collision Prevention Program

1 INTRODUCTION

1.1 BACKGROUND

The race for transportation automation and the mass deployment of Connected and Autonomous Vehicles (CAVs) technology is rising, not least due to their perceived social and economic benefits. CAV technologies are expected to create a paradigm shift urged by global socio-economic and environmental megatrends to enhance traffic safety, mobility, economic competitiveness, supply-chain efficiency, and air quality (AASHTO, 2020; European Commission, 2019). The United Kingdom, European Union, Australia, China, and the United States (U.S.) have adopted national visions and roadmaps to meet the needs of CAVs through synchronized research, standards development, and pilot deployments (AASHTO, 2020; Australian Transport and Infrastructure Council, 2019; European Commission CORDIS, 2019; Gouda et al., 2023; Zenzic, 2019).

With recent breakthroughs in CAV technologies and available self-driving vehicle solutions (Elliott et al., 2019; Gouda, Mirza, et al., 2021; Tesla, 2023; Waymo, 2023), public and private agencies and infrastructure owners/operators (IOOs) started contemplating the role of infrastructure in automation. They have also asked the question, how do we prepare for autonomous vehicles? (3M, 2020; AASHTO, 2020; NACTO, 2019; NCHRP, 2020; Transport Canada, 2023; US DOT FHWA, 2020; US DOT National Science and Technology Council, 2020). Questions that, without doubt, require a shared vision and collaboration between involved stakeholders, including IOOs and the automotive industry (AASHTO, 2020; NCHRP, 2020; Transport Canada, 2023; US DOT FHWA, 2020)

Despite recent advances in the automotive industry and related research (Janai et al., 2020; Tesla, 2023; Waymo, 2023), a fully autonomous system in complex and unstructured environments has not been realized to date due to limitations of autonomous vehicle (AV) technologies (Elliott et al., 2019; Gouda et al., 2023; Janai et al., 2020). The enormous scale of computation and power required for AVs to operate safely in real-time and at high levels of autonomy is beyond the capabilities of any battery-powered vehicle control system (Janai et al., 2020; L. Liu et al., 2021). As such, offloading expensive computation workloads to smart

infrastructure is an active area of research (Janai et al., 2020; L. Liu et al., 2021; US DOT National Science and Technology Council, 2020).

Recent approaches to allow high levels of autonomy shifted towards the enhancement of road infrastructure readiness for AVs by Infrastructure Owners/Operators (IOOs) (AASHTO, 2020; Farah et al., 2018; Gopalakrishna et al., 2019; Gouda et al., 2023; Gouda, Mirza, et al., 2021; Janai et al., 2020; NCHRP, 2020; TomTom, 2023; US DOT FHWA, 2020, 2021). Readiness is defined as taking a proactive approach in implementing the road infrastructure changes required for CAVs, as opposed to being reactive to challenges encountered as AVs are deployed (AASHTO, 2020; FTVG, 2021; US DOT FHWA, 2021). At a conceptual level, enhancing road infrastructure readiness by IOOs has three approaches (AASHTO, 2020; NCHRP, 2020). First, improving vehicle-to-infrastructure connectivity (V2I), known as "talking to the road." Second, enhancing the vehicle's ability to monitor the driving environment, or "seeing the road." Finally, "simplifying the road" through modifying roadway and roadside geometric design to support navigation by CAVs. The ultimate goal is to control the vehicle's Operational Design Domain (ODD) to realize the full potential of CAV technologies. ODD defines the environment within which an autonomous vehicle can operate safely (Colwell et al., 2018; NCHRP, 2020; US DOT FHWA, 2021).

Digital maps and road design datasets are essential in allowing high levels of autonomous driving (Elghazaly et al., 2023; Farah et al., 2018; Here, 2023; Ilci & Toth, 2020; Janai et al., 2020; NVIDIA, 2023; TomTom, 2023). If available, such information supports the vehicle system with attributes, at the centimeter to millimeter-level accuracy, about lane gradient, lane curvature, roadside features inventory (e.g., traffic signs), speed limits, and dynamic traffic and weather information (Farah et al., 2018; Here, 2023; Z. Liu et al., 2020; NVIDIA, 2023; TomTom, 2023). Light detection and ranging (LiDAR) data is a practical source for such information. LiDAR utilizes laser scanning equipment, global positioning systems (GPS), and navigation technologies to obtain intensity and positional information of surrounding features. The output of the LiDAR scanning process is a rich 3D point cloud of the surveyed objects that supports the digitization of the environment from the collected data (S. A. Gargoum & El Basyouny, 2019; Gouda et al., 2023; Hinks et al., 2015; Laefer, 2020; Vo et al., 2019; Zhang et al., 2019).

Collected sensor data by CAVs are processed using a powerful onboard computer system and complex algorithms designed to support real-time localization, planning, and decision-making (Elliott et al., 2019; Janai et al., 2020; NACTO, 2019; Tesla, 2023; Waymo, 2023). Occlusion is

one of AV performance's most challenging safety risks (Baumgartner et al., 2020). Local AV collision avoidance systems (CAS) could fail if subjected to unexpected objects/road-users beyond their sensor range or vision. This phenomenon is often referred to as "the Line-of-Sight (LOS) limitation" or "Occlusion limitation" in automotive engineering research (Baumgartner et al., 2020; Gábor Orosz et al., 2017; Ge et al., 2018; He et al., 2019). In IOOs current road design standards, the corresponding concept is known as "Sight distance," which is a core element of road design guidelines and existing road infrastructure design (AASHTO, 2020; Alberta Infrastructure and Transportation, 2022; Khoury et al., 2019; Khoury & Amine, 2019; S. Wang & Yu, 2019).

Sight distance is defined as the visible length of a roadway in front of a vehicle from a vantage point on the travel lane (AASHTO, 2018; Alberta Infrastructure and Transportation, 2022). Examples of sight distance design requirements include but are not limited to stopping sight distance (SSD) and passing sight distance (PSD) (AASHTO, 2018). SSD is the required distance for a driver to completely stop before hitting an obstacle, such as crossing animals, pedestrians, and vehicles (AASHTO, 2018). Road designers verify that the available sight distances (ASD) are always higher than the required SSD at any point on the road. Sight distance requirements mainly depend on the speed limit, the driver's perception-reaction time (PRT), pavement surface longitudinal coefficient of friction, deceleration rate, and grade of the road segment (AASHTO, 2018). Currently, roadway design guidelines are defined with human perception capabilities in mind, which differ from how AVs scan and react to the environment.

Another vital element in roadway design is roadside clear zones, defined as the traversable areas on either side of a roadway, including shoulders (US DOT FHWA, 2023). Clear zones provide time for deceleration and recovery of an errant vehicle. However, if filled with obstructions or rough terrain, they reduce the ASD, threaten fatal collisions (e.g., vehicle run off the road collisions, animal collisions, etc.), and provide no alternative avoidance option for drivers in case of an on-road obstruction (AASHTO, 2018; Alberta Infrastructure and Transportation, 2007).

Due to the tremendous cost of building highway infrastructure that meets design guidelines and limiting factors, design exceptions are common with approval by IOOs (Alberta Infrastructure and Transportation, 2018). For instance, roadside clear zones often do not meet guideline recommendations due to right-of-way costs and planned or conflicting utilities. Similarly, sight distance requirements are unmet due to roadside obstacles (e.g., vegetation, mountainous terrain, other utilities) and the high cost of building a smooth road alignment (Alberta Infrastructure and

Transportation, 2007, 2018). Furthermore, pavement deterioration and changes during road maintenance over the project life may cause deviation from design standards. Limited sight distance is a leading cause of collisions on highways, and wildlife-animal collisions account for more than 50% of vehicle collisions on Alberta highways, with more than 1,000 accidents per month (S. A. Gargoum, El-Basyouny, et al., 2018; Government of Alberta, 2013).

Reported failures involving AVs during testing raised concerns about their compatibility with existing infrastructure (California DMV, 2023; Dadvar & Ahmed, 2021). While there are several probabilistic methods developed in CAV research for handling urban short-range occlusion in intersections at low speeds (< 30 kph), these approaches do not provide a practical, safe, and complete solution to the problem (Shetty et al., 2021). State-of-the-art research recommended using offline maps to extract physical infrastructure occlusions before dealing with dynamic occlusions (Schorner et al., 2019; M. Y. Yu et al., 2019). Besides, long-range, high-speed occlusion is not sufficiently addressed due to several limitations.

AV sensors' mapping performance, from the vehicle point of view, degrades significantly at long ranges and high speeds, with each sensor type having different limitations (Ilci & Toth, 2020; Kumar et al., 2020; Mohammed et al., 2020; Patole et al., 2017; Sensible4, 2020; US DOT FHWA, 2021). Dynamic occlusions hinder the vehicle's ability to perceive the physical environment and static occlusions, both essential for path planning and collision-avoidance maneuvers (Ilci & Toth, 2020; Wong et al., 2021). The required 3D reconstruction in real-time to perceive the long-range physical environment is computationally infeasible (Ilci & Toth, 2020; Janai et al., 2020; L. Liu et al., 2021; Tesla, 2019). In addition, without the knowledge of the environment beyond the occlusions, AVs cannot safely modulate their speeds and plan their path.

Traffic signs information, lane geometry and markings are also identified as key requirements for safe navigation by AVs (Gopalakrishna et al., 2019; Gouda et al., 2023; Lawson, 2018; US DOT FHWA, 2021). These road features' current design, placement and guidelines are also based on human driver requirements (AASHTO, 2018; FHWA, 2009). Speed limit compliance, lane-keeping, simultaneous localization and mapping (SLAM), and hazard warning for AV systems rely on these road signs and markings. Reviewing road signs, markings, and geometry by IOOs for AVs is essential for their safety (Lawson, 2018; US DOT FHWA, 2021).

Addressing the gap between existing road infrastructure design and CAVs is a trending topic suggested in recent studies and major initiatives (AASHTO, 2020; Gopalakrishna et al.,

2019; Gouda et al., 2023; Gouda, Mirza, et al., 2021; NCHRP, 2020; Senate Canada, 2018; Transport Canada, 2019; US DOT FHWA, 2021). Such efforts brought together representatives from the IOOs, the automotive industry, and academia to define the changes required by IOOs to prepare roadways for the safe rollout of CAVs (AASHTO, 2020; NCHRP, 2020; US DOT FHWA, 2021). The key aspects identified include the network-level physical infrastructure occlusions, road geometric design, travel lanes and pavement markings design, roadside design, traffic signs placement, and the deployment of countermeasures required to improve design readiness for CAVs at substandard locations (AASHTO, 2020; Farah et al., 2018; Gopalakrishna et al., 2019; Gouda et al., 2023; Gouda, Mirza, et al., 2021; Janai et al., 2020; NCHRP, 2020; TomTom, 2023; US DOT FHWA, 2021). This thesis aims to address the gap identified in recent initiatives by developing quantitative methods to assess existing road design for the deployment of AVs on a network-level and to explore the possible practical countermeasures to improve compatibility at substandard locations.

1.2 PROBLEM STATEMENT

Existing road infrastructure is designed based on human constraints. This includes the design of highway alignments, horizontal curves, vertical curves, roadside clear zones, placement of traffic signs, etc. With the increasing interest in the mass deployment of AV technologies, IOOs need to meet the needs of AVs. Recently, collaboration between public and private agencies, IOOs, and AV industry experts provided recommendations for IOOs to assess and improve the AV readiness of existing transportation infrastructure. Currently, the required design performance assessment on a network-level is a challenge. This work aims to develop methods to assess road design performance for the deployment of AVs and to explore the possible practical countermeasures to improve readiness. To reach this goal, the following research problems are covered:

1.2.1 Problem One: Understanding the Relationship Between AVs and Road Design Features

There is an urgent need to understand how AV technologies will impact design guidelines and the risk of their deployment on existing roads. Limited number of studies and initiatives explored the impact of autonomous vehicles on road design (AASHTO, 2020; Khoury & Amine, 2019; Mcdonald, 2018; NCHRP, 2020; Ray, 2017; US DOT FHWA, 2021; S. Wang & Yu, 2019). A

comprehensive review of these efforts is needed to inform the development of performance assessment analytics on existing roadways.

1.2.2 Problem Two: Large-Scale Occlusion Assessment of the General Road Environment

Occlusion due to the physical infrastructure is a significant limitation requiring input from IOOs. As the deployment of CAV technologies is on a network-level, the development of new methods for occlusion mapping must serve the ability to test long road segments automatically within a reasonable amount of time. Recent research has recommended the use of offline digital maps to detect occlusions and develop methods for identifying unsafe locations for AVs (Gouda et al., 2023; Schorner et al., 2019; US DOT FHWA, 2021; M. Y. Yu et al., 2019). LiDAR digitization is one way to satisfy this requirement. Due to its millimeter-level accuracy, cost-saving in data collection, collection speed, and widespread data collection capabilities, research and practice have gravitated towards the use of LiDAR to perform large-scale infrastructure asset management, condition assessment and maintenance (S. A. Gargoum, El Basyouny, et al., 2018; Gouda, Chowdhury, et al., 2021; Gouda, Epp, et al., 2022; Gouda et al., 2020; Kilani et al., 2021; Laefer, 2020). Novel LiDAR applications in transportation engineering and research include sight distance analysis, intersection sight distance analysis, roadside clear zones mapping, asset management, conflict-based traffic safety management, traffic surveys and vertical clearance mapping (Agina et al., 2021; S. A. Gargoum, El Basyouny, et al., 2018; S. A. Gargoum & El Basyouny, 2019; S. Gargoum & El-Basyouny, 2017; Gouda, Mello, et al., 2021; Gouda, Shalkamy, et al., 2022).

1.2.3 Problem Three: Road and Roadside Design Assessment for AVs

While a few recent studies discussed the problem, all studies focus on the issue from either a qualitative view or a minor assessment of the impact on specific guidelines design equations. Regarding the design aspect of existing roads, no studies provide a performance-based and quantitative measure of existing roads' design performance for AVs. Several government entities and design standards development groups (e.g., AASHTO) called for research on this problem. Suggested changes to road design equations can be combined with the extracted information from point cloud data for a performance assessment of existing roadways. For instance, the use of AV PRT, deceleration rates, road and roadside grades, available sight distances based on vehicle

sensors, and the visibility of roadside zones can be used to assess road design performance and detect substandard locations for AVs.

1.2.4 Problem Four: Traffic Signs Assessment

Transportation agencies follow the Manual on Uniform Traffic Control Devices for Streets and Highways (MUTCD) for the design and placement of traffic signs (FHWA, 2009). MUTCD is designed based on human constraints. Moreover, severe weather conditions, changes in the surrounding environment, accidents, and maintenance, among other reasons, lead to changes in traffic signs placement. Hence, traditional surveying methods are performed by IOOs to build traffic sign inventories (TSIs) in order to maintain traffic signs.

Lawson (Lawson, 2018) discussed the importance of traffic signs to AVs for avoiding collisions. AVs can only properly read a traffic sign when it is correctly placed, oriented, and unobstructed. Current studies do not consider the parameters of autonomous vehicles when defining the visibility of a traffic sign. Therefore, it is crucial to investigate if the current placement of traffic signs is adequate for an autonomous vehicle to process and react to highway sign information (Guo et al., 2019).

1.2.5 Problem Five: Pavement Marking Assessment

Reliable lane marking detection is a key component in bridging the gap between approximate knowledge of an AV's location using, for example, a GPS, to exact positioning (Laconte et al., 2022; US DOT FHWA, 2021). Connected and automated vehicle (CAV) systems have been designed with the constraints of human-friendly infrastructure. For instance, state-of-the-art research and practice in AV motion/speed planning approaches rely on either traffic signs designed for human drivers or parsing human GPS speed profiles from naturalistic data and using it for AV systems training (Anastassov et al., 2017; Claussmann et al., 2020; Rodrigues et al., 2018; TomTom, 2023; Xinli et al., 2016). Furthermore, in a highway setting, enough contextual clues and lane markings are provided, given that humans can drive safely in their lane through curves or locations of limited pavement markings visibility (AASHTO, 2018; A. S. Huang et al., 2009). The same cannot be said for autonomous driving systems, which have been proven to have trouble staying within their lane when road markings are ambiguous, eroding, or on combinations of horizontal and vertical curves as their lane-keeping systems fail (IIHS, 2018; B. Li et al., 2018;

Tak et al., 2022; Tudose, 2019). In such situations, path planning algorithms lack the intuition innate in humans, making them unsafe.

Pavement markings are identified as the most important infrastructure feature for autonomous vehicles (AVs) operations (US DOT FHWA, 2021). Current roadway infrastructure is plagued by a variety of issues, making the pavement marking detection task significantly complex and difficult (e.g., lighting conditions) (Burghardt et al., 2021; Pike et al., 2018; Zheng et al., 2019). Obstructed pavement markings within the vehicle sensors' field of view (FOV) have been recently promoted as a potential concern for AVs deployment (US DOT FHWA, 2021). Recent research suggests that addressing this problem in order to improve AV sensors' monitoring of road markings is a key responsibility of IOOs (US DOT FHWA, 2021). Ideally, CAVs must be able to provide a safe and comfortable experience without user intervention, which correlates directly to their ability to plan their trajectory based on the information they collect (Elbanhawi et al., 2015; H. Liu et al., 2019).

1.2.6 Problem Six: Readiness Improvement Countermeasures

A crucial aspect of this research is to explore suitable countermeasures to solve the identified issues at substandard locations for CAVs. Literature and practice reviews are carried out to synthesize the potential and relevant countermeasures.

1.3 OBJECTIVES

For AVs to realize their full potential, challenges facing their deployment in as many situations as possible must be addressed. Research has shown that a complete assessment of road environments is yet to be achieved. Such an understanding could vastly expand the domain in which AVs could safely operate. The ultimate goal of this work is to understand and address the gap between infrastructure design and CAV technologies. Using this information, developing methods for detecting substandard locations for AVs proactively and quantitatively can be performed. This would help stakeholders (e.g., IOOs, automotive industry professionals, etc.) make informed decisions about infrastructure upgrades. The objectives of this thesis are aligned with the problems identified in the Problem Statement section. These objectives are to 1) Review the road design features needed to be reviewed by IOOs for CAVs in recent initiatives and studies, 2) develop

automated methods to assess design performance for CAVs on a large scale for the different design features, 3) present case studies where substandard locations are identified, and 4) Provide a synthesis of design improvement countermeasures. Figure 1 shows a simplified flowchart of the general steps carried out in this thesis.

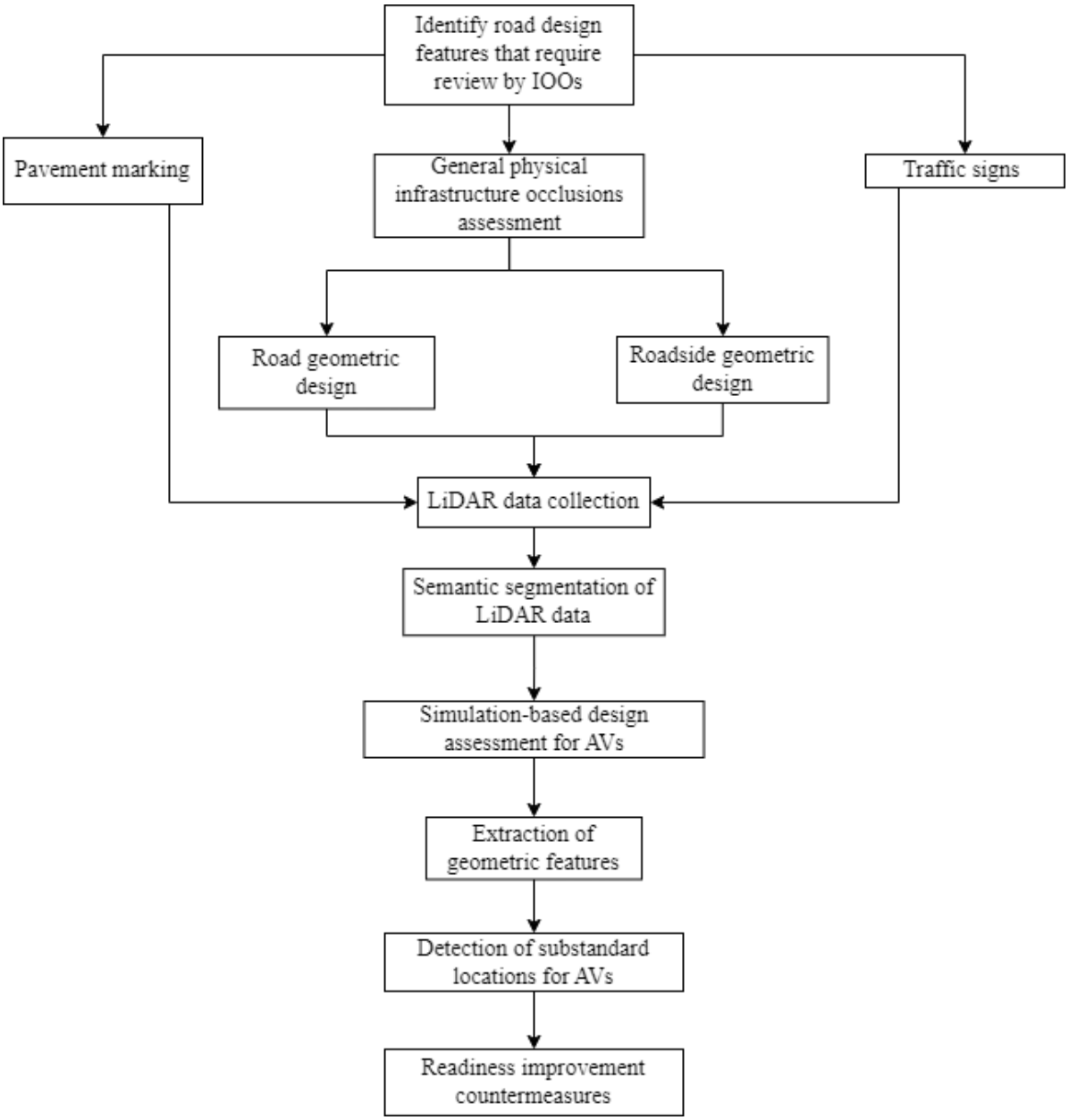


Figure 1 General flowchart of readiness assessment

This work proposes a simulation-based approach that utilizes LiDAR data to study the interaction between AVs and road design elements.

1.4 THESIS STRUCTURE

The remainder of this thesis is divided into six different chapters. Details of the topics covered in each chapter are described below:

Chapter 1 presents an introduction to the thesis, background information related to the research, the research problem, the main objectives, and the thesis structure. It shows a background of recent trends by IOOs to identify their role in the AV ecosystem. In addition, it shows the key identified aspects of the interaction between AVs and existing road infrastructure that require input from IOOs. Based on this background, the chapter summarizes the problem statements this work covers.

Chapter 2 provides a detailed literature review of the topics addressed in this thesis. This chapter covers background and reviews on autonomous vehicles, the impact of CAVs on road design, the impact of CAVs on roadside design, pavement markings and CAVs, and the use of LiDAR in road design assessment.

Chapter 3 examines the simulation-based method for occlusions and geometric design information mapping and roadway assessment. The chapter goes over the data structures and the creation of the convex hulls/polyhedrons used to search for occlusions to targets on the road and roadside environment. Details of the voxel-based raycasting comparison method are presented. Finally, three case studies for detecting locations with substandard conditions for AVs are presented.

Chapter 4 covers traffic signs assessment into two sub-sections: 1) semantic segmentation and 2) traffic signs assessment. The semantic segmentation section shows a background about extracting traffic signs from point cloud data and recent methods in the literature. Then, it covers the data collection and description and how the training dataset used in this thesis was developed. The proposed method, results and discussion, conclusions and future work are finally presented. The remainder of the chapter is dedicated to the traffic signs assessment for Avs, covering the method, results, discussion and conclusions.

Chapter 5 looks into the pavement marking assessment methodology. The method shows the creation of a field of vision on the pavement surface that is used to search for occlusions between the modelled vehicle sensors and the lane marking. Once locations with occlusions are detected, a Bézier curve-based path is predicted to find the vehicle trajectory. Using the trajectory, the radius and centripetal acceleration are mapped along the trajectory and compared to suggested values in road design guidelines and research to assess the segment performance. Finally, results and conclusions are presented.

Chapter 6 includes a summary of the research conducted in this thesis, a discussion of the contributions of this work, and a discussion of topics for future research. It also includes the readiness improvement countermeasures promoted in practice and the literature. It shows the different countermeasures that can be applied at detected substandard locations identified through the methods presented in this thesis.

2 LITERATURE REVIEW

2.1 AUTONOMOUS VEHICLES

According to the Society of Automotive Engineers (SAE), there are six levels of vehicle automation (SAE International, 2019; Saeed et al., 2021). These levels range from 0 to 5. Level 0 refers to no automation. Levels 1 and 2 refer to an autonomous vehicle system monitored by a human driver who is assisted by advanced driver assistance systems (ADAS) or relinquishes non-fallback-control (steering, acceleration, and braking) to the ADAS. Levels 3 to 5 refer to autonomous vehicle systems that consider less human driver input, including monitoring of the driving environment and vehicle operation. In levels 4 and 5, this also includes fallback-control, with 5 being completely autonomous.

Autonomous collision avoidance systems (CAS) rely mainly on the data collected by sensors aboard the vehicle to identify hazards and safe areas, such that the proper operational commands can be translated into safe vehicle manoeuvres to avoid collisions (Elliott et al., 2019; J. Liu et al., 2017). Therefore, the data collection capability and accuracy of such sensors are the most significant factor in the effectiveness of CAS (Elliott et al., 2019; Seif & Hu, 2016).

To enhance the effectiveness of CAS, vehicle-to-everything communications (V2X) can be utilized to receive external information on road conditions, obstacles, etc. (3M, 2023; Elliott et al., 2019; L. Liu et al., 2021; Mobileye, 2021). V2X includes vehicle-to-vehicle (V2V), vehicle-to-infrastructure (V2I), vehicle-to-pedestrian (V2P), and vehicle-to-network (V2N) communications (Elliott et al., 2019). All components of CAV technologies have advantages and limitations. No single component is adequate in completely supporting high levels of autonomous driving (3M, 2023; Elliott et al., 2019; Z. Liu et al., 2020; Seif & Hu, 2016).

For example, CAS cannot be designed solely to use V2N communication, as network outages and delays could completely impair the CAS. Additionally, not all obstructions or road features are part of the network, such as pedestrians, animals, non-connected objects, etc. (3M, 2023; Elliott et al., 2019; NCHRP, 2020). In such a case, the CAS must be able to resort back to routine autonomous operation using the vehicle on-board sensors (Elliott et al., 2019). It has also been recently recommended that CAVs should be compatible with the use of Artificial Intelligence (AI) to analyze and identify pedestrian gestures and behavior, as well as erratic animal behavior, to recognize additional hazards each may pose (Elliott et al., 2019).

Occlusion remains one of the largest challenges facing AV safety. Recent studies employ a variety of strategies to combat occlusion (Baumgartner et al., 2020; Hubmann et al., 2019; Orzechowski et al., 2018; Schorner et al., 2019; M. Y. Yu et al., 2019). (Shetty et al., 2021) show that state-of-the-art methods are based on estimating the state and behavior of occluded road users using AV system observations in limited and controlled ODDs within travel lanes. The study concluded that these approaches do not provide a practical, safe, and complete solution to the problem and can lead to a significant number of collisions in the general road environment. The paper discussed several scenarios presented in previous research. It proved mathematically that ensuring safety using probabilistic methods in real-time is impossible even if a crash risk comparable to current observed crash rates is accepted. The main cause of this is that the AV would not have enough time to take evasive actions. Assuming an AV reaction time of 0.83 s, the discussed scenarios included occluded vehicles and pedestrians and the inability to accurately predict vehicle actions based on past observations (as in (Baumgartner et al., 2020)).

While the paper proposed a crash probability in several scenarios, the authors suggested that road geometry and conditions, unexpected behaviours of road users, and AV limitations (sensing limitations and perception failures) can increase the probability of collisions. The authors concluded that reliance on I2V (infrastructure-to-vehicle) connectivity by installing sensors on the roads and V2V communications are a viable solution to resolve the occlusion problem. Research has also shown that accounting for all possible occlusions in the field is computationally impossible (Orzechowski et al., 2018).

Recent studies have recommended leveraging prior information about the environment using offline maps to minimize the problems caused by occlusion and to develop methods for identifying unsafe locations (Schorner et al., 2019; M. Y. Yu et al., 2019). Addressing limitations such as the computationally expensive and infeasible reconstruction of the environment in real-time and sensor technology limitations through leveraging prior maps, assessing the general road environment (travel lanes as well as roadside zones), and redesigning focusing on minimizing hazardous occlusions are examples of topics future research could explore to safely expand the reach of CAVs.

2.2 IMPACT OF CAVS ON GEOMETRIC DESIGN

Although highway design has evolved over many years through research and practice, it has generally developed around the characteristics of human drivers and their constraints. Now, with the expected increase of AVs deployment, recommendations are adapting to accommodate them. Agencies, such as AASHTO, are considering new evidence and data from emerging research to inform new geometric highway design standards (AASHTO, 2020).

The effects of AVs on highway geometric design have been explored by (Khoury et al., 2019; Khoury & Amine, 2019). By applying conservative estimates for AV reaction time and sensor height to current AASHTO geometric design equations, a redesign of a mountainous road in Hasbaya, Lebanon, was performed. Alignment redesign was based on the calculated SSD and rates of curvature (K) values for AVs. It was found that highway construction costs and environmental impacts could be reduced when designed for use by AVs due to fewer amounts of cut and fill required to reach the subgrade level. The authors suggested that future research considers road grades as AASHTO, and recent studies assume a level terrain.

Similar work by (Wang & Yu, 2019) studied the impact of AVs with different levels of autonomy on geometric design. Appropriate design criteria for each level, 0 to 5, such as reaction time and sensor height, were used. Similarly, to (Khoury et al., 2019), it was concluded that as the autonomy level increased, so did the design speed of the roadway, while the length of crest and sag curves and the SSD decreased. An earlier study by (Mcdonald, 2018) suggested that the SSD for AVs decreases due to reduced reaction time (0.3 s and 1.0 s).

(Gábor Orosz et al., 2017; Ge et al., 2018) tested vehicle reaction to a stopped vehicle occluded by road geometry due to a combination of a horizontal and vertical curve. They concluded that using V2X communications to notify a CAV about the occluded object avoids severe braking that would be unsafe for passengers.

2.3 IMPACT OF CAVS ON ROADSIDE DESIGN

Roadside design, especially on highways, is an area of attention for CAVs. Generally, a roadside should be constructed as wide and forgiving as reasonably possible while still considering physical, economic, and environmental constraints (Alberta Infrastructure and Transportation, 2007).

Clear zones are provided adjacent to roadways in an attempt to provide a traversable area, void of obstacles, to accommodate errant vehicles so that they can safely return to the roadway or come to a complete stop (Alberta Infrastructure and Transportation, 2007). Often, clear zones consist of a shoulder and some form of recoverable, non-recoverable, and traversable slopes. Clear zone width is determined by the road design speed, annual average daily traffic (AADT), and steepness of the embankment/slope. Where adequate clear zones are not available, additional safety countermeasures (e.g., guard rails) are required (Alberta Infrastructure and Transportation, 2007).

In recent studies, the recoverability of AVs across various levels of autonomy has been researched. (Favarò et al., 2019) analyzed data from 2014 to 2017 regarding the disengagement of autonomous technology (AUT) in levels 2 and 3 AVs on public roads in California. The data recorded the time to takeover (TTO), i.e., the reaction time of the driver to take fallback control of AV in off-nominal scenarios. This is an area of concern for lower-level AVs that relinquish software control of the vehicle back to the human driver when threats or hazards outside of normal operating conditions occur. It was found that the average TTO was between 0.83 s and 1.0 s for drivers trained to operate AVs. The authors noted the TTO for the average untrained driver is expected to be higher; thus, this is likely a minimum average expected TTO. It was also found that over longer time spans, as drivers became more experienced with monitoring the AUT, their TTO slightly decreased, contrary to previous estimates that comfortability and trust of the AUT increased TTO.

With the increase of autonomy in AVs, systematic fallback and recovery control methods are being researched. Work by (Freeman et al., 2016) simulated two control algorithms, sliding control and linear quadratic control, for use in AV run-off-road (ROR) recovery systems. ROR crashes are often the result of human error, i.e., poor driving performance, leading up to the incident (Freeman et al., 2016). As such, this error can be reduced by allowing a sufficiently advanced AV ROR recovery system to perform instead of a human driver. This would apply to AVs categorized level 4 and higher as fallback control of the vehicle at these levels is the responsibility of the system.

(Freeman et al., 2016) used a custom dynamic vehicle model to predict and calculate the behavior of the vehicle and its system in real-time. This model was run through multiple ROR scenarios with each of the sliding controller and linear quadratic controller. The sliding controller

implemented successful recoveries for nominal ROR scenarios but lacked in performance with varying roadway and vehicle conditions. The linear-quadratic controller performed more consistently across the tested ROR scenarios, recovering 25% faster while using 70% less steering than the sliding control method. The linear-quadratic method is supported by the authors as a recommended ROR recovery controller for autonomous vehicles. It should be noted that the development and hardware implementation of the system for further research is required to validate the method.

On the other end, roadside clear zone design has always developed around human drivers and is yet to change to accommodate AVs. Though, with lower-level AVs on the road, longer reaction times to off-nominal scenarios are to be expected, TTO plus normal human reaction time, ~2.5 s for stopping (AASHTO, 2018), as human drivers are still responsible for fall-back control of the vehicle. It may be beneficial for clear zone design to be modified as AVs become more conventional transportation tools. Similarly, as higher-level AVs develop and become more conventional as well, their automated ROR and other recovery systems may be advanced enough to also require different safety protocols than those that exist today. Whether such safety protocols become stricter, more lenient, or both, it is imperative that roadside safety is adequate to accommodate the deployment of all levels of AVs on the road.

Through Vehicle-to-Infrastructure (V2I) connectivity, CAVs can receive external information about the road and roadside environment (J. Chen et al., 2019; Malik et al., 2020; Razali et al., 2020; Tak et al., 2020; S. Wang & Li, 2019). In (J. Chen et al., 2019), roadside LiDAR sensors are used to track and detect vehicle trajectories. The authors note that current CAV applications are constrained by the resolution of data input as CAVs require a network passing high-resolution micro traffic data (HRMTD). This means real-time traffic data shared across all road users and related infrastructure. It has been shown in the past that LiDAR is capable of providing HRMTD (J. Chen et al., 2019). Their method processes LiDAR data by performing a region of interest (ROI) selection, ground surface filtering, point clustering, object classification, and lastly, vehicle tracking. Though this method was for computing vehicle trajectories to analyze traffic patterns, the authors mention other possible applications of HRMTD using roadside LiDAR for future research.

Some of these applications include predicting pedestrian trajectories at intersections and urban settings, predicting animal trajectories near rural highway settings, and performing traffic

micromanagement in high-traffic areas. Infrastructure like this provides additional warning of obstacles and other hazards to CAVs that they would not be able to detect using the CAV local system. It may also enable remote control and/or suggestive manoeuvres for CAVs in certain hazardous or high-traffic areas. This can be especially useful on highway segments/locations that have high animal traffic on and near the roadway. These applications are suggested as future research or possibilities and are not included in the study. Therefore, the costs, limitations, and validity of such applications are merely speculative and require further analysis to be justified before the practicality of their deployment in critical settings can be discussed.

Although animal fencing is one of the most used safety techniques for mitigating wildlife-vehicle collisions, they are not 100% effective and are typically installed where wildlife road mortality is already high due to the significantly high fencing cost (Ascensão et al., 2021; J.W. Gagnon et al., 2020; M.P. Huijser et al., 2015; Spanowicz et al., 2020; WCPP, 2023b). (M.P. Huijser et al., 2015) suggested that road conditions, sightlines, visibility distance and grades of the road and roadside, traffic speeds, and driver abilities can be used as proactive criteria for good practice in deploying collision mitigation measures (e.g., fencing and detection systems). Noteworthy is that the yearly cost of wildlife-vehicle collisions in Canada is estimated at $800 million, with a significant number of high-severity collisions (causing driver injury or death) (Desjardins Insurance, 2021; WCPP, 2023a).

While AVs introduce new safety concerns, innovation and continuous research have shown the potential of solutions and should be considered with the evolution of AVs. Collision avoidance systems, vehicle recovery systems, improving connectivity at hazardous locations, and road/roadside redesign are important topics for AVs as agencies and IOOs continually push for their deployment. It is uncertain which measures will be considered essential and which will be discretionary. As such, it is an emerging research area proving the advantages of different techniques, practices, and technologies.

2.4 PAVEMENT MARKINGS AND CAVS

Many industry experts proclaim that modifying, updating and standardizing lane markings to facilitate their detection is paramount in the mass deployment of AVs (US DOT FHWA, 2021). In bridging the gap between approximate knowledge of a CAV's location using, for example, a high-definition (HD) map or GPS, to exact positioning, reliable detection of lane markings is often a

key component (US DOT FHWA, 2021). Machine vision is a markedly popular solution for many advanced driver assistance systems (ADAS), such as lane-departure warning and lane-keeping assistance (Pike et al., 2018). These tasks are further convoluted when considering the different types of lane markings to recognize, where varying colours and patterns indicate different and relevant information.

Despite the importance of CAVs detecting lane markings, current roadway infrastructure is plagued by various issues, making that task significantly more complex and difficult. Lighting conditions can substantially affect the contrast between markings and the road. During the daytime, depending on the sun's angle or in the presence of overhead lamps, glare can reduce detectability (Zheng et al., 2019). In the nighttime, a lack of contrast between markings and adjacent pavement is also an issue that persists under wet conditions (Burghardt et al., 2021; Pike et al., 2018). Furthermore, no standard is used to determine the point at which lane markings have faded enough to warrant maintenance (Pike et al., 2018; US DOT FHWA, 2021). Lastly, and most relevant to IOOs, are the cases where the geometry of the road reduces the sight distance to the point where safe and comfortable manoeuvring is impossible within a reasonable reaction time (Pike et al., 2018; US DOT FHWA, 2021).

Industry experts have identified a few key areas in which the quality of lane markings and their compatibility with ADAS could be developed. Ideas such as creating a uniform design and quality for lane markings (i.e., fixed-width) (US DOT FHWA, 2021), ensuring marking design has suitable contrast in all adverse conditions (i.e., wetness, night, etc.) (Pike et al., 2018), and developing standards for minimum reflectivity where maintenance becomes mandatory have been put forth for the sake of simplifying the job of ADAS. Suggestions regarding the road geometry and standards specific to lane markings were first promoted by recent FHWA reports (US DOT FHWA, 2021).

Because of their critical importance in localization, several studies have explored various methods to detect or extract lane markings (Burghardt et al., 2021; Zheng et al., 2019). Regarding sensor types, the most popular option is to use cameras. The generalized procedure many approaches follow is to take an input image, convert it to a greyscale for computational efficiency purposes, apply various filters to remove noise and use Hough transform to detect edges within the processed image. Despite being the most popular method, there are several limitations to taking

an image-based approach. Firstly, environmental effects, such as rain or fog, tend to reduce accuracy by reducing image quality (Zheng et al., 2019). Additionally, most algorithms are well suited to detect straight lines but show poorer results for curved lines (Kaur & Kumar, 2015).

Other studies have shown LiDAR to be a viable method for lane marking extraction. With the advantage of reconstructing the 3D surrounding environment, LiDAR point cloud data can leverage geometric features unique to roadways to extract drivable regions. The returning laser from a scanner also carries intensity information. Intensity has been shown to successfully distinguish lane markings from the rest of the pavement (Jung & Bae, 2018; Y. Wang & Tsai, 2018).

2.5 TRAFFIC SIGNS AND CAVS

Much of the existing research on autonomous vehicle sign detection has been focused on image processing algorithms rather than the optimal positioning of the signs themselves. This literature review concentrated on existing research on traditional detection methods to better understand potential difficulties with these algorithms concerning sign positioning. In addition, state-of-the-art research on traffic sign occlusion detection is presented.

(Shladover & Bishop, 2015) presented a paper addressing predicted future issues regarding the introduction of autonomous vehicles. One topic mentioned was the importance of traffic signs for connected autonomous vehicles (CAV). The authors highlighted how traffic sign recognition is necessary for vehicles in areas with a high sign density, like construction zones, or while they are disconnected from the CAV network and cannot receive road information digitally, such as during times of poor connectivity due to hacking, adverse weather conditions, etc. In these cases, sign recognition is essential as a reliable source of real-time information. Sign data from individual vehicles can then be uploaded to the network and aggregated to update digital sign inventories in real-time. The paper concluded that accurate autonomous vehicle sign recognition is critical for CAV deployment as the vehicles, by design or circumstance, may not always be connected to a network and, therefore, will rely on on-vehicle sensors for information.

Research done by (Wahyono et al., 2014) investigated a method of traffic sign recognition by autonomous vehicles using cascade support vector machines (SVMs). Signs were identified by segmenting images by colour, isolating known sign shapes, and identifying differentiating patterns

on the faces of signs. Segmentation was handled primarily using a Histogram of Oriented Gradient (HOG). The completed program was run on a database of 300 images, with sign recognition and detection rates of 90.82% and 89.12%, respectively. The paper also noted how rotation and occlusion of traffic signs complicated the image processing techniques used.

(Walters et al., 2019) reviewed the difficulties connected autonomous vehicles face in rural areas. They noted that CAVs were prone to become disconnected from the V2I network in these areas and were reliant entirely on onboard sensor systems and GNSS for navigation. In these cases, identification of road signs allowed CAVs to intake road regulatory information and make positional adjustments during "blind time" when global navigation satellite system (GNSS) signals were temporarily interrupted. The authors pointed out how the development of robust sign detection programs is required to ensure safety during extreme weather conditions, which generally lower the accuracy of existing sign identification techniques. The authors recommended regular inspection and cleaning of rural road signs to improve their visibility and identification by CAVs.

(L. Chen et al., 2012) investigated multi-sensor system design for autonomous vehicle environmental perception. The authors noted how sensors with high contrast improve sign recognition, as most existing detection algorithms rely heavily on colour matching. This was tested by constructing a multi-sensor array of visual and laser scanners on a vehicle and then processing the sign data using Speeded Up Robust Features (SURF). The algorithm operated by identifying regions of interest in the images (ROI locking) and then segmenting out the individual colours. Detection rates for signs were 94.3%, taking 50ms per frame.

(Geese et al., 2018) investigated sensor performance in autonomous vehicles and advanced driver assistance systems. The study focused on a comparison between human observers and visual sensors. They recognized that the human eye has much better contrast detection abilities than existing image processing techniques and can better adapt to unideal conditions. The paper found that digital sensors and image processing algorithms struggled to locate the parts of an image that should be denoised, especially for dark images.

(Wali et al., 2019) investigated current difficulties with vision-based sign detection systems. They identified that the traditional analysis methods of detection, tracking, and classification could be improved by merging steps together and allowing, for instance, the

classification of a sign to be used to update its detected shape. The study also examined environmental effects on sign detection. Inconsistent lighting, sign clustering, and sign rotation, among other factors, generally decreased the accuracy of detection algorithms. It was also identified that colour and shape remain the most common metrics detection algorithms use, and that convolutional neural networks produce the most accurate results.

(S. Wu et al., 2015) investigated traffic sign visibility using retro-reflectivity and ground plane proximity. Signs were ranked based on vertical driver view angle, planarity concerning the travel direction, and view distance, among other factors. Rankings were done based on sign visibility. Test sign data was collected using a mobile laser scanning (MLS) system that utilizes laser scanners, digital cameras, and a wheel-mounted distance measurement indicator. A sample of 100 sign images was tested both by the proposed algorithm and a human reviewer. Signs were categorized by visibility rankings, with less than a 5% discrepancy between human and computer rankings observed.

(Hirt et al., 2022) investigated the occlusion detection of traffic lights and signs caused by vegetation using LiDAR data and 3D City models. The model included the location and orientation of streets, traffic lights, and traffic signs. The method used ray-tracing in an occupancy grid generated by the voxelization of the LiDAR data to identify occluded traffic signs. The authors found that their method is feasible for occlusion detection by vegetation in the urban street scenario. (Kilani et al., 2021) and (Gouda, Chowdhury, et al., 2021; Gouda, Mello, et al., 2021) used voxel-based raycasting for sight distance analysis and obstruction detection on urban and rural roads. (P. Huang et al., 2017) detected occluded traffic signs using 3-D point clouds and trajectory data acquired by MLS. Signs extraction was based on both the reflectance and geometric features of signs. Using the trajectory and traffic sign data, a hidden point removal algorithm was adopted to detect occlusions.

In conclusion, sign recognition is essential for the safe deployment and operation of autonomous vehicles. Additionally, several factors, including obstructions, rotation, color condition, shape consistency, lighting, and weather conditions, affect the performance of sign detection. The placement of signs influences occlusions and rotation and is relatively independent of the detection systems and algorithms used by autonomous vehicles. There is a lack of prior research regarding physical sign visibility that considers autonomous vehicle sensor characteristics

despite recommendations of previous studies. As such, this thesis aims to develop a simulation-based method to assess existing traffic sign infrastructure design for AV deployment using LiDAR data. In addition, the proposed method is compared to a redesigned state-of-the-art voxel-based raycasting approach to account for AV sensors.

2.6 USING LIDAR IN ROAD DESIGN ASSESSMENT

As arguably one of the most important elements of highway design, sight distance (SD) assessment received significant attention in the literature.

State-of-the-art SD research relies on using LiDAR for the automated assessment of ASD on roadways. (Khattak & Shamayleh, 2005) are one of the earliest studies to use aerial LiDAR data to assess compliance to SSD and PSD on highways. The method proved the feasibility of using LiDAR data to examine sight distances, which is much less time-consuming and more accurate than manual field surveying.

Utilizing the capabilities that had been proved viable with aerial LiDAR, (Castro et al., 2014) developed an automated approach to determine ASD. Using the LiDAR data, a digital terrain model (DTM) with a grid spacing of 1 m was generated, and sight distances to targets were calculated at different observer points, 5 m apart, along the vehicle trajectory. Though this method was effective, the authors suggested that the use of digital surface models (DSM), which considers elements above the ground surface as well, should be researched to better capture the surroundings of the roadway. This was later done by (Castro et al., 2016), who considered elements above the ground, where DSMs were compared against a DTM on the same segments. It was found that the sight distances calculated on the DSMs were shorter due to trees and vegetation along the roadways.

Another method proposed in 2018 (Jung et al., 2018) assessed the 3D ASD at intersections. The method voxelated 3D point cloud data of an intersection collected using LiDAR. By slicing this data on an x-y grid plane at different driver heights/perspectives, sightlines could protrude from the driver's perspective at some angular resolution along the x-y plane until intersecting with a voxel. The ASD was then calculated using Bresenham's algorithm. The authors suggested that a better modelling method is required as the inaccuracy of the angular resolution increases with sight distance; the sightlines diverge. (González-Gómez & Castro, 2020) used DTM and raycasting in

ArcGIS to assess ASD on roundabouts. Recent work by (S. A. Gargoum & Karsten, 2021) used a raycasting approach to assess available sight distances on highways using LiDAR data.

Using LiDAR for geometric highway design and sight distance assessment has become much more commonplace, supported by over a decade of research. To that end, all of the abovementioned studies in this section have only considered human drivers, their requirements, and their constraints. Recent studies used DTMs and DSMs to analyze 3D surfaces. DSMs can cause some inaccuracy (mostly false positives) as they are generated using triangulation and represented as a triangulated irregular network (TIN). DSMs enabled better ASD analysis techniques but do not properly represent overlapping surfaces when projected onto a 2D plane and, therefore, are not a true 3D representation of the surface (de Santos-Berbel et al., 2014; Jung et al., 2018). Finally, state-of-the-art studies use a raycasting voxel-based method, which suffers from major limitations (S. A. Gargoum & Karsten, 2021; Gouda, Chowdhury, et al., 2021).

(Gouda, Chowdhury, et al., 2021; Gouda, Mirza, et al., 2021) proposed the first simulation-based approaches for the assessment of road design for CAVs deployment using 3D point cloud data. AV sensors and computer system capabilities are modelled on existing roads, and then obstructions to vehicle sensors in the travel lane are queried. (Gouda, Mirza, et al., 2021) solved major limitations in previous studies and reduced the processing times of ASD assessment by 73% (35.5 seconds for a 10 km segment with 73,803,254 million points).

2.7 SUMMARY

In conclusion, the identified key elements of the infrastructure of importance to AV operation are: i) occlusion due to physical infrastructure, ii) road geometric design, iii) roadside and clear zone design, iv) pavement markings, and v) traffic signs. It was found that occlusion handling using the local AV system from the vehicle point of view is drastically limited and computationally infeasible. Recent research recommended using prior/offline maps to extract unsafe occlusions to the vehicle sensors in the general road environment. Several countermeasures at such unsafe locations are recommended, such as simplifying road design and improving V2I connectivity. However, no methods are developed to perform this assessment.

In terms of the impact on road geometric design, all studies focused on studying the impact in terms of changes to design guidelines equations/parameters with no focus on testing existing designs. Roadside clear zones are prompted as a need for CAVs as these areas would allow AV

run-off-road (ROR) recovery systems to avoid collisions. In addition, studies recommended using roadside sensors at substandard roadside locations to warn AVs of road conditions and existing wildlife animal traffic. The use of roadside visibility and grades as criteria for roadside fencing on rural highways was also suggested.

The placement of traffic signs and visibility of road surface and pavement markings are identified as some of the most important infrastructure elements to be addressed by IOOs (US DOT FHWA, 2021). For instance, if lane marking is occluded due to road features, the AV system would not be able to follow the road ahead and perform appropriate/comfortable manoeuvres. There is currently no method to perform this test in the literature.

Recent studies in road design assessment focus on using LiDAR to develop automated testing methods based on human driver constraints. Due to LiDAR's millimeter-level accuracy, reductions in data collection cost, and widespread data collection capabilities, civil engineering research has gravitated towards point cloud data. Before the work presented in this dissertation, no methods were developed to assess roads' design compatibility with AVs, and the applicability of existing methods to perform this task was not explored.

3 Occlusion Assessment and Road Design Features

The following explanation reflects the principles used to develop a testing approach for road design performance for AVs. The method was developed, taking into account recommendations of recent studies for future research. This methodology serves the purpose of evaluating the visibility of phantom targets on the road and roadside from the perspective of an AV travelling along a roadway. Figure 2 illustrates the high-level procedural flow of how data is processed in this methodology.

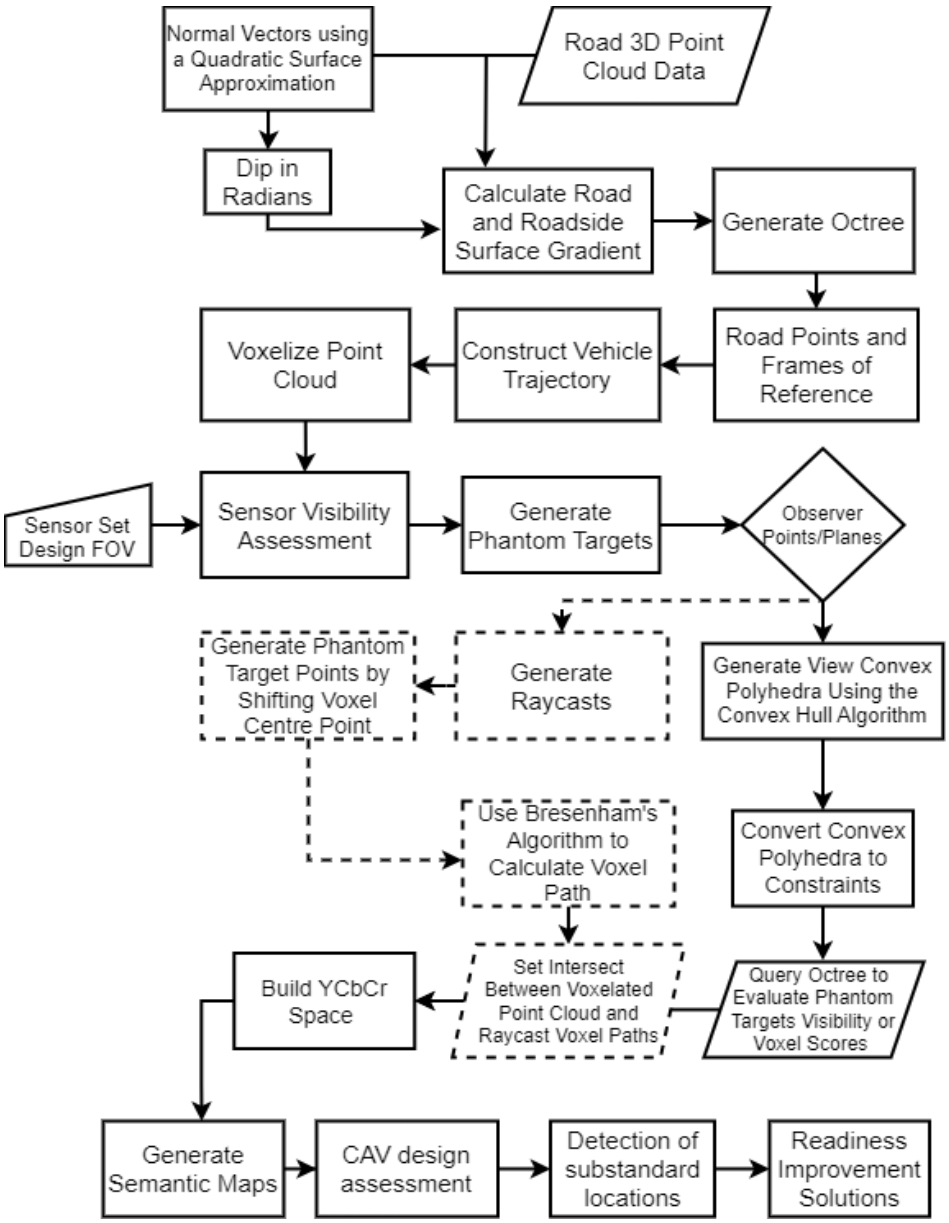


Figure 2 High-level procedural overview

3.1.1 Point Cloud Data

The input required for this methodology is reasonably dense point cloud data of roadways. Mobile LiDAR data collected from the perspective of a vehicle is effective in accomplishing this as it accurately captures relevant roadside features, trees, vertical surfaces, etc.

3.1.2 Data Structures

The proposed method uses a combination of an octree and voxels to compute sight distance. Though they are separate data structures, their relationship acts similarly to a modified voxel-octree (Laine, 2010), as they are computed over the same point set. Intuitively, a voxel-octree would be an octree organization of voxels, but this method uses the data structures for different purposes and thus calculates them separately over the point cloud data. Voxels are used for target placement and output representation, whereas the octree is used for obstruction querying, which is unrelated to the voxel structure itself. Obstruction querying with an octree is done by defining a set of plane equations, called constraints, to create a bounded solid, as shown in Figure 3.

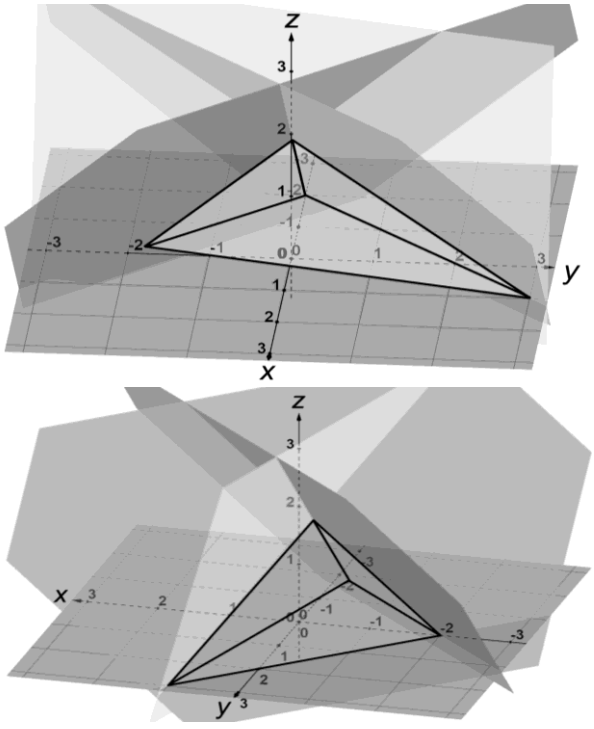


Figure 3 4 Half-spaces intersecting to bound a tetrahedron (two angles)

The intersection of a set of half-spaces in 3D space forms a convex polyhedron. In this case, with four planes making up four sides of the polyhedron, it is a tetrahedron. If the tetrahedron is not unbounded, i.e., bounded by these half-spaces, it is considered a closed object and can act as a 'bounded volume' input into the octree.

All points of the original input cloud were represented by three double precision floating point values, providing the $\boldsymbol{x}$, $\boldsymbol{y}$, and $\boldsymbol{z}$ coordinates of the point's location. Given the minimum positional coordinates in each dimension of the point set $\boldsymbol{x_{min}}$, $\boldsymbol{y_{min}}$, and $\boldsymbol{z_{min}}$, voxel coordinates $\boldsymbol{x_{vox}}$, $\boldsymbol{y_{vox}}$, and $\boldsymbol{z_{vox}}$ were given by the following equations.

$$\boldsymbol{x_{vox}} = \left\lfloor \frac{\boldsymbol{x - x_{min}}}{\boldsymbol{g}} \right\rfloor \tag{1}$$

$$\boldsymbol{y_{vox}} = \left\lfloor \frac{\boldsymbol{y - y_{min}}}{\boldsymbol{g}} \right\rfloor \tag{2}$$

$$\boldsymbol{z_{vox}} = \left\lfloor \frac{\boldsymbol{z - z_{mi}}}{\boldsymbol{g}} \right\rfloor \tag{3}$$

The complete set of voxel coordinates, $\boldsymbol{V}$, represents a simplified version of the input point cloud in voxelated form, where nearby points are aggregated together into cubes with a side length of $\boldsymbol{g}$. It should be noted that these are floored coordinates, i.e., they are the minimum corner of the voxel (in all coordinate directions). In the case of this thesis, the sight distance evaluation is computed over a voxelated point cloud at 0.10 m grid size ($\boldsymbol{g = 0.1\ m}$), which was an appropriate balance between computational speed and results quality (Figure 4).

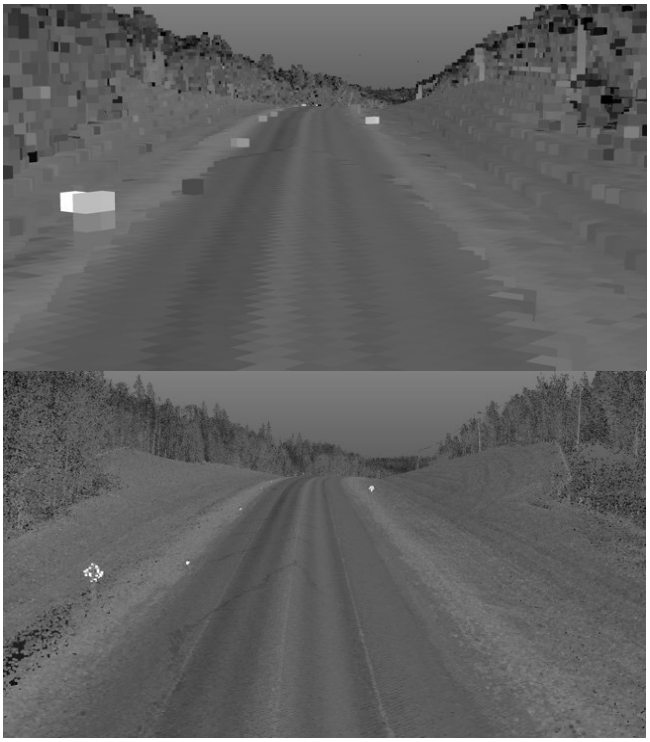


Figure 4 Comparison of 1.0 m (top) to 0.10 m (bottom) voxel grid size

3.1.3 Grade Percentage Assessment

Though this method focuses on sight distance assessment, grade percentage is also assessed on the input data. First, normal vectors are fitted to the point cloud using a quadric surface approximation oriented in the positive z-direction (upwards). These normal vectors can then be converted to the dip in radians from the positive z-axis, which provides the angular offset from the normal upward direction across the approximated surface. The slope of the approximated surface as a grade percentage is then the tangent of these dip values. The grade percentage can be aggregated and applied to the voxelated output set to be included in the results and the sight distance evaluation.

3.1.4 Road Points and Frames of Reference

To determine the view perspective of a vehicle, a parametric curve $\vec{\boldsymbol{P}}(\boldsymbol{t})$ must be fit to the roadway, which reflects the path of a vehicle along the pavement. In this work, since mobile LiDAR data was used to develop the input point cloud, it was utilized to create $\vec{\boldsymbol{P}}(\boldsymbol{t})$ in a manner similar to (Gouda, Mirza, et al., 2021). From the resulting parametric curve, a set of road points

$\boldsymbol{R} = \{\boldsymbol{r_1}, \boldsymbol{r_2}, \ldots, \boldsymbol{r_n}\}$, as shown in Figure 5 and the normalized forwards, leftwards, and upwards vectors , $\overrightarrow{\boldsymbol{f_i}}, \overrightarrow{\boldsymbol{l_i}}, \overrightarrow{\boldsymbol{u_i}}$, at each road point are defined (Figure 6).

Figure 5 Road points

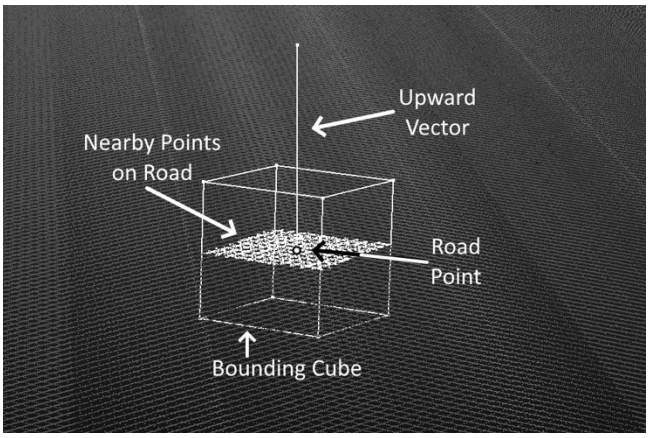


Figure 6 Upward vector calculation

3.1.5 Observer Points and Observer-planes

For each road point $\boldsymbol{r_i}$ spaced 1 m apart along the road, an observer point is given by translating upwards by a fixed height $\boldsymbol{o_i} = \boldsymbol{r_i} + \boldsymbol{1.2} \cdot \overrightarrow{\boldsymbol{u_i}}$, see Figure 7.

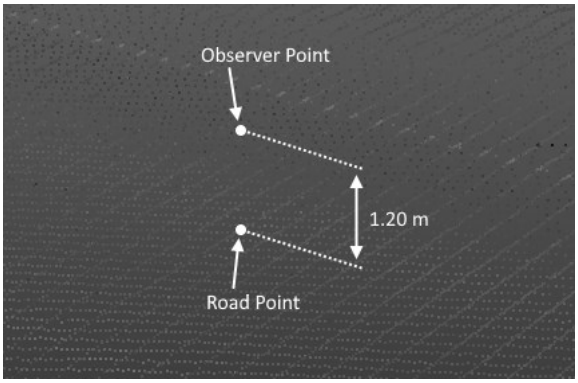


Figure 7 Illustration of observer point

An observer-plane (Figure 8) can be centered around the observer aligned with its respective leftward and upward vectors $\overrightarrow{\boldsymbol{l_i}}$ and $\overrightarrow{\boldsymbol{u_i}}$ and the normal of the plane represented by the observer's forward vector $\overrightarrow{\boldsymbol{f_i}}$. The width and height of the observer-plane, $\boldsymbol{w}$, and $\boldsymbol{h}$, are 3.75 m and 0.5 m, respectively. This width extension of the observer-plane provides a representation of a vehicle sensor across the width of the lane. The corner points of the observer-plane are calculated as follows:

$$\boldsymbol{c_1 = o_i + \overrightarrow{l_i} \cdot \frac{w}{2} + \overrightarrow{u_i} \cdot \frac{h}{2}} \quad (4)$$

$$\boldsymbol{c_2 = o_i + \overrightarrow{l_i} \cdot \frac{w}{2} - \overrightarrow{u_i} \cdot \frac{h}{2}} \quad (5)$$

$$\boldsymbol{c_3 = o_i - \overrightarrow{l_i} \cdot \frac{w}{2} - \overrightarrow{u_i} \cdot \frac{h}{2}} \quad (6)$$

$$\boldsymbol{c_4 = o_i - \overrightarrow{l_i} \cdot \frac{w}{2} + \overrightarrow{u_i} \cdot \frac{h}{2}} \quad (7)$$

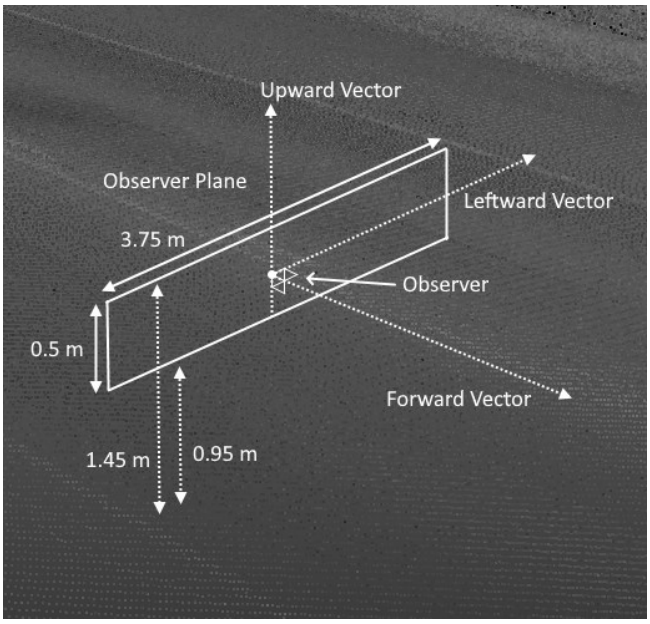


Figure 8 Illustration of observer-plane

Using an observer-plane considers the worst-case scenario out of all possible sensor positions at every interval in the lane, as it aggregates all the view perspectives from inside the plane into one result.

3.1.6 Sensor Visibility Assessment

The first analysis of the voxelated data is the sensor visibility assessment. This assessment determines which voxels are in the field of view (FOV) of the vehicle sensors along the trajectory, i.e., in the FOV from any one or more observers. If a voxel is not in the FOV of any observers, it is deemed 'not visible' to autonomous vehicle sensors and is not further analyzed for sight distance. Performing this assessment refines the voxelated cloud, increasing the performance of the rest of the analysis.

The FOV used in this thesis was modelled after real-world autonomous vehicle sensor set specifications (Tesla, 2023). Details regarding sensor modelling are derived in (Gouda, Mirza, et al., 2021).

Figure 9, Figure *10*, and Figure *11* illustrate the sensor spherical coordinate system, as well as the sensor set FOV.

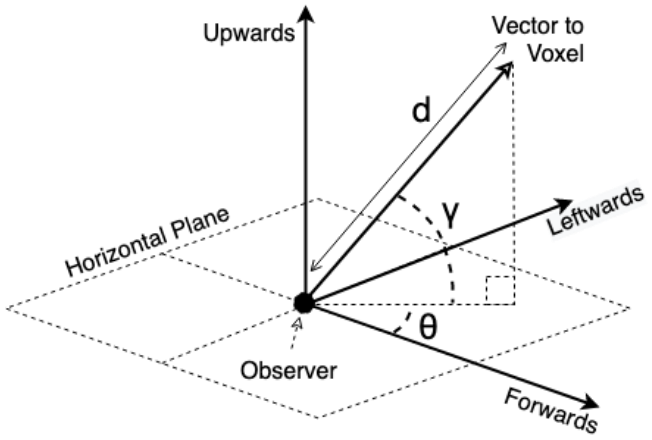


Figure 9 Spherical coordinate system for sensor bounds

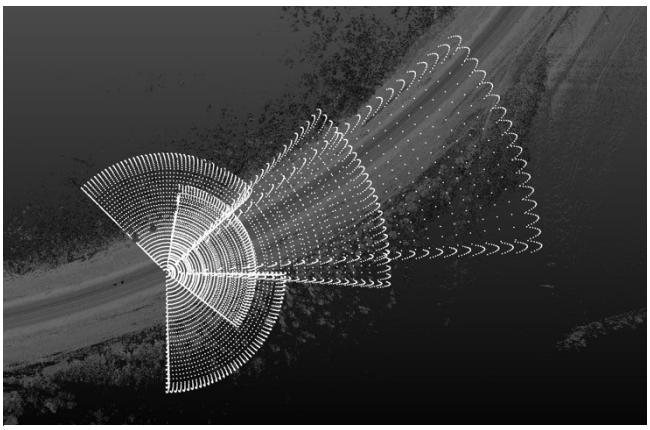


Figure 10 Sensor FOV (above)

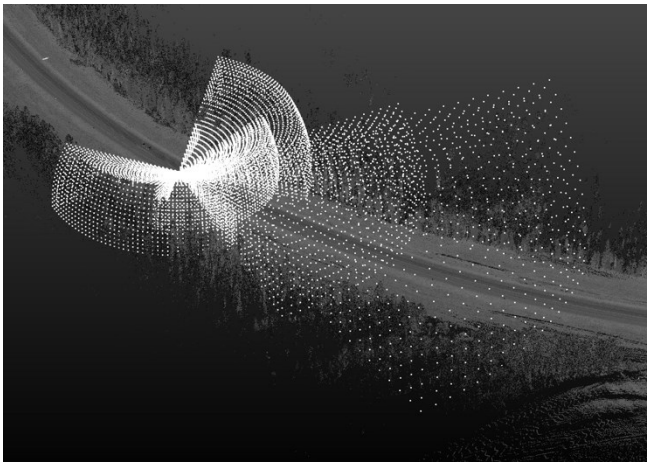


Figure 11 Sensor FOV (side)

3.1.7 **Target Planes**

In a voxelated point cloud, obstructing points exist inside voxels. Due to this, when analyzing the sight distance of a voxel, any obstruction searches or queries must not look inside the specific voxel itself.

3.1.7.1 Translation Approach

Each target plane $\boldsymbol{s_{i,j}}$ corresponds to a specific observer $\boldsymbol{o_i}$ and voxel $\boldsymbol{v_j}$ pair. The relative vector from the voxel to the observer $\overrightarrow{\boldsymbol{r_{i,j}}} = \boldsymbol{o_i} - \boldsymbol{v_j}$ is used as the normal vector of the target plane, which is translated some distance above the top of the voxel. For example, 0.38 m (Alberta Infrastructure and Transportation, 2022) was used for testing and development. The plane is then oriented with the z-axis $[\mathbf{0}, \mathbf{0}, \mathbf{1}]$ such that the horizontal components of $\overrightarrow{\boldsymbol{r_{i,j}}}$ are maintained, and the target plane $\boldsymbol{s_{i,j}}$ faces the observer. The cross product of the normal vector $\overrightarrow{\boldsymbol{r_{i,j}}}$ and the upward vector $[\mathbf{0}, \mathbf{0}, \mathbf{1}]$ provides the horizontal vector of the target plane, which allows the corners of the target plane to be calculated similarly to the observer-plane. Unlike the observer-plane, which has a fixed size, the target plane is calculated at the same width and height of the voxel grid size, as illustrated in Figure 12.

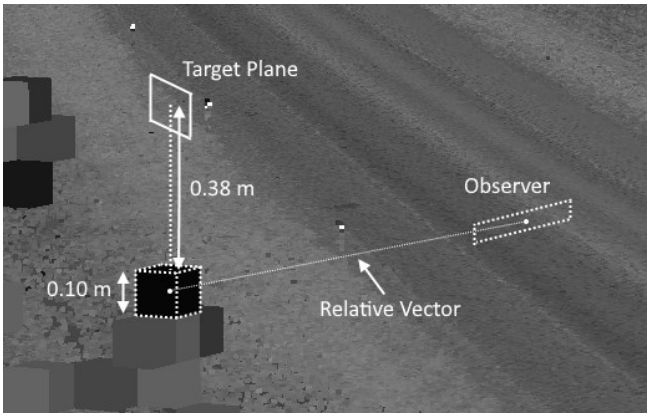


Figure 12 Illustration of voxel target plane

The target plane is translated 0.38 m upward so that the observer is looking above the ground, or the surface of the voxel being analyzed.

3.1.7.2 Voxel-Side Approach

An alternative to the translation approach is the voxel-side approach. The voxel-side approach uses the sides of the voxel as target planes. By determining the three physically detectable sides of the voxel, three target planes can be calculated for each voxel-observer pair.

Each target plane $\boldsymbol{s_{i,j,[1-3]}}$ is part of a set of 3 planes $\boldsymbol{S_{i,j}}$ corresponding to a specific observer $\boldsymbol{o_i}$ and voxel $\boldsymbol{v_j}$ pair. Due to a voxel's cubic geometry, the set of planes $\boldsymbol{S_{i,j}}$ is bounded at three, as it is only possible to see three sides of a cube from any arbitrary angle.

There are a few problems with this approach that cause inaccuracy in sight distance assessment due to the density of point cloud data. When creating a convex polyhedron to the vertical standing sides of the voxel, this approach works very similarly to the translation approach, but when creating the convex polyhedron to the horizontal sides of a voxel, the top or the bottom, the convex polyhedron can be quite narrow. In other words, the height or z-difference of the convex polyhedron is minimized when the voxel and observer are at similar heights. Therefore, the volume of the convex polyhedron is also minimized and acts more like a thin sheet than a view convex polyhedron. When this occurs, the convex polyhedron can sometimes slice between points that would normally be considered a surface, and thus, an obstruction. Due to this phenomenon, voxels that would normally be obstructed are recorded as visible on the top or bottom side due to the density of points obstructing the voxel. This is common with vegetation/foliage that is not as dense as ground surfaces, causing inaccurate inflation of ASDs.

3.1.8 View Convex Polyhedron

In analyzing sight distance, the motivation of this method was to use a three-dimensional representation of 'view' rather than a ray or line to better capture how sensors interact with their environment. Regardless of the observer-target view-perspective shape, the only condition was that the volume needed to be bound by planes such that the octree could interpret the constraints. Using a convex hull algorithm to create a view shape ensures that any convex polyhedron generated will meet such criteria. This allows the view perspective to be represented by any convex structure.

3.1.8.1 Convex Hulls

A set of points $\boldsymbol{P}$ in 3D space is considered convex if it contains every line segment connecting every point-to-point pair in $\boldsymbol{P}$:

$$\{\boldsymbol{P} \subseteq \boldsymbol{R^3} \mid \overline{\boldsymbol{pq}} \subseteq \boldsymbol{P}, \forall\, \boldsymbol{p}, \boldsymbol{q} \in \boldsymbol{P}\} \tag{8}$$

The convex hull (Figure 13) of set $\boldsymbol{P}$ in 3D space is defined by the four axioms below and equation 9 (E. Welzl & B. Gärtner, 2013; F.P. Preparata & M.I. Shamos, 1985):

1. It is the minimal convex set containing all points in $\boldsymbol{P}$.
2. It is the intersection of all convex supersets of $\boldsymbol{P}$.
3. It is the set of all convex combinations of points in $\boldsymbol{P}$.
4. It is the union of all tetrahedrons with vertices in $\boldsymbol{P}$.

$$\boldsymbol{C} \equiv \left\{ \sum_{\boldsymbol{k=1}}^{\boldsymbol{N}} \boldsymbol{\lambda_k}\, \boldsymbol{p_k} : \boldsymbol{\lambda_k} \geq \boldsymbol{0} \;\wedge\; \sum_{\boldsymbol{k=1}}^{\boldsymbol{N}} \boldsymbol{\lambda_k} = \boldsymbol{1} \right\} \tag{9}$$

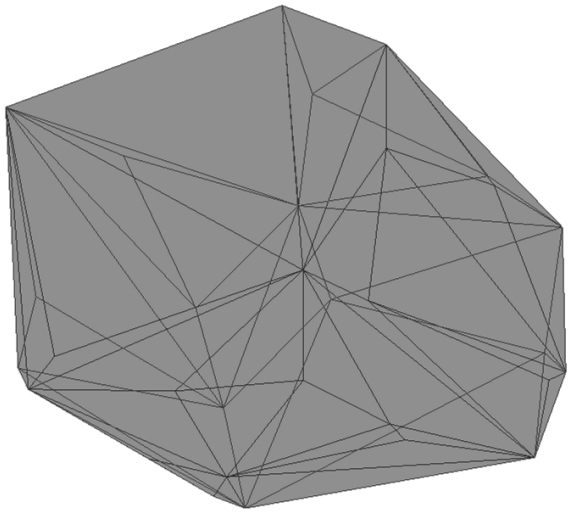


Figure 13 3D Convex hull of random 100-point cloud (transparent)

A triangular mesh can be calculated that contains all individual points $\boldsymbol{p_k}$ in the set, either as part of the mesh exterior or inside it, which can represent the convex hull of the set of points. Each face of the mesh is a triangle defined by three points $t_j = [\boldsymbol{p_m}, \boldsymbol{p_n}, \boldsymbol{p_o}]$. Every triangle also defines three edges $\overline{\boldsymbol{p_m p_n}}$, $\overline{\boldsymbol{p_n p_o}}$, $\overline{\boldsymbol{p_o p_m}}$, which are each shared with one other triangle in the mesh. At the end of the calculation, all edges will be 'closed,' or in other words, shared between two triangles, such that some volume of space is bounded by the mesh.

3.1.8.2 Defining the Hull Points

The view convex polyhedron is created as a convex hull around the corner points of the target plane $\boldsymbol{s_{i,j}}$ and either around the observer point itself for a total of five points, or the corner points of the observer-plane, around $\boldsymbol{o_i}$, for a total of eight points.

Determining the voxel target plane corners follows a similar approach to that of the observer plane. First, the middle point of the target plane is calculated by translating from the voxel's coordinates. The location of the target plane will differ based on the approach used. The translation approach moves from the centre of the voxel upward, whereas the voxel sides approach can be in any of six directions aligned with the coordinate axis.

3.1.8.3 Generating the Convex Hull

3.1.8.3.1 Initialization

In generating the convex hull of the set of points $\boldsymbol{P}$, a gift-wrap algorithm was developed to calculate the mesh. The initialization step of the gift-wrap algorithm involves finding the first triangle $\boldsymbol{t_1}$ in the mesh. This starts by locating a point with the most extreme coordinates in some dimension. In the case of this work, the point $\boldsymbol{p_{m1}}$ with the minimum $\boldsymbol{x}$ value is chosen. The second point $\boldsymbol{p_{n1}}$ in $\boldsymbol{t_1}$ is chosen by the smallest angle relative to the YZ – plane from the newly formed edge $\overline{\boldsymbol{p_{m1}p_{n1}}}$. Therefore, the first edge is guaranteed to be on the exterior of the hull. The third point $\boldsymbol{p_{o1}}$ in $\boldsymbol{t_1}$ is found by crossing vector $\overrightarrow{\boldsymbol{p_{m1}p_{n1}}}$ with a vector $\vec{\boldsymbol{e}} = \overline{\boldsymbol{p_{m1}p_{n1}}} - [\boldsymbol{1,0,0}]$ such that the result is tangent to the hull. By crossing this result again with $\overrightarrow{\boldsymbol{p_{m1}p_{n1}}}$ the result $\vec{\boldsymbol{n}}$ is now a vector normal to the hull. Lastly, for every point left in the set, the cross product $\overrightarrow{\boldsymbol{v_k}} = \overrightarrow{\boldsymbol{p_{m1}p_k}} \times \overrightarrow{\boldsymbol{p_{m1}p_{n1}}}$ provides another normal vector facing the opposite direction, that adding the point $\boldsymbol{p_k}$ would create. By taking the largest dot product between $\vec{\boldsymbol{n}}$ and any $\overrightarrow{\boldsymbol{v_k}}$, $\boldsymbol{p_k}$ can be distinguished as the point $\boldsymbol{p_{o1}}$ in the first triangle $\boldsymbol{t_1}$.

3.1.8.3.2 Iteration

The iteration process involves closing the edges of the mesh by selecting a point $\boldsymbol{p_k}$ in the set $\boldsymbol{P}$ that forms a new triangle. The new triangle will share at least one of the unclosed edges already existing in the mesh. The last iteration should close the final three remaining unclosed edges such

that all edges in the mesh are shared by two triangles, resulting in a closed surface, the convex hull around $\boldsymbol{P}$.

The triangle $\boldsymbol{t_1}$ added in the initialization created three unclosed edges, $\overline{\boldsymbol{p_{m1}p_{n1}}}$, $\overline{\boldsymbol{p_{n1}p_{o1}}}$, $\overline{\boldsymbol{p_{o1}p_{m1}}}$. To close an edge, the normal vector of the triangle $\overrightarrow{\boldsymbol{n_1}}$ should be found, which is done by crossing the edge vectors of the triangle: $\overrightarrow{\boldsymbol{n_1}} = \overrightarrow{\boldsymbol{p_{m1}p_{n1}}} \times \overrightarrow{\boldsymbol{p_{n1}p_{o1}}}$. Then calculate, for every other point that isn't on $\boldsymbol{t_1}$, the normal vector that the new point would create, $\overrightarrow{\boldsymbol{v_k}} = \overrightarrow{\boldsymbol{p_\iota p_{m\iota}}} \times \overrightarrow{\boldsymbol{p_{m1}p_{n1}}}$ just as done in the initialization step. The largest of the dot products between $\vec{\boldsymbol{n}}$ and any $\overrightarrow{\boldsymbol{v_k}}$ will distinguish the point in $\boldsymbol{t_2}$ that should close the edge $\overline{\boldsymbol{p_{m1}p_{n1}}}$, which can subsequently be removed from the list of unclosed edges.

Therefore, $\boldsymbol{t_2}$ is added to the mesh as $[\boldsymbol{p_{m1}}, \boldsymbol{p_k}, \boldsymbol{p_{n1}}]$ and thus, two more unclosed edges $\overline{\boldsymbol{p_{m1}p_k}}$ and $\overline{\boldsymbol{p_kp_{n1}}}$ are added to the list. The iteration will continue until all the edges are closed, and the mesh should contain a unique set of triangles defining the point connections of the convex hull. The following is a pseudocode description of this algorithm. Figure 14 provides a visual for an observer mesh in iteration. Figure 15, Figure 16, and Figure 17 display the convex hull polyhedron around and observer and voxel target planes from different angles; they display the result of this algorithm for a single target plane.

giftwrap3d

Inputs:

$\boldsymbol{P}$: An indexed set of input points, $\boldsymbol{p_k} \in \mathbb{R}^3$

Returns:

$\boldsymbol{C}$: A convex hull of $\boldsymbol{P}$ defining triangular edge surfaces by their corners, indexed from $\boldsymbol{P}$

$\boldsymbol{C} \leftarrow \emptyset$

unclosed_edges $\leftarrow \emptyset$

find_first_plane( ):

$p_{m1} \min_{x(P)}$)

For p_k in $P \setminus (p_{m1})$ **do**

$temp \leftarrow angle_{YZ(\overline{p_{m1}p_k})}$

$p_{n1} \leftarrow min(temp, p_{n1})$

$\vec{e} \leftarrow \overline{p_{m1}p_{n1}} - [1,0,0]$

$\vec{n} \leftarrow cross(\overline{p_{m1}p_{n1}}, \vec{e})$

For p_k in $P \setminus (p_{m1} \cup \mathrm{p_{n1}})$ **do**

$\overrightarrow{v_k} \leftarrow cross(\overrightarrow{p_{m1}p_k}, \overrightarrow{p_{m1}p_{n1}})$

$temp \leftarrow dot(\overrightarrow{v_k}, \vec{n})$

$p_{o1} \leftarrow max(temp, p_{o1})$

$\boldsymbol{C \leftarrow t_1 \leftarrow (p_{m1}, p_{n1}, p_{o1})}$

$unclosed\_edges \leftarrow (\overline{p_{m1}p_{n1}}, p_{o1})$

$unlcosed\_edges \leftarrow (\overline{p_{n1}p_{o1}}, p_{m1})$

$unlcosed\_edges \leftarrow (\overline{p_{o1}p_{m1}}, p_{n1})$

While $unclosed\_edges \neq \emptyset$ **do**

$(e, p_3) \leftarrow unclosed\_edges(1)$

pop_edge(1)

$\vec{n} \leftarrow cross(e, \overline{e(2)p_3})$

For p_k in $P \setminus (e \cup p_3)$ **do**

$\boldsymbol{\overrightarrow{v_k} \leftarrow cross(\overrightarrow{p_k e(1)}, \vec{e})}$

$temp \leftarrow dot(\vec{n}, \overrightarrow{v_k})$

$p_{new} \leftarrow max(temp, p_{new})$

$\boldsymbol{C \leftarrow t_i \leftarrow (e(1), p_{new}, e(2))}$

If $\overline{e(1)p_{new}} \notin unclosed\_edges$ **do**

$unclosed\_edges \leftarrow (\overline{e(1)p_{new}}, e(2))$

else *pop_edge*$(\overline{e(1)p_{new}})$

If $\overline{p_{new}e(2)} \notin$ *unclosed_edges* **do**

unclosed_edges $\leftarrow$ $(\overline{p_{new}e(2)}, e(1))$

else *pop_edge*$(\overline{p_{new}e(2)})$

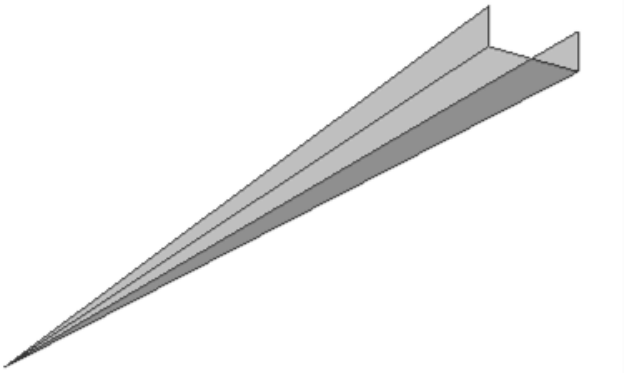


Figure 14 (Not-to-scale) observer-target mesh in iteration (3 triangles)

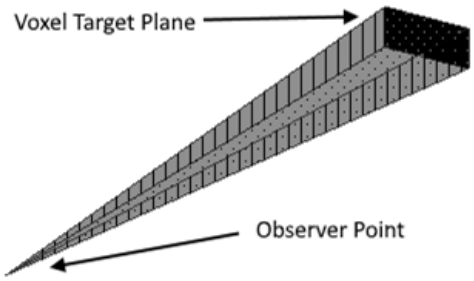


Figure 15 (Not-to-scale) convex hull polyhedron around an observer & voxel target plane

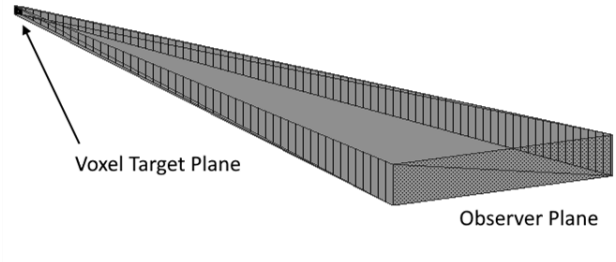


Figure 16 Convex hull polyhedron around an observer-plane & voxel target plane – angle 1

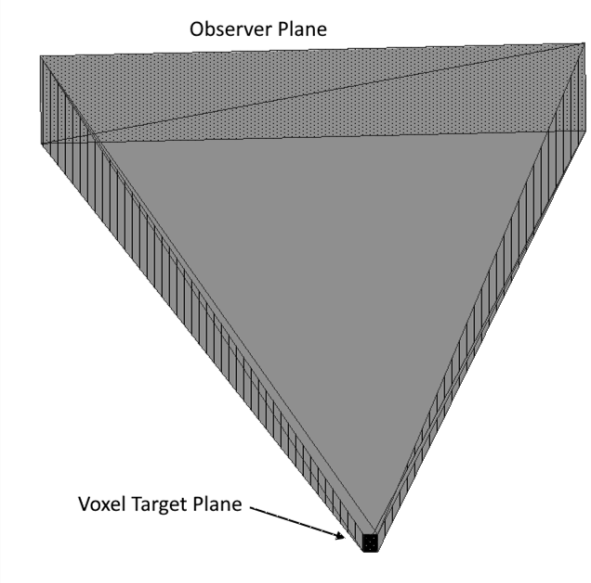


Figure 17 Convex hull polyhedron around an observer-plane & voxel target plane – angle 2

3.1.8.4 Generating Constraints

To evaluate the view convex polyhedron between an observer-voxel pair, the convex polyhedron is converted to a set of half-space constraints. The closure of half-spaces makes up the bounding volume $\boldsymbol{C_{i,j}}$.

3.1.9 Sight Distance Evaluation

To evaluate the sight distance to a voxel, two techniques were used: non-continuous sight distance and continuous sight distance. The non-continuous technique scores each voxel $\boldsymbol{v_j}$ by the number of observers $\boldsymbol{o_i}$ where $\boldsymbol{v_j}$ is in the sensor range and is not obstructed by any point $\boldsymbol{p_k}$. If no points are returned, the area described by the target plane $\boldsymbol{s_{i,j}}$, is visible and unobstructed, adding 1 m to the sight distance of the voxel, i.e., its 'score'. The continuous technique operates similarly to the non-continuous technique but additionally keeps track of the longest consecutive stretch of observers that have an unobstructed view.

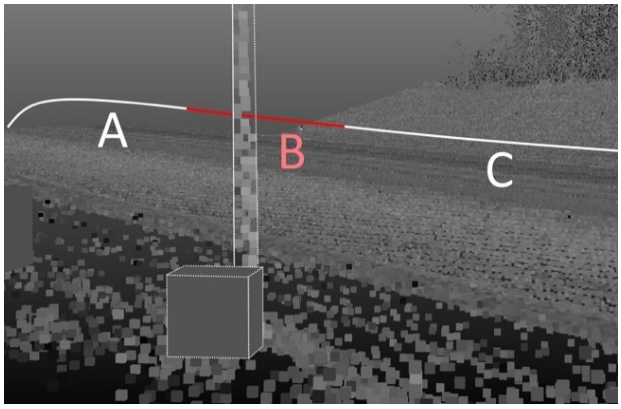


Figure 18 Continuous versus non-continuous sight distance example (0.1 m voxel size)

Figure 18 shows three sections along the path of observers, A, B, and C. In the non-continuous technique, the highlighted voxel would have a 'score' of all the observers in sections A and C, but in the continuous technique, it would have a 'score' of the observers in either section A or C, whichever is longer.

3.1.10 Verification

Sight distance evaluation using octree queries is verified manually by checking the coordinates of returned obstructions and confirming they do exist inside the convex polyhedron. In doing so, a number of layers can be generated and overlayed onto the original point cloud to visually confirm accuracy as well. These layers describe a separate object pertaining to the view convex polyhedron in question and the subset of points returned as obstructions from the octree query. Aligning these files and overlaying them onto the original point cloud provides visual confirmation that the returned obstructions do exist inside the view convex polyhedron. Some examples of detected obstructions are shown in Figure 19, Figure *20*, Figure *21*, and Figure *22*.

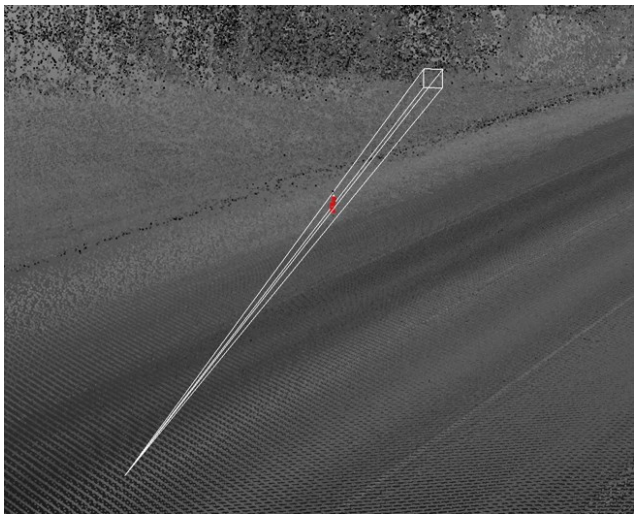


Figure 19 Obstructed view convex polyhedron from observer due to roadside reflector

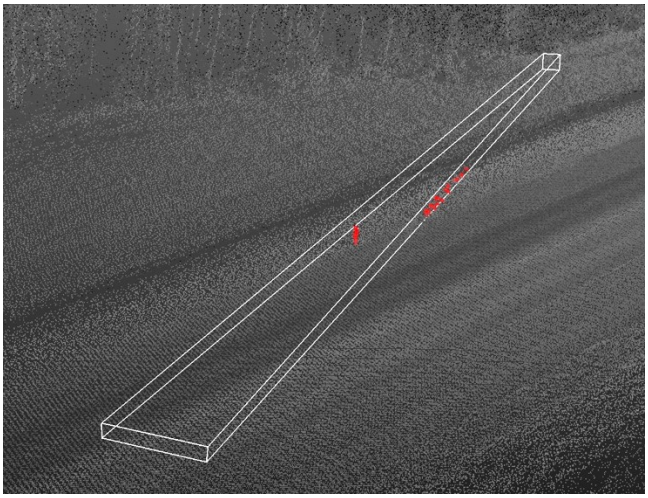


Figure 20 Obstructed view convex polyhedron from observer plane due to roadside

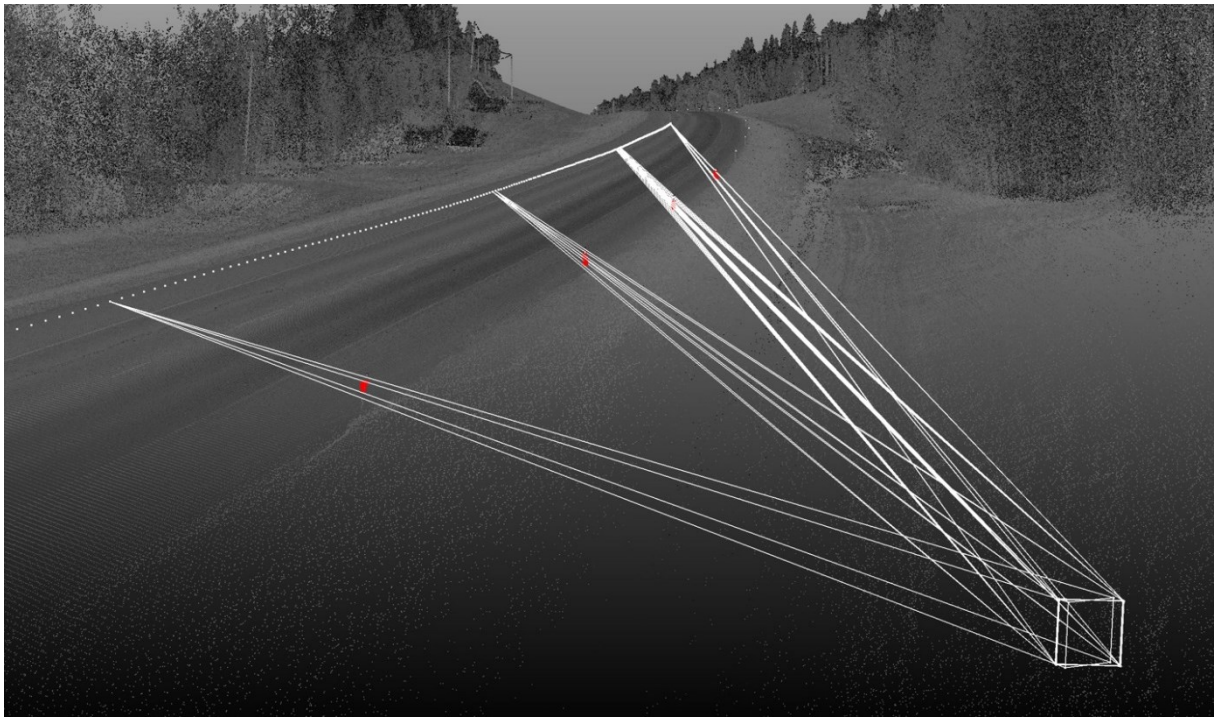


Figure 21 Set of obstructed view convex polyhedrons from observer points

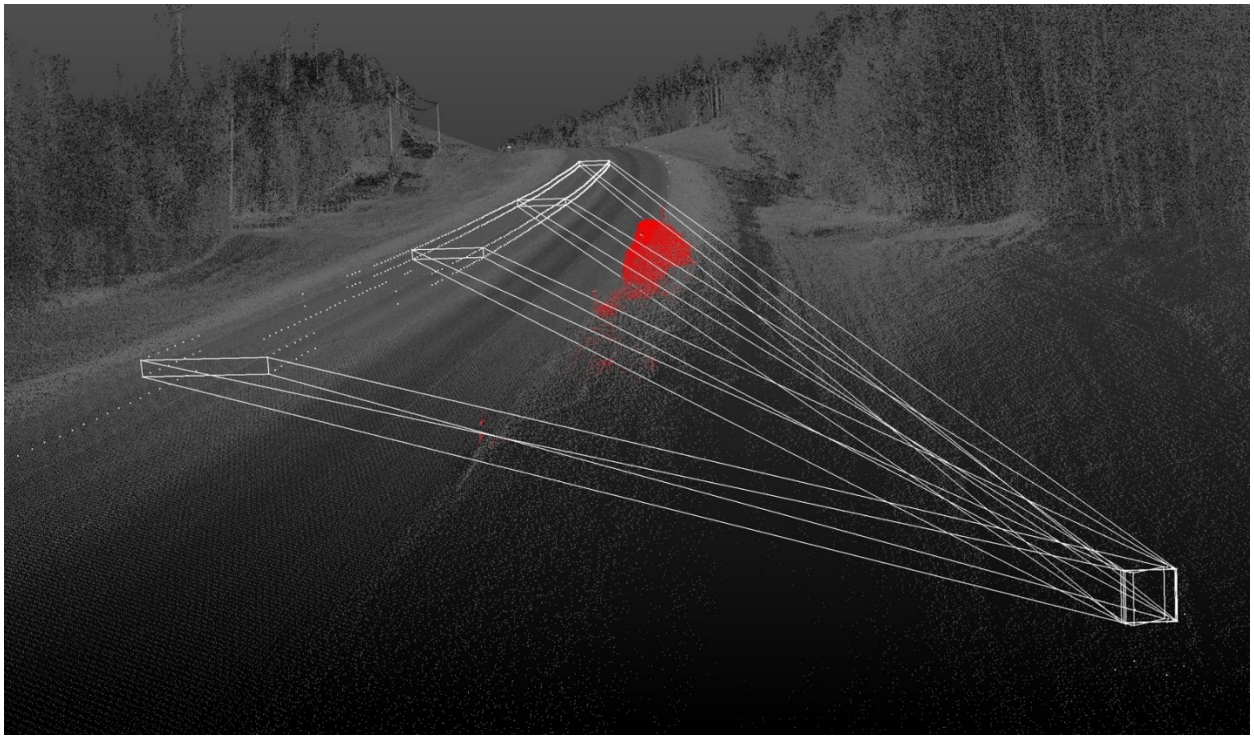


Figure 22 Set of obstructed view convex polyhedrons from observer plane

3.1.11 The Raycast Method

This raycast method is compared to the view convex polyhedron method outlined in this thesis for assessing the sight distance of an entire road environment. This raycasting method is developed by modifying the state-of-the-art version in (Gouda, Chowdhury, et al., 2021). Details of the raycasting methods are not covered to economize on space. In Figure 23, a standard voxel size shows how, for each observer, an analysis is done for obstructions on the sightline to a voxel.

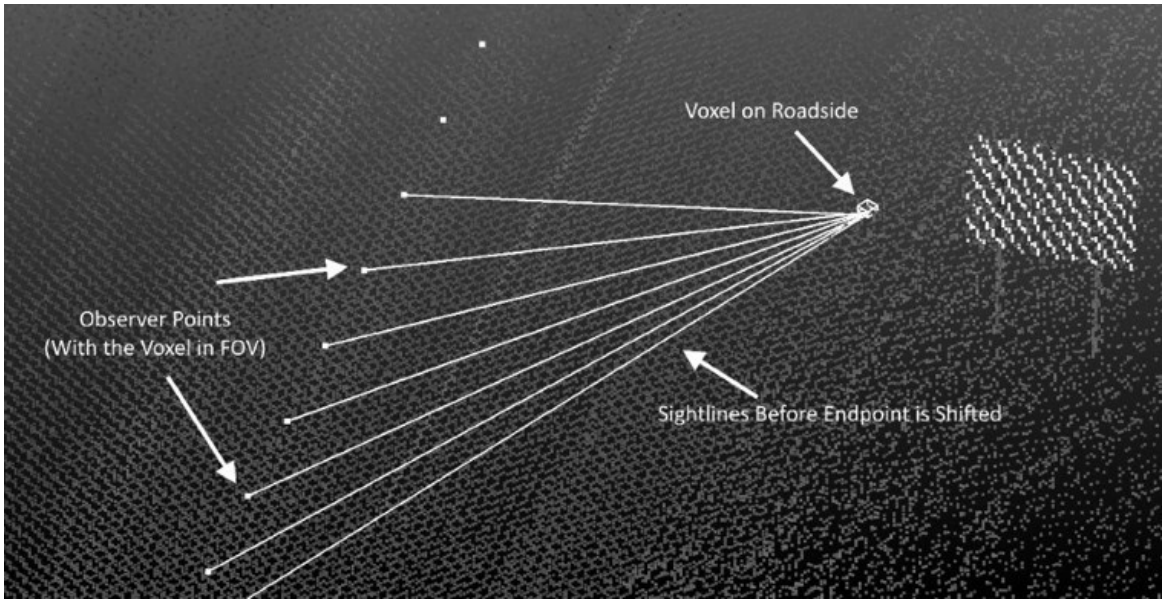


Figure 23 Observer sightlines to a voxel

3.1.12 Comparison with the Raycast Method

This section discusses the advantages of the view convex polyhedron method described in section 3. For the purpose of a fair comparison, only the observer points were used for evaluating sight distance as the observer plane analysis is not possible/infeasible in the raycast method, requiring infinite rays. This is because the raycast method does not check for obstructions using a volume-related process like in the convex-hull method. To analyze sight distance with an observer plane using the raycast method would require rays to be cast from every possible location within the plane.

3.1.12.1 Processing Times

As shown in Table *1*, the processing time per voxel is dependent on different factors for each method. It can be noticed that the increase in processing time for the raycast method to Run 4 is greater than in the convex hull method (~16.7x vs ~14.6x, respectively). Of all the runs, Run 4 was performed on a full segment at the proper voxel grid size and observer spacing. Therefore, other than specific or extreme cases, the convex hull method usually outperforms the raycast method regarding processing time due to its independence from the voxel grid size. The machine used for the testing incorporated two Nvidia RTX 2080Tis, an Intel Core i9-9960X, and 128 GB ram.

Table 1 Processing times, convex hull (observer point) versus raycast

Parameter	Run 1	Run 2	Run 3	Run 4	Run 5
Voxels ($\boldsymbol{v}$)	2,398,673	2,398,673	2,040,635	20,078,505	20,078,505
Voxels Grid Size ($\propto \boldsymbol{t}$)	0.1 m	0.1 m	1.0 m	0.1 m	0.1 m
Observer Spacing ($\propto \boldsymbol{b}$)	1 m	2 m	1 m	1 m	2 m
Points in Cloud	4,071,266	4,071,266	34,584,002	34,584,002	34,584,002
Distance of Road Segment (m) :	467	467	4,004	4,004	4,004
X-Sectional Avg. Dens. (points/m) : [Points in Cloud / Distance of Seg.]	8.7 e+03	8.7 e+03	8.6 e+03	8.6 e+03	8.6 e+03
Raycast Processing Time (s) :	4,623.91	2,609.58	334.72	77,307.93	21,852.60
Convex Hull Processing Time (s) :	2,802.94	2,064.17	2,621.64	40,966.42	24,872.37

3.1.12.2 Results

The comparison of the convex hull method with raycast was analyzed both qualitatively and quantitatively. The qualitative analysis involved side by side comparison of sight distance (and grade) heatmaps to identify differences and similarities.

3.1.12.2.1 YCbCr for Comparisons

YCbCr is a family of colour spaces that can be used for scoring. It is used in this method to show colour scores in three dimensions, i.e., mapping up to three variables to a colouring scale without loss of data or redundancy. Figure 24 shows an example of a YCbCr Heatmap Key. Again, to economize on space, the mathematical derivations and basics of YCbCr scoring can be found in (International Telecommunication Union, 2011). In this work, the voxel grades and scores are normalized and mapped non-linearly to the colour space by the following equations:

$$Y = Y_{min} + (Y_{max} - Y_{min}) * i \quad (\mathbf{10})$$

$$C_R = C_{Rmin} + (C_{Rmax} - C_{Rmin}) * \left(\frac{1}{1 + e^{\frac{h_s - s}{p}}} \right) \quad (\mathbf{11})$$

$$C_B = C_{Bmin} + (C_{Bmax} - C_{Bmin}) * \left(1 - e^{\frac{log(0.5)}{h_g} * g} \right) \quad (\mathbf{12})$$

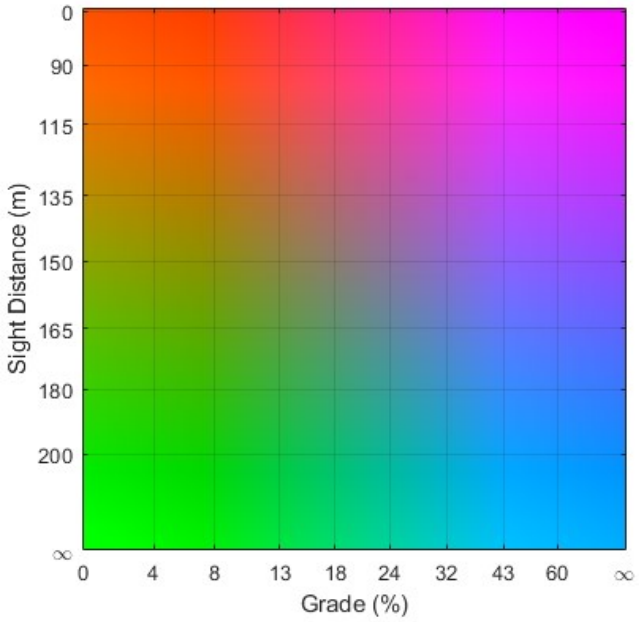


Figure 24 Heatmap key (154±30m / 20% grade)

3.1.12.2.2 Qualitative Comparison

Figure 25, shows some important discrepancies between the two methods but misses some that are more apparent in Figure 26. The scaling of these heatmaps hides some of the discrepancies, as most of the colour changes occur around a sight distance of 154 m. So, differences in sight distance that centre around a sight distance well above or below 154 m are difficult to distinguish as their colour difference are minimized. This happens on the trees, especially where the continuous sight distance for each method is usually relatively low compared to the scale but may still be quite different.

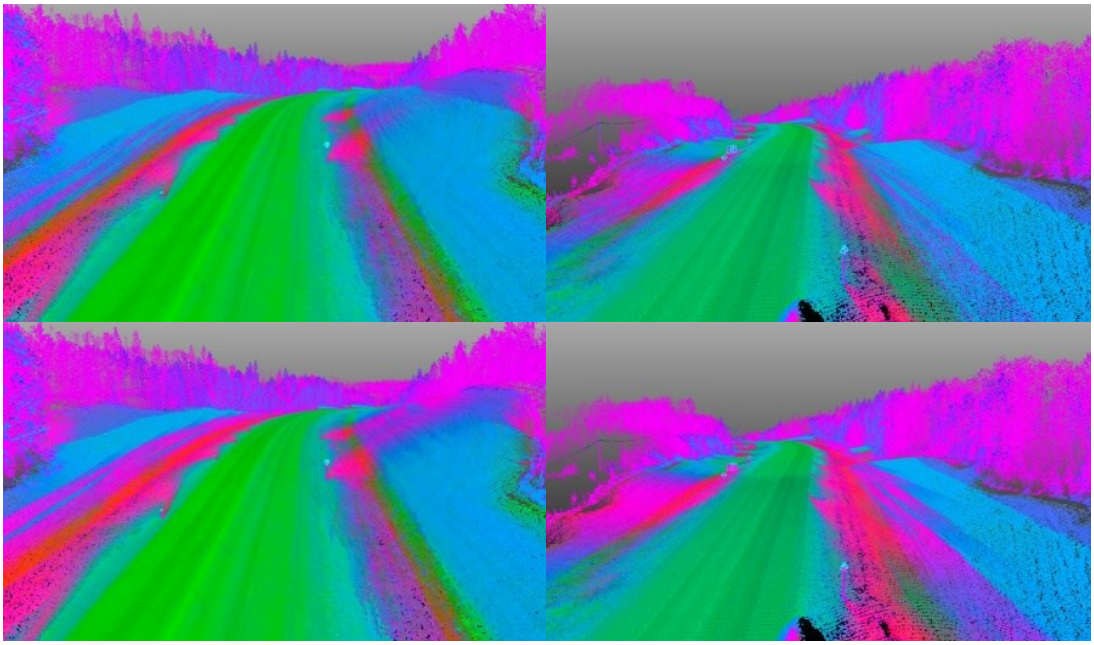


Figure 25 Convex hull method (top) versus raycast (bottom) continuous sight distance heatmap (0.1 m voxel size)

3.1.12.2.3 Quantitative Comparison

For the quantitative analysis, more heatmaps were generated over the voxelated point cloud to highlight the differences (in red) and similarities (in white) in sight distance at certain locations.

Figure 26 Raycast versus convex hull sight distance difference heatmap (0.1 m voxel size). Scale range [white]: <10 m dif. to [red]: >40 m dif.

As seen in Figure 26, sight distances relating to reasonably flat surfaces on and around the road surface are quite similar. However, there are two classes of surfaces that produce quite different sight distances in each method that would not as easily be noticed in the side-by-side comparison:

surfaces partially shielded from the FOV, such as ditches, and steep grade surfaces, such as trees. These are common surfaces near highways, and therefore, the more accurate method should be considered.

3.1.12.2.4 Accuracy

The convex hull method used in this thesis consistently produces longer sight distances than the raycast method. The cause of this discrepancy was investigated to determine which method is subjected to fewer inaccuracies.

3.1.12.2.5 Obstructions

The first difference between the two methods that must be noted is how they each identify obstructions. The raycast method performs a voxel traversal along the vector from the observer to the voxel target. The voxel target is just a point identifying the centre of a voxel position that is usually translated above the original voxel.

If the voxel traversal along the observer-target vector intersects a voxel from the voxelated point cloud, that voxel will be returned as an obstruction. This means that the accuracy of the raycast method is dependent on voxel size, as it is directly proportional to the volume of the traversal object.

From visual inspection, there are significantly more raycasts obstructed in Figure 27 than convex polyhedrons shown in Figure 21.

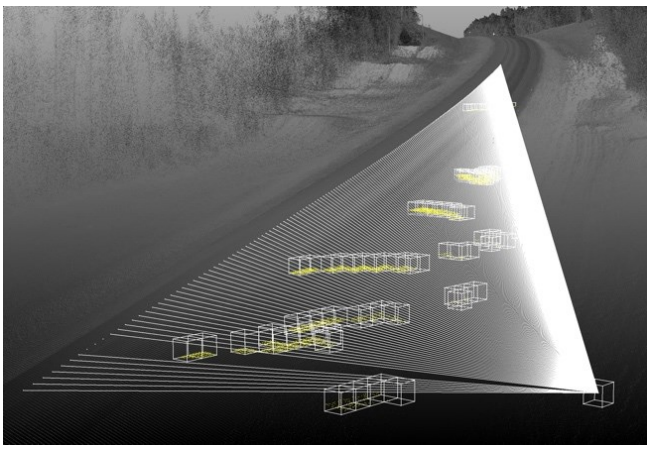


Figure 27 Set of obstructed raycasts

In the convex hull method, obstructions are specific points returned from the octree query of the volume bounded by each convex polyhedron. Consequently, only the convex polyhedron volume is related to the voxel size regarding the size of the target plane. Obstructions, on the other hand, are independent of the voxelated point cloud. Therefore, the more empty space a voxel contains, the greater the difference between these methods. Since the empty space inside a voxel also increases with voxel size, the discrepancy between them is proportional to voxel size.

3.1.12.2.6 Raycast Clipping

Analysis and verification at voxels with large differences in sight distance between the two methods, such as the voxel in Figure 21 and Figure 27, provide evidence to show the inaccuracy of the raycast method. 'Clipping' occurs when a raycast passes through the empty space of a nonempty voxel (Figure 28). In contrast, the convex polyhedron does not intersect any point in the point cloud and would not be declared obstructed (Figure 29).

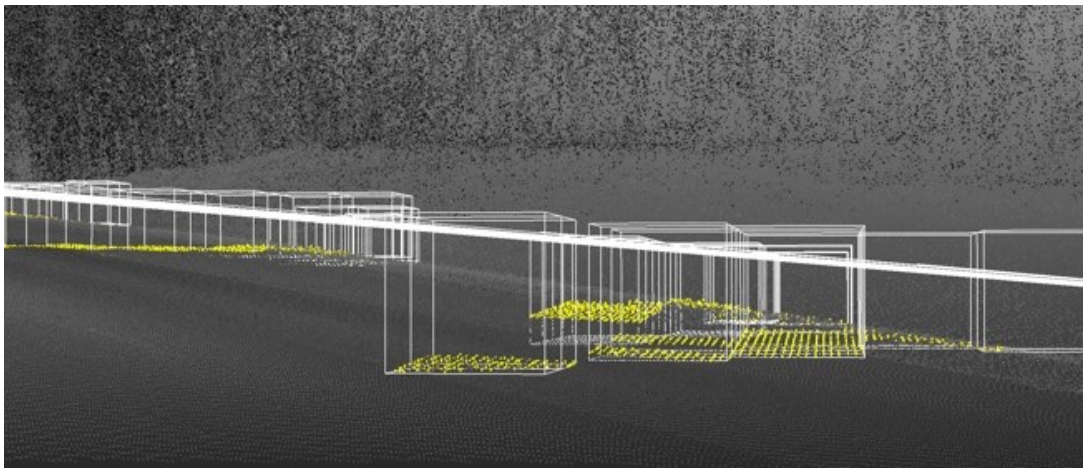


Figure 28 Illustration of raycast clipping

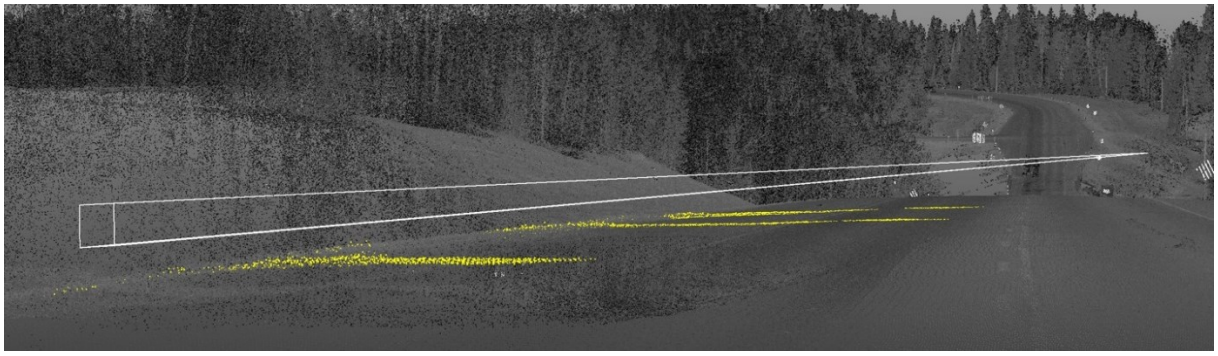


Figure 29 View convex polyhedron over clipping point

3.1.13 Data Collection

Alberta Transportation collected LiDAR point cloud data from rural highways in Alberta, Canada, in 2020. The data was collected using a multifunctional surface profiling vehicle, Tetra Tech PSP-7000, equipped with a REIGL VMX-450 system. Surveys were performed at speeds of up to 100 km/h in normal traffic flow. The collected data was segmented into separate 4 km section LAS files. At 90 km/h, the point density ranges from 150 to 1000 points/m^2, saved in files of approximately 500 MB each.

Four km, 8 km, and 10 km sections from Alberta Highways 40, 32, and 11, respectively, were further split into 4 km segments and considered in this section.

3.1.14 Case Studies

The method outlined has numerous applications that help identify locations requiring further safety assessment on the road and roadside.

3.1.14.1 Clear Zones & Roadside Safety

Locations of high grade, low visibility, or combinations of such can be identified using the method proposed in this thesis. From this, clear zone and roadside safety assessments can be made. Some of these locations would be sharp slopes, fixed objects, etc.

Side slopes steeper than a 3:1 width-to-height ratio, or 33% grade, are considered a hazard due to the increased possibility of vehicle rollover. Safety precautions are therefore required. If the hazardous slope is less than 14 m in height, either the side slope grade can be reduced to an

adequate level or barrier protection is required. However, if the hazardous slope is greater than 14 m in height, barrier protection is suggested to be installed regardless of the grade. With these requirements, the clear zone must also be free of obstacles, such as trees over 100 mm in diameter, utility poles/supports, non-breakaway signs or other pole-like structures, water hazards, boulders taller than 100 mm, etc. The method presented in this work can help users identify areas of interest that either meet these specifications or require further safety infrastructure, as shown in the following figures.

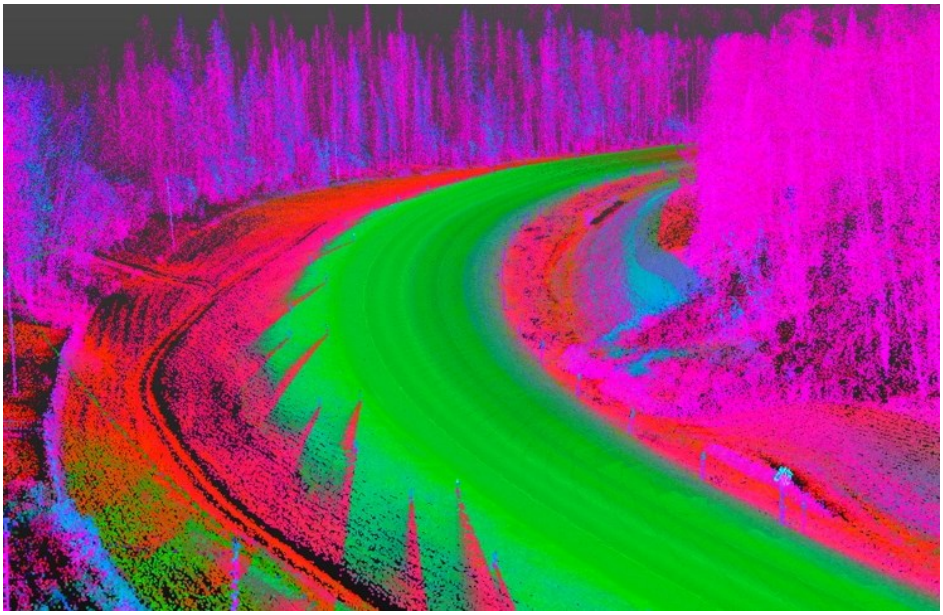


Figure 30 Steep, low visibility roadside with reflectors

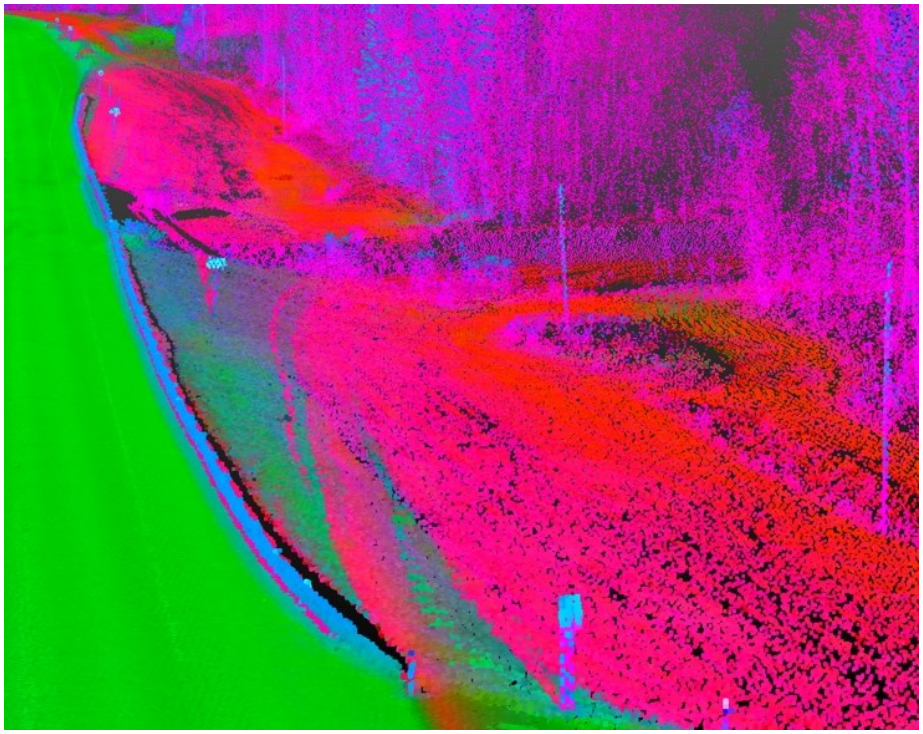


Figure 31 Steep, low-visibility roadside

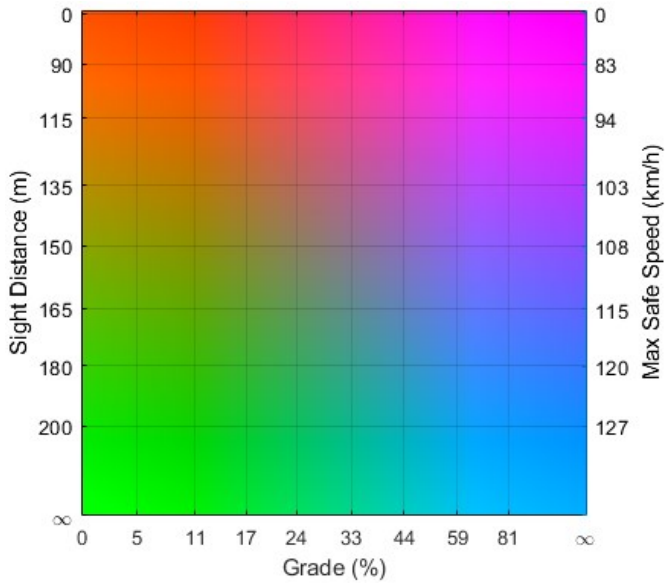


Figure 32 Heatmap key (154±30m / 30% grade)

Figure 30 and Figure 31 depict two roadside sections detected as low visibility and/or high grade. Locations of poor visibility within the vehicle sensors on the road and roadside can be directly identified. These locations may require a decrease in the speed limit or the implementation of other safety protocols or infrastructure elements (e.g., wildlife fencing, animal detection sensors, removal of obstructions, etc.). On the other hand, the opposite conclusion may also be drawn,

where some sections may be evaluated for higher ASD to increase the maximum and recommended speed limits for CAVs.

According to AASHTO, human drivers are expected to react in 2.5 s under most road conditions, whereas recent works suggest that AVs can reach a reaction time as low as 0.5 s, which is considered in this analysis. Therefore, the following AASHTO equation was used to calculate stopping sight distance (SSD) or maximum speeds as suggested in recent studies (AASHTO, 2018):

$$0.278Vt + \frac{V^2}{254\left[\frac{a}{9.81} \pm G\right]} = d \qquad (\mathbf{13})$$

$\boldsymbol{V}$ refers to the speed of the vehicle (km/h). $\boldsymbol{t}$ refers to the perception reaction time, i.e., 2.5 s for a human driver and 0.5 s for an AV. $\boldsymbol{a}$ refers to the deceleration rate of the vehicle (m/s^2). The deceleration rate of 3.4 m/s^2 recommended by AASHTO was used to ensure the safety and comfort of human passengers. This deceleration rate is considered a physical constraint of the vehicle but is still based on characteristics of human driver performance. Lastly, $\boldsymbol{G}$ refers to the grade of the roadway as a decimal. Equation 13 can turn the heatmaps to maximum safe speeds, as shown in Figure 32.

Figure 33, Figure *34*, and Figure *35* identify a horizontal curve around a mountainside, which causes a reduced sight distance to a portion of the road and the roadside, again given by the key in Figure 32.

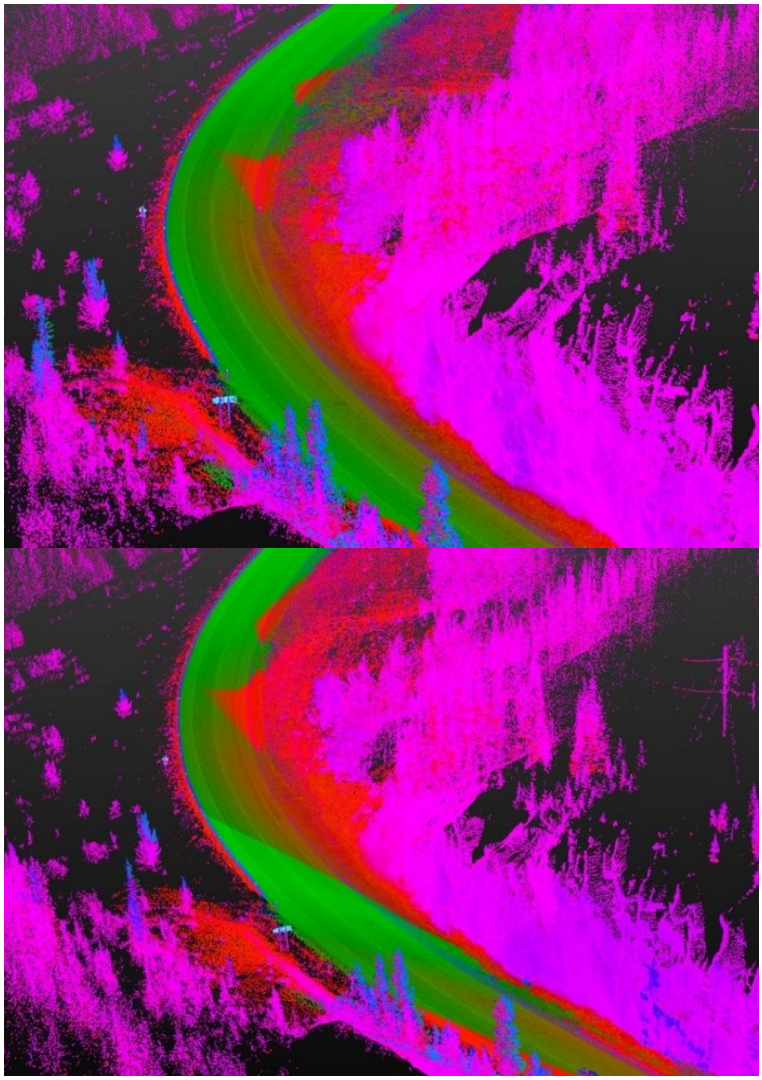


Figure 33 Obstructed due to mountain side (heatmaps – top observer point, bottom observer plane)

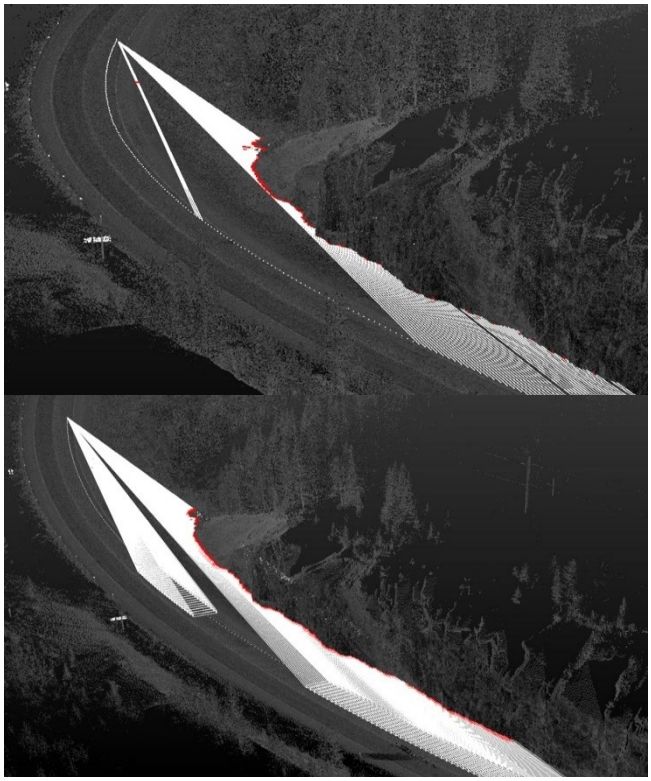


Figure 34 Obstruction due to mountain side (convex polyhedrons angle 1 – top observer points, bottom observer planes)

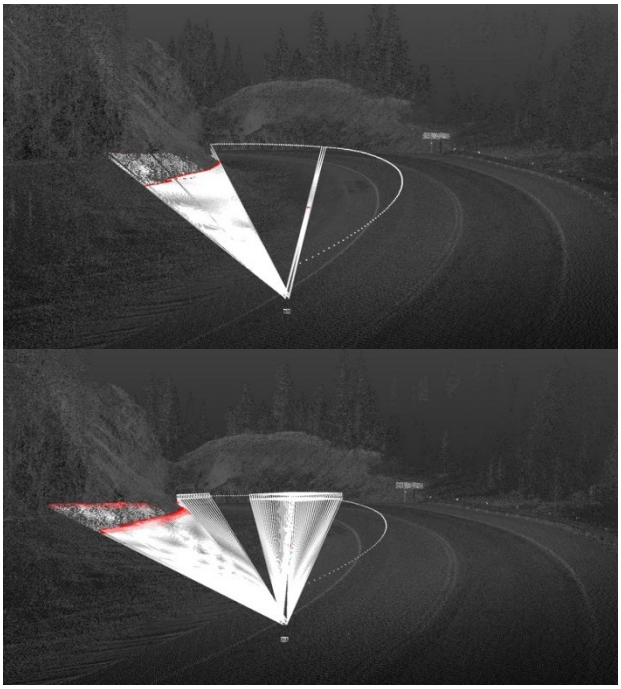


Figure 35 Obstruction due to mountain side (convex polyhedrons angle 2 – top observer points, bottom observer planes)

As this location is in an area where wildlife animal activity is expected, the purple area, with high grade and low visibility, and red areas would require safety countermeasures, such as wildlife fencing, animal detection sensors or reducing speed limit (smart warning sign). This is based on the ASD to the roadside, not the travel lane.

3.1.14.2 Wildlife Sight Distance & Target Heights

Following on, this section evaluates sight distance at multiple target heights above the ground to identify animal visibility. Sight distance assessments were performed at several target heights, starting with a base of 0.38 m and following at heights of 0.53 m, 0.70 m, 1.3 m, and 1.4 m. These heights correspond to various animals found commonly in Alberta, Canada (deer, bear, elk, and moose). The following figures demonstrate the output along a highway segment at different animal heights.

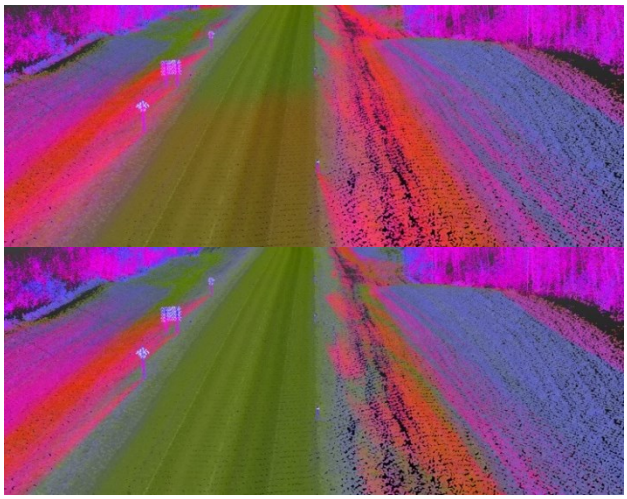


Figure 36 Comparison of target height 0.38 m (top) vs 1.40 m (bottom)

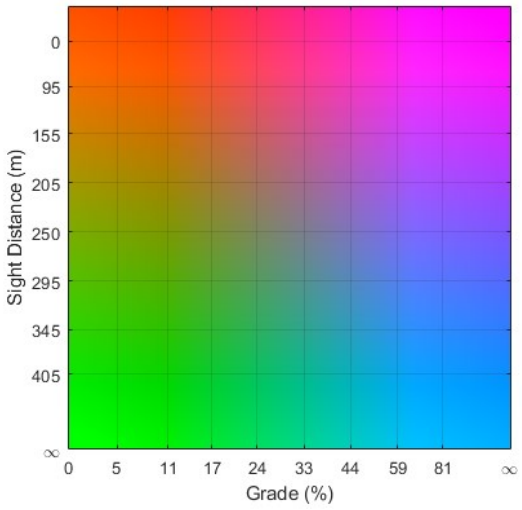


Figure 37 Heatmap Key (246±100m / 30% Grade)

The target heights compared in Figure 36 are coloured according to the key shown in Figure 37. Appropriate countermeasures may be applied depending on the expected species in a region.

3.1.14.3 Pavement Surface Assessment

In addition, the proposed method can be used to perform analytics specific to the pavement surface to provide data-driven insights for maintenance operations. For example, Figure 38 and Figure 39 show two sections along the same segment with limited visibility and high grades. A recent USDOT FHWA collaborative initiative between IOOs and AV industry professionals concluded that removing occlusions to lane markings is a recognized measure to improve highways for CAV readiness and compatibility by IOOs. Hence, the ultimate benefit of this pavement surface analysis is studying occlusion to lane markings.

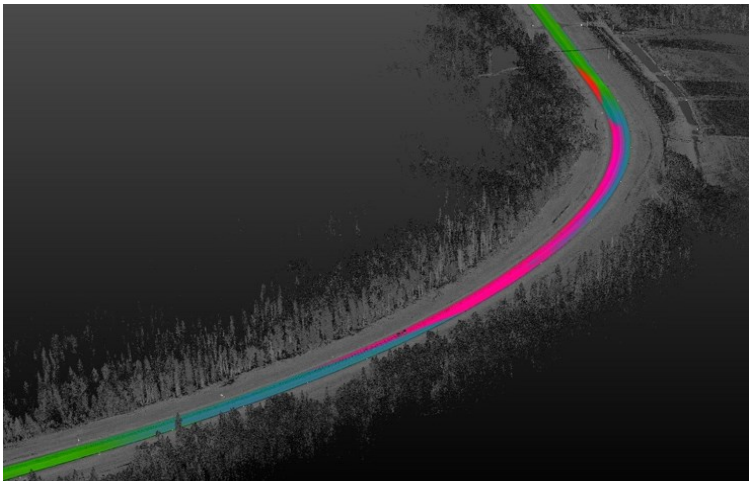


Figure 38 Pavement assessment on a curve

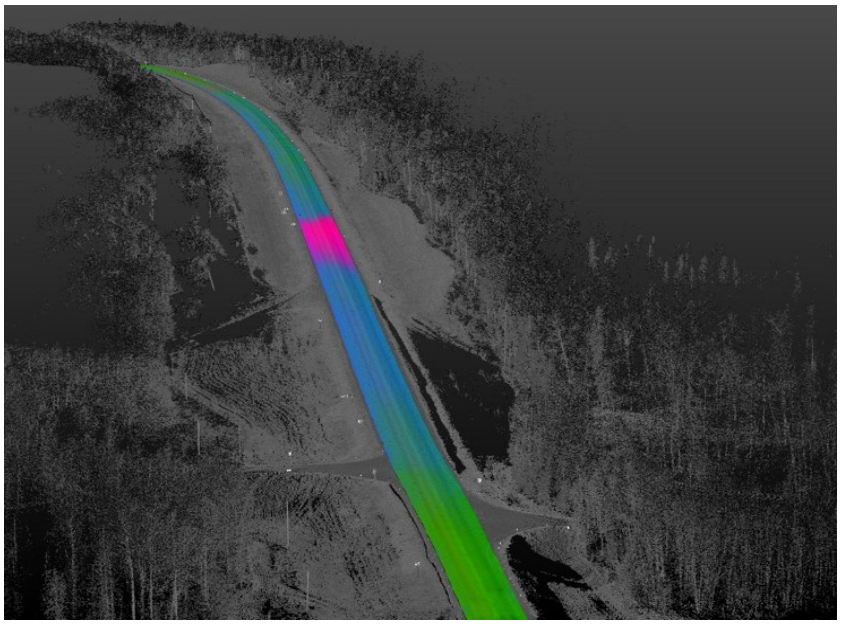


Figure 39 Pavement assessment on a slope

3.1.15 Conclusions

This section proposed a holistic simulation-based method for mapping existing road and roadside design information based on AV sensor set design and autopilot parameters using ultra-dense point clouds. As demonstrated here, the mapped data can then be used to assess road and roadside design for AV deployment and identify substandard locations for AVs. The proposed method simulates virtual AV sensors' field of vision, originating from different points in the travel lane and phantom targets on the voxelated road and roadside environment. Convex hulls and octrees are then used to score voxels based on the visibility of every phantom target in view of a convex polyhedron. Following that, a quadratic surface model of the road and the roadside surface was utilized to generate normal vectors. Then, a series of arithmetic operations were applied to map road/roadside gradients. Next, a semantic segmentation variant of PointNet++ is applied to classify points into road and roadside points. Finally, the different dimensions (visibility score, gradient, intensity, and classification) were used to generate maps based on a unique YCbCr colour space value. A different state-of-the-art raycasting approach used in lane-level sight distance research and also suggested in state-of-the-art AV urban occlusion handling research was developed and improved to perform a general road/roadside analysis. This approach was compared to the proposed method.

Three highway segments in Alberta, Canada, with a total length of 22 km, were used for testing. In general, It was found that the proposed method outperforms the raycasting method in

terms of performance and processing time at small voxel grid sizes and observer spacing. Besides, the proposed approach has other advantages, such as 1) the ability to account for different sensor origins in the travel lane, 2) avoiding false clipping in voxel-based raycasting methods, 3) the possibility to generalize to different shapes of sensor sets and targets required by a user, and 4) accurately detects occlusions regardless of the voxel size used.

4 TRAFFIC SIGNS

4.1 SEMANTIC SEGMENTATION

Following state-of-the-art research and practice, this work uses advanced professional-grade mapping scanners to digitize the road infrastructure. The collected scans are 3D points with several features related to the returned laser pulse to the scanner. The scans are not ready for direct use or assessment. The automated semantic segmentation step is necessary to turn the data into subsets for further processing and scene understanding. These subsets locate every point in the point cloud into its relevant class (e.g., pavement, lane marking, road signs, roadside, etc.).

Several methods were utilized in this research to perform semantic segmentation on the collected data. Some of these methods are heuristic models, and some are deep-learning-based. The method proposed in this section was developed using traffic sign data due to the extensive work needed to build rural training datasets manually.

4.1.1 Background

Traffic signs provide meaningful information about the conditions and limitations of the road. Current practices in road maintenance and asset management involve a tremendous amount of manual surveying work by transportation agencies to build traffic sign inventories (TSIs) (CoE, 2023; S. A. Gargoum & El Basyouny, 2019). For instance, the City of Edmonton (CoE) maintains about 130,000 traffic signs, performs regular surveys, and responds to more than 2,000 sign maintenance jobs annually (CoE, 2023). TSIs compile and maintain information about traffic sign location, placement, dimensions, orientation, visibility, and classification (Karsten, 2019). TSIs are essential for a successful, reliable, and cost-effective Transportation Asset Management (TAM) plan and to ensure that traffic signs regularly meet standard traffic sign placement guidelines and the needs of drivers (Karsten, 2019). Given the size of current global transportation networks, traditional manual surveying of signs is time-consuming, labour-intensive, inefficient, suffers from low accuracy, causes traffic disruptions, and is economically infeasible (S. A. Gargoum & El Basyouny, 2019; Gouda, Epp, et al., 2022; Karsten, 2019). To address this, government agencies have started to digitize their infrastructure network while developing alternative sign-detection methods (Gudigar et al., 2016).

Through their digitization efforts, governments have collected data on road infrastructure through images and videos. Machine learning techniques such as support vector machines, random forests, neural networks, convolutional neural networks (CNN), and other classification algorithms have been applied to image and video data and achieved satisfactory detection rates (A. Adam & Ioannidis, 2014; Kuo & Lin, 2007; Lee & Kim, 2018; Pei et al., 2013; Y. Yuan et al., 2019; Zaklouta & Stanciulescu, 2012; Zhu et al., 2016). Image and video-based methods, however, suffer greatly from variations in environmental conditions, ambient light levels, sign orientation, and cleanliness (Howe, 2006). An alternative is LiDAR, a technology that had increased interest in the past few years (Habib et al., 2019, 2020; L. Ma et al., 2018; Williams et al., 2013). LiDAR is an optical sensor that transmits laser beams toward surrounding targets and records features related to the surveyed points. The return strength of a reflected laser pulse at a point is known as "intensity," which depends on the reflectivity of the surveyed object (ArcMap, 2019). LiDAR utilizes laser scanning equipment, global positioning systems (GPS), and navigation technologies to obtain intensity and positional information of surrounding features. The output of the LiDAR scanning process is a rich 3D point cloud of the surveyed objects that can be used in the automated extraction of road and roadside features (Agina et al., 2021; S. A. Gargoum, El Basyouny, et al., 2018; Gouda et al., 2020, 2023; Kilani et al., 2021).

Several researchers have studied how such data can be used to automate the detection of traffic signs (Ai & Tsai, 2014; X. Chen et al., 2009; Y. Z. Chen et al., 2007; S. Gargoum et al., 2017; S. A. Gargoum & El Basyouny, 2019; S. Gargoum & El-Basyouny, 2017), and it was found to be a potentially cost-effective alternative compared to conventional approaches, with reductions in the processing time of up to 76% (Ai & Tsai, 2014). LiDAR data, with millimeter-level accuracy and widespread data collection capabilities, represents an alternative to collecting traffic sign information required for building TSIs (Karsten, 2019). Emerging technologies like LiDAR have recently shifted research trends to applying deep learning on 3D point clouds (Qi, Su, et al., 2017; Qi, Yi, et al., 2017). New neural network architectures have been developed to operate on 3D data to perform tasks such as shape reconstruction and point cloud segmentation (Qi, Su, et al., 2017; Qi, Yi, et al., 2017; Yang et al., 2018; W. Yuan et al., 2018). However, point clouds are an unordered set of data points in 3D space, constraining the neural network's architectural design. Most existing methods typically transform 3D point cloud data to a regular 3D voxel grid, such as SegCloud (Tchapmi et al., 2017) and VoxelNET (Y. Zhou & Tuzel, 2018), or to collections of

images (multi-view representations) (Su et al., 2015), and provide these as input to their deep neural networks (Briechle et al., 2019; Griffiths & Boehm, 2019). This additional preprocessing step yields unnecessarily expanded data and affects existing features associated with the point data (Qi, Su, et al., 2017).

Since 2017, PointNet++ (Qi, Yi, et al., 2017) has been considered the state-of-the-art deep hierarchical neural network architecture due to its accuracy when analyzing standard datasets in small scenes (Griffiths & Boehm, 2019; Qi, Yi, et al., 2017). PointNet++ (Qi, Yi, et al., 2017) operates directly on unordered 3D points without converting the data into other shape representations, such as 3D voxels or multi-view data. However, it has been difficult to apply the same framework for semantic segmentation on large scenes because of the increased computational requirements, which will typically not fit into the memory of an economic Graphical Processing Unit (GPU), and the significant reduction in segmentation accuracy if low-density scans are used (Briechle et al., 2019; Qi, Yi, et al., 2017). As such, (Qi, Yi, et al., 2017) recommended further research exploring how to apply PointNet++ to larger metric scenes while maintaining its accuracy. In addition, a review by (L. Ma et al., 2018) recommended further research on deep learning applications for extracting road features from point cloud data, which poses challenging scene sizes for such algorithms. (Griffiths & Boehm, 2019) reviewed deep learning techniques for point cloud classification and segmentation, and it was concluded that recent advances in the PointNet++ architecture demonstrated a benchmark performance on standard datasets. Further, it was found that the direct processing of unordered point clouds, established by PointNet and PointNet++, is potentially a very effective solution for point cloud segmentation for future research (Griffiths & Boehm, 2019).

In this thesis, the semantic segmentation variant of PointNet++ (Qi, Yi, et al., 2017) is adopted, with several augmentations to its original implementation. This work is an attempt to apply recently developed and highly accurate point-based direct processing deep learning on large-scale scenes for traffic sign extraction from LiDAR data. Further, a practical methodology for extracting and preparing training datasets is proposed. Finally, a series of validation tests were run to investigate the accuracy of this new approach. Due to the flexibility of deep learning, the proposed approach would avoid limitations in previous research, such as point cloud transformation to volumetric representations or multi-view data, data filtering, and manual thresholding (e.g., intensity thresholding). As such, it would improve the detection accuracy of

sign points and processing time. The performance of the proposed method is compared against other similar methods on an existing benchmark dataset. The method is also compared against SqueezeSegV2 (B. Wu et al., 2019), a state-of-the-art convolutional neural network (CNN) capable of accurately segmenting rasterized 3D image-like tensors representing the visible point cloud from the perspective of a vehicle.

4.1.2 Previous work

Several studies proposed different methods for detecting traffic signs using LiDAR data. (Y. Z. Chen et al., 2007) was one of the earliest studies to propose a vehicle-borne system consisting of LiDAR, cameras, and a navigation unit. Collected LiDAR data was filtered to remove the surroundings (vegetation, buildings, etc.) objects. The remaining points were efficiently clustered according to point distance and projected onto the horizontal plane, and then traffic sign candidates were obtained based on their strong linear features. Laser points were projected onto images to verify the candidate sign points. The study did not discuss success rates.

(X. Chen et al., 2009) filtered point clouds based on the distance to sensors, sensor angle, and intensity. Next, points were clustered and projected in a 3-D grid structure to retain high-intensity clusters. Finally, RANSAC and other geometric filters were applied to remove outliers. Tested on a 7-miles highway segment, the method achieved a 98.0% detection rate.

(Riveiro et al., 2015) applied an intensity filter to detect potential sign points, followed by a density-based algorithm to segment the point clouds. Clusters were further filtered by the number of points, dimensional constraints, and Principal Component Analysis (PCA) to identify sign points. The algorithm yielded variable success rates based on the type of road and sign.

(Arcos-García et al., 2017; Soilán et al., 2016) focused on detecting traffic signs in 3D point clouds in urban and highway environments. Ground points were removed from the LiDAR data, and then an intensity filtering technique was applied to the remaining points. DBSCAN-based clustering and PCA were used to isolate each traffic sign. Tested on an urban road and a highway in Spain, the method achieved precisions of 86.1% and 92.8%, respectively.

(Wen et al., 2016) employed a 'terrain filter' (i.e., repeated local horizontal fuzzy plane fitting) to locate off-ground points that were then clustered using Euclidean distance clustering. Linearly structured objects were separated by geometric filtering, and traffic signs were detected by an intensity filter. Tested on three roads in Xiamen, China, the method achieved an F1 score of

92.7%, which was attributed to incomplete point clouds and non-reflective side signs. (P. Huang et al., 2017) adopted a similar approach and achieved an F1-score of 94.8%. The authors attributed false positives to strong reflective billboards and a few oversized traffic poles.

(Guan et al., 2018), with prior knowledge of pole height and road width, filtered out the majority of ground and non-ground points. Then, the traffic sign interest regions were extracted from the non-ground points according to intensity information and geometrical structures. Finally, a refinement procedure was used to further remove non-traffic-sign clusters and noise. The authors reported a detection accuracy of 86.8%.

More recently, (Javanmardi et al., 2018) proposed a method that detected traffic signs and light poles from LiDAR data. A surface reconstruction algorithm was used to extract the orientation of the points, and *k*-means clustering was applied to extract road points. A sliding cuboid was designed to search for objects with a high elevation, and RANSAC was employed to remove non-planar false candidate points. In the post-processing step, a modified growing region was used to remove the outlier points around the candidates. Predefined thresholds based on sign and pole dimensions were used to remove false object candidates. The method achieved success rates of 94.5% in traffic sign detection.

(Y. Yu et al., 2016) developed a traffic sign detector using a bag-of-visual-phrases (BoVPs) representation of the LiDAR point cloud. They first supervoxelized the MLS data to construct feature regions and built a visual phrase dictionary. Semantic objects were segmented, supervoxelized, featured, and quantized to form BoVPs representations, detecting signs according to the similarity between the BoVPs and the semantic objects. The average recall, precision, and F-score were 0.95, 0.94, and 0.95, respectively, but their approach increased computational complexity.

In summary, most methods for sign detection using LiDAR data are based on intensity filtering and density-based clustering (Ai & Tsai, 2012; Arcos-García et al., 2017; X. Chen et al., 2009; S. Gargoum et al., 2017; Ghallabi et al., 2019; Guan et al., 2018; P. Huang et al., 2017; Riveiro et al., 2015; Soilán et al., 2016; Vu et al., 2013; Wen et al., 2016). Other studies used plane fitting methods based on PCA (Arcos-García et al., 2017; Soilán et al., 2016; Vu et al., 2013) or RANSAC (X. Chen et al., 2009; Ghallabi et al., 2019; Javanmardi et al., 2018). Real-time supervised learning (L. Zhou & Deng, 2014), eigenvalue analysis, and graph-analysis (Bremer et al., 2013) were also used. Some methods involve the voxelization or supervoxelization of the

LiDAR data (Guan et al., 2018; P. Huang et al., 2017; Y. Yu et al., 2016). Comparisons are made between the proposed method and 13 existing methods, including deep learning methods, via evaluation of the Paris-Lille-3D point cloud classification benchmark (Roynard et al., 2018). This benchmark is a "large and high-quality ground-truth urban point cloud dataset for automatic segmentation and classification." The LiDAR scanner used to acquire this dataset was a Velodyne HDL-32E. Each part of the dataset is stored in a separate PLY-file. Each point has ten attributes, including x, y, z, reflectance (intensity), label and class. The road and roadside objects were segmented manually. While a rural benchmark would be preferred, Paris-Lille-3D was selected due to a lack of existing rural point cloud classification benchmarks. An additional comparison is made with SqueezeSegV2, which uses a CNN for semantic segmentation of road objects in rasterized 3D image-like tensors.

4.1.3 Data collection and description

LiDAR data collection was performed by Alberta Transportation between 2013 and 2015 using Tetra Tech PSP-7000, a proprietary multifunction pavement surface profiling vehicle. The vehicle is equipped with a REIGL VMX 450 system to collect 360° LiDAR point clouds on rural highways in the province of Alberta, Canada. Surveys were conducted in normal traffic flow at posted speed limits of up to 100 km/h. Collected data for a given highway were saved in 4 km segments in multiple LAS files: each is about 500 MB of data. Surveys performed at 90 km/h generated point densities in the range of 150 to 1,000 points/m^2. In this thesis, data were collected on ten highway segments, with a total length of approximately 40 road-km. Seven of these road segments were used to develop the training dataset, and three were used as a validation dataset. Figure 40 shows data collected on Highway 1A in Alberta.

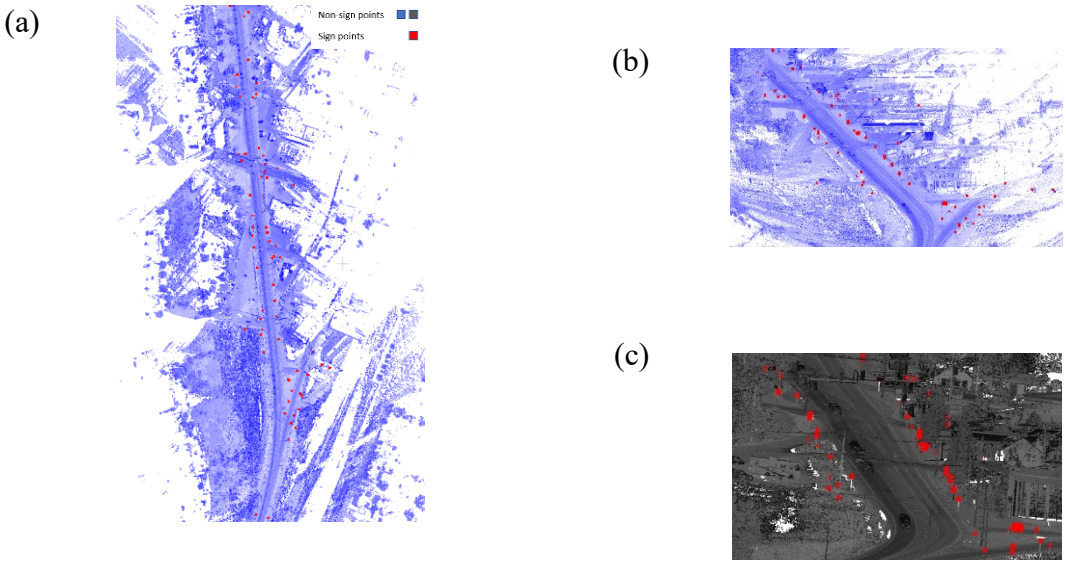


Figure 40 Labelled data on a 450 m section of Highway 1A-4

Five segments were collected on Highway 01A, located northwest of the City of Calgary (CoC), with a speed limit of 90 km/h. The LAS point files for those segments consisted of 160.2 million points. Two segments on Highway 28A extended for a length of 8.0 km with a speed limit of 100 km/h. The segments belong to a two-lane road located northeast of the CoE. Located north of the CoC, a segment on Highway 02A had a right-angle turn. One segment was collected on Highway 01, a divided multilane located about 100 km east of the CoC. One segment on Highway 28 is located about 20 km north of the CoE.

4.1.4 Training dataset development

To develop a training dataset, several filters are usually applied to the original point cloud to extract sign data using several point features, such as roughness and *z*-gradient. It is worth noting that applying intensity filters for data labelling is not recommended since lower intensity signs and points within individual signs are usually lost, which negatively impacts the performance of the neural network. In the proposed method, several variables are used in a trial-and-error process until all sign points are extracted. The method is designed to minimize false negatives without undue regard for the large number of false positives it produces (as these may be easily removed manually). Below is a definition of the most used variables in the filtering process (CloudCompare, 2023):

- *Roughness*: is equal to the distance between a point and the best fitting plane to its neighbours.
- *Gradient*: gradient is the rate of change in any feature value at a point (e.g., elevation) compared to its neighbours.
- *z-gradient*: is the gradient of the elevation of a point relative to its neighbours.
- *Number of neighbour density (N):* counts the number of neighbour points to each point in the point cloud within a sphere of radius *R*.

Several other variables were explored (e.g., linearity, verticality, surface density, and volume density), but details are not mentioned to economize on space; further details can be found in (CloudCompare, 2023; Hackel et al., 2016). After points are filtered using the above measures, the remaining points are clustered based on a minimum distance and the number of points within each cluster using Connected Component Analysis. This process is similar to the method commonly applied to image labelling (CloudCompare, 2023). CloudCompare, an open-source 3D point cloud editing and processing software, was used to visualize and label the point clouds (CloudCompare, 2023). All connected components were visually reviewed to verify their class. Finally, manual edits and revisions were performed on the labelled data to correct any misclassifications. Google Street View (Google Maps, 2023) was continuously used in all steps to validate the labelling output. Figure 40 (a) and (b) show examples of the labelled training data on a 450 m stretch on segment four of Highway 01A. All sign points are marked in red, and all road points are marked in light blue. Figure 40 (c) shows the road intensity points on the same segment. All points on the 40 km segments were labelled into two groups: sign points and non-sign points.

4.1.5 Methodology

4.1.5.1 PointNet and PointNet++

PointNet (Qi, Su, et al., 2017) is a neural network architecture capable of approximating a function f that operates on an unordered point set. Let $\{x_1, \dots, x_n\}$ be the point set, so that each $x_i \in \mathbb{R}^N$ is a point with N features. Then PointNet learns to approximate f by learning two other functions $h: \mathbb{R}^N \to \mathbb{R}^K$ and $\gamma: \mathbb{R}^K \to \mathbb{R}^L$ according to:

$$f(\{x_1, \dots, x_n\}) \cong \gamma\left(\max_{i=1,\dots,n}\{h(x_i)\}\right), \tag{14}$$

Where h is a multi-layer perceptron that operates on a per-point scale, converting the N input point features to K point features. γ is another multi-layer perceptron, this one operating on the global scale, converting the K world features obtained by the element-wise maximum of the per-point features into the L outputs of f.

PointNet++ is an extension to PointNet that processes a set of points sampled in a metric space in a hierarchical fashion. Compared to PointNet, PointNet++ has a better ability to abstract local patterns and generalize to unseen cases. PointNet++ also has robust performance on point clouds with varying sampling densities.

PointNet++ consists of three main parts:

4.1.5.1.1 Set Abstraction Layers

A set abstraction layer converts an input point set into a down-sampled point set, each point of which represents the centroid of a neighbourhood layer's input points. Each output point is given an output feature, a feature vector learned by a PointNet to summarize the neighbourhood.

Four set abstraction layers are used in this work's PointNet++ implementation. The first samples 1,024 neighbourhood centroids with a neighbourhood radius of 0.1. The next three use sample sizes and radii of 256 and 0.2, 64 and 0.4, and finally 0.8 and 16. The multi-layer perception h of the PointNet in the first set abstraction layer has three layers of 32, 32, and 64 units. The multi-layer perceptrons corresponding to the PointNets in the next set abstraction layers have 64, 64, and 128 units; 128, 128, and 256 units; and 256, 256, and 512 units. K is 32 for each layer.

4.1.5.1.2 Feature Propagation Layers

A feature propagation layer functions somewhat like an inverse set abstraction layer: it takes as input a point set of N_{l-1} neighbourhood features and converts these features into features corresponding to the N_l centroids of the next-smallest neighbourhoods (note $N_{l-1} \leq N_l$). The N_{l-1} features are initially assigned by averaging the nearest k input features weighted by inverse distance. Next, the newly generated feature vectors are concatenated with the output feature vectors from the set abstraction layer (from the first part of the network) that corresponds to the same neighbourhood scale. Finally, each of the concatenated feature vectors is run through the same small multi-layer perceptron (a single multi-layer perceptron is used per feature propagation layer).

Four feature propagation layers are used in the version of PointNet++ implemented in this thesis. The multi-layer perceptrons used in each have 256, 256 units; 256, 256 units; 256, 128 units; and 128, 128, 128 units.

4.1.5.1.3 Output Fully Connected Layers

The output of the feature propagation stack is a 128-element feature vector for each point. Each of these is passed through a fully connected layer with 128 units, a dropout layer with a keep probability of 0.5, and a final fully connected layer with two units.

The resulting two-element feature vector corresponding to each point is passed through a softmax function. That is, the predicted one-hot classification y_{pred} for each point is calculated from the corresponding output z from the final fully connected layer according to:

$$y_{pred,i} = \frac{e^{z_i}}{\sum_{i=1}^{2} e^{z_j}} \text{ for } i = 1,2, \tag{15}$$

The network architecture is visualized in Figure 41.

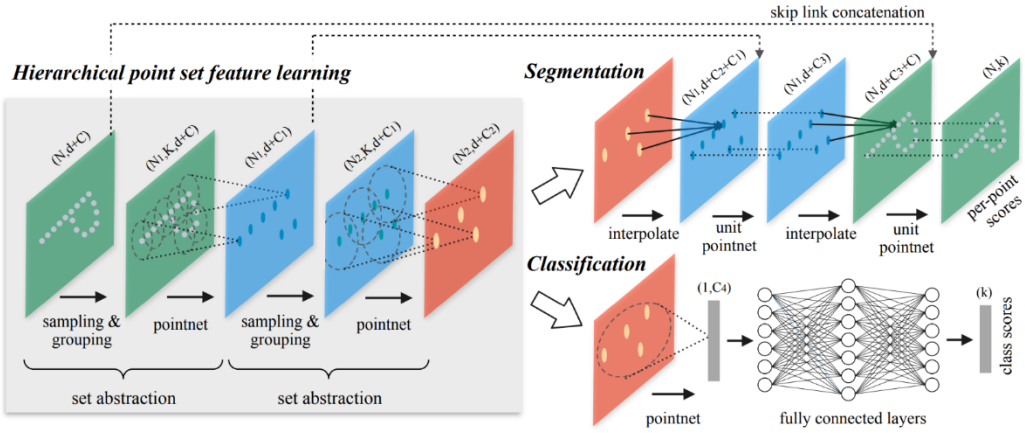


Figure 41 PointNet++ architecture (Qi, Yi, et al., 2017)

PointNet++ is trained by minimizing the total cross-entropy between the one-hot representation of each point`s classification and the softmax output. For a single point, this works out to:

$$L_{point} = -\sum_{c=1}^{2} y_{true,i} \log(y_{pred,i}), \tag{16}$$

where y_{true} is $(0, 1)^T$ if the point is a sign and $(1, 0)^T$ otherwise.

4.1.5.1.4 Changes to the PointNet++ implementation

In this thesis, a number of changes were made to (Qi, Yi, et al., 2017) method, as described below.

- *Expanded Processing Volume:* Qi et al.'s segmentation network operates on small portions of a scene individually (1.5m by 1.5m by 3m 'cubes'). This is increased to 10m x 10m x 10m cubes, with the justification that the original cube size is too small to capture crucial contextual hints obtainable only by examining a larger volume. The choice of 10m was based on providing a clear view of traffic signs within the input cube volumes and is reinforced by the sensitivity analysis presented in this section.
- *Down-sampling:* The larger volume consequentially contains too many points to fit the necessary PointNet++ calculations inside the memory of an economic GPU, so the cubes are down-sampled before feeding them to the neural network. The down-sampling approach used deals with the drastically varying point densities in different parts of the same cube (e.g. of the pavement vs. a sign pole). For each 0.2m by 0.2m by 0.2m space, the point with the highest intensity is kept, and the rest are discarded. In order to obtain the fixed-cardinality input required by PointNet++, 4096 (or 8192) points are then uniformly sampled (with replacement) from the remaining points. The cube size was chosen to maximize both training speed and detection performance. A sensitivity test on this cube size in the sensitivity analysis section demonstrates the effectiveness of this cube size.
- *Extra Features:* Unfortunately, in the down-sampling step, details are thrown away to deal with computation and memory requirements of the neural network. This becomes problematic when, for example, one cannot use down-sampled points to distinguish a traffic signal from an adjacent sign. Additional local geometric features, such as intensity, z-gradient, and roughness, are therefore calculated **before the down-sampling is performed**. At each point, the features are calculated based on neighbouring points within a radius R. While fine shape details about points are lost due to the down-sampling process, geometric features retain and feed the network with critical information about neighbouring points.
- *Oversampling Sign Examples:* Even after down-sampling removes issues of unbalanced density, the vast majority of points do not belong to signs. This is resolved by oversampling

examples that include signs. Each cube given to the model for training is designated as a 'sign' (contains sign points) or as 'other' (e.g., poles, pavement, cars, etc.), and each training example given to the model is equally likely to be either one of these cube types.

- *Changes to Training Details:* Irrelevant point cloud details are pre-emptively removed by discarding all points more than 20 meters away from the trajectory of the vehicle in an automated process (this buffering occurs in both the training and validation data). Each cube used for training is centred on the origin and scaled so that $x, y, z \in [-1\text{m}, 1\text{m}]$. Furthermore, data augmentation is used as follows:
- *Jitter*: Randomly sampled shift of points in all directions. the x, y, z coordinates of each point are jittered according to a normal distribution with a mean of 0.005 m (on average the points will be shifted by 0.005 m) (clipped between [-0.01, 0.01], any shift that would result in a change greater than 0.01 meters or less than -0.01 is not allowed). Certain random shifts are introduced to improve the robustness and generalization of the model (Qi et al., 2016).
- *Rotation*: Azimuth rotation refers to a rotation of the object around the z-axis, while elevation rotation involves a rotation around the x or y axis. The points are rotated, so the model would learn to be invariant to z-axis rotations as well as small tilts. This is done by rotating about the z-axis (Azimuth rotation) by a random angle within $[0, 2\pi]$, then rotating about the y-axis (Elevation Rotation) by a random angle within $\left[-\frac{\pi}{8}, \frac{\pi}{8}\right]$, and finally by rotating again about the z-axis (Azimuth rotation) by a random angle within $[0, 2\pi]$. It is worth mentioning that the data augmentation, jitter, and rotation are recommended by (Qi et al., 2016; Qi, Yi, et al., 2017). The change in the rotation about the y-axis is done to improve the detection of tilted traffic signs. The augmentation improves the model's ability to classify 3D shapes as it introduces variability and perspective into the training data.

The model was trained in 200 epochs, each epoch consisting of 845 batches of 24 cubes (sampled as described above). The Adam (Kingma & Ba, 2014) optimizer was used with a learning rate that starts at 0.001 and halves every 15 epochs. The loss function minimized during training was unweighted cross-entropy. The GPU used is a GeForce RTX 2080 Ti with 11 GB VRAM.

SqueezeSegV2 Data Preparation

SqueezeSegV2 accepts rasterized 3D image-like tensors representing the visible point cloud from the perspective of the vehicle as it travels along its trajectory (B. Wu et al., 2018, 2019). Each cell of a tensor contains information about the 3D point that it represents. A series of these tensors are then passed into the neural network to train a model, which is then evaluated on identically formatted input to assess its performance.

4.1.5.2 Understanding SqueezeSegV2 input

Before training or evaluation of SqueezeSegV2 models can take place, 3D LiDAR point clouds must be converted into a series of tensors of size $H \times W \times C$, where H and W are the height and width of the tensor, respectively. The last dimension, C, contains all the necessary information and features about the visible point in that cell. To echo the dimensions used in Wu et al., tensors of size $64 \times 512 \times 6$ were created. The six fields stored in each pixel are the x, y, and z position of the point relative to the observer (point of observation), intensity, distance from the observer, and classification value. In this case, a classification value of 1 would indicate a point on a sign, while all other points have a classification value of 0. This information is illustrated in Figure 42.

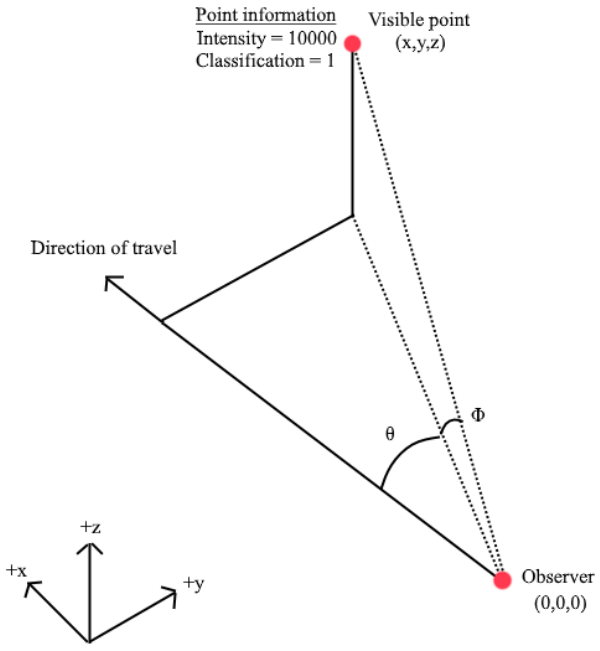


Figure 42 Variable information for SqueezeSegV2

Choosing where in the first two dimensions of a tensor to situate the information of a given point is intuitively based on the azimuth (side-to-side) and elevation (bottom-to-top) angles of the point relative to the observer, illustrated in Figure 42.

$$\boldsymbol{\theta = arcsin\left(\frac{y}{\sqrt{x^2+y^2}}\right)} \quad \textbf{(17)}$$

$$\boldsymbol{\varphi = arcsin\left(\frac{z}{\sqrt{x^2+y^2+z^2}}\right)} \quad \textbf{(18)}$$

Because the tensor only has a finite number of pixels in which to encode point information, each cell has a range of azimuth and zenith angles for which it will store point information. This range of angles is dependent on the horizontal and vertical field of view, as well as the width and height of the tensors. In this comparison, to be as representative of the original SqueezeSegV2 performance as possible, the tensor height, tensor width, and horizontal field of view were drawn from Wu et al., while the vertical field of view and look-ahead distance were drawn from the Velodyne HDL-32E LiDAR sensor specifications. The observer spacing was chosen to replicate the size of the dataset used to train SqueezeSegV2. All chosen constants in the rasterization process are shown in Table 2.

Table 2 Constants Used in the SqueezeSegV2 Rasterization Process

CONSTANT	VALUE
TENSOR HEIGHT (H)	64
TENSOR WIDTH (W)	512
TENSOR DEPTH (C)	6
LEFT VIEWING ANGLE	45°
RIGHT VIEWING ANGLE	45°
UP VIEWING ANGLE	2°
DOWN VIEWING ANGLE	25°
LOOK-AHEAD DISTANCE (LAD)	100 m
OBSERVER HEIGHT	1.2 m
OBSERVER SPACING	2 m

4.1.5.2.1 Rasterization preparation

With an understanding of the input format, a procedure for creating input tensors can now be developed. However, a few challenges need to be addressed before rasterization can begin. First, because the entirety of the point cloud exists already in the data and is not given incrementally as the vehicle moves, a trajectory must be extracted. Second, the coordinate reference frame of a point cloud is constant. This clashes with the trajectory-dependent, shifting reference frame needed for rasterization. To fix this, a sense of directionality must be established at each point along the trajectory. This is done by treating the observer as the origin of the 3D space and calculating a set of forward, rightward, and upward vectors as the basis for this 3D space.

The algorithm needed to solve both these issues is detailed in (Gouda, Mirza, et al., 2021). In brief, the algorithm first extracts uniformly spaced points situated directly below the scanner to get the trajectory. The upward vectors are calculated as the positive z normal of the best least-squares fit plane of nearby points. The forward vectors are calculated by subtracting the upward component of the trajectory from the derivative of the trajectory. Lastly, the rightward vectors are calculated as the cross-product of the forward and rightward vectors.

4.1.5.2.2 Rasterization Procedure

With a set of observers along the vehicle's trajectory extracted along with a sense of directionality established at each of those observers, rasterization can begin. The following procedure is applied to each observer along a segment's trajectory.

For a given observer, its "view frustum" represents all visible points that a LiDAR scanner could hypothetically detect. The shape of the view frustum is defined by the sensor specifications outlined in Table 2. With those specifications, a lopsided pyramid whose apex is situated on the observer can be created. An example of a view frustum can be seen in Figure 43.

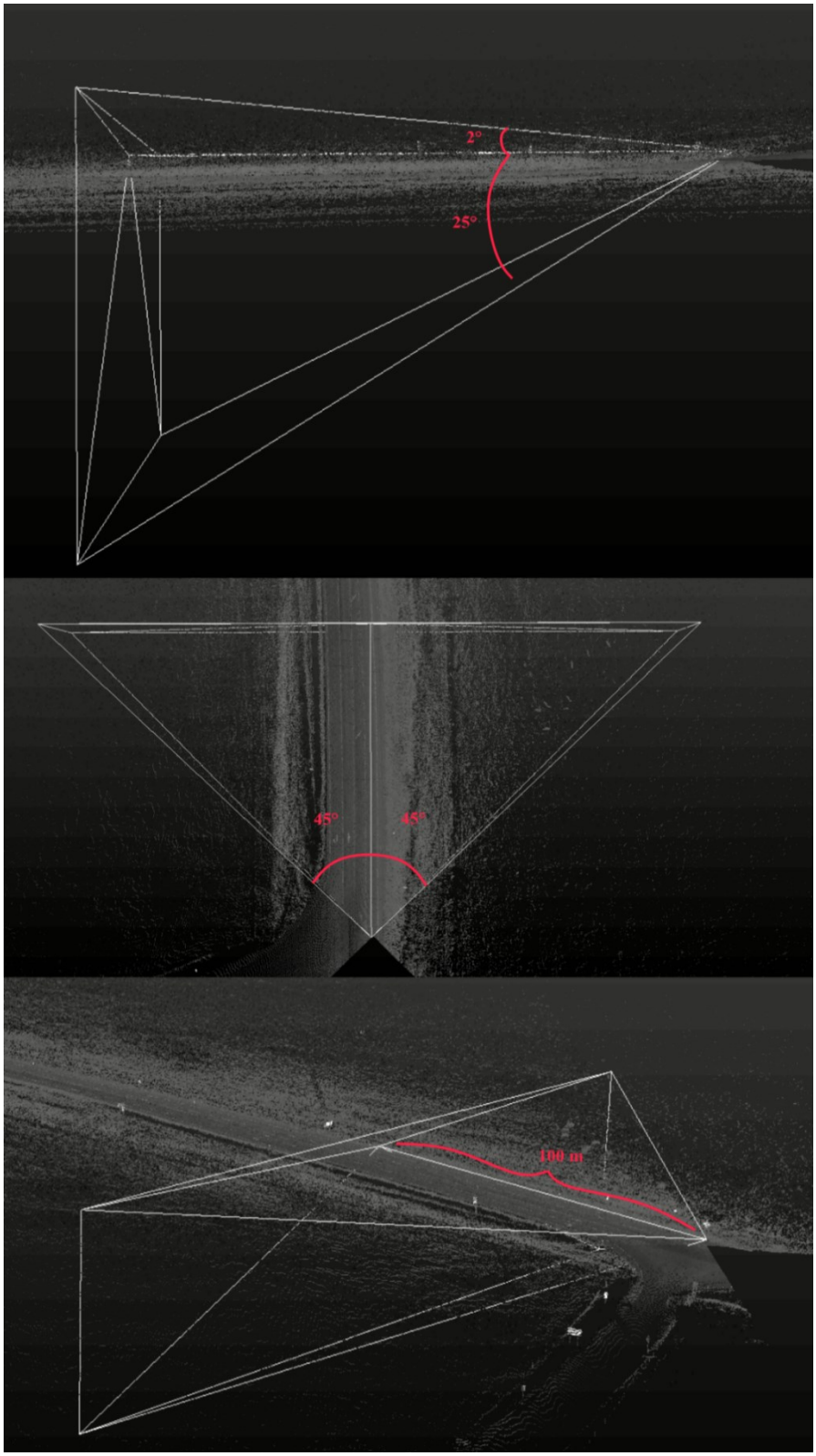


Figure 43 View frustum as seen from the side (top image), above (middle image), and from a distance (bottom image)

Using an octree to query points inside a frustum, all potentially visible points from the observer are collected. To query the points from the octree, the equations of planes representing the view frustum's faces are needed. As the directionality of the vehicle is known at that point, as well as the angular dimensions of the frustum, a set of translations needed to go from the apex of the frustum to its four corners can be established. Figure 44 shows a visualization of the frustum with translations and vertex points illustrated in the figure.

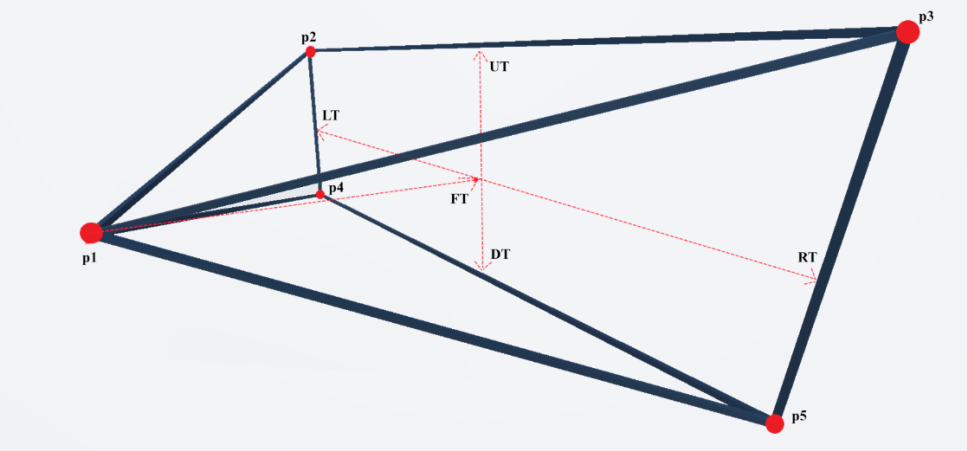


Figure 44 View frustum with translation direction and vertex points

$$Directionality = \begin{bmatrix} forward \\ rightward \\ upward \end{bmatrix} = \begin{bmatrix} f_x & f_y & f_z \\ r_x & r_y & r_z \\ u_x & u_y & u_z \end{bmatrix} \quad (19)$$

$$Forward\ translation\ (FT) = forward * LAD \quad (20)$$

$$Upward\ translation\ (UT) = upward * (LAD * \tan(up\ angle)) \quad (21)$$

$$Downward\ translation\ (DT) = -upward * (LAD * \tan(down\ angle)) \quad (22)$$

$$Leftward\ translation\ (LT) = -rightward * (LAD * \tan(left\ angle)) \quad (23)$$

$$Rightward\ translation\ (RT) = rightward * \big(LAD * tan(right\ anle)\big) \quad (24)$$

$$p_1 = observer \tag{25}$$

$$p_2 = observer + FT + UT + LT \tag{26}$$

$$p_3 = observer + FT + UT + RT \tag{27}$$

$$p_4 = observer + FT + DT + LT \tag{28}$$

$$p_5 = observer + FT + DT + RT \tag{29}$$

With the locations of all five vertices defined, so too are the equations of all five faces of the view frustum, as only three points are needed to fully define the equation of a plane.

All points within the view frustum must then be rasterized. A cell in the tensor requires that the relative *x, y,* and *z* coordinates to the observer, distance from the observer, intensity, and classification value of a point be provided, the last two being directly provided by reading in the point cloud. This leaves only the relative coordinates and distance to the point to be found.

To do this, the directionality matrix must be used once again, as it provides a basis for the relative coordinate system whose origin is the observer. To transform the coordinates from their global coordinates in the standard basis *E*, to their relative coordinates in the observer directionality basis *B,* the following equations are used:

$$[x, y, z]_E = [x, y, z]_{global} - [x, y, z]_{observer} \tag{30}$$

$$[x, y, z]_B = [x, y, z]_E * \begin{bmatrix} f_x & f_y & f_z \\ r_x & r_y & r_z \\ u_x & u_y & u_z \end{bmatrix}^T \tag{31}$$

From here, the distance of the point is found as

$$distance = norm([x, y, z]_B]) \tag{32}$$

One might naively set the cell's value to the rasterized information and move on to the next point. However, this opens the possibility of overwriting another point's rasterized information. To best model a LiDAR sensor's real-life capabilities, only the point closest to the observer should be saved. All other points which fit into the same cell and are further from the observer are behind the closest point and can be considered occluded. A visualization of this phenomenon is shown in Figure 45, wherein the green point is the closest to the observer and is kept, and the red points

behind it are occluded and discarded. All the necessary geometric features associated with the chosen point are also kept during this process and placed into the corresponding spot in the input tensor described above.

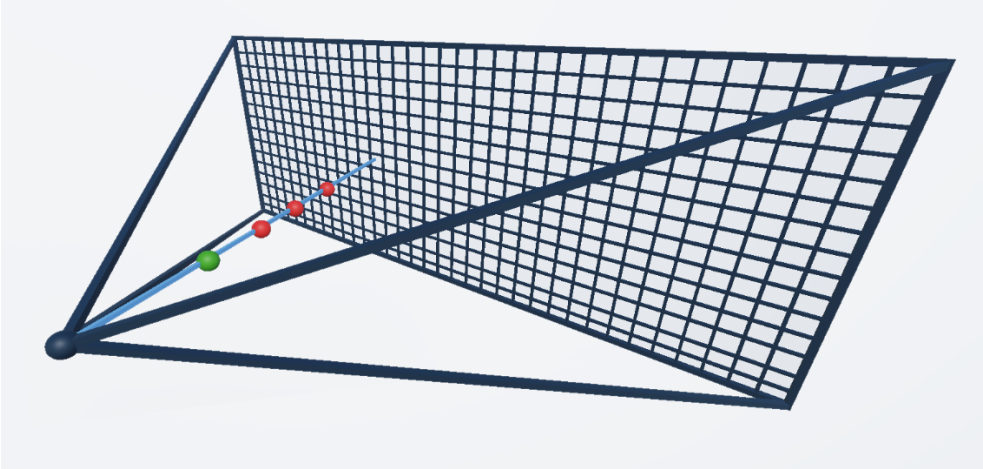


Figure 45 SqueezeSegV2 cell point selection

4.1.5.3 Model performance assessment

Different combinations of input features and number of input points were investigated. True positives (TP), false positives (FP), and false negatives (FN) were calculated per sign (as in most previous studies) and per point in an automated process as follows: first, the points the model predicts as sign points, and the ground-truth sign points, are extracted. A true positive refers to a cluster in the model output of at least 10 points that approximately overlaps a cluster in the ground truth file. Clusters are found by separating points into groups that have no neighbouring points closer than 1m from another cluster. 'Approximately overlapping' in this case means that the bounding boxes a_1, a_2 and b_1, b_2 satisfy:

$$\frac{\|a_1 - b_1\| + \|a_2 - b_2\|}{\|a_2 - a_1\| + \|b_2 - b_1\|} < 0.5, \tag{33}$$

where a_1, a_2 are the minimum and maximum bounds of the ground truth cluster, and b_1, b_2 are the minimum and maximum bounds of the model output cluster. A false negative is a cluster in the ground truth points that does not approximately overlap with any cluster in the model output, and

a false positive is a cluster in the model output that corresponds with no approximately overlapping cluster in the ground truth points.

The precision, recall, and F1 scores are calculated for each input and feature configuration and for each test file. They are calculated from the counts of true positives, false positives, and false negatives found using the automated procedure detailed above:

$$precision = \frac{true\ positives}{true\ positives + false\ positives}, \tag{34}$$

$$recall = \frac{true\ positives}{true\ positives + false\ negatives}, \tag{35}$$

$$F1 = 2 \cdot \frac{precision \cdot recall}{precision + recall}, \tag{36}$$

Finally, a permutation test was used to assess the statistical significance of the difference between the models' precision, recall, and F1-score according to the methodology proposed by (Yeh 2000). A comparison to benchmark results on the Paris-Lille-3D (Roynard et al., 2018) dataset is presented in the results section. A comparison to SqueezeSegV2 is also presented.

4.1.6 Results and discussion

To establish a baseline with which to compare the proposed method, a model with none of the changes made to the original PointNet++ implementation was trained – i.e., Cube processing volume of 1.5 m x 1.5 m x 3 m, no geometric features included in the training data, even sampling of all processing cube examples, no removal of points beyond the buffer zone, 8192 points per input volume, and no jitter and rotation of points. Table 3 A and B summarize the results of this model on each of the three validation segments, along with the overall performance. The results show that using PointNet++ without any modifications is not at all suitable for larger scenes. Compared to the total amount of ground truth signs and sign points, the baseline model made very few to no predictions. In fact, not a single sign was predicted on a per-sign basis, so no precision or F1-score value could be calculated while all recall values were 0. On a per-point basis, there was a nearly negligible number of predictions, leading to near-zero values for precision, recall and F1-score across all segments. It becomes apparent that the proposed implementation changes to PointNet++ are required to yield favourable results.

Table 3 Baseline Model Results and Baseline Model Variation Results

Table 3A Baseline model validation results (sign-based)						
Segment	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
28	0	39	0	N/A	0.000	N/A
2A	0	43	0	N/A	0.000	N/A
1A	0	40	0	N/A	0.000	N/A
Overall	0	122	0	N/A	0.000	N/A
Table 3B Baseline model validation results (point-based)						
Segment	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
28A	0	2059	4	N/A	0	N/A
2A	0	1874	0	N/A	0	N/A
1A	2	2130	0	1	0.001	0.002
Overall	2	6063	4	0.333	0.0003	0.0006

Table 3C Baseline model variation results (sign-based)							
Cube size	**Features**	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
Experimental results (28A)							
1.5x1.5x3	Intensity	38	1	32	0.543	0.974	0.697
1.5x1.5x3	Intensity, Z-gradient	38	1	180	0.174	0.974	0.296
5x5x5	None	18	21	3	0.857	0.462	0.600
10x10x10	None	25	14	9	0.735	0.641	0.685
15x15x15	None	34	5	5	0.872	0.872	0.872
Experimental Results (2A)							
1.5x1.5x3	Intensity	39	4	134	0.225	0.907	0.361
1.5x1.5x3	Intensity, Z-gradient	40	3	947	0.041	0.930	0.078
5x5x5	None	11	32	10	0.524	0.256	0.344
10x10x10	None	31	12	39	0.443	0.721	0.549
15x15x15	None	38	5	9	0.809	0.884	0.844

Experimental Results (1A)							
1.5x1.5x3	Intensity	39	1	33	0.542	0.975	0.696
1.5x1.5x3	Intensity, Z-gradient	38	2	457	0.077	0.950	0.142
5x5x5	None	8	32	4	0.667	0.200	0.308
10x10x10	None	33	7	8	0.805	0.825	0.815
15x15x15	None	33	7	3	0.917	0.825	0.868
Experimental Results (Overall)							
1.5x1.5x3	Intensity	116	6	199	0.368	0.951	0.531
1.5x1.5x3	Intensity, Z-gradient	116	6	1584	0.068	0.951	0.127
5x5x5	None	37	85	17	0.685	0.303	0.420
10x10x10	None	89	33	56	0.614	0.730	0.667
15x15x15	None	105	17	17	0.861	0.861	0.861
Table 3D Baseline model variation results (point-based)							
Cube size	**Features**	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
Experimental results (28A)							
1.5x1.5x3	Intensity	1924	135	2797	0.408	0.934	0.568
1.5x1.5x3	Intensity, Z-gradient	1930	129	7238	0.211	0.937	0.344
5x5x5	None	589	1470	453	0.565	0.286	0.380
10x10x10	None	924	1135	542	0.630	0.449	0.524
15x15x15	None	1466	593	156	0.904	0.712	0.797
Experimental Results (2A)							
1.5x1.5x3	Intensity	1754	120	5840	0.231	0.936	0.371
1.5x1.5x3	Intensity, Z-gradient	1773	101	29150	0.057	0.946	0.108
5x5x5	None	268	1606	361	0.426	0.143	0.214
10x10x10	None	1055	819	1321	0.444	0.563	0.496
15x15x15	None	1663	211	282	0.855	0.887	0.871

Experimental Results (1A)							
1.5x1.5x3	Intensity	2056	76	1692	0.549	0.964	0.699
1.5x1.5x3	Intensity, Z-gradient	2066	66	14421	0.125	0.969	0.222
5x5x5	None	430	1702	238	0.644	0.202	0.307
10x10x10	None	1496	636	628	0.704	0.702	0.703
15x15x15	None	1664	468	123	0.931	0.780	0.849
Experimental Results (Overall)							
1.5x1.5x3	Intensity	5734	331	10329	0.357	0.945	0.518
1.5x1.5x3	Intensity, Z-gradient	5769	296	50809	0.102	0.951	0.184
5x5x5	None	1287	4778	1052	0.550	0.212	0.306
10x10x10	None	3475	2590	2491	0.582	0.573	0.578
15x15x15	None	4793	1272	561	0.895	0.790	0.839

A sensitivity analysis on the original PointNet++ implementation were performed to show the necessity and relative impact of including the proposed changes. The cube size was varied, and different combinations of features were tested on the baseline PointNet++ implementation. The results of these tests are shown in Table 3 C and D. With respect to cube size, additional models were trained with cube sizes of 5m x 5m x 5m, 10m x 10m x 10m, and 15m x 15m x 15m. As cube size increased, precision, recall and F1 scores all improved, reaching their highest scores on the 15m x 15m x 15m model. For sign-based results, precision increased from a low of 0.614 to a high of 0.861. Recall improved from 0.303 to 0.861, and F1 score went from 0.420 to 0.861. Similarly, point-based results showed precision ranging from 0.550 to 0.895, recall in the 0.212-to-0.790 range and F1 scores going from 0.306 to 0.839. As with the sign-based results, the 15m x 15m x 15m model had the highest scores out of the tested models. This cube size was finally tested on the modified PointNet++ models (sensitivity analysis section), which showed that a cube size of *10m x 10m x 10m* yielded better results once the rest of the proposed changes were implemented.

Two models with additional features– trained with a cube size of 1.5m x 1.5m x 3m – showed improvements in precision (0.368), recall (0.951) and F1 score (0.531). Hence, the inclusion of extra features also improved the performance compared to the baseline model.

To show the effect of the proposed changes, Table 4, sections A and B summarize the results for each of the three validation segments and the overall performance of all segments with various models. Figure 46 shows the model output on the validation segment on Highway 1A. The blue colour marks the extent of the buffer zone, and detected traffic signs are marked in red. The three validation segments contain 122 traffic signs on 12 kilometres of rural highways. The precision, recall, and F1-score for each model were compared with varying local geometric features as inputs during the training and inference phases. The models were also trained with different input point cloud densities, with either 4,096 or 8,192 points sampled and passed to the model per cube. The overall section of Table 4 shows, in brackets, the *p*-value of the significance of the changes in precision, recall, and F1-score. The test was performed to compare the model with the highest precision, recall, and F1-score with all other models.

Table 4 Results of each validation segment and overall performance on all segments

Table 4A Validation Results (sign-based)							
Features	**Points**	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
Experimental Results (28A)							
Intensity	8192	35	4	6	0.854	0.897	0.875
Intensity, z-gradient	8192	36	3	5	0.878	0.923	0.9
Intensity, roughness	8192	33	6	10	0.767	0.846	0.805
Intensity, z-gradient	4096	38	1	2	**0.95**	**0.974**	**0.962**
Intensity, roughness	4096	35	4	5	0.875	0.897	0.886

Intensity, roughness, z-gradient	4096	37	2	6	0.86	0.949	0.902
Experimental Results (2A)							
Intensity	8192	41	2	2	0.953	0.953	0.953
Intensity, z-gradient	8192	42	1	0	**1**	0.977	0.988
Intensity, roughness	8192	43	0	1	0.978	**1**	0.989
Intensity, z-gradient	4096	43	0	0	**1**	**1**	**1**
Intensity, roughness	4096	43	0	3	0.93	**1**	0.963
Intensity, roughness, z-gradient	4096	42	1	1	0.977	0.977	0.977
Experimental Results (1A)							
Intensity	8192	39	1	8	0.83	0.975	0.897
Intensity, z-gradient	8192	40	0	1	**0.976**	**1**	**0.988**
Intensity, roughness	8192	38	2	7	0.844	0.95	0.894
Intensity, z-gradient	4096	40	0	2	0.953	**1**	0.976
Intensity, roughness	4096	40	0	3	0.93	**1**	0.964
Intensity, roughness, z-gradient	4096	40	0	3	0.93	**1**	0.964
Experimental Results (Overall)							

Intensity	8192	115	7	16	0.878 (<0.001)	0.943 (0.03)	0.909 (<0.001)
Intensity, z-gradient	8192	118	4	6	0.952 (0.30)	0.967 (0.25)	0.959 (0.16)
Intensity, roughness	8192	114	8	18	0.864 (<0.001)	0.934 (0.03)	0.898 (<0.001)
Intensity, z-gradient	4096	121	1	4	**0.968**	**0.992**	**0.98**
Intensity, roughness	4096	118	4	11	0.915 (0.07)	0.967 (0.50)	0.94 (0.06)
Intensity, roughness, z-gradient	4096	119	3	10	0.922 (0.05)	0.975 (0.25)	0.948 (0.03)

Table 4B Validation Results (point-based)

Features	**Points**	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
Experimental Results (28A)							
Intensity	8192	1094	155	267	0.804	0.876	0.838
Intensity, z-gradient	8192	1956	103	204	0.906	0.950	0.927
Intensity, roughness	8192	1922	137	233	0.892	0.933	0.912
Intensity, z-gradient	4096	1968	91	173	**0.919**	**0.956**	**0.937**
Intensity, roughness	4096	1917	142	173	0.917	0.931	0.924
Intensity, roughness, z-gradient	4096	1959	100	219	0.899	0.951	0.925
Experimental Results (2A)							
Intensity	8192	1797	77	221	0.890	0.959	0.923

Intensity, z-gradient	8192	1819	55	78	0.959	0.971	0.965
Intensity, roughness	8192	1774	100	232	0.884	0.947	0.914
Intensity, z-gradient	4096	1855	19	18	**0.990**	**0.990**	**0.990**
Intensity, roughness	4096	1786	88	129	0.933	0.953	0.943
Intensity, roughness, z-gradient	4096	1796	78	122	0.936	0.958	0.947
Experimental Results (1A)							
Intensity	8192	2063	69	175	0.922	0.968	0.944
Intensity, z-gradient	8192	2099	33	98	0.955	0.985	0.970
Intensity, roughness	8192	2070	62	64	0.970	0.971	0.970
Intensity, z-gradient	4096	2116	16	98	**0.956**	**0.992**	**0.974**
Intensity, roughness	4096	2068	64	63	0.970	0.970	0.970
Intensity, roughness, z-gradient	4096	2104	28	86	0.961	0.987	0.974
Experimental Results (Overall)							
Intensity	8192	4954	301	663	0.882	0.943	0.911
Intensity, z-gradient	8192	5874	191	380	0.939	0.969	0.954
Intensity, roughness	8192	5766	299	529	0.916	0.951	0.933

Intensity, z-gradient	4096	5939	126	289	**0.954**	**0.979**	**0.966**
Intensity, roughness	4096	5771	294	365	0.941	0.952	0.946
Intensity, roughness, z-gradient	4096	5859	206	427	0.932	0.966	0.949

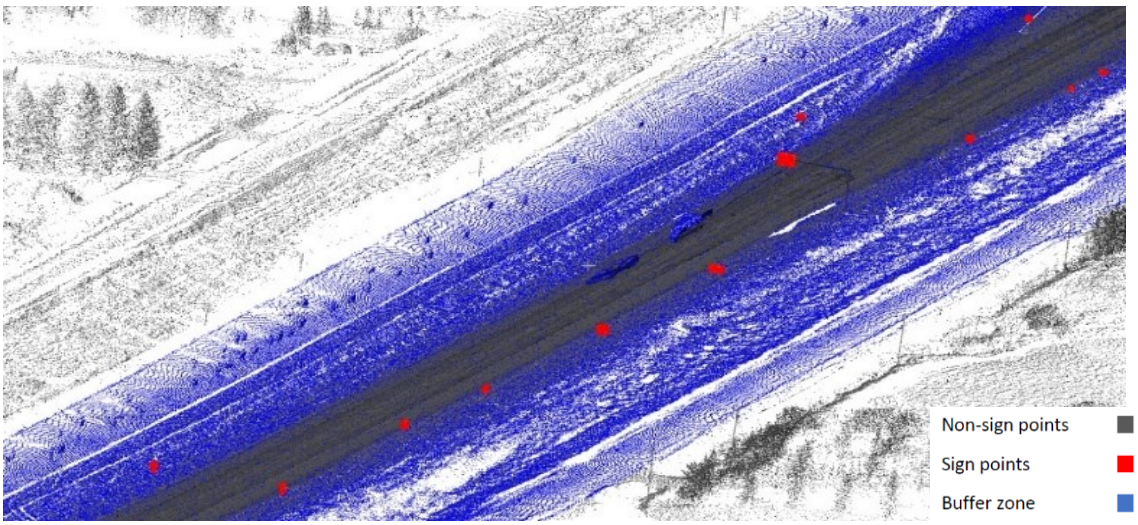


Figure 46 Model output on Highway 01A

Across all three validation segments, the models with intensity and *z*-gradient as additional geometric features perform the best in terms of precision (0.97 per sign and 0.954 per point), recall (0.992 per sign and 0.979 per point), and F1-score (0.980 per sign and 0.966 per point). This is the opposite of the results for the baseline PointNet++ models that were trained with these same features. The baseline PointNet++ model trained with only intensity as a feature performed better than the model with both intensity and z-gradient (Table 3). This is due to the modifications made to PointNet++, which include increasing the cube size to allow the network to better utilize the z-gradient values during training. A larger cube size results in a wider range of z-gradient values in each cube, which can help the model better distinguish between sign and non-sign points. The larger cube size also increases the probability of a sign being completely contained within one

cube. However, the sensitivity tests presented below show that increasing the cube size beyond a certain volume can have an adverse effect on the performance of the model.

For highway 1A when considering the per sign analysis, the model using intensity and *z*-gradient performs better with 8192 points per input volume (precision: 0.976, F1-score: 0.988) than a lower point density of 4096 (precision: 0.953, F1-score: 0.976) due to a higher number of false positives. For highway segments 28A and 1A, the lower point density performs better when adding intensity and *z*-gradient. In the per-point analysis, the model using intensity and z-gradient with 4096 points per input volume outperforms all other models on the three segments.

In the overall per-sign analysis, decreases in precision can be observed for the sign detection model, which was trained using intensity-only (0.878) versus a combination of intensity and roughness (0.864). A similar effect is observed when comparing the models trained using intensity and z-gradient (0.968) versus intensity, z-gradient, and roughness (0.922). The lower performance from the models trained with roughness appears to be caused by the higher likelihood of these models to produce false positives. This can be concluded from the recall value, which remains similar to the better-performing models. The improvement in precision and F1-score due to the use of intensity and z-gradient, with 4096 points, is statistically significant at the 95% confidence level compared to all other models, except for the model trained with the same features and higher point density. Similar performance can be observed in the per-point analysis. However, the model with intensity only shows the lowest precision (0.882), recall (0.943), and F1-score (0.911).

The model using intensity-only does not perform well in comparison, but the relative change in performance can be evaluated by adding roughness as opposed to adding z-gradient in the sign detection model. For this comparison, the point density with 8192 points per input volume is kept constant, and the per sign results are used. For the addition of roughness, a 2% decrease in precision is observed, as well as a 1% decrease in recall and a 2% decrease in the F1-score. The addition of *z*-gradient leads to an 8% increase in precision, a 2% increase in recall, and a 5% increase in the F1-score. These results indicate that z-gradient is a more helpful geometric feature when compared to roughness for detecting traffic signs.

When a lower point density (i.e., 4096 points per input volume) is used, the addition of the roughness to z-gradient and intensity leads to a 5% decrease in precision, a 2% decrease in recall,

and a 4% decrease in the F1-score. This indicates that roughness is not a helpful feature for detecting traffic signs.

In the per-point comparison, the model using intensity-only shows the lowest performance compared to all models. For the models using an 8192 point density, the addition of z-gradient increases precision by 6.5%, recall by 3%, and F1-score by 5%. For the models using 4096 points per input volume, the addition of roughness to z-gradient and intensity decreases precision by 2.5%, recall by 1.5%, and F1-score by 2%.

As evident from the above results, the proposed approach demonstrates remarkable performance on all validation segments, with per sign precision, recall, and F1-score values in the ranges of 0.95-to-1, 0.974-to-1, and 0.962-to-1, respectively. A similar per-point performance is observed with precision, recall, and F1-score values in the range of 0.92-to-0.99, 0.956-to-0.992, and 0.937-to-0.99, respectively. This also shows a substantial improvement relative to the unmodified PointNet++ models.

Evaluation of the proposed methodology was also done on the Paris-Lille-3D dataset using intensity and z-gradient-based models. A consistent point density of 4096 was used for all models due to superior performance compared to 8192-point models on the highway test data. Further demonstration of the effectiveness of the 4096-point density is found in the sensitivity analysis section. Models were trained and tested under conditions based on the standard scoring system used in the Paris-Lille-3D benchmark. The benchmark groups sign recognition with poles and traffic lights, not testing signs exclusively. Training and testing of the program were therefore done with all three object types classified using the standardized training dataset to provide an accurate comparison with existing studies. Three test files were included in the benchmark: one in Dijon, France, and two in Ajaccio, France. Runtimes of each test file averaged 5774 seconds, which is lower than the average runtime of the submitted methods on the benchmark site. Overall results show high recall rates of up to 88.1%. All models do, however, compare favourably to existing studies on the benchmark (Roynard et al., 2018), which have recall rates ranging from 37.7% to 79.7%.

4.1.6.1 *Sensitivity Analysis*

As with the baseline model, a series of sensitivity tests were done on the best-performing model to study the relative impact of the PointNet++ implementation changes. The impact of

changing the processing cube size was investigated by training models that partitioned the data into 5 m x 5 m x 5 m and 15 m x 15 m x 15 m cubes and comparing them to the best model, which had a cube size of 10 m x 10 m x 10 m. The results of this test are shown in detail in Table 5. The sign-based results show a 7.64% decrease in precision, a 2.52% decrease in recall and a 5.20% decrease in F1 score for the 5 m x 5 m x 5m cubes, while the 15 m x 15 m x 15 m results show an average 4.15% decrease across precision, recall and F1 score. Meanwhile, the point-based results show a similar trend. In comparing models, one can observe a 7.80% decrease in precision, a 1.17% decrease in recall, and a 4.61% decrease in F1 score for the 5 m x 5 m x 5 m model, and a 4.33% decrease in precision, a 1.10% decrease in recall, and a 2.73% decrease in F1 score for the 15 m x 15 m x 15 m model. This suggests that 10 m x 10 m x 10 m is an effective size to divide point cloud data when it comes to predicting traffic signs.

Table 5 Results of Cube Size Variation Test

Table 5A Validation Results (sign-based)						
Cube Size	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
Experimental Results (28A)						
5x5x5	37	2	6	0.860	0.949	0.902
10x10x10	38	1	2	0.95	0.974	0.962
15x15x15	37	2	5	0.881	0.949	0.914
Experimental Results (2A)						
5x5x5	41	2	6	0.872	0.953	0.911
10x10x10	43	0	0	1	1	1
15x15x15	40	3	4	0.909	0.930	0.920
Experimental Results (1A)						
5x5x5	40	0	2	0.952	1	0.976
10x10x10	40	0	2	0.953	1	0.976
15x15x15	39	1	0	1	0.975	0.987
Experimental Results (Overall)						
5x5x5	118	4	14	0.894	0.967	0.929
10x10x10	121	1	4	0.968	0.992	0.98
15x15x15	116	6	9	0.928	0.951	0.939

Table 5B Validation Results (point-based)						
Cube Size	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
Experimental Results (28A)						
5x5x5	1948	111	370	0.840	0.946	0.890
10x10x10	1968	91	173	0.919	0.956	0.937
15x15x15	1981	78	252	0.887	0.962	0.923
Experimental Results (2A)						
5x5x5	1810	64	294	0.860	0.966	0.910
10x10x10	1855	19	18	0.990	0.990	0.990
15x15x15	1790	84	192	0.903	0.955	0.928
Experimental Results (1A)						
5x5x5	2110	22	139	0.938	0.990	0.963
10x10x10	2116	16	98	0.956	0.992	0.974
15x15x15	2102	30	118	0.947	0.986	0.966
Experimental Results (Overall)						
5x5x5	5868	197	803	0.880	0.968	0.921
10x10x10	5939	126	289	0.954	0.979	0.966
15x15x15	5873	192	562	0.913	0.968	0.940

The impact of changing the point density per processing cube was investigated by training models with 2048 and 16384 points in addition to the 8192-point and 4096-point models. The results of this sensitivity test are detailed in Table 6. Sign-based results show that, compared to the best model, which had a point density of 4096, the model with a point density of 2048 showed an average decrease of 2.52% across all metrics. The model with a point density of 8192 decreased the least, with a drop of 1.65% in precision, 2.52% in recall and 2.55% in F1 score. Finally, the model with a density of 16384 showed a 6.92% drop in precision, a 2.52% decrease in recall and a 4.80% decrease in F1 score. Similar to the cube size sensitivity test, the point-based results are roughly the same in precision compared to the sign-based results but show a smaller reduction in recall relative to the best model. The 2048-point density model showed a 2.17% decrease in precision, a 0.06% decrease in recall, and a 1.11% decrease in F1 Score. The 8192-point

test showed a 1.57% decrease in precision, a 1.02% decrease in recall, and a 1.24% decrease in F1 score. Lastly, the 16384-point test showed a 6.171% decrease in precision, a 0.381% decrease in recall, and a 3.37% decrease in F1 score.

Table 6 Results of Density Variation Test

Table 6A Validation Results (sign-based)						
Density	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
Experimental Results (28A)						
2048	38	1	7	0.844	0.974	0.905
4096	38	1	2	0.950	0.974	0.962
8192	36	3	5	0.878	0.923	0.900
16384	38	1	5	0.884	0.974	0.927
Experimental Results (2A)						
2048	40	3	0	1	0.930	0.964
4096	43	0	0	1	1	1
8192	42	1	0	1	0.977	0.988
16384	40	3	5	0.889	0.930	0.909
Experimental Results (1A)						
2048	40	0	0	1	1	1
4096	40	0	2	0.953	1	0.976
8192	40	0	1	0.976	1	0.988
16384	40	0	3	0.930	1	0.964
Experimental Results (Overall)						
2048	118	4	7	0.944	0.967	0.955
4096	121	1	4	0.968	0.992	0.980
8192	118	4	6	0.952	0.967	0.959
16384	118	4	13	0.901	0.967	0.933
Table 6B Validation Results (point-based)						
Density	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
Experimental Results (28A)						

2048	2009	50	282	0.877	0.976	0.924
4096	1968	91	173	0.919	0.956	0.937
8192	1956	103	204	0.906	0.950	0.927
16384	1989	70	306	0.867	0.966	0.914
Experimental Results (2A)						
2048	1812	62	53	0.972	0.967	0.969
4096	1855	19	18	0.990	0.990	0.990
8192	1819	55	78	0.959	0.971	0.965
16384	1813	61	169	0.915	0.967	0.940
Experimental Results (1A)						
2048	2113	19	89	0.960	0.991	0.975
4096	2116	16	98	0.956	0.992	0.974
8192	2099	33	98	0.955	0.985	0.970
16384	2113	19	218	0.906	0.991	0.947
Experimental Results (Overall)						
2048	5934	131	424	0.933	0.978	0.955
4096	5939	126	289	0.954	0.979	0.966
8192	5874	191	380	0.939	0.969	0.954
16384	5915	150	693	0.895	0.975	0.933

In the down-sampling process, the cube size used in the best model was 0.2m x 0.2m x 0.2m. The effects of changing this cube size are shown in

Table *7*. Both the point-based results in

Table 7A and the sign-based results in

Table 7B show a small decrease in recall and a large decrease in both precision and F1 scores for both the 0.1 and 0.4 models relative to the best model, which has a cube size of 0.2.

Table 7 Down-sampling Cube Size Sensitivity Results

Table 7A Validation Results (sign-based)						
Cube Size	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
Experimental Results (28A)						
0.1x0.1x0.1	37	2	7	0.841	0.949	0.892
0.2x0.2x0.2	38	1	2	0.950	0.974	0.962
0.4x0.4x0.4	36	3	8	0.818	0.923	0.868
Experimental Results (2A)						
0.1x0.1x0.1	40	3	5	0.889	0.930	0.909
0.2x0.2x0.2	43	0	0	1	1	1
0.4x0.4x0.4	39	4	5	0.886	0.907	0.897
Experimental Results (1A)						
0.1x0.1x0.1	36	1	4	0.900	0.973	0.935
0.2x0.2x0.2	40	0	2	0.953	1	0.976
0.4x0.4x0.4	36	1	5	0.878	0.973	0.923
Experimental Results (Overall)						
0.1x0.1x0.1	113	6	16	0.876	0.950	0.911
0.2x0.2x0.2	121	1	4	0.968	0.992	0.980
0.4x0.4x0.4	111	8	18	0.860	0.933	0.895
Table 7B Validation Results (point-based)						
Cube Size	**TP**	**FN**	**FP**	**Precision**	**Recall**	**F1**
Experimental Results (28A)						
0.1x0.1x0.1	1959	100	391	0.834	0.951	0.889
0.2x0.2x0.2	1968	91	173	0.919	0.956	0.937
0.4x0.4x0.4	1966	93	423	0.823	0.955	0.884
Experimental Results (2A)						
0.1x0.1x0.1	1793	84	220	0.891	0.957	0.923
0.2x0.2x0.2	1855	19	18	0.990	0.990	0.990
0.4x0.4x0.4	1792	82	423	0.857	0.956	0.904

Experimental Results (1A)						
0.1x0.1x0.1	2047	27	187	0.916	0.987	0.950
0.2x0.2x0.2	2116	16	98	0.956	0.992	0.974
0.4x0.4x0.4	2052	22	257	0.889	0.989	0.936
Experimental Results (Overall)						
0.1x0.1x0.1	5799	211	798	0.880	0.965	0.921
0.2x0.2x0.2	5939	126	289	0.954	0.979	0.966
0.4x0.4x0.4	5810	197	978	0.856	0.967	0.908

A series of rotations is done on the data, both on the y-axis and the z-axis. To show the advantage of including this rotation, a version of the best-performing model was trained with the rotation removed. This model obtained overall sign-based precision, recall and F1 scores of 0.884, 0.951 and 0.915. Compared to the best model, this one showed an 8.71% decrease in precision, a 4.17% decrease in recall, and a 6.61% decrease in F1 score. The point-based results were 0.915, 0.965, and 0.921, representing a decrease of 4.04%, 1.42% and 4.62%, respectively.

4.1.6.2 SqueezeSegV2 Results and Discussion

To determine the best possible comparison between SqueezeSegV2 and the proposed modified PointNet++, ten models were trained for different lengths of time, represented by steps. For the sake of comparison, all SqueezeSegV2 models are trained using the same segments as the proposed modified PointNet++ models used in their training. The segments used in testing were also the same across both methods. The first model was trained for 5,000 steps to provide a sufficiently large sampling of the dataset. The number of steps was incremented by 5,000 for each subsequent model, finishing with a model of 50,000 steps, the default number listed in the SqueezeSegV2 code. The results in Table 8A show the cell-based precision, recall and F1 scores of the models. The table shows that the scores decrease for the later models, likely because of overfitting to the training segments. The models with the best overall performance were the models trained for 10,000 steps. This model took approximately 50 minutes to train and evaluated each segment in 25 seconds. It had average precision, recall and F1 scores of 0.959, 0.836 and 0.892, respectively.

Compared to the best model of the proposed modified PointNet++, which had scores of 0.968, 0.992 and 0.980, the performance of SqueezeSegV2 is lower.

Table 8 SqueezeSegV2 Results

Table 8A SqueezeSegV2 Base Results				
Steps	**Segment**	**Precision**	**Recall**	**F1**
5000	1A	0.972	0.854	0.909
	2A	0.978	0.750	0.849
	28A	0.964	0.728	0.830
10000	1A	0.948	0.906	0.927
	2A	0.966	0.806	0.879
	28A	0.964	0.796	0.872
15000	1A	0.875	0.948	0.910
	2A	0.849	0.861	0.855
	28A	0.831	0.911	0.869
20000	1A	0.878	0.945	0.910
	2A	0.864	0.852	0.858
	28A	0.905	0.884	0.894
25000	1A	0.878	0.889	0.883
	2A	0.854	0.682	0.758
	28A	0.882	0.879	0.880
30000	1A	0.810	0.887	0.847
	2A	0.778	0.754	0.766
	28A	0.866	0.882	0.874
35000	1A	0.783	0.829	0.805
	2A	0.717	0.636	0.674
	28A	0.793	0.775	0.784
40000	1A	0.803	0.911	0.854
	2A	0.753	0.767	0.760

	28A	0.853	0.855	0.854
45000	1A	0.786	0.820	0.803
	2A	0.661	0.679	0.670
	28A	0.843	0.798	0.820
50000	1A	0.750	0.864	0.803
	2A	0.570	0.726	0.639
	28A	0.828	0.871	0.849

Table 8B SqueezeSegV2 Extra Features Results

Extra Features	**Segment**	**Precision**	**Recall**	**F1**
Roughness	1A	0.923	0.861	0.891
	2A	0.975	0.744	0.844
	28A	0.963	0.788	0.867
Z-Gradient	1A	0.961	0.751	0.843
	2A	0.988	0.574	0.726
	28A	0.963	0.657	0.781
Roughness, Z-gradient	1A	0.968	0.843	0.901
	2A	0.977	0.748	0.847
	28A	0.960	0.753	0.844

To provide a more accurate comparison between SqueezeSegV2 and the modified PointNet++, the SqueezeSeg program was slightly modified to accept tensors of larger dimensions. The internal network structure was not modified, only the dimensions of the tensors that are passed into the network. As a result, SqueezeSegV2 models could be trained on data with more features than were used in (B. Wu et al., 2018). Thus, three more models were trained with additional features in the input tensors. In addition to intensity– the first model was trained with roughness as an additional feature, the second with z-gradient and the third with roughness and z-gradient. This allowed us to compare the modified PointNet++ program with SqueezeSegV2 using the same features for both programs. Table 8B summarizes the results from these models. The results show that adding additional features can increase the precision of the model when compared to the unmodified SqueezeSegV2 (0.959 to 0.971) at the expense of recall (0.836 to 0.661) and,

consequently, F1 score (0.892 to 0.783). That is, the models with extra features are more likely to underpredict sign points compared to the original SqueezeSegV2 model. Overall, although adding features improved part of the performance of SqueezeSegV2, it still cannot replicate the results produced by the proposed model, which had respective precision, recall and F1 scores of 0.968, 0.992 and 0.980.

Computation Time

The modified PointNet++ implementation significantly improved the computational requirements of the original. The baseline, unmodified PointNet++ model with a cube size of 1.5m x 1.5m x 3m and a point density of 8192 required 11.5 minutes per training epoch and 67 minutes to evaluate each segment. The improved PointNet++ model with a cube size of 10m x 10m x 10m and a point density of 4096 required only 7 minutes per training epoch and 1 to 1.5 minutes to evaluate each segment. Both PointNet++ models trained for 200 epochs, for total training times of 38.3 hours and 23.3 hours, respectively. The best performing SqueezeSegV2 model trained for approximately 50 minutes and evaluated each segment in 25 seconds, though not as effective as the best model in classification performance.

4.1.7 Conclusions and future research

This thesis demonstrated that the PointNet++ neural network architecture, with adjustments, can achieve remarkable results on large metric scale scenes, i.e., extraction of traffic signs from LiDAR point clouds. The use of local geometric features, calculated before the down-sampling of the point cloud, was found to significantly improve the neural network's performance in terms of precision, recall, and F1-score. Several models with different combinations of local geometric features (i.e., intensity, roughness, and *z*-gradient) and different numbers of points were trained using labelled data from seven highway segments (28 km total) in Alberta, Canada. The models' performances were tested on three segments (approximately 12 km) based on the precision, recall, and F1-score metrics. The statistical significance of the model variations was determined using permutation tests. Sensitivity tests were also applied to the PointNet++ model to justify the proposed adjustments. Another neural network architecture, SqueezeSegV2, was tested against the PointNet++ model to see how they would compare. The best-performing SqueezeSegV2 model

showed a decrease in precision, recall, and F1 score relative to the best-modified PointNet++ model.

Overall, in the per-sign analysis, the intensity and *z*-gradient models with 4096 points significantly outperformed other models in precision, recall, and F1-score on all validation segments. The sign detection performance on three segments showed a 99.2% recall and 98% F1-score. The intensity with roughness combination yielded a lower performance compared to the intensity-only and the intensity with *z*-gradient models. Overall, compared to the intensity model, the inclusion of *z*-gradient significantly increased the precision by 9%, the recall by 4.9%, and the F1-score by 7.1%.

Similar conclusions were observed in the per-point analysis with 98% recall and 97% F1-score on the three validation segments. However, the intensity-only model yielded the lowest performance compared to all other models regarding precision, recall, and F1-score. Overall, compared to the intensity model, including *z*-gradient significantly increased the precision by 8.2%, the recall by 4.0%, and the F1-score by 6.0%. The proposed model showed high performance in accuracy and processing time in comparison to previous studies. A four-km road segment could be segmented in approximately 60 to 100 seconds, with 95% to 100% sign detection accuracy.

Due to the flexibility of deep learning approaches and their ability to avoid potentially restrictive manual preprocessing, such as thresholding, the model avoids certain limitations in previous research. For example, intensity thresholding, as used by most previous studies, is inherently unable to detect low-intensity signs correctly. In contrast, the model proposed by this work was observed to accurately classify sign points regardless of their intensity level. Additionally, any sign type given during training may be classified by this approach, such as overhead signs, which was a limitation discussed in (S. Gargoum et al., 2017). Further, due to the high accuracy in detecting sign points, this approach can help in the automatic development of TSIs with accurate sign locations, placement, dimensions, orientation, visibility, and shape. It is worth noting that, owing to the high scanning speed on rural highways, point density decreases as the exposure time to the scanner decreases, hindering the ability to classify signs using LiDAR data. This is typically not a concern in urban areas as lower speed limits allow the same scan rate to collect a higher density of points (Karsten, 2019). As such, if video data is collected, the detected sign locations and points can help detect the video frames wherein the contents of the detected

signs can be classified using image processing, producing the last requirement for building a TSI. Another interesting application would be studying traffic sign occlusions and vision obstructions for human drivers and autonomous vehicles, which would help develop performance-based sign placement guidelines.

This work facilitates several opportunities for future research. First, applying a similar approach to extracting road features is suggested (e.g., pavement surface, road markings, etc.). In addition, testing the possibility of performing semantic segmentation on several features simultaneously would be another challenge worthy of investigation.

4.2 METHODOLOGIES

Two different methodologies were used in this thesis to analyze the visibility and sight distance of the signs along several road segments. They were validated and compared to determine the more accurate method. The first method involved a voxelization and ray casting technique, whereas the second method utilized an octree organization and convex hull view frustum to determine visibility. The following subsections describe the shared and varying processes that each of these methods uses. A high-level flowchart of the methodologies is shown in Figure 47. The data structures used are similar to section 5.A1.2.

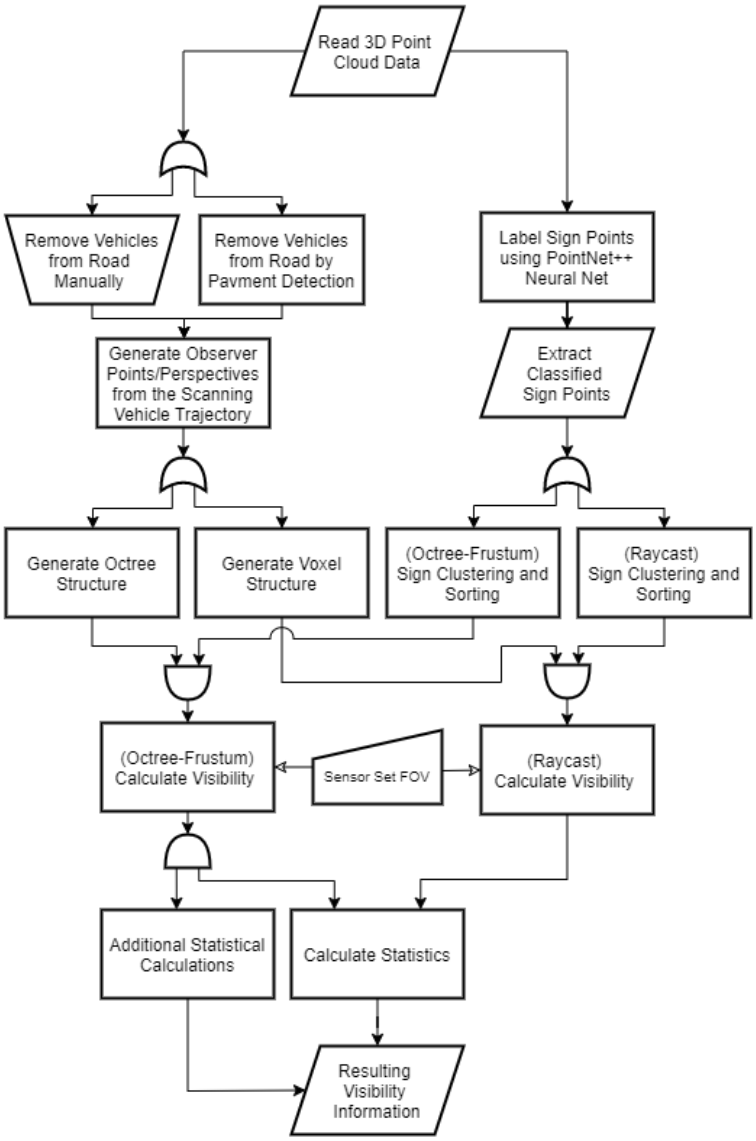


Figure 47 Methodology flowchart (comparing both methods)

4.2.1 Pre-processing

Prior to using this proposed methodology, the input point cloud (L) must have all sign points labelled. The signs were labelled using a PointNet++ neural network (Gouda, Epp, et al., 2022) and trained on point clouds with road signs labelled manually in CloudCompare. Further labelling and filtering were done based on parameters including roughness, gradient, and point density. This filtered data was then run through a modified PointNet++ neural network with increased

processing volume and random sign rotations to improve the identification of misaligned signs. Network configuration was done to optimize accuracy at the expense of false positives, as these were easily removed manually.

Classification of test data segments was then done following an automated process of filtering to format data similarly to the training data. After all the sign points were classified, the original point cloud is filtered to contain only sign points, resulting in a new point cloud denoted S. Then, S was filtered such that only points with intensity greater than $\frac{l_{avg}}{4}$ were considered, which yields a new point cloud S_{new}.

The last pre-processing step was to remove any vehicles that were on the road. Other vehicles act as obstructions, but since they don't always exist there, they cause false positives. Vehicles were removed by first detecting the pavement from the input point cloud. The pavement is characterized by a maximum z-gradient norm (K_1), density power (K_2), density power per z-gradient norm deviations (K_3), roughness deviation (K_4), and density gradient deviations (K_5), as a function of the number of lanes selected. The values used are shown in the table below.

Table 9 Parameters used for pavement filtering

Number of lanes	K_1	K_2	K_3	K_4	K_5
1	0.0675	1.3	-1.175	0.925	-
2	0.0675	1.1	-1.15	0.875	-0.005
More	0.065	1.25	-1.225	0.925	-

All values with a z-gradient norm greater or equal to (K_1) were filtered out. Point cloud volume density was calculated for a radius of $0.45\ m$, and the roughness was calculated for a radius of $1.0\ m$.

After this, the density power per z-gradient norm (DZG) deviations filter was applied, the DZG property was defined for each point i as shown in equation (37). These values were then used to compute the average ($\overline{DZG}$) and the standard deviation ($\sigma(DZG)$) of DZG, and all points i satisfying equation (38) were filtered out.

$$DZG_i = \frac{Density_i^{K_2}}{cosh\left\|\vec{\nabla} z_i\right\|} \tag{37}$$

$$DZG_i < \overline{DZG} + K_3 \cdot \sigma(DZG) \tag{38}$$

Two more filters were applied using the same notation of standard deviation for the previously applied variables. After doing so, all points with roughness greater than or equal to $K_4 \cdot \sigma(roughness)$ were filtered out. When the density gradient ($\vec{\nabla} density$) deviations were used, the last filter removed all points satisfying equation (39), thus leaving the remaining points located on the pavement of the road.

$$\left\|\vec{\nabla} density\right\|_i \geq \overline{\left\|\vec{\nabla} density\right\|} + K_5 \cdot \sigma\left(\left\|\vec{\nabla} density\right\|\right) \tag{39}$$

After filtering for the pavement was performed, vehicles were found by searching above the pavement for point clusters. Those that exist between 1 - 2 m above the pavement were classified as vehicles and were removed from the input point cloud L. In any cases where vehicles were not removed, sometimes due to low density, they were removed manually. The equations and parameters used in this section were developed through trial and error.

4.2.2 Defining observers

In this work, observers were defined as a set of points following a vehicle's trajectory that simulates the visibility positions for an autonomous vehicle. First, the car's trajectory was extracted from LiDAR data by taking points with zero scan angle and sorting them by GPS time. All selected points were then smoothed using a window to create a curve that describes the car's trajectory. Along the curve, points were sampled every 1 m and shifted up by 1.2 m, leading to the set of observers (O).

To explore the full potential of a point cloud, a second set of observers (O_{opp}) was defined to consider a car driving in the opposite direction of the first set. This was done by simply shifting the original observer set, O, 4 m in their leftward directions and flipping their orientations and order. This works on standard, non-separated, two-way, two-lane highways due to the simple geometry of the road surface (Figure 48). In other cases, a manual method was used by selecting control points in the opposite lane. Observer points were then generated by interpolating 1 m apart between the control points and translating them upwards of 1.2 m.

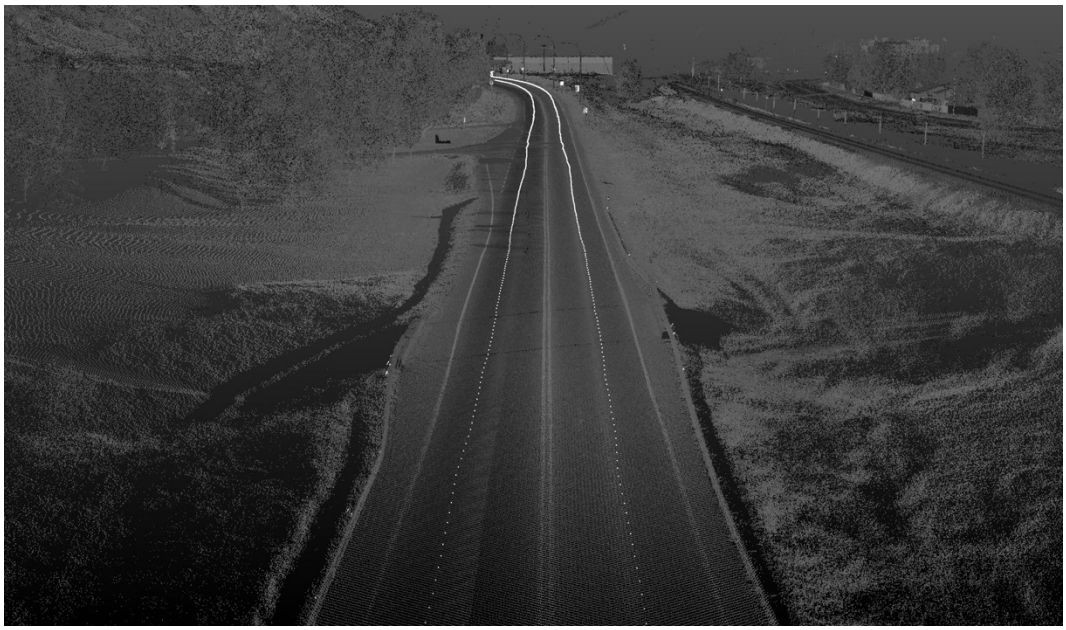


Figure 48 Observer points in both directions along a two-way two-lane road

4.2.3 Sign clustering and sorting

a) Raycast Method – Sign Clustering and Sorting

In the raycast method, signs were clustered based strictly on proximity to each other. Points were clustered such that no two clusters were within 0.5 m of each other. After defining these separate clusters as individual signs, their mean centre points were calculated. Using the centre points of these sign clusters, the closest observer to the sign was calculated and used to determine

the sign's distance from the road. Signs beyond 20 m from the road were not considered, and such clusters were removed from the sign set. The remaining sign centre points formed the set C_{all}.

A set of signs $S \subset C_{all}$ was created by sorting signs according to a given set of observers O or O_{opp} to obtain an ascending sign set according to the closest observer. First, the trajectory vector ($\vec{t}$) for each observer point o_i was calculated as $\vec{t} = o_{i+1} - o_i$. After, the relative position vector ($\vec{r}$) of a sign $s_j \in S$ and o_i was calculated as $\vec{r} = s_j - o_i$.

Then, the observer's index that led to the minimum value of the dot product $\vec{t} \cdot \vec{r}$ for a selected sign s_j was recorded. The procedure described was repeated for all signs, leading to a relation of each sign and the index of the observer that is the closest projection of s_j along the vehicle's trajectory. This relation made it possible to sort S according to the ascending observers' index.

To consider observers driving in both directions for the same point cloud, C_{all} was divided into two disjoint sets, C and C_{opp}, containing only the signs that were related to each set of observers. First, C_{all} was sorted according to O to have a global index for each sign. After, for the projection p of each sign $c \in C_{all}$ in O calculated to sort signs, the coordinates of c in the coordinate system centred in p and oriented following O was computed.

The orientation angle (θ) of the observer $o_i \in O$ located in p was calculated by taking the vector between the next observer o_{i+1} and o_i and calculating its angle with respect to the xy plane. Therefore, for $\vec{w} = (x_w, y_w, z_w) = o_{i+1} - o_i$, the angle θ was described as shown in equation 40.

$$\theta = atan2\left(\frac{y_w}{x_w}\right) \tag{40}$$

After, the homogeneous transformation matrix (H_W^P) from the coordinate system (P) centred in p to the global coordinate system (W) was calculated. First, using θ the rotation matrix from P to W along the z axis (R_W^P) was described as shown in equation 41.

$$R_W^P = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}^T \tag{41}$$

Defining d_P^W as the vector from the origin of W to the origin of P, follows that $d_P^W = p$. Equation 42 shows H_W^P in function of R_W^P and d_P^W.

$$H_W^P = \begin{bmatrix} R_W^P & -R_W^P \cdot d_P^W \\ 0 & 1 \end{bmatrix} \tag{42}$$

The matrix H_W^P calculated was applied in sign c to find the coordinates of c from the perspective of p (c'). If c' had negative y coordinate, it was placed on the right side of p and therefore $c \in C$. Otherwise, $c \in C_{opp}$. At this point, there were signs separated into C and C_{opp}, but the order of signs in each set and their index were relative to the first sort. Thus, C was sorted according to O and C_{opp} according to O_{opp} to adjust the index of signs and their order.

b) Octree-Frustum Method – Sign Clustering and Sorting

In the octree-frustum method, signs were obtained by clustering points in S_{new} using DBSCAN such that each point in the cluster (or sign) was located at least 0.5 m away from all points in another cluster. Utilizing an orthogonal-least-squares-fit (affine-fit), a plane and four corners can be assigned to these arbitrary point clusters such that they represent the sign structure and area. The orthogonal fit returns three vectors of importance, the normal vector (n) to the sign plane and two tangential vectors (t_1, t_2) corresponding to the principal directions of the shape. Also returned is the mean point (p_{mean}) of all sign points $p \in P_{sign}$. With this information, the corner points (c_1, c_2, c_3, c_4) and the centre of the sign (c_{mean}) can be calculated as follows:

$$A_{min} = \min_{p \in P_{sign}} t_1 \cdot (p - p_{mean})$$

$$A_{max} = \max_{p \in P_{sign}} t_1 \cdot (p - p_{mean})$$

$$B_{min} = \min_{p \in P_{sign}} t_2 \cdot (p - p_{mean})$$

$$B_{max} = \max_{p \in P_{sign}} t_2 \cdot (p - p_{mean})$$

$$c_1 = p_{mean} + A_{min} t_1 + B_{min} t_2$$

$$c_2 = p_{mean} + A_{max} t_1 + B_{min} t_2$$

$$c_3 = p_{mean} + A_{max} t_1 + B_{max} t_2$$

$$c_4 = p_{mean} + A_{min} t_1 + B_{max} t_2$$

$$c_{mean} = \frac{1}{4}(c_1 + c_2 + c_3 + c_4) \tag{43}$$

Figure 49 illustrates how this fit is applied to a point cluster.

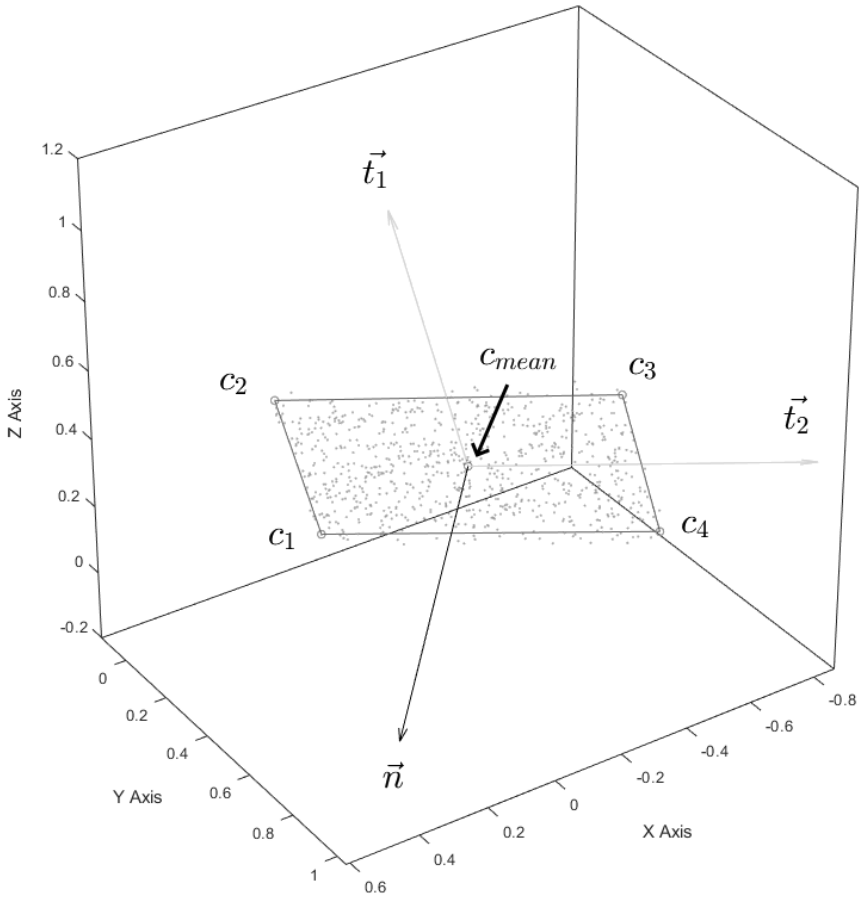


Figure 49 Sign point cluster fit, displaying corner points, mean point, and corresponding computed vectors

Additionally, shifted corner points can be calculated using two scaling factors (s_1, s_2) to decrease the area of the sign. This may be useful when the details of the sign are perhaps constrained to a smaller area than the physical size of the sign, such as when the sign has a large border. The shifted corner points (p_1, p_2, p_3, p_4) can be calculated as follows:

$$
\begin{aligned}
p_1 &= c_1 + \frac{1-s_1}{2}(c_2 - c_1) + \frac{1-s_2}{2}(c_4 - c_1) \\
p_2 &= c_2 + \frac{1-s_1}{2}(c_1 - c_2) + \frac{1-s_2}{2}(c_3 - c_2) \\
p_3 &= c_3 + \frac{1-s_1}{2}(c_4 - c_3) + \frac{1-s_2}{2}(c_2 - c_3) \\
p_4 &= c_4 + \frac{1-s_1}{2}(c_3 - c_4) + \frac{1-s_2}{2}(c_1 - c_4)
\end{aligned} \tag{44}
$$

These formulas stretch the border around the sign by a ratio of s_1 in the direction of t_1 and by a ratio of s_2 in the direction of t_2, both centred around the mean point.

The clusters were filtered such that only ones with centre points within 20 m of their closest observer were considered to filter out distant signs from the road, the filtered centre points forming a set C_{all} similar to the raycast method.

$$
o_{close} = \arg\min_{o \in O} \| c_{mean} - o \| \tag{45}
$$

Additionally, each sign is either on the left or the right side of the road. This is determined based on the dot product of the vector from its closest observer to the sign and the leftward vector for that observer.

$$
\text{left} \Leftrightarrow l_{close} \cdot (c_{mean} - o) > 0 \tag{46}
$$

Identifying which side of the road the sign is on allows the signs to be filtered such that one side of the road may be analyzed later by splitting C_{all} into C and C_{opp} corresponding to signs on the right-hand side of the respective observer set O or O_{opp}. Lastly, the angle of the sign to the roadway can be determined. If the angle is greater than 90°, the normal is flipped to the other side of the sign on the assumption that it would not make sense for a sign to face away from the road.

$$\theta = \operatorname{acos}\left(\frac{n \cdot (c_{mean} - o)}{\|c_{mean} - o\|}\right) \tag{47}$$

The created set of signs, $S \subset C_{all}$, is sorted according to a given set of observers O or O_{opp} to obtain an ascending sequence of signs according to the closest observer indexes.

4.2.4 Defining sensor sets

The analysis of visibility by an autonomous vehicle is strongly dependent on the number of sensors and their field of view. This can vary between vehicles, which is why a simulation-based approach with configurable sensor characteristics is valuable. The number of sensors, the description of the vertical and horizontal limits, and the range for each sensor used forms a sensor set. Figure *50* shows how the individual sensor boundaries are defined.

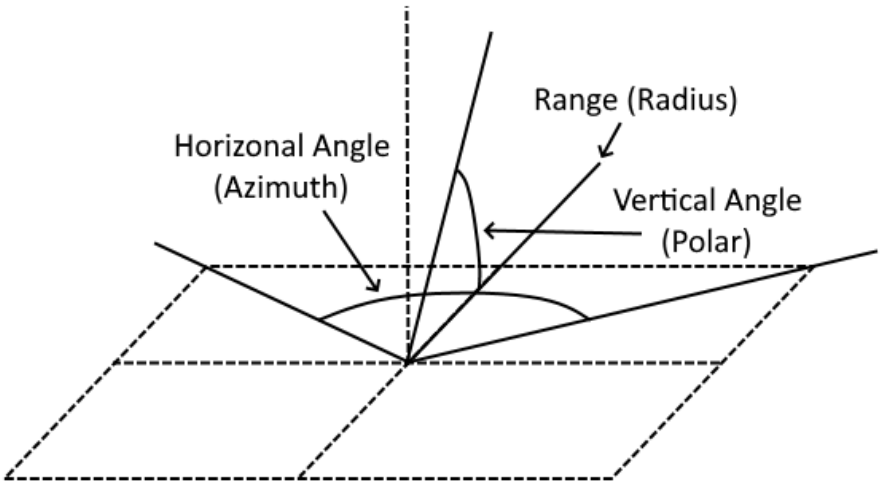


Figure 50 Sensor angles using the spherical coordinate axis

Generally, sensors are defined in spherical coordinates. This suggests that the horizontal and vertical limits or boundaries are given as an azimuthal or polar angle, respectively, and the forward vector being 0° at any given observer. Therefore, each sensor has a minimum and maximum angle in each of these directions, with upwards and clockwise from the forward vector considered as the positive directions for these angles. The range of sensors also varies, bounding the distance of the individual sensor FOV arcs created by these angles and causing the FOV to appear as a section from a sphere with the range as the radius. Figure *51* shows what this looks like, provided by the sensors in Table 10.

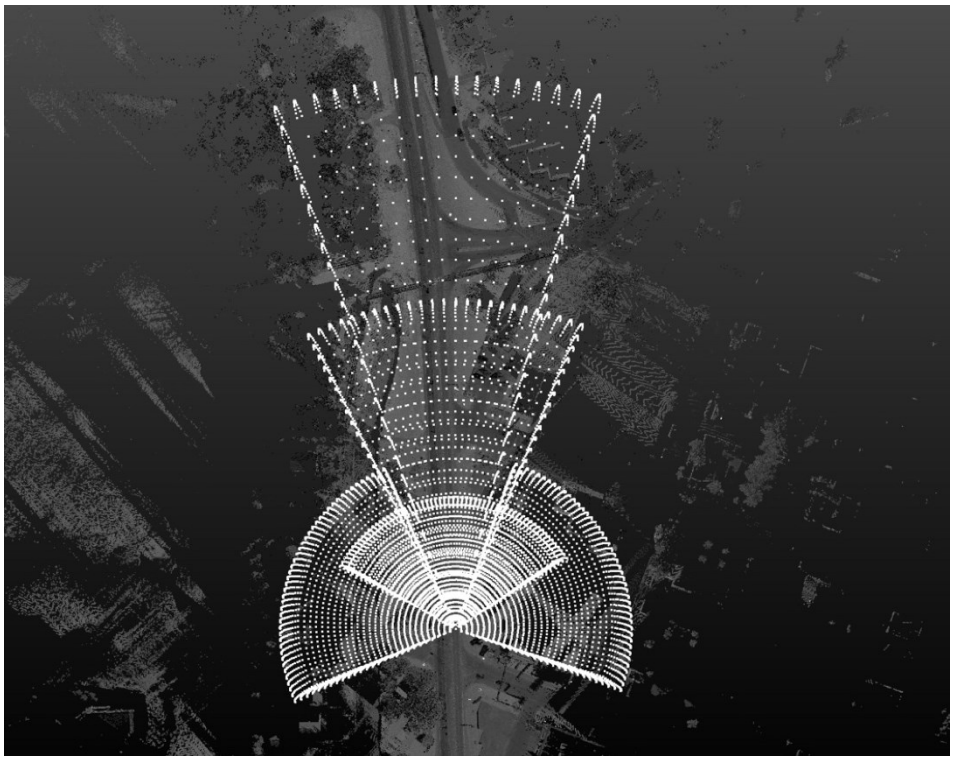


Figure 51 Example sensor set from above, given the sensors referenced in Table 11

4.2.5 Calculating Visibility

c) Raycast Method – Calculating Visibility

After data pre-processing, we can determine all observers of each set that were visible from each sign point $c_j \in C$ or $c_j \in C_{opp}$. For every observer $o_i \in O$ or $o_i \in O_{opp}$, all signs $c_j \in C$ or $c_j \in C_{opp}$ were considered depending on which direction was being analysed. If the sign c_j was not visible from o_i, the next sign c_{j+1} was analysed. In this autonomous vehicle scenario, a sign not visible from one observer means it was not within any of its sensor ranges or viewing angles for the selected sensor set centred at the observer point.

For all signs within a sensor range, Bresenham's line algorithm (Bresenham, 1965) was used to cast a ray from o_i to c_j and check each ordered voxel r on the ray for occlusion from a voxel in L. If there was an obstruction, sign c_j was not visible from o_i. Otherwise, o_i was visible from c_j. The process was repeated for each sign until the last one was analysed for each set. For each possible direction was defined a matrix of visibility (V or V_{opp}) in which the element of the

i-th line and j-th column (v_{ij}) was equals to 1 if the sign j was visible from the observer i. Otherwise, $v_{ij} = 0$.

d) Octree-Frustum Method – Calculating Visibility

For every observer $o_i \in O$ or $o_i \in O_{opp}$, all signs $c_j \in C_{all}$ were considered. If the sign c_j was not visible from o_i, the next sign c_{j+1} was analysed. Exactly as in the raycast method, a sign not visible from one observer means that it was not within range of any individual sensors or viewing angles for the sensor set selected centred at the observer point.

For all signs within a sensor range, a volume bounding frustum was created and queried against an octree organization of the point cloud L to determine if any obstructions existed between the observer o_i and the sign c_j. If there was an obstruction, sign c_j was not visible from o_i. This view frustum is a rectangular pyramid starting at the observer and ending just before the sign face so as not to intersect the points that make up the sign itself, which would cause errors in the evaluation of obstructions.

The apex of the view frustum is an observer, and the base of the view frustum is made up of a shifted version of the scaled corner points p_1', p_2', p_3', p_4' discussed in the previous section. The shift or cut-off of these points is given by the following equation and is repeated for each scaled corner point. Figure 52 shows an example of a view frustum with a shift amount of 1 m and stretch factors $\boldsymbol{s_1 = s_2 = 1/\sqrt{2}}$.

$$p_k' = o_i + (p_k - o_i)\left(1 - \frac{\text{shift amount}}{\|c_{mean} - o_i\|}\right) \tag{48}$$

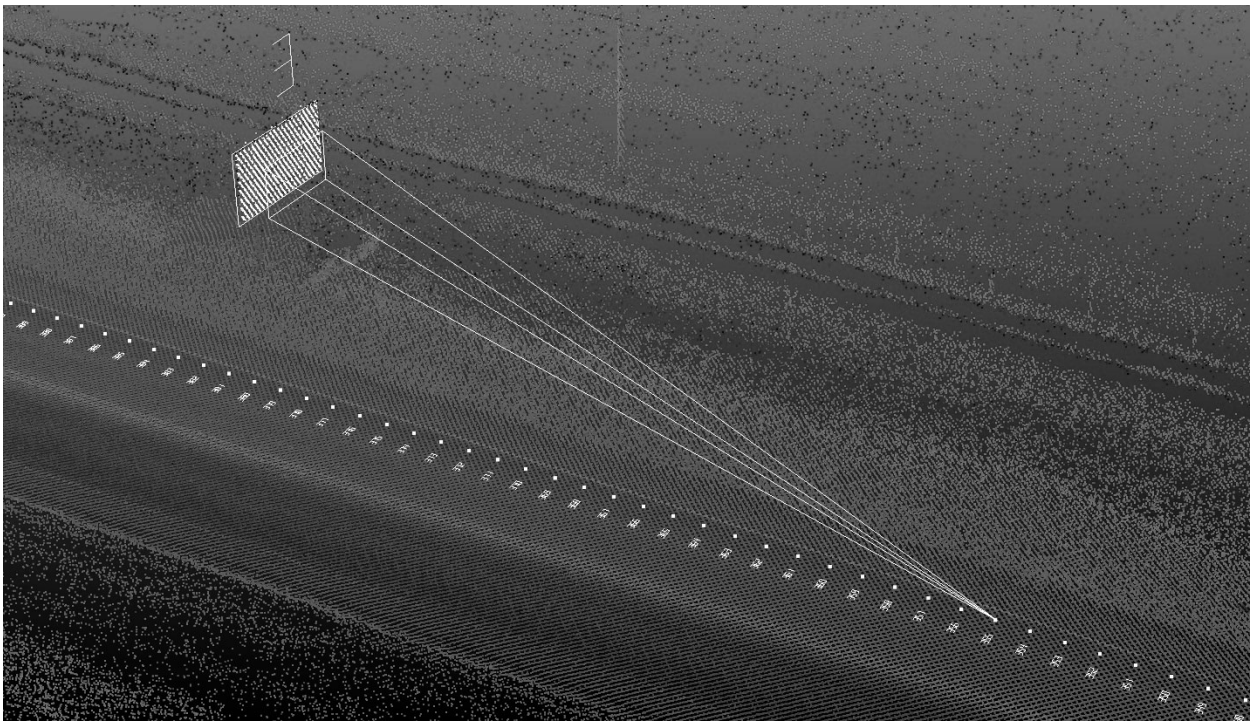


Figure 52 View frustum with a shift amount of 1 m and stretch factors $s_1 = s_2 = 1/\sqrt{2}$

To determine the intersection of half-spaces that constrain this frustum, a point inside it must be found. In turn, this would be inside all the half-spaces required to define the constraints of this volume. A point that is guaranteed to be inside this frustum is the average of the 5 points that make up the corners, given by the following equation.

$$m = \frac{p_1' + p_2' + p_3' + p_4' + o_i}{5} \tag{49}$$

Therefore, the intersection of half-spaces defining the constraints of the view frustum is given by:

$$H_{frustum} = f(p_1', p_2', p_3', m) \cap f(o_i, p_1', p_2', m) \cap f(o_i, p_2', p_3', m) \cap f(o_i, p_3', p_4', m) \cap f(o_i, p_4', p_1', m) \tag{50}$$

$H_{frustum}$ can then be passed to the octree of L to query for any points that exist inside this volume. Any points that are inside the frustum are obstructing the view of the observer to the sign. For each possible direction, a matrix of visibility (V or V_{opp}) was defined in which the element of the i-th line and j-th column (v_{ij}) was equal to 1 if the sign c_j was visible from the observer o_i, otherwise, $v_{ij} = 0$. It should be noted that in each direction V or V_{opp} could be additionally segmented and/or filtered by C or C_{opp} to only the signs on the right side of the road. However, the computation evaluates the visibility of all signs for the given observer set.

4.2.6 Calculating Statistics

The calculated visibility score matrices were used to generate various statistics, such as total visible distance and time, for each sign. If the sum of visibility scores for all observers in relation to a sign c_j was equal to 0, then the sign was not visible to any observer o_i. In this case, the next sign c_{j+1} was taken to be analysed. Otherwise, the statistics for the sign were calculated.

To calculate the statistics for a sign, the first visible observer o_f and the last visible observer o_l were used to define a set of observers, $U = \{o_f, o_{f+1}, \dots, o_{l-1}, o_l\}$, between the first visible and last visible observer, inclusive. Then, all visible observers $U' = \{o_u \mid o_u \in U, v_{uj} = 1\}$ were found.

The sets U and U' were used to calculate statistics related to visibility. The first statistic calculated was the total visible distance of the sign, given by $|U'|$ multiplied by the observer interval of 1 m. After that, the maximum speed was defined using the total visible distance and dividing it by the sensor reaction time (0.5 s). This speed limit represents the maximum speed an autonomous vehicle may drive in order to have enough time to process the given signs.

Later, the relation between U and U' was used to generate other information about signs. The number of obstructions obscuring a sign along the path from o_f to o_l were calculated as $|U - U'|$. Finally, the proportion of visible observers was calculated as $|U'|/|U|$.

To compare the results to characteristics of autonomous vehicles, a manual check for the speed limit of a segment mapped by the point cloud was made using Google Street View. The design speed for urban areas was adopted as 75 km/h, while a design speed of 135 km/h was used for rural areas.

The design speed of the area can be used to calculate the distance needed for the autonomous vehicle to react to a sign (reaction distance). The referred calculation was done by dividing the sensor reaction time by the design speed (in meters per second). The reaction distance was used to compare if the total visible distance for a sign was enough to guarantee safe conditions for an autonomous vehicle. This can be considered the minimum decision sight distance (DSD), given a reaction time, defined by the American Association of State Highway Transportation Officials (AASHTO, 2018). Similarly, a comparison between the total visible time for each sign and the sensor reaction time was made. Thus, the total visible time for a sign was defined as the total visible distance divided by design speed.

e) Octree-Frustum Method – Continuous Sight Distance (Alternate Sight Distance)

In the octree-frustum method, the sight distance was calculated the same way as in the raycast method, as the sum of observers that had an unobstructed view frustum of the sign given by the visibility matrix V or V_{opp}. Additionally, the continuous sight distance was calculated, i.e., the longest consecutive set of observers with an unobstructed view of the sign. The continuous sight distance SD_c is equivalent to the longest consecutive set of indexes from U that intersect U'. This is described by equation 15.

$$SD_c = max(\sum_{i=0}^{|U|} Ui \qquad (51)$$

$$o_i \in U' \wedge o_{i-1} \in U'$$

4.3 RESULTS AND DISCUSSION

4.3.1 Test Segments

The proposed method was applied to three segments of Highway 01A in Alberta, Canada. First, a segment from km 12 to km 16 (segment one) was selected because it has a high number of signs. Two other segments were considered: one from km 0 to km 4 (segment two) and one from km 8 to km 12 (segment three). As each 4 km segment was analyzed for both possible directions, the segments generated results for 24 km. All test segments were collected in 2020 by Alberta Transportation using a REIGL VMX 450 LiDAR system capable of collecting 360° point cloud

data. Point cloud densities ranged between 150 points/m2 to 1,000 points/m2 depending on survey speed and distance from the scanner.

4.3.2 Sensor set used

For all types of sensors analyzed, an angular resolution of 0.1 degrees was adopted for both vertical and horizontal ranges. This work also considered vertical angles ranging from -60° to 30° for all sensors. The referential adopted for orientation was the x axis pointed in the front of the vehicle and the z axis pointing up, perpendicular to the ground.

The set was a representation of a combination of different sensors (Tesla, 2023). Table 10 shows a description of the sensor set, and Figure *53* shows a representation of an observer field of view using the sensor set.

Table 10 Description of the horizontal angle limits and range for each component of the sensor set

Sensor name	Horizontal angle limits (degrees)	Range (m)
Narrow forward (1)	-17.5° to 17.5°	250
Main forward (2)	-25° to 25°	150
Wide forward (3)	-60° to 60°	60
Side sensor 1 (4)	-112.5° to -22.5°	80
Side sensor 2 (5)	22.5° to 112.5°	80

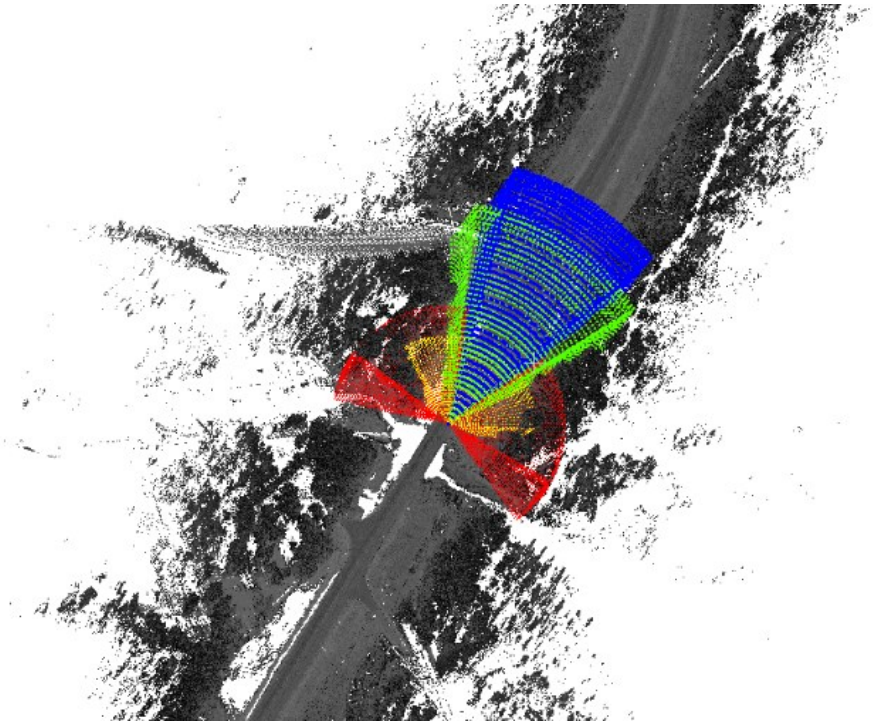


Figure 53 Representation of each sensor field of view in a different colour for the sensor set

4.3.3 Comparison of Methods

Each of the raycast and octree methods analyzed the sight distance to signs along a roadway segment. These methods were then compared against each other to determine inaccuracies that may have caused discrepancies in the results. The following section displays and discusses several figures comparing the different results of each method in one direction on the same 4 km segment from Highway 1A (segment one). Figure 54 shows the comparison of sight distance on segment one, set C, between the two methods.

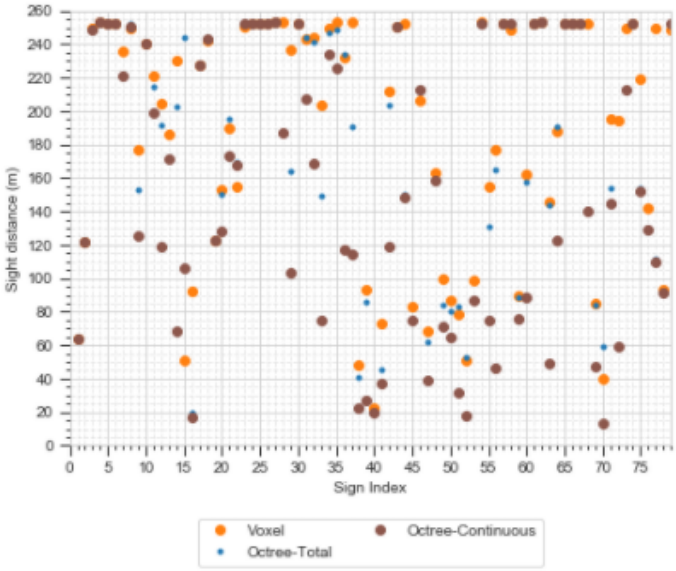


Figure 54 Octree vs. Voxel Sight Distance Comparison on Segment One, Set C

Three distinct types of dots can be seen in Figure 54, representing the raycast method, the octree method with continuous sight distance, and the octree method with total sight distance. Most of the discrepancies between the two octree methods are of less importance as they operate differently. It is possible that the continuous sight distance to some arbitrary sign equals 1 m, while the total sight distance equals half of the maximum sensor range, 125 m, for a difference of 124 m, in the case where consecutive observers alternate between obstructed and unobstructed. This could occur if a sign was viewed through a fence line. The concerning discrepancies are between the total sight distance of the raycast method and of the octree method, as these results are expected to be the same.

Some of the results with large differences between the two methods were excluded. For example, Sign 33 was excluded as it was not a traffic sign and was distanced from the road, and sign 41 was excluded as it was for the perpendicularly connecting road.

The first sign analyzed from set C was sign 21, shown in Figure 55. The octree total sight distance was calculated at 195 m, the octree continuous sight distance was calculated at 173 m, and the voxel total sight distance was calculated at 190 m.

Figure 55 Highway 1A Segment One, Sign 21

The obstructions for sign 21 are mostly applicable to sign 22 as well, as it is positioned approximately 75 m behind. Additionally, sign 21 blocks the view of sign 22, justifying the lower sight distance calculated in both methods.

Sign 21 is blocked by several obstacles, including a guardrail, a pole, and some other points along the roadside. Displayed in Figure 56 are these points, and Figure 57 shows a section of the raycasts as well as a section of the octree frustums obstructed by them.

Figure 56 Sign $c_{21} \in C$, obstructing points

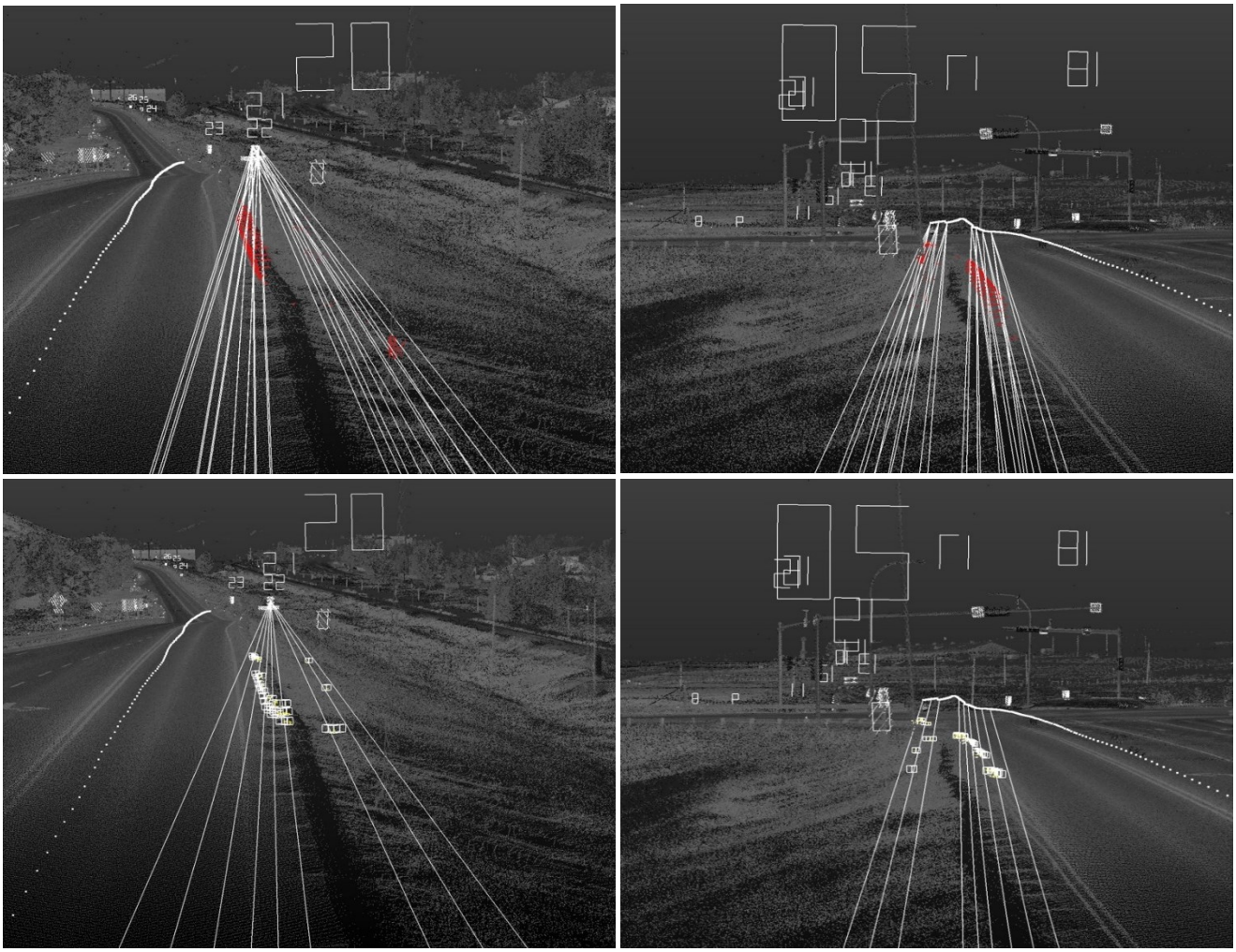


Figure 57 Sign $c_{21} \in C$, raycasts and frustums from the forward and reverse-facing perspectives

The voxels, represented by the encased yellow points, are those obstructing raycasts (bottom row of Figure 57); the red points obstructing the frustums (top row of Figure 57). Though they seem to overlap almost identically, five more raycasts are obstructed than frustums, causing a small difference in the calculated sight distances. This provides evidence of the difference in precision between the two methods.

The raycast method uses voxels and Bresenham's line algorithm, an incremental error algorithm, to detect obstructions based on the best approximation. This can occur when the points aggregated into a voxel group exist near one boundary of the voxel, and the raycast passes through or near an opposite boundary of that voxel. Bresenham's line algorithm may calculate such a voxel

as being in the trajectory of the raycast when the points inside the voxel did not actually intersect the ray and were, in fact, a small distance away from it. This shall be referred to as 'clipping,' as in the ray clipped a voxel that contains empty space adjacent to and or near the ray.

In the octree method, clipping does not occur. The octree method bounds a specific volume independent of the point cloud itself, which allows for a precise point query into the octree against a set of equalities. By detecting possible obstructions against a set of equalities, the accuracy of the results is dependent strictly on the precision of the input point cloud.

Sign 44 was also analyzed, as shown in Figure 58. The raycast method did not calculate any obstructions and recorded a maximum total sight distance of 252 m. The octree method recorded 150 m and 148 m for the total and continuous sight distance, respectively. The difference in total sight distance of 102 m was from sign 43, blocking a small portion of sign 44. The difference between the two methods is shown in Figure 58.

Figure 58 Sign $c_{44} \in C$ obstruction.

The last few signs of the set C encountered a similar problem, as shown in Figure 59. The raycast method only evaluates the visibility of the sign center point, whereas the octree method considers the entire sign area, thus accurately detecting more obstructions.

Figure 59 Sign $c_{77} \in C$ obstructions

In many cases, the results of the two methods overlapped with a mean percent difference of only 1.2%. Still, it can be seen from the preceding examples that, in some cases, a large discrepancy in the results is observed due to the limitations of the voxel-based raycasting method.

Most of the sign information needs to be visible for a certain amount of time to allow the driver to process the information and possibly take some form of action. Therefore, the octree method should be considered a more realistic representation of sign visibility as it encompasses a better majority of sign information. As a result of the precision of LiDAR data in depicting a road environment in 3D, the proposed octree-based evaluation method surpasses the state-of-the-art raycast method used in previous studies in accuracy and provides a more realistic and practical result regarding sight distances.

4.3.4 Case study on segment one

Figure *60* shows the results of the proportion of in-range observers for Highway 01A using the sensor set and analyzing the sets O and C. The values shown by the referred figure indicate the existence of some signs that have a significant number of obstruction events compared to visible events.

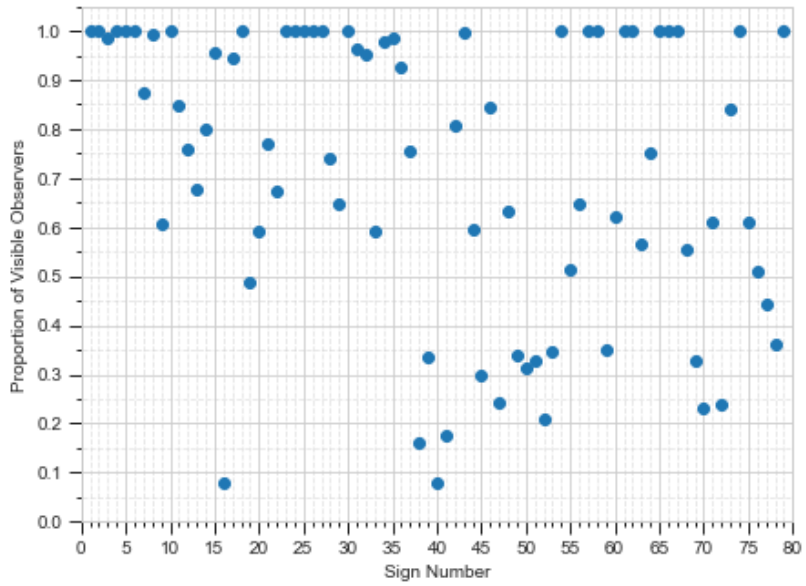


Figure 60 Proportion of visible observers for each sign number using the sensor set and considering O and C

Using the number of visible observers, the sight distances were calculated and shown in Figure *61*. To calculate the reaction distance, the regions from sign 1 to sign 26 and from sign 71 to sign 79 were rural, while the region from sign 27 to sign 70 was an urban area.

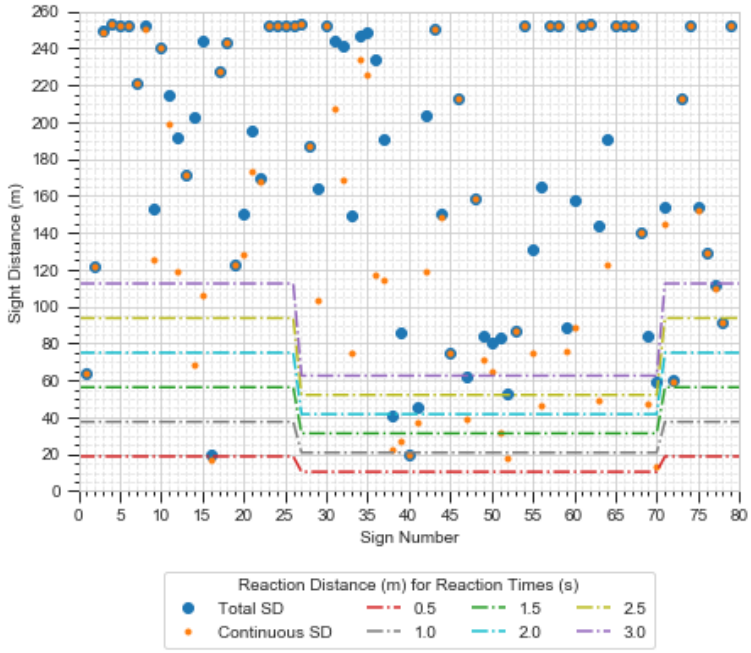


Figure 61 Total visible distance for each sign calculated using the sensor set

A few signs along the segment were not visible for a long enough distance to be analyzed by a road user for certain reaction times at the given speeds. Signs that are low in both Figure *60* and Figure *61* would be of concern. The total visible time was computed, and the results are shown in Figure *62*. Using Figure *60*, Figure *61*, and Figure *62*, a manual search was done for signs that had poor visibility in comparison to other signs.

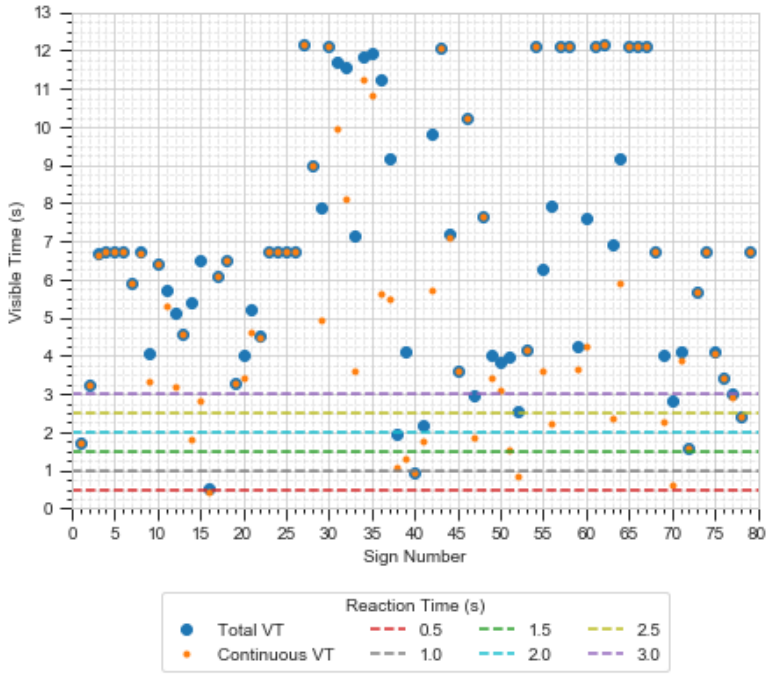


Figure 62 Total visible time for each sign calculated using the sensor set

Figure *63* shows a comparison of signs 77 and 78 along the trajectory of a vehicle. First, the point cloud data shows sign 77 (green), covering parts of sign 78 (pink) from the perspective of an observer. Figure *63* shows the same location when it was possible to see the same situation identified using the point cloud (middle image). Finally, the third image shows the same location after sign 77 was removed, and sign 78 was replaced. This example shows the proposed methodology automatically extracting the position of a sign replaced by Alberta Transportation. Several other cases on the rest of the data are presented in Figure 64, Figure *65*, Figure *66*, and Figure *67*.

Figure 63 A comparison for the location of signs 77 and 78. On the left, the point cloud data. In the middle, an image of the location before the change. On the right, the same location after sign 77 was removed

Figure 64 Sign obstructed by a roadside tree

In the first image of Figure *65*, the point cloud data are shown. In this image, sign 10 (pink) is occluded by sign 8. The middle and right images show the location before and after sign eight was changed to improve its sight distance.

Figure 65 A comparison of the location of signs 10, 9 and 8

Figure 66 A comparison for the location of sign 31, middle and left show the sign after and before the change

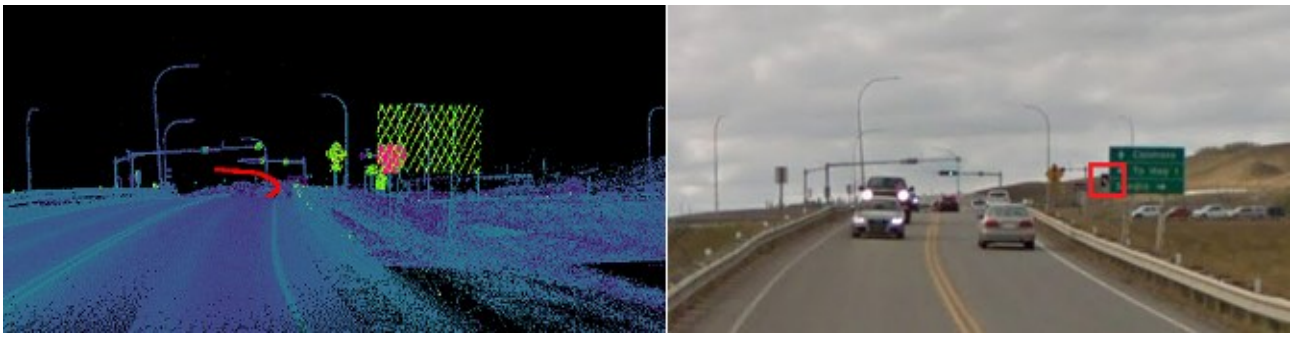


Figure 67 Sign 37 occluded by sign 36 in the point cloud data (left) and a picture of the road (right)

4.4 CONCLUSIONS AND LIMITATIONS

The widespread introduction and support for autonomous vehicles is growing as manufacturers race to release higher-level AVs to consumers. Despite this, government agencies have been slow to assess whether the existing road infrastructure can safely handle these vehicles. Though this is the case, research has determined that AVs will rely on signs for road and safety information comparably to humans until the infrastructure, environment, and AVs become fully connected to each other and the road network. Such a network will enable the information to be passed over wireless connections and shared with relevant connected entities. Until then, road users will continue to rely on visual recognition of signs for an enormous amount of vital information such as speed limit compliance, lane-keeping, and hazardous warnings.

Though AV sensors and human eyes are different, research has also determined that AVs and humans read signs using similar methods, requiring them to be properly placed, oriented, and unobstructed, as highly visible signs can be effective in mitigating accidents. Despite this, the assessment of sign placement, orientation, and visibility relies on the manual efforts of traffic departments, making the task exceedingly tedious, time-consuming, and costly.

This section proposed a novel automated approach to assess the visibility of signs in the road environment based on AV characteristics. Though the methodology was focused on AVs the method is still applicable for human drivers. The approach uses mobile LiDAR data of 3 different road segments in Alberta, Canada, as input, detecting signs using a PointNet++ neural network. Visibility was assessed by sampling perspectives from the scanning vehicle's trajectory and calculating view frustums to signs in the FOV of typical AV sensors. The sensor set FOV was configurable since multiple sensor setups were tested and compared. The input cloud was

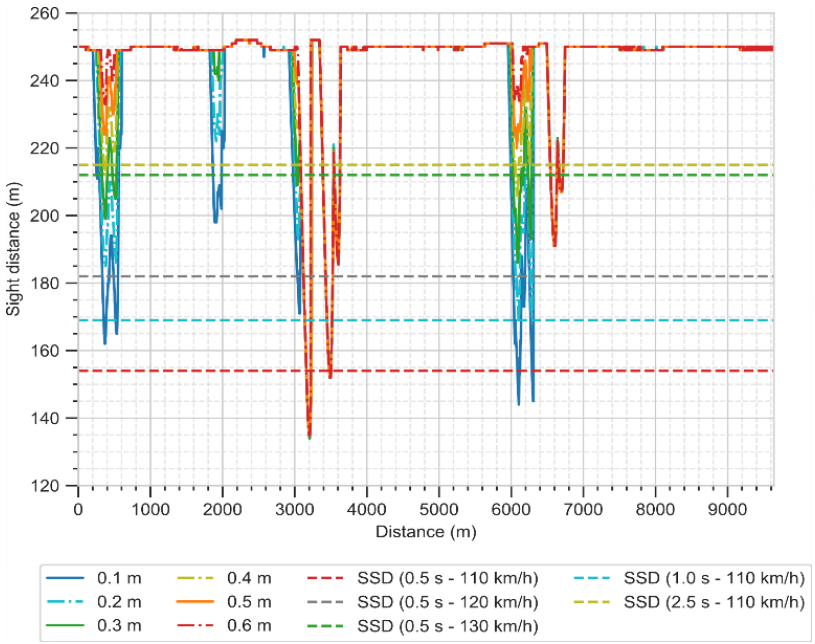
organized into an octree data structure such that the calculated view frustums were spatially queried for obstructions. Frustum placement and query configuration accommodated noise in the input data to be mitigated in the results, allowing for several statistical analyses, including visible distance, visible time, continuous visible distance, maximum speed, and more.

The proposed method was compared to an adapted raycasting approach, which voxelates the input cloud and detects obstructions using Bresenham's line algorithm. It was determined that the octree method is superior as it considers a 3D view perspective and the area of the sign face, whereas the raycasting approach only considers the centre point of a sign and thus acts more as a point-to-point view perspective. Additionally, though the results were fairly comparable, the octree method was determined to be more precise than the raycasting method, especially evident in extreme cases of discrepancy.

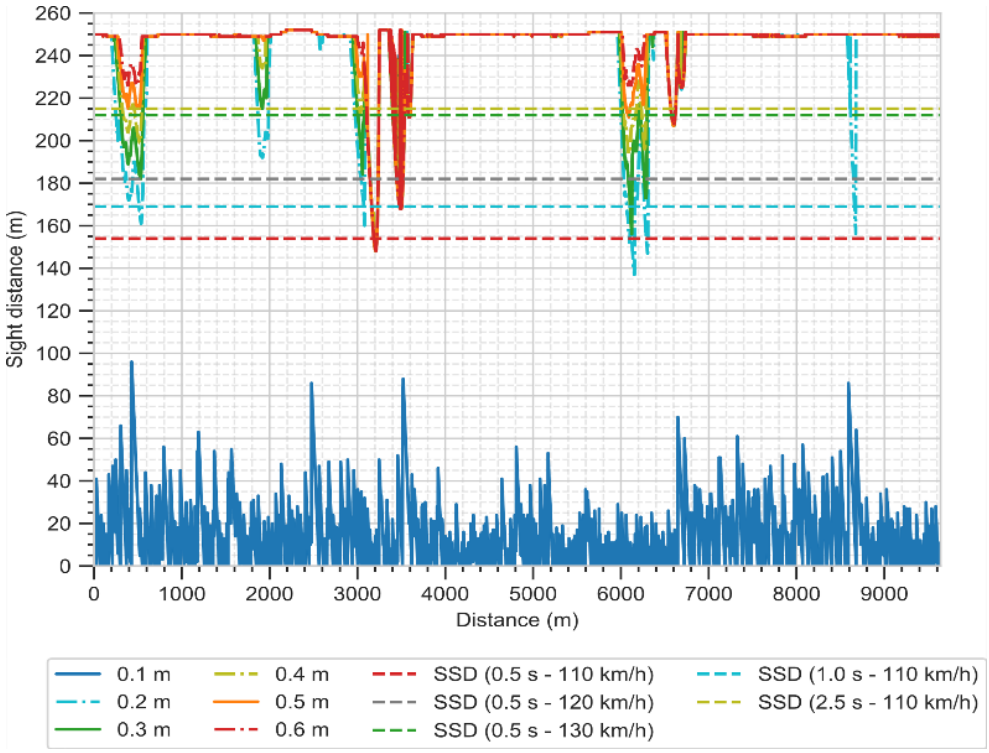
The novel methodology introduced in this work automatically identified multiple sign locations that had reduced visibility to AV sensors. Furthermore, three of the identified sign locations had already been adjusted by Alberta Transportation since the data was collected, providing evidence of this method's validity. The limitations of this method are that it does not classify signs by type or relevance to the road user, only by proximity to the road. Thus, signs irrelevant to the road user or not providing safety information were still included in the results and must be ignored or removed by manual inspection. For example, such signs may be calculated as having limited visibility to the analyzed vehicle trajectory but were only relevant to other directions of travel, which may or may not have adequate visibility of the sign. Therefore, future research should include automating signs' classification and relevance to the road user so that only relevant signs are considered for the analyzed trajectory.

5 PAVEMENT MARKING ASSESSMENT

The proposed marking occlusion detection methodology simulates AV sensors and then assesses the visibility of virtual target planes as the vehicle traverses a travel lane. Octrees are used to query occlusions to vehicle sensors within view frustums between sensors' origins and target planes. Any point detected within the view frustum is detected as an obstruction. As such, the method does not transform point clouds into volumetric voxel representations. The voxels, used in state-of-the-art research on sight distance assessment, prevent the assessment of the pavement surface visibility as any occupied voxel that intersects with a raycasted line would be detected as an obstruction, even if points are located at the lower edge of a voxel. To simplify, Figure 68 shows the difference between the two approaches at different target heights on a 10 km highway segment in Alberta, Canada. As evident from the graphs, it is impossible to feasibly perform a voxel-based raycasting assessment for target heights lower than 0.10m (the lowest voxel size commonly used in voxel-based research) (Gouda, Chowdhury, et al., 2021). In addition, voxel-based raycasting methods suffer from other limitations such as 1) the unnecessarily large processing time; 2) vertical and horizontal angular resolution of raycasting; 3) false-negative obstruction detections; 4) false-positive obstruction detections; 5) inaccurate sight distance estimates, etc. (Gouda, Mirza, et al., 2021).



a) Octree-based (Gouda, Mirza, et al., 2021)



b) Voxel-based (Gouda, Chowdhury, et al., 2021)

Figure 68 Available sight distance at various target heights on a 10 km highway segment in Alberta, Canada

The "target planes", which are essentially checkpoints for how far on the pavement surface a sensor can see at a given point, would lie flat to the ground. Axiomatically, this is because the interest in this work is the visibility of lane markings. The flowchart of the simulation-based method proposed in this section is shown in Figure 69.

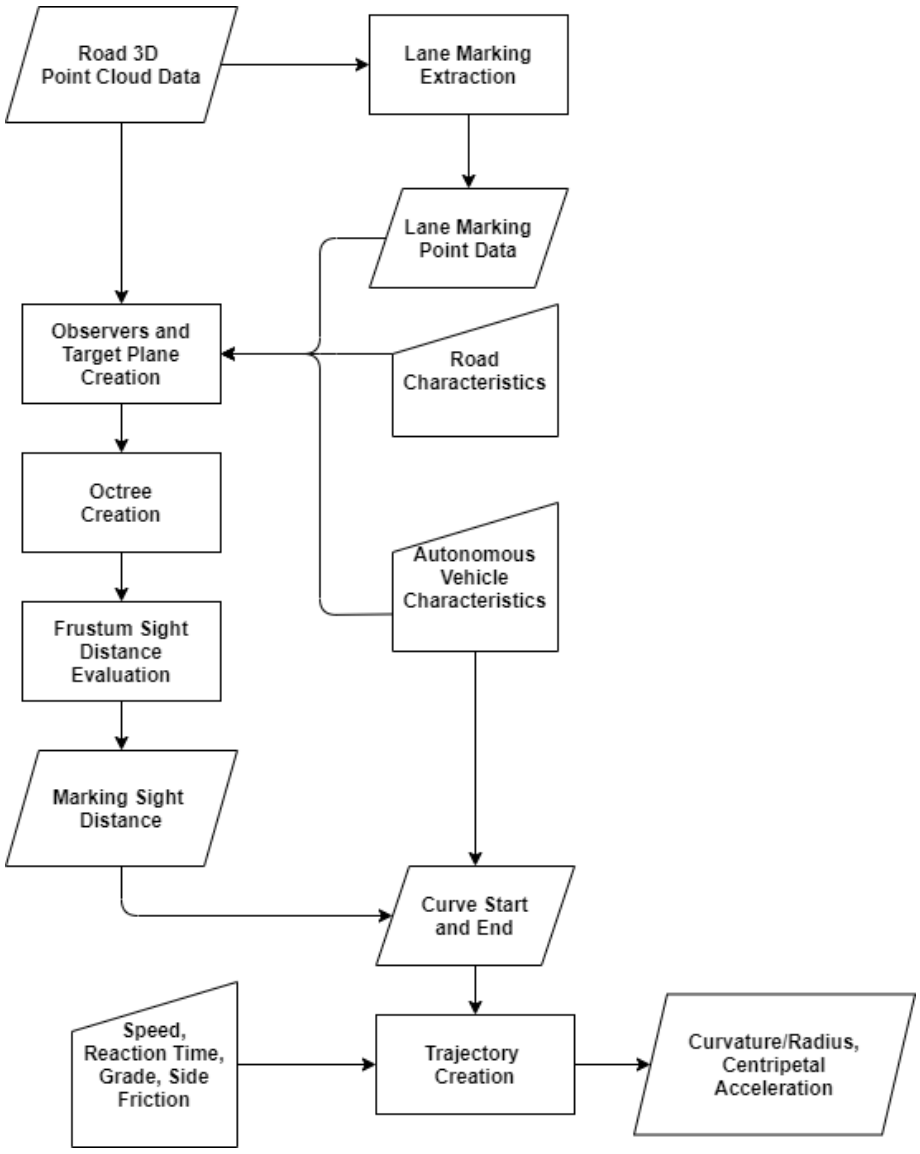


Figure 69 Flowchart of the lane marking assessment

5.1 METHODOLOGY

Pavement extraction was performed using the method explained in section 4.2.1.

5.1.1 Observer Points and Headings

To begin with, a method that gives 3D points (observer points, $o_i \in O$) representing the sensors' point of view must be devised. Additionally, a 3D reference frame consisting of a car's forward, leftwards, and upward directions must be calculated for each point along the sensors' path. With

the above information on the AV's path, directionality is fully established along a vehicle's trajectory.

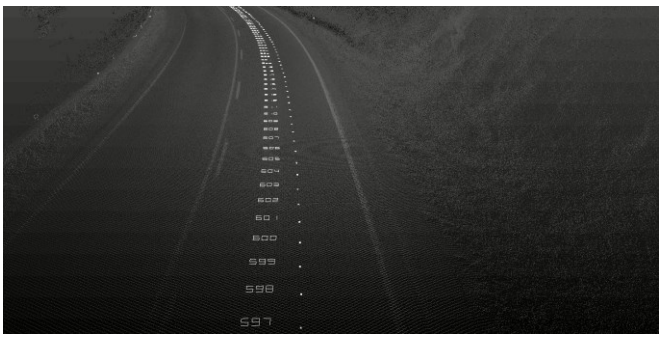


Figure 70 Labelled road points

In brief, the way this method works is by taking points directly beneath the vehicle scanners (scanning angle of zero) and fitting a smooth curve to them using regression techniques, which gives a set of road points r_i uniformly spaced at 1 m intervals, as shown in Figure 70. The smoothing utilizes the time each point was scanned and a new set of times corresponding to a given sample rate. By considering every point recorded within a certain time period, a regression smoothing can generally be achieved by forming a system of linear equations and solving by Cramer's rule. An interpolant is calculated for every time in the new set. For a given interpolant, the new time is denoted t_0, the number of points in the current window as n, the real times corresponding to said points with t_0 subtracted as a column vector $\vec{t}$, and the matrix of points themselves as R. Then,

$$\overrightarrow{t_2} = \vec{t} * \vec{t}\ (element - wise) \tag{52}$$

$$s_1 = \sum_{i=1}^{n} \overrightarrow{t_i} \tag{53}$$

$$s_2 = \sum_{i=1}^{n} \overrightarrow{t_{2_i}} \tag{54}$$

$$s_3 = \overrightarrow{t_2}^T \vec{t} \tag{55}$$

$$s_4 = \overrightarrow{t_{2_i}}^T \overrightarrow{t_2} \tag{56}$$

$$\overrightarrow{sr} = (\sum_i R_{i1}, \sum_i R_{i1}, \sum_i R_{i2}) \tag{57}$$

$$sr_t = \vec{t}^T R \tag{58}$$

$$sr_{t2} = \overrightarrow{t_2}^T R \tag{59}$$

Using these variables, we can explicitly solve for c in a least-squares solution to a quadratic.

$$det_{denom} = s_4 s_2 n + s_3 s_1 s_2 + s_2 s_3 s_1 - s_2 s_2 s_2 - s_3 s_3 n - s_4 s_1 s_1 \tag{60}$$

$$det_{num} = s_4 * s_2 * \overrightarrow{sr} + s_3 * sr_t * s_2 + sr_{t2} * s_3 * s_1 - sr_{t2} * s_2 * s_2 - s_3 * s_3 * \overrightarrow{sr} - s_4 * sr_t * s_1 \ (* element - wise) \tag{61}$$

$$interpolant = det_{num}/det_{denom} (* element - wise) \tag{62}$$

The resulting smooth path is used to obtain the correctly spaced road points through a linear interpolation.

The upwards vectors, $\overrightarrow{u_i}$, were obtained as the positive z normal of a best least-squares plane fit to other nearby points in a cube with edges of length 2 m centred on r_i. The forwards vectors, $\overrightarrow{f_i}$, were obtained by taking the derivative of the parametrization of the road points curve and subtracting its upward component. Lastly, leftwards vectors $\overrightarrow{l_i}$ are computed with the cross-product $\overrightarrow{l_i} = \overrightarrow{u_i} \times \overrightarrow{f_i}$. Each of the directional vectors are normalized to a magnitude of 1 m.

Observer points are defined as road points plus an observer height multiple of the upward vectors, $o_i = r_i$ + observer height $* \overrightarrow{u_i}$. Their purpose is to model the point of view of a typical AV. In this work, the observer height was set to 1.1 m as a conservative estimate of an AV's sensor height (Tesla, 2023).

5.1.2 Left and Right Target Planes

Targets to detect from said observer points must be defined. Since the goal is to detect how far the road surface and its markings are visible, target planes that lie flat to the road serve this purpose. A necessary precursor to defining these target planes is constructing three sets of target points for left t_{Li}, right t_{Ri}, and centre t_{Ci}.

$$t_{Ci} = r_i + u_i * \text{target height} \tag{63}$$

$$t_{Li} = t_{Ci} + l_i * \text{left lane width} \tag{64}$$

$$t_{Ri} = t_{Ci} - l_i * \text{right lane width} \tag{65}$$

This is calculated for each road point. A series of points following the trajectories of the left and right lanes are generated from segmented lane marking point data obtained by PointNet++. These series of points are then used to determine the left and right lane widths for every road point in the point cloud by finding the distance to the closest lane marking point on the left or right side accordingly. By doing so, the accuracy of the ground planes is greatly improved and consistently covers the entire width of the travel lane.

The target height constant was set to be as low and as close as possible to the road surface before yielding nonsensical results due to pavement surface roughness. The next step is defining two target planes for each road point, T_{L_i} and T_{Ri}, which represent the surface of the road. A plane, in this case, is given by an ordered set of four points, all coplanar, which define the corners of a quadrilateral $T = (p_1, p_2, p_3, p_4)$.

Each left target plane is roughly defined using two consecutive centre target points and left target points. $T_{Li} \approx \left(t_{Li}, t_{Ci}, t_{C(i+1)}, t_{L(i+1)}\right)$. An issue with this approximation is that, in general, these four points may not be coplanar, which can lead to unbounded equations later in the analysis. To correct this, the fourth point, $t_{L(i+1)}$, is projected into the plane defined by the prior three. The equations for doing this for four points (p_1, p_2, p_3, p_4) are as follows.

$$n = (p_1 - p_2) \times (p_3 - p_2) \tag{66}$$

$$p_4' = p_4 - n * \frac{(p_4 - p_2) \cdot n}{\|n\|^2} \tag{67}$$

The same equations are used to define the right target planes except with the left target points substituted with the right target points.

5.1.3 Octrees and Constraints

Lane marking occlusion evaluation is done using octrees. Octrees search for points in an intersection of half-spaces called a constraint $C = h_1 \cap h_2 \cap ... \cap h_n$. Each half-space h is a set of points satisfying:

$$h(a,b,c,d) = \{(x,y,z) \in \mathbb{R}^3 | ax + by + cz \geq d\} \tag{68}$$

Simply, a half-space is all the points "above" a plane in 3D space, given some orientation.

Arbitrary (and possibly unbounded) convex polytopes can be formed using an intersection of half-spaces. One can imagine making a cube by using six such half-spaces with three pairs of parallel planes orthogonal to each other. An explicit example of such a cube is the following.

$$\begin{aligned} C = h(1,0,0,0) \cap h(-1,0,0,-1) \cap h(0,1,0,0) \cap h(0,-1,0,-1) \cap h(0,0,1,0) \\ \cap h(0,0,-1,-1) \end{aligned} \tag{69}$$

The above corresponds to the points satisfying the following equations.

$$x \geq 0 \tag{70}$$
$$-x \geq -1 \tag{71}$$
$$y \geq 0 \tag{72}$$
$$-y \geq -1 \tag{73}$$
$$z \geq 0 \tag{74}$$
$$-z \geq -1 \tag{75}$$

Note how $-x \geq -1 \Leftrightarrow x \leq 1$. This is a 1x1x1 cube with opposing corners at $[0,0,0]$ and $[1,1,1]$.

Constraints, as defined above, are equivalent to systems of inequalities, which are typically expressed as the matrix multiplication $\boldsymbol{Ax} \leq \boldsymbol{b}$.

Given three arbitrary points in a plane u_1, u_2, u_3 not collinear and a point within the half-space u_{in}, the coefficients (a, b, c, d) for a half-space can be found using the following equations.

$$(a', b', c') = (u_1 - u_2) \times (u_3 - u_2) \tag{76}$$

$$d' = u_2 \cdot (a', b', c') \tag{77}$$

$$k = sign(a', b', c') \cdot u_{in} - d' \tag{78}$$

$$(a, b, c, d) = k(a', b', c', d') \tag{79}$$

Lastly, as $(a, b, c) \cdot u_{in} \geq d$ is wanted if it is found that $(a', b', c') \cdot u_{in} < d'$, multiplying both sides by -1 reverses the inequality. This way, the normal of the plane is guaranteed to be towards the centre of the frustum.

Both cases can be expressed as a multiplication by a factor k equal to the sign of the expression $(a', b', c') \cdot u_{in} - d$.

5.1.4 Left and Right View Frustums

View frustums (Figure 71) used are conic quadrilateral pyramids with the base as a target plane and the tip as an observer point. By taking $f(u_1, u_2, u_3, u_{in})$ to be a function that gives a half-space given three points in the plane and a point inside the half-space using the previously defined procedure, it follows that the frustum from any observer point o_i to any target plane $T = (p_1, p_2, p_3, p_4)$ is given by the following two equations.

$$m = \frac{p_1 + p_2 + p_3 + p_4 + o_i}{5} \tag{80}$$

$$\begin{aligned} C_{frustum} = f(p_1, p_2, p_3, m) \cap f(o_i, p_1, p_2, m) \cap f(o_i, p_2, p_3, m) \\ \cap f(o_i, p_3, p_4, m) \cap f(o_i, p_4, p_1, m) \end{aligned} \tag{81}$$

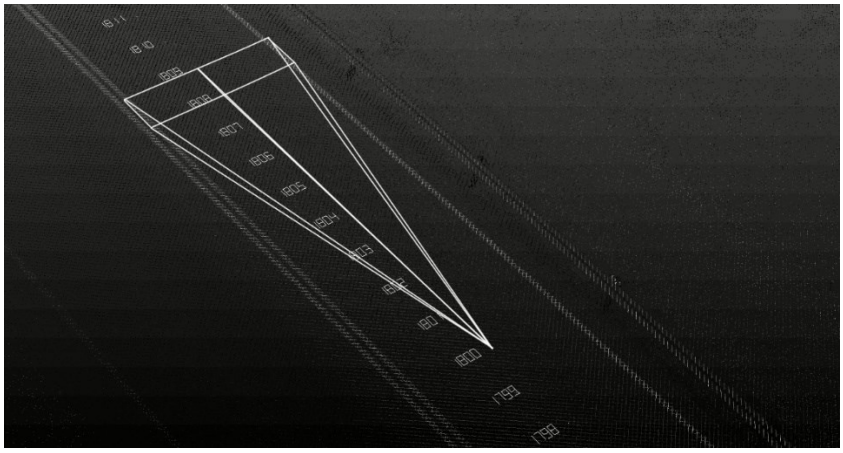


Figure 71 Left and right view frustums 9 m away from an observer

5.1.5 Evaluation

To evaluate marking sight distance for each of the n observer points, the left and right target planes are constructed up to k meters ahead. For each of these frustums, the octree is used to determine if any points are contained within the frustum. If they are, then that view is considered blocked. Otherwise, it is considered visible.

This measure of visibility is used to determine marking sight distance for each road point by checking for the nearest target plane whose view is blocked and then finding the distance from the road point to the prior plane. This number is analogous to the furthest continuous field of visibility of the road surface for an AV's sensor. Separate sight distances are produced for the left and right sides. Since road points are spaced in 1 m intervals, that is the maximum resolution of the marking sight distance evaluation. However, this is deemed sufficient for practical purposes.

5.1.6 Path Prediction

A path prediction method was devised to investigate the consequences that limited marking sight distances might have on AVs. The path depends on how fast the onboard computer can theoretically calculate it, how fast the vehicle is going, and how far the vehicle's sensors can see. From the path, measures relevant to rider comfort, such as radius, curvature, and centripetal acceleration, were extracted. The methodology was primarily applied to detected locations with a slope followed by a turn to model a particularly challenging situation to react to in time.

5.1.6.1 Bézier Curves

Many recent studies on intelligent path-planning algorithms have used Bézier curves to model smooth trajectories (Bulut, 2021; Fakhfakh et al., 2020; H. Li et al., 2019; Zheng et al., 2020). This state-of-the-art approach is computationally inexpensive relative to other path-planning methods, making it desirable for real-time situations. Given $n + 1$ control points C_i in $\mathbb{R}^2$, a Bézier curve of degree n can be parametrically represented by the function

$$B(t) = \sum_{i=0}^{n} \binom{n}{i} t^i (1-t)^i C_i \tag{82}$$

where $t \in [0,1]$.

Bézier curves have several properties that make them uniquely suited for path planning and smooth trajectories. First, they are guaranteed to touch both the first and last control points, C_0 & C_n. Second, the curve is tangent to the first edge $C_0 \rightarrow C_1$ at the first control point C_0, as well as a tangent to the last edge $C_{n-1} \rightarrow C_n$ at the last control point C_n. Lastly, the curve is guaranteed to be contained by the convex hull of all its control points.

To find the curvature at any point along the curve, the first and second derivatives, themselves being Bézier curves of lesser degree, must be found. Once found, the curvature is represented by the function

$$k(t) = \frac{\dot{x}(t)\ddot{y}(t) - \dot{y}(t)\ddot{x}(t)}{\left(\left(\dot{x}(t)\right)^2 + \left(\dot{y}(t)\right)^2\right)^{\frac{3}{2}}} \tag{83}$$

With the curvature at every point along the Bézier curve found, the radius can be calculated by taking the reciprocal of the curvature.

$$R(t) = \frac{1}{k(t)} \tag{84}$$

Given the speed at which the vehicle is travelling, v, the centripetal acceleration can be found at any point along the curve as well.

$$\propto = \tan^{-1}(e) \tag{85}$$

$$CA(t) = \frac{v^2}{R(t)} * \cos(\propto) \quad (86)$$

5.1.6.2 Control Points

Control points were identified based on Sakai et al.'s Bézier Path Planning Python module (Sakai et al., 2018). This module automatically generates a 3rd degree Bézier curve (i.e., four control points) given the location and direction of the starting and ending points, as well as an 'offset' parameter that controls how wide the turn will be. The starting point's location was determined by finding the point where marking visibility was first detected around a curve labelled "the visibility point." A simulation starting point is located upstream of the visibility point where the vehicle will be travelling at a given speed with a given reaction time. The vector between the starting and visibility points provides the starting angle.

The ending point is decided based on a two-step process. The first step is based on the marking sight distance evaluation and is the point of furthest visibility on the outside of the turn from the visibility point. The second step involves superimposing an AV's sensor detection range from the visibility point onto the point cloud and extracting the fields of vision of the AV sensors. If the point of furthest visibility is contained within any of the sensors' detection ranges, it is chosen as the location of the ending point. If it exists beyond the sensor's range, the vanishing point at the intersection of the sensor's range and the path of the road points was identified as the ending point (Andrei et al., 2022; Fakhfakh et al., 2020). The vector between the ending and 3rd road points downstream provides the ending angle. Figure 72 shows the visibility point and the point of furthest visibility.

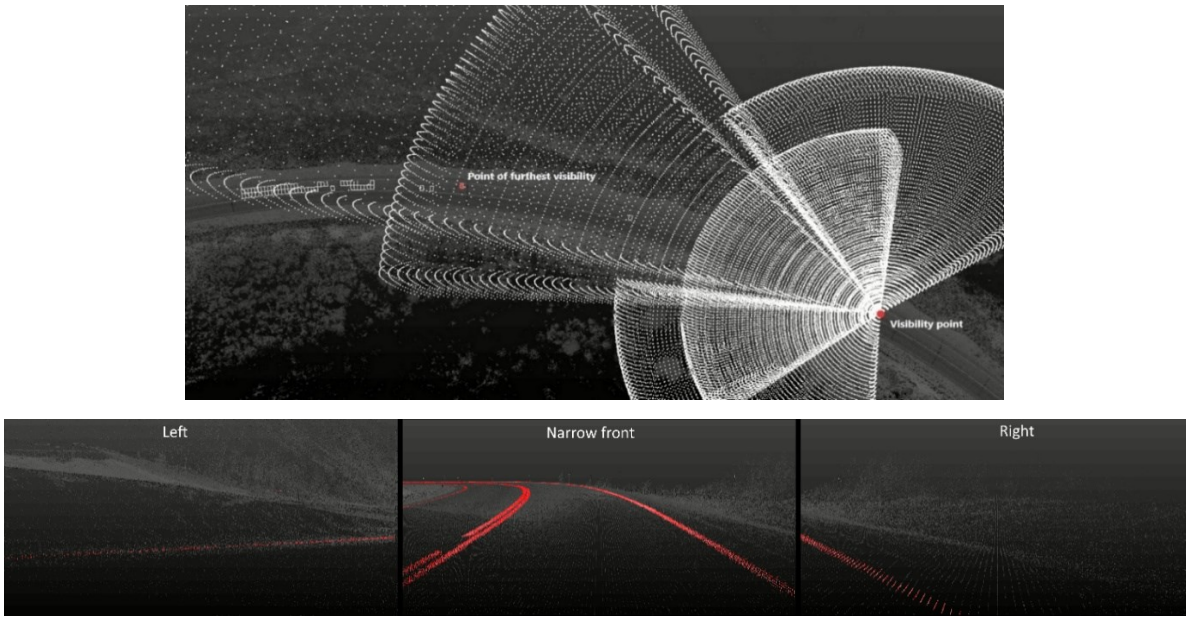


Figure 72 The detection of the point of furthest visibility from the visibility point

The offset controls the width of the turn by controlling where the middle two control points of the Bézier curve will be placed. A higher offset will make the control points stay closer to the start and end, and a lower offset will make them go farther. This results in a more shallow curve for higher offsets, while lower offsets will be more triangular and pronounced.

5.2 RESULTS

5.2.1 Data collection

The LiDAR point cloud data used in the analysis were collected in 2020 by Alberta Transportation throughout rural highways in Alberta, Canada, using Tetra Tech PSP-7000, a multifunction pavement surface profiling vehicle. The vehicle was equipped with a REIGL VMX 450 LiDAR system, capable of collecting 360° point cloud data. Point cloud densities ranged between 150 points/m^2 to 1,000 points/m^2 depending on survey speed and distance from the scanner. All surveys were done at normal traffic flow speeds, reaching as high as 100 km/h. Data was divided into 4 km segment LAS files, which were approximately 500 MB each.

5.2.2 Test segments

Marking sight distance and path planning evaluations described in the method above were done on three different curves on three different segments from rural roads located within Alberta:

1. *Highway 563:* A curve from a 4 km segment on this rural road was used for analysis (Figure 73). This segment is on the outskirts of Calgary and has the test segments' sharpest curve. The speed limit is 60 km/h, with a posted advisory speed of 35 km/h on the curve.
2. *Highway 1A:* A curve from a 2 km segment on this highway was used for analysis (Figure 74). This segment is in the foothills between Calgary and Canmore and has many turns and rolling terrain. The curve of interest is not as sharp as the first segment's curve but has a higher speed limit. The speed limit is 80 km/h, with a posted advisory speed of 65 km/h on the curve.
3. *Highway 775:* A curve from a 2 km segment on this highway was used for analysis (Figure 75). This segment is also in the foothills in the far southwest of Alberta. Much like the second segment's curve, this curve is not as sharp and has a higher speed limit. The speed limit is 100 km/h, with a posted advisory speed of 75 km/h on the curve.

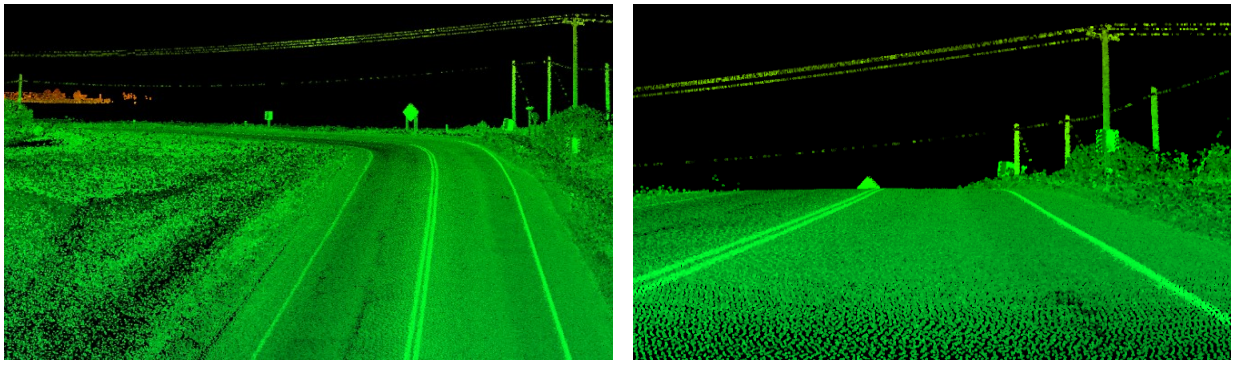


Figure 73 Curve of interest of Highway 563

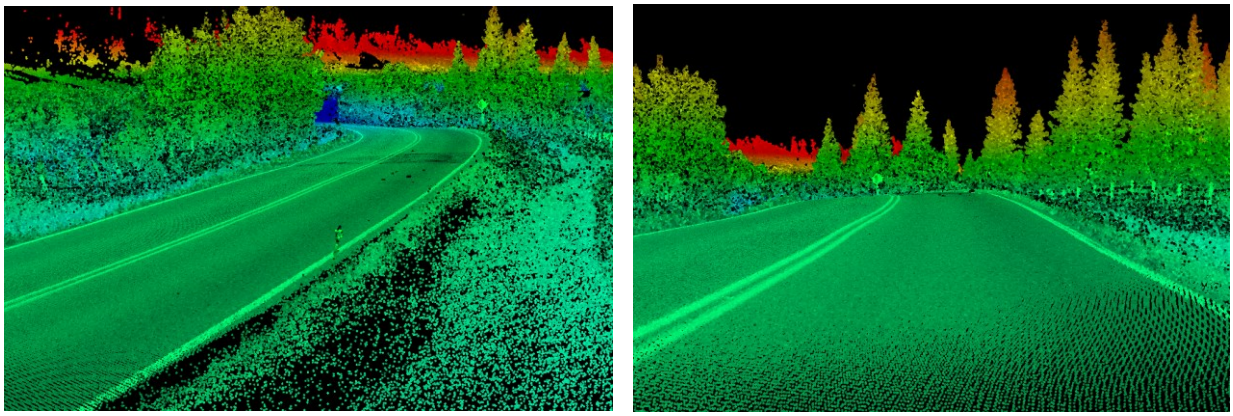


Figure 74 Curve of interest on Highway 1A

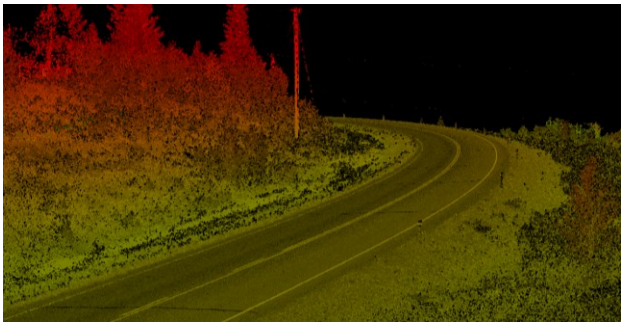


Figure 75 Curve of interest on Highway 775

5.2.3 Sensor values

To model the sensing capabilities of an AV, the following parameters in Table 11 were chosen for five different sensors based on the specifications of (Tesla, 2023). Figure 76 depicts the meaning of each of the parameters. Side and bird's eye view perspectives are given in Figure 77 of what a sensor layout with these parameters would look like.

Table 11 Sensor parameters

Sensor Set					
Sensor	$\boldsymbol{\theta_{min}}$	$\boldsymbol{\theta_{max}}$	$\boldsymbol{\gamma_{min}}$	$\boldsymbol{\gamma_{max}}$	$\boldsymbol{d}$
1	-17.5	17.5	-60	30	250
2	-25	25	-60	30	150
3	-112.5	-22.5	-60	30	80
4	22.5	112.5	-60	30	80
5	-60	60	-60	30	60

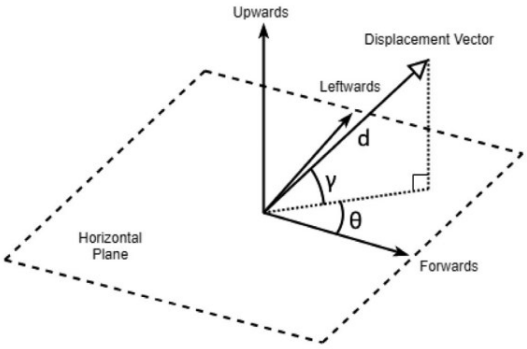
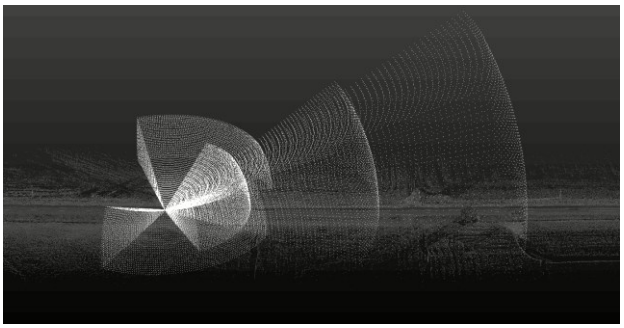
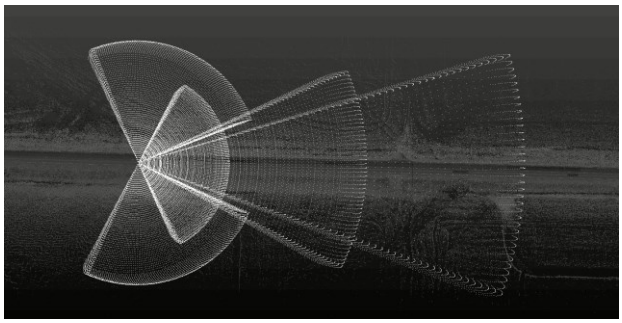


Figure 76 Sensor parameter visualization



(a): Side perspective



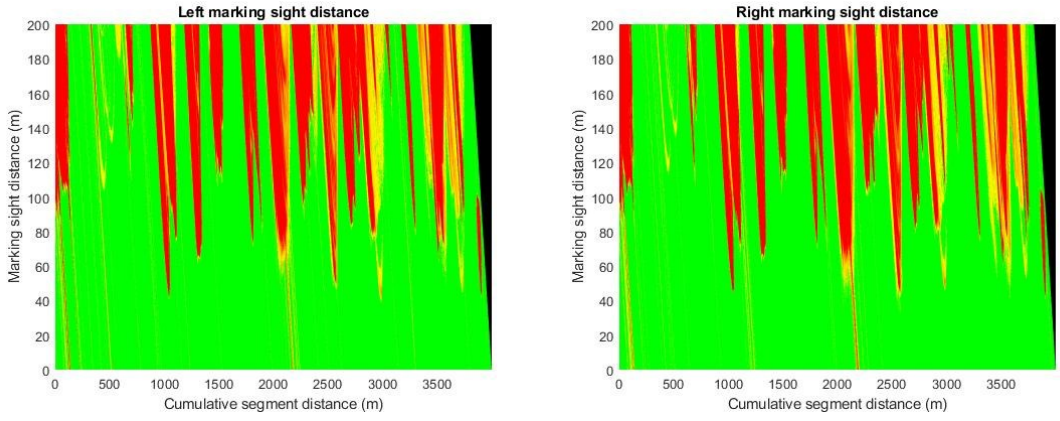
(b) Bird's eye perspective

Figure 77 Sensor layout

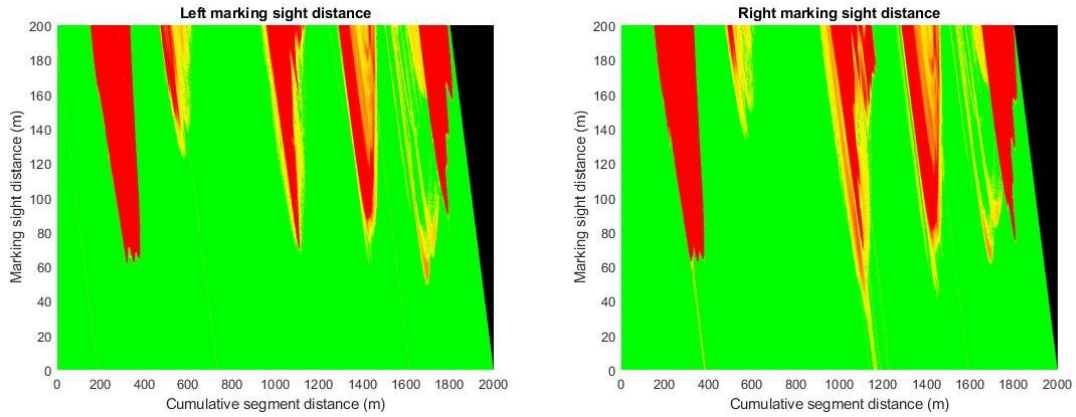
5.2.4 Lane marking occlusion evaluation

To detect locations with limited lane marking visibility, the lane marking occlusion method was performed on several highway segments in Alberta, Canada. A colour map was produced for each evaluated segment to visualize the changes and spikes in marking sight distance. The height of the green along the x-axis represents the sight distance, up to a maximum of 200 m. Segments with dips in marking visibility in the 50-70 m range were queried as potential candidates for further analysis. Manual verification of the point cloud was performed to narrow down segments for further analysis. Segments with a vertical crest curve followed by a tight horizontal curve were singled out as situations where an AV would have difficulty reacting to the lane markings in time while calculating a trajectory. At tested locations, the AV sensor view of the pavement after the crest was continuously hindered along the distance upstream of the crest.

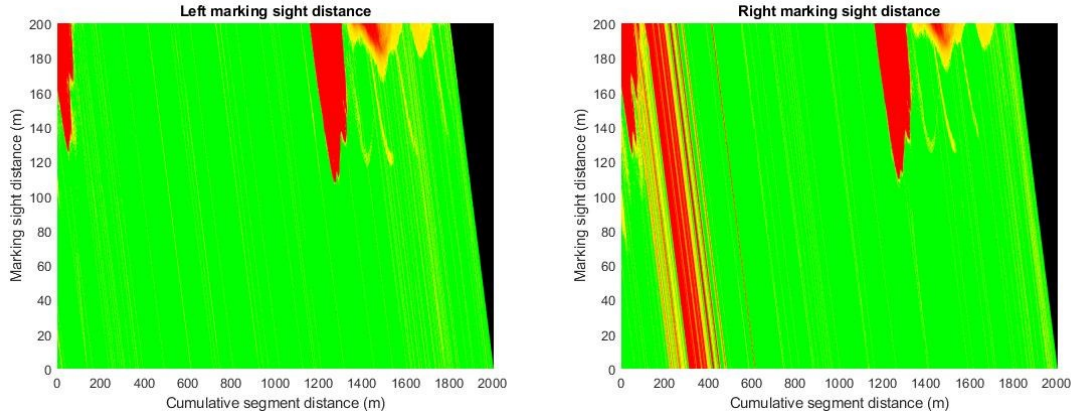
Figure 78 shows the colour maps for the test segments.



(a) Highway 563



(b) Highway 1A



(c) Highway 775

Figure 78 Colour maps of the three segments' sight distance

5.2.5 Path Prediction

The curves of interest on the chosen segments were then analyzed with the path prediction algorithm detailed in the methodology. For each curve, seven different speeds at five different reaction times were considered for 35 unique paths. The reaction times were the same across all the curves, going from 0.2 s to 1.1 s in 0.3 s increments, and are intended to represent the time between the first detection of the lane markings and mechanically starting the expected manoeuvre (Gouda, Chowdhury, et al., 2021; Gouda, Mirza, et al., 2021; Khoury et al., 2019; Khoury & Amine, 2019; Mcdonald, 2018; Saeed, 2019; S. Wang & Yu, 2019). Figure 79 shows a sample of the paths from Highway 563. The ranges of speeds analyzed encompassed both the posted recommended speed for that curve and the actual speed limit of the road. Table 12 shows the

speeds used for each segment. The green highlight denotes the recommended speed, while the orange highlight denotes the actual speed limit.

Table 12 Speeds analyzed for each segment (km/h)

Highway 563	Highway 1A	Highway 775
30	50	60
35	60	70
40	65	75
50	70	80
60	80	90
70	90	100

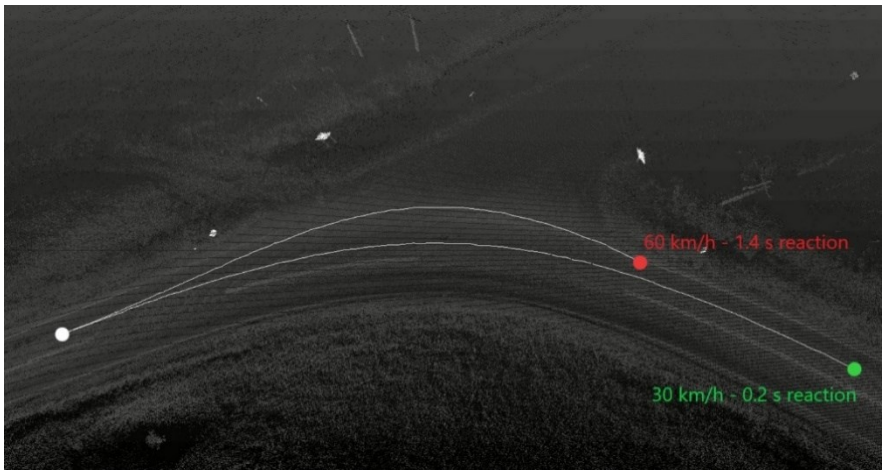


Figure 79 Highway 563 sample trajectories

5.2.6 Radius

The radius of each path at each point along its trajectory was plotted against T, the same dimensionless parameter used in the general form of a Bezier curve, which can considered as a stand-in for time. Thresholds for the minimum turning radius under different conditions were overlayed on the plots as a measure of the severity of the turn. The minimum required radii were determined using the AASHTO equation.

$$R_{min} = \frac{v^2}{127(e + fs)} \tag{87}$$

where v is the speed in km/h, e is the superelevation, and fs is the side friction factor, which varies based on the condition of the road (dry, wet, ice, etc.) as outlined in the AASHTO guidelines (AASHTO, 2018; Ghandour et al., 2010; NCHRP, 2009). The grades of the road surfaces were extracted using a quadratic surface model and a set of arithmetic operations to determine the superelevation. Figure 80, Figure 81 and Figure *82* show the radii for all paths and all reaction times on the curves of interest for Highway 563, Highway 1A, and Highway 775, respectively.

5.2.7 Centripetal acceleration

Centripetal or lateral acceleration is a simple extension of the above results for radius but is more relevant in the discussion of driver and passenger comfort. It has been plotted for each curve in the same fashion as the radius plots. Different thresholds of the centripetal acceleration of $1.8\ m/s^2, 3.6\ m/s^2, 5.0\ m/s^2$ have been added, representing comfortable, relatively uncomfortable, and uncomfortable levels of lateral acceleration, respectively. Values exceeding $5.0\ m/s^2$ are considered unbearable (Xu et al., 2018). Figure 83, Figure 84, and Figure 85 show centripetal acceleration at each point along the curves.

5.3 DISCUSSION

5.3.1 Radius

Examining the results of the trajectory radius, the first two curves show a distinctly U-shaped graph, regardless of speed and reaction time. This indicates that the severity of the turn is worst around the middle of the turning manoeuvre and softer around the ends when the vehicle is straightening out. The higher speed graphs, however, decreased at the end of the predicted trajectory and were especially pronounced at slower reaction times. This results from the vehicle needing to realign itself with the road by continuing to turn farther than its desired finishing angle before turning in the opposite direction and straightening out, as shown in Figure 86. The third curve appears to be an outlier, with its shape resembling the second half of the U-shape shown in the first two graphs. One possible explanation could be that the point of further visibility only

comes into view when the shape of the road is already turning at its sharpest, providing little time for the vehicle to ease into the turn.

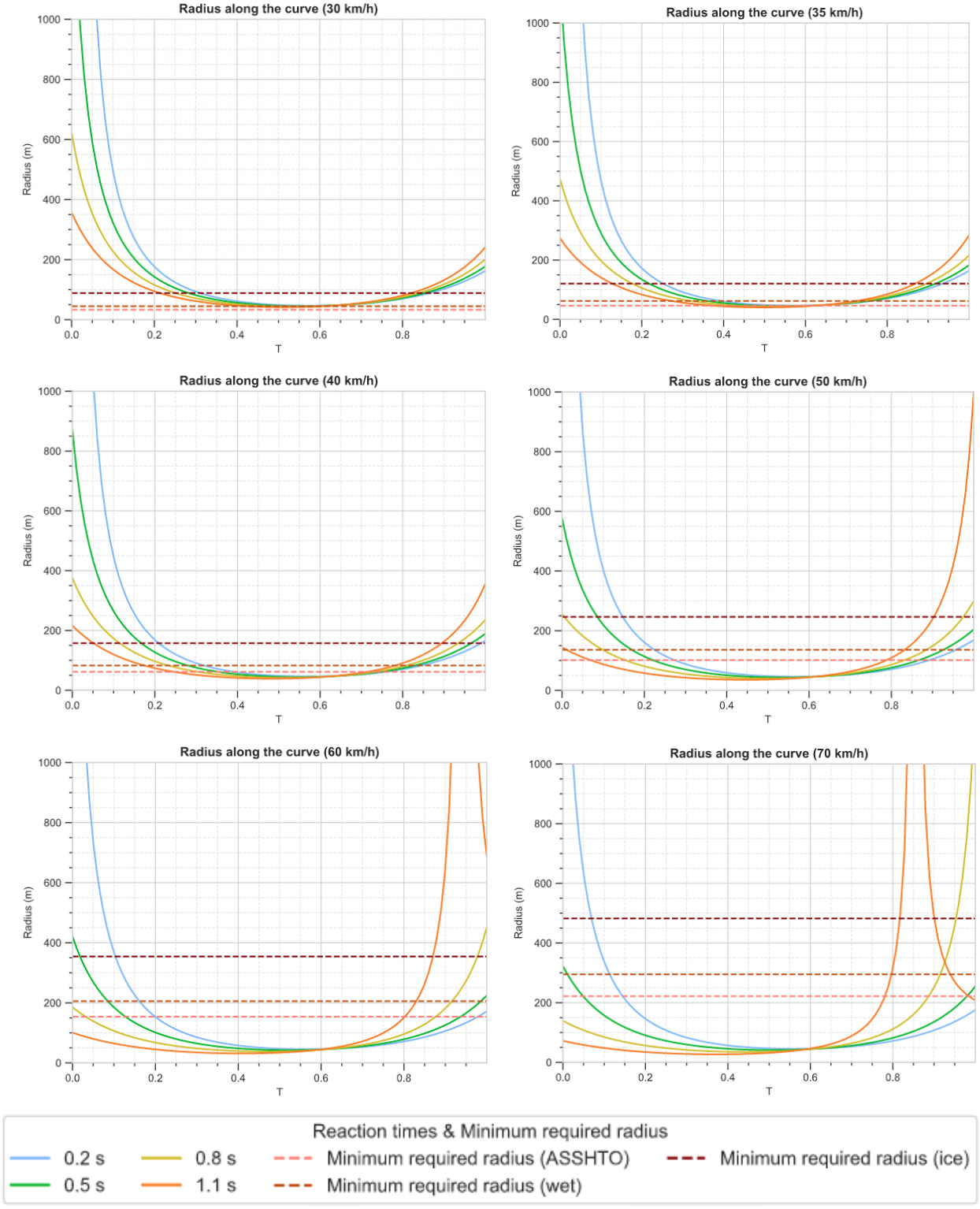


Figure 80 Radius results for Highway 563

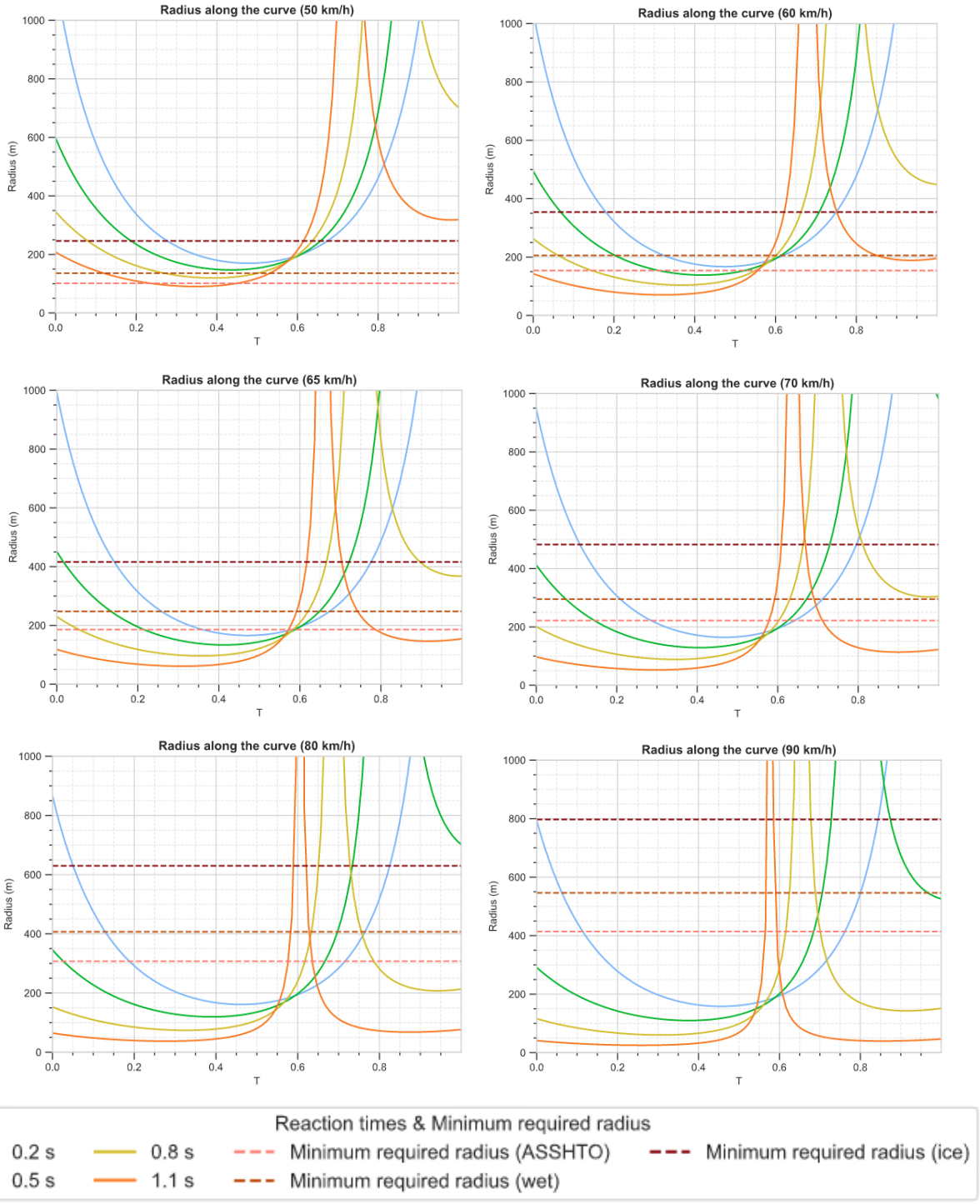


Figure 81 Radius results for Highway 1A

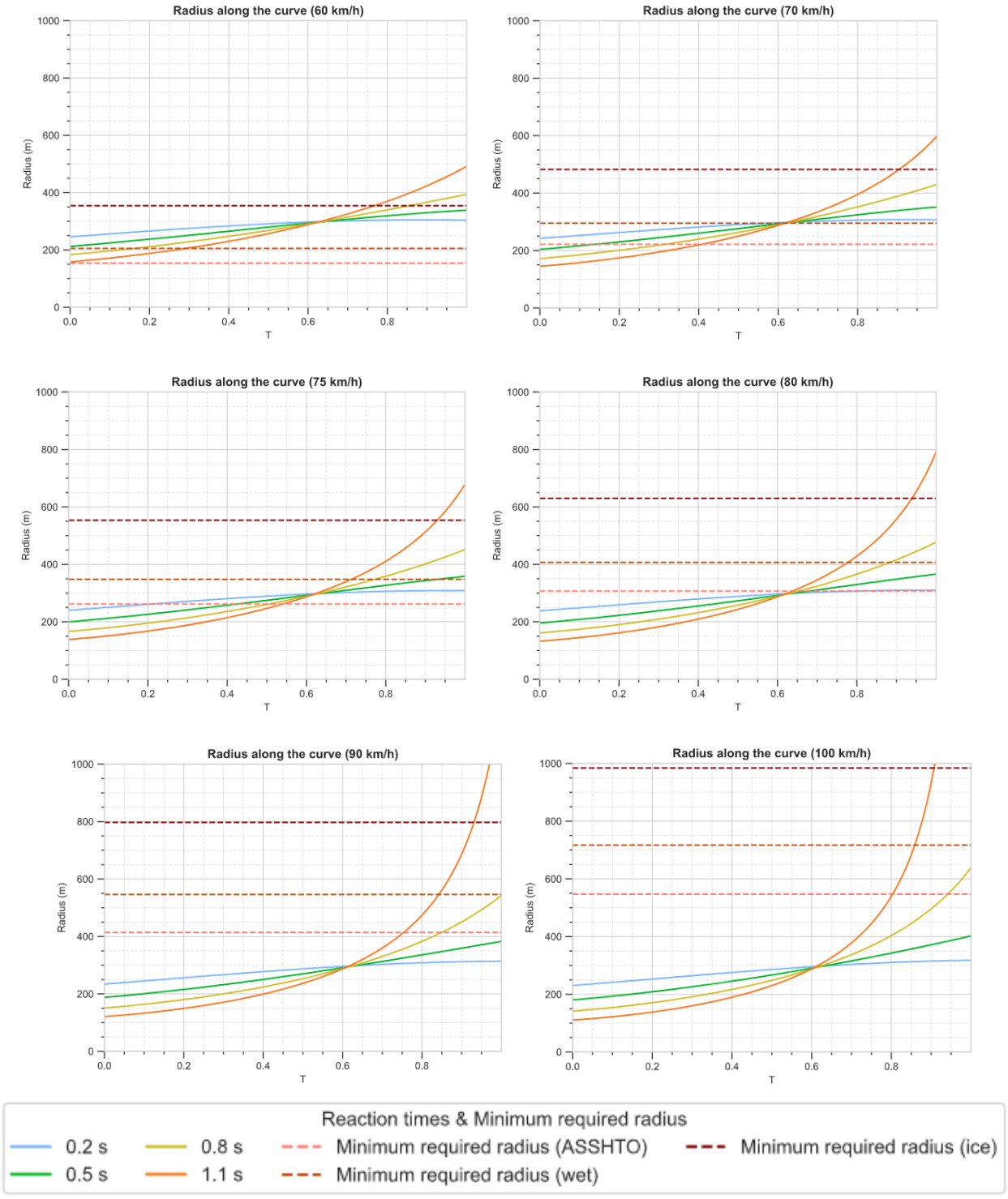


Figure 82 Radius results for Highway 775

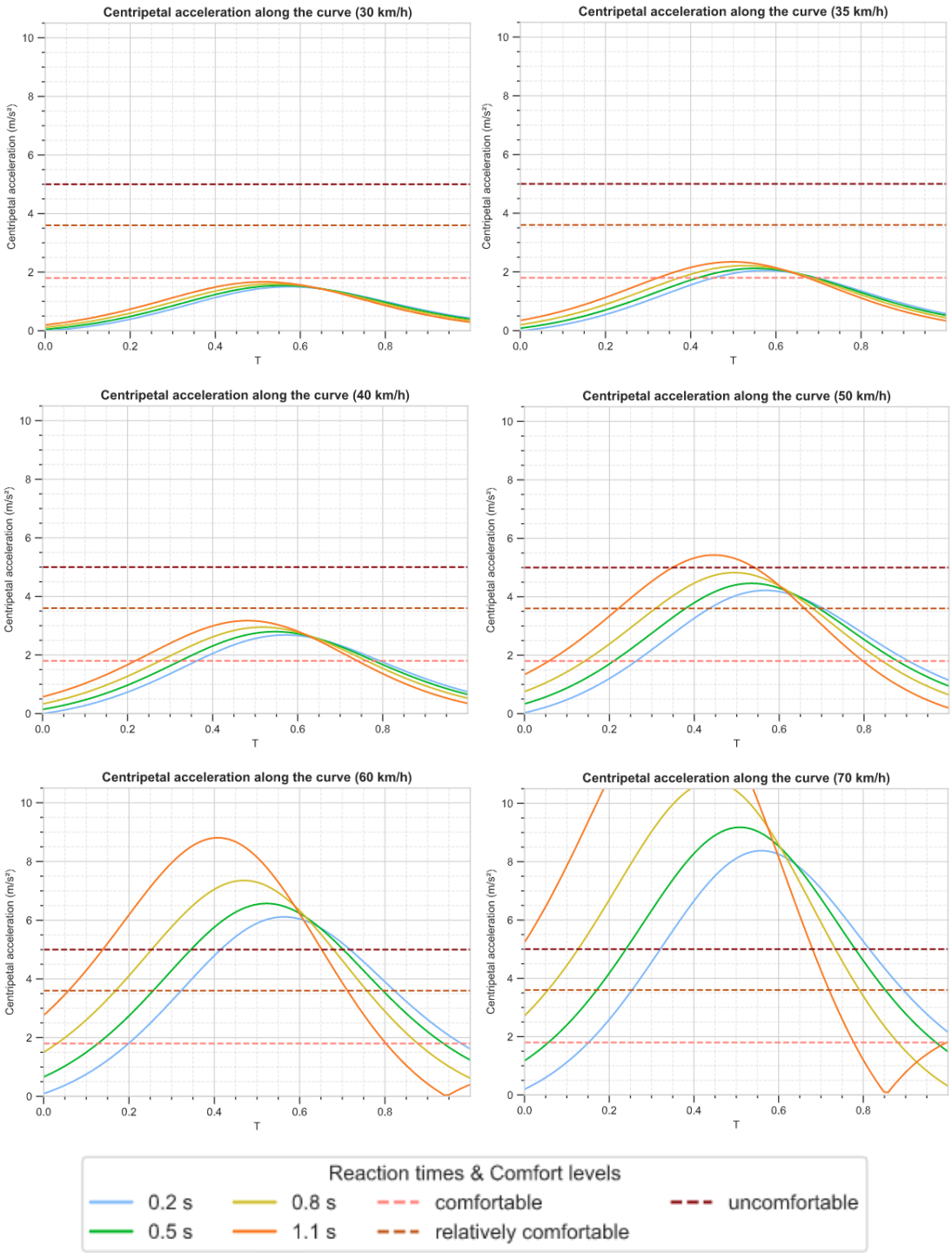


Figure 83 Centripetal acceleration results for Highway 563

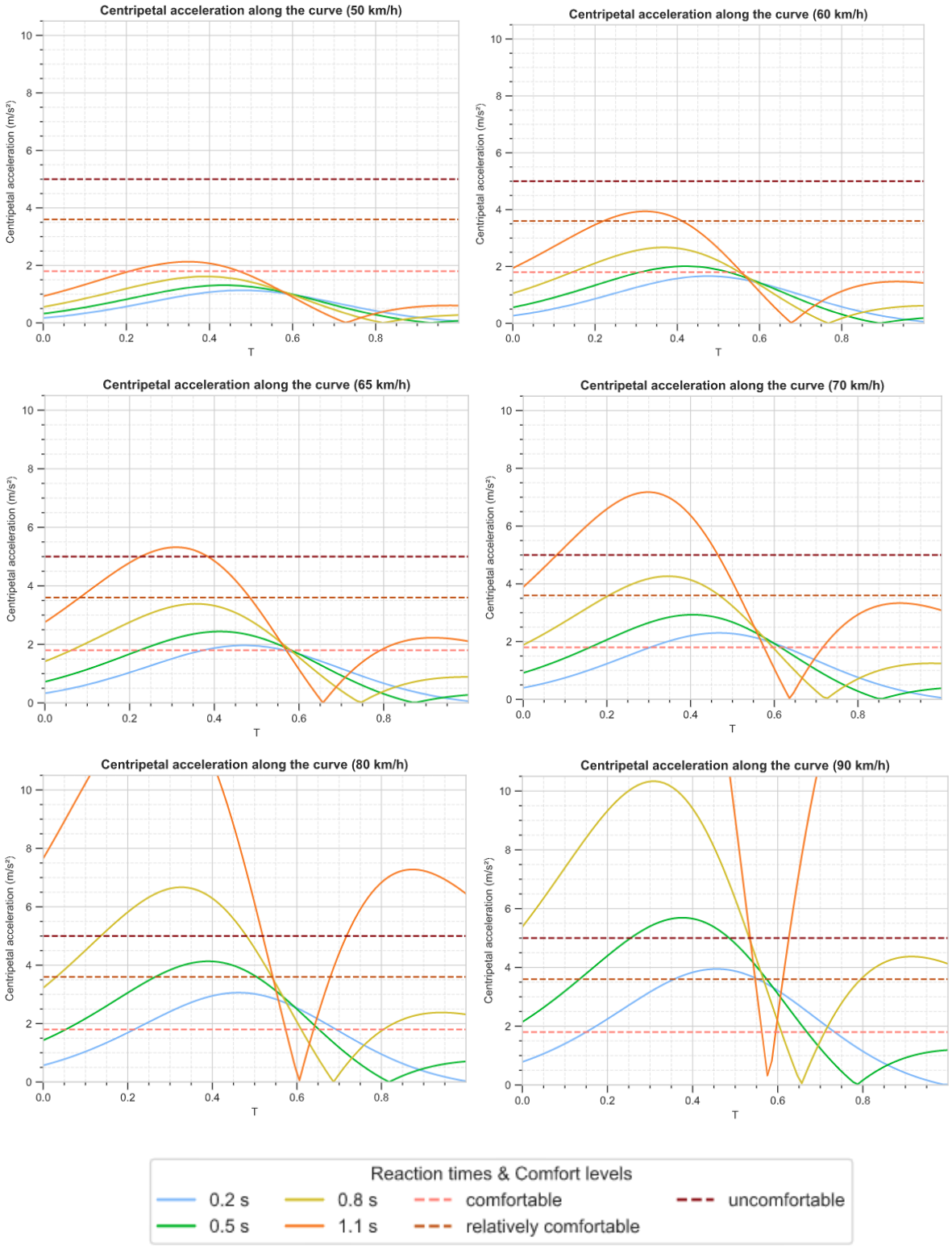


Figure 84 Centripetal acceleration results for Highway 1A

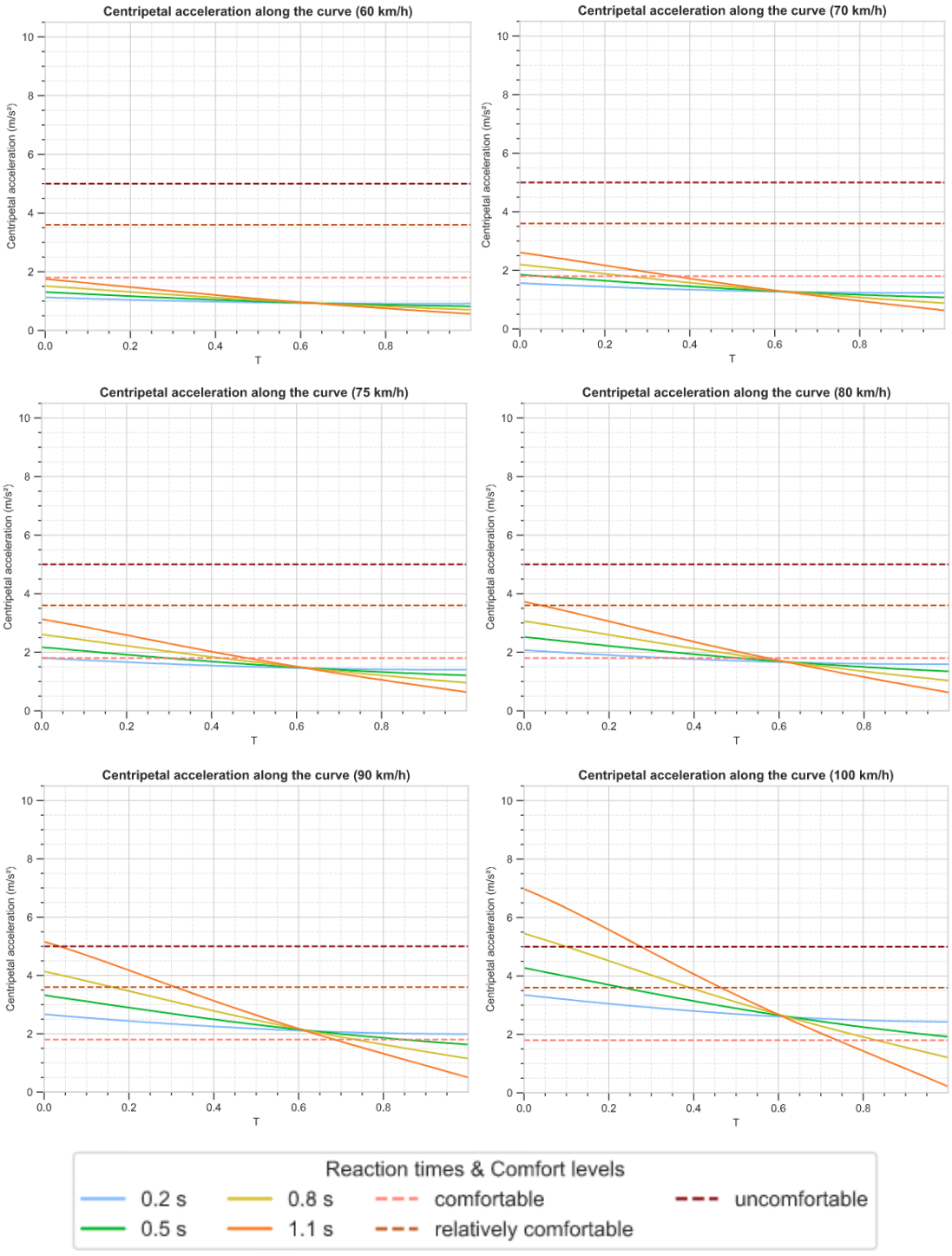


Figure 85 Centripetal acceleration results for Highway 775

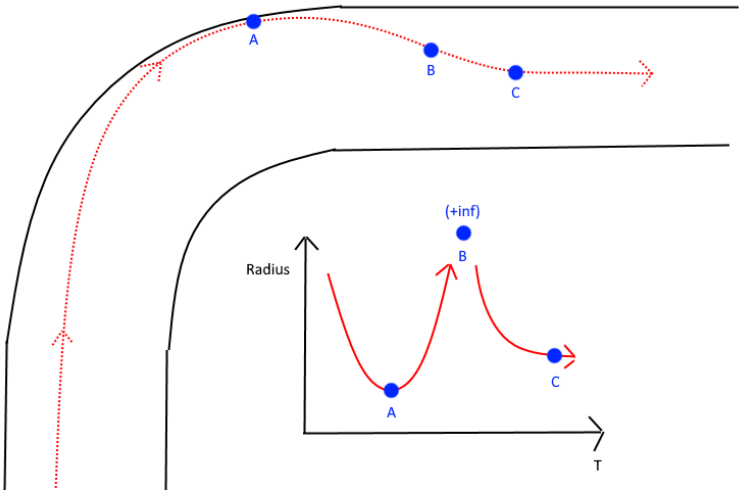


Figure 86 Oversteering correction

The suitability of the turning radius can be determined by comparing it to the plotted thresholds. With this comparison, most speed and reaction time combinations fall below the minimum recommended turning radius in all conditions. Only trajectories below the recommended speed limit show at least one reaction time fringing on the AASHTO conditions threshold at their worst point. The radius recommendations in worse conditions (i.e., wet, ice) are largely unmet, regardless of speed and reaction time. This suggests that extra considerations for minimum turning radius or maximum speed limit need to be made in situations where sight distance is significantly reduced. This is especially important in locations where the local climate frequently and predictably leads to adverse road conditions.

5.3.2 Centripetal acceleration

The centripetal acceleration graphs are simply the reciprocal of the radius graphs multiplied by a constant. It follows that when the curve is at its sharpest, the radius is at its lowest while the centripetal acceleration is at its highest, hence the inverted U-shape of the centripetal acceleration graphs.

Rider comfort thresholds are another way to judge the suitability of the road geometry. Between the first two curves, the matching of speed to curve position relative to thresholds is not

as consistent as with radius. For example, at the recommended speed for the first curve, all reaction times show centripetal acceleration values that violate the comfort threshold. At the recommended speed for the second curve, the centripetal acceleration values are evenly spread between relatively comfortable and unbearable levels. This can be explained by the centripetal acceleration graphs having constant thresholds, while the radius graph thresholds scale with the speed of the vehicle.

The uncomfortable and unbearable thresholds are breached at and below the posted speed limit for two curves at the slowest reaction times. Meanwhile, the fastest reaction times are generally slightly above comfortable levels at the recommended speed limit. This observation reinforces the idea that as the computational abilities of AVs evolve through more powerful hardware and more efficient algorithms, their reaction time will decrease, allowing for increases in speed while maintaining acceptable levels of rider comfort.

5.4 CONCLUSIONS

This thesis presented a method to identify road locations where lane marking visibility is compromised. Using LiDAR point cloud data, the vehicle's trajectory was extracted, and with the help of octrees, iterative queries on how far down the road segment the lane markings can be seen at each point along the road can be made. This method was used to identify segments with inclines followed by curves, which pose a particularly challenging situation for AV sensors relying on line-of-sight sensor-based monitoring.

Three highway segments in Alberta - a 4 km segment on Highway 563, a 2 km segment on Highway 1A, and a 2 km segment on Highway 775 - were identified as having such a problem and investigated further for the consequences of having substandard lane marking conditions. Different reaction time and speed combinations were used to generate potential paths an AV might take on the selected segments. Bézier curves were chosen as the AV path generation algorithm due to their computational efficiency for real-time purposes and useful geometric features. Radius and centripetal acceleration graphs were then generated at each point along the curves and compared against key thresholds to assess the severity of the curves.

Results show that the analyzed most predicted trajectories on the chosen curves were potentially dangerous or uncomfortable for the passengers. Regarding minimum recommended radius standards, all trajectories fell below the ideal conditions threshold at their worst point at the

recommended curve speeds, regardless of reaction time. In unideal road conditions, the disparity between the trajectory's radius and the recommended radius grows to unsafe levels. Regarding centripetal acceleration, threshold comparisons indicate that rider discomfort was generally observed at around the recommended curve speeds and is particularly egregious at higher reaction times.

These results indicate a need for an assessment of roadways for their AV deployment readiness, particularly to ensure ample room for AVs to view lane markings and safe and comfortable trajectories. The marking sight distance calculation methodology can be used as a tool for such assessment, as well as motivation for taking a proactive approach in updating the roadway design guidelines and implementing countermeasures (e.g., V2I connectivity improvement, adjusted advisory speed limits, etc.) to better meet the needs of AVs. The conclusions of this chapter agree with the findings of the previous field testing projects and provide a novel method to proactively identify substandard locations for AVs (Tak et al., 2022).

6 CONCLUSIONS

6.1 RESEARCH SUMMARY

This thesis presents methods for assessing road design features performance under AV driving scenarios based on the recommendations of recent initiatives and studies. A holistic simulation-based approach has been adopted where AV sensor sets and autopilot parameters are modelled in a digital replica of existing infrastructure collected using LiDAR data. The digital scans collected are ultra-dense point cloud data collected with high-precision professional-grade scanning systems.

The developed methods perform the following:

- Semantic segmentation of point cloud data to extract features of interest
- Assessment of occlusions due to the physical infrastructure on the road and roadside environment
- Extraction of road and roadside grades using a quadratic surface model
- Combining the occlusion and grade metrics using YCbCr scoring spaces to facilitate performance evaluation
- Assessment Traffic signs placement within the field of vision of AV sensorsPavement markings occlusions from the FOV of vehicle sensors

The developed approach simulates virtual AV sensors' field of vision, originating from different points in the travel lane and phantom or known targets on the voxelated road and roadside environment. Unlike the common use of a single data structure, creating a combination of Octrees and Voxels is designed in a relationship that acts similarly to what could be best described as a modified voxel-octree. Convex hulls and Octrees are then used to score voxels based on the visibility of every target in view convex polyhedron. A quadratic surface model of the road and the roadside surface is used to generate normal vectors. Then, arithmetic operations were applied to map road/roadside gradients. A semantic segmentation variant of PointNet++ was developed to classify points into pavement, signs and roadside points. Finally, design metrics are mapped based on a unique YCbCr score space value. An improved state-of-the-art raycasting approach was

developed and utilized to perform a general road/roadside analysis. A comparison with the proposed methods and manual assessment was carried out for validation.

Several highway segments in Alberta, Canada, were used for testing the developed methods. In general, it was found that the proposed methods outperform the raycasting methods in terms of performance and processing time at different voxel grid sizes and observer spacing. Besides, the proposed approaches have other advantages, such as 1) the ability to account for different sensor origins in the travel lane, 2) avoiding false clipping in voxel-based raycasting methods, 3) the possibility to generalize to different shapes of sensor sets and targets required by a user, and 4) accurately detects occlusions regardless of the voxel size used, which is a significant limitation in previous studies. A framework of IOOs design improvement countermeasures is presented to assist in making performance-based decisions regarding smart digital and physical infrastructure upgrades. Different case studies, where substandard locations for AVs were identified, are presented based on the proposed methods. These case studies include:

- Clear zones and roadside safety assessment
- Wildlife sight distance assessment
- Pavement surface assessment
- Pavement markings assessment
- Traffic signs assessment

A framework of IOOs design improvement countermeasures is presented to assist in making performance-based decisions regarding smart digital and physical infrastructure upgrades.

6.2 READINESS IMPROVEMENT COUNTERMEASURES

A framework of collected readiness improvement solutions is presented in Figure 87. The outputs of this research can help stakeholders (e.g., IOOs and automotive industry professionals) apply countermeasures to improve readiness for CAV deployment. Countermeasures in the figure are categorized with four different types of labels. The first category denotes who the countermeasure concerns, whether it be IOOs (“A”), AV professionals (“B”), or both (“C”). The second category denotes if the countermeasure should be enacted in the early stages of deployment (mixed traffic) (“1”), after full deployment (“2”), or in all stages (“3”). The third category denotes whether the

countermeasure aims to prevent all collisions ("I") or wildlife-vehicle collisions ("II"). The last category denotes whether the countermeasure requires network coverage ("**"), is independent of network coverage (no marking), or is enhanced by network coverage ("*"). It is worth noting that network coverage is not expected to be fully available in the early stages of AV deployment, especially on rural highways (Wsp, 2020).

Wildlife-vehicle collision mitigation measures should consider animal traffic in a region. It is worth mentioning that wildlife animal traffic is significant in Alberta and accounts for 50% of the yearly collisions in the province. As such, wildlife-vehicle collision mitigation measures would be crucial in Alberta. In addition to solutions available in references previously cited in other chapters, several more countermeasures were presented in Figure *87* (3M, 2021; C. Adam et al., 2021; Baigas & Gunson, 2021; Druta & Alden, 2020; Gibson, 2017; Government of Canada, 2022; Pagany, 2020; Weerasinghe et al., 2021).

Multiple technologies must be fused together to provide information redundancy to CAVs from the roadway environment (3M, 2023; Cui et al., 2019; Z. Liu et al., 2020; US DOT FHWA, 2021). Combining these technologies would create a safer environment where CAVs can operate, making CAVs more reliable and mitigating potential failures. A recent major US DOT FHWA collaborative initiative between IOOs and AV industry professionals (US DOT FHWA, 2021) concluded that the maintenance of high-quality and visible lane marking, including removing obstructions to improve lane marking and pavement surface visibility, is one of the most acknowledged measures to improve highways CAV readiness and compatibility by IOOs.

6.3 RESEARCH CONTRIBUTIONS

For AVs to realize their full potential, challenges facing their deployment in as many situations as possible must be addressed. Research has shown that a complete assessment of road and roadside environments, especially on highways, has yet to be achieved. Such an understanding could vastly expand the domain in which AVs could safely operate. The contributions of this work are as follows:

- Proposes a novel simulation-based method for assessing existing road and roadside design based on AV sensor set design and autopilot parameters using ultra-dense point cloud data.

- Proposes a proactive method for identifying substandard locations for AVs on both the road and roadside environment and directly addresses recommendations of recent studies.
- Applies to ultra-dense point clouds collected either by probe mapping vehicles or crowdsourced from CAV fleets.
- Provides a holistic and novel solution to several limitations and gaps presented in state-of-the-art CAV and smart infrastructure research and practice.
- Can map long and short-range physical infrastructure occlusions to vehicle sensors in the general road and roadside environments.
- Supports a data-driven deployment of roadside countermeasures to mitigate AV collisions.
- The method can also generate sensor FOVs from different origins on the travel lane.
- Proposes a method to create performance-based speed profiles for AVs, unlike standard methods that rely on either traffic signs designed for human drivers (AASHTO, 2018) or parsing human GPS speed profiles from naturalistic data and using it for AV systems training to overcome local AV system limitations (Anastassov et al., 2017; Claussmann et al., 2020; Rodrigues et al., 2018; TomTom, 2023; Xinli et al., 2016).
- Help stakeholders (e.g., IOOs, automotive industry professionals, etc.) make informed decisions regarding smart infrastructure upgrades and investments.

6.4 LIMITATIONS AND FUTURE RESEARCH

While the work presented in this thesis overcomes limitations in previous studies, there is room for improvement, including i) assessing the performance of more road design features such as the decision sight distance (DSD) and passing sight distance (PSD) based on AV parameters, ii) Improving the processing time of the simulation, iii) Improving the semantic segmentation accuracy and classification capabilities and iv) automating the adjustment process of existing designs through the use of generative-design.

For the extraction of road features, the limitation of the method is that it does not classify features by type. For future research, applying an explainable artificial intelligence (AI) approach

can help understand the relationship between changes to a deep neural network architecture and the model output performance; however, current developments for the integration of explainable AI in 3-D data remain fairly unexplored (Matrone et al., 2022). For instance, signs irrelevant to the road user or not providing safety information were still included in the analysis but will need to be ignored by manual inspection. For example, such signs may be calculated as having limited visibility to the analyzed vehicle trajectory but were only relevant to other travel directions, which may or may not have adequate visibility of the sign. Therefore, future research should involve automating the classification of sign types.

Furthermore, using generative design principles can enhance the process of modifying pre-existing designs. The endeavour to identify the most optimal solutions that align with specific design criteria and constraints necessitates the exploration of diverse design alternatives. As a methodology, generative design involves using algorithms and computational techniques to create and appraise design proposals. Integrating generative design, leveraging algorithmic and computational capabilities, emerges as a promising strategy for tackling the complexities inherent in highway design. This approach empowers engineers to scrutinize many scenarios and pinpoint efficient, cost-effective solutions.

Using vehicle intelligence data, a geographic information system (GIS) tool can be used to screen/rank locations with safety issues. For instance, locations with AV failures, harsh braking, excessive lane changes, or disengagements can be queried for further analysis. Using point cloud data, the proposed methods can be used to analyse these screened locations to better understand their performance and make necessary decisions to improve their safety. Such data can be obtained from government agencies that collect AV failures data, such as the California Department of Motor Vehicle Collisions or vehicle intelligence companies that crowdsource data from sensors mounted on AVs (California DMV, 2023; Dadvar & Ahmed, 2021). The high frequency of recorded events at a location requires a safety audit of such cases by IOOs. This approach connects this work with the safety management process adopted by municipalities in their road safety plans.

It is worth noting that the different thresholds of the centripetal acceleration representing comfortable, relatively uncomfortable, uncomfortable, and unbearable levels of lateral acceleration are not based on AV data (Xu et al., 2018). As such, it is recommended for future

research to study the relationship between centripetal acceleration and comfort levels while accounting for the difference in the location of passengers in an AV.

Optimizing the locations of roadside sensors to improve coverage is another topic for future research. The developed method can be used to assess sensor coverage from different locations on the roadside. Furthermore, the maximum coverage possible using more than one sensor can be explored to optimize the use of limited resources available for transportation agencies. Roadside sensors (e.g., cameras, LiDAR, etc.) can be used for different purposes, such as scanning the roadside occluded areas to send information about potential road users to connected vehicles. In some cases, these sensors are synchronized and coordinated. Optimizing the relationship between their coverage and number is an emerging issue in the literature (Y. Ma et al., 2022).

The huge size of point cloud data is another limitation that makes the storage, transfer and processing of such data a challenge. Collected point cloud data during scanning are usually stored on the collection platform with huge sizes in terms of terabytes. Further research is needed to find ways to reduce the file sizes and allow a cloud-based solution to transfer the data in real-time or near real-time that provides access by different teams. This is important as identifying road safety issues and the timely deployment of countermeasures is crucial for public safety.

Due to the high cost of professional-grade scanning systems, researchers and practitioners have started exploring the possibility of using low-cost scanning equipment to collect LiDAR data, similar to those used in HD mapping for AVs (NVIDIA, 2023). As such, applying the proposed methods on low-cost scans is suggested to investigate the accuracy and performance of the output. In addition, the proposed work can help build HD maps for autonomous vehicles by mapping design information on the road and roadside environments and supporting the AV system with digital semantic information to improve its performance.

The high performance of the octree-based evaluations shed light on the possibility of extracting other road geometric features using this approach. For instance, mapping highways vertical clearance and the structure of bridges/tunnels could be done at unprecedented processing speeds and accuracy. Querying vegetation encroachment locations is also important in recent studies that can benefit from this method.

As this work focused on the geometric design aspect of roadways, future research on the structural performance of pavements under AV driving scenarios is suggested. In addition, the impact of AV deployment on pavement design guidelines is worth further investigation. Finally, combining design performance metrics into a readiness index for roadway segments can assist stakeholders in making planning and maintenance decisions for existing road infrastructure. This way, mapping performance metrics on a GIS with AV failure information can provide a holistic view of the existing infrastructure performance and the relationship between readiness indices and other road geometric and AV failure parameters.

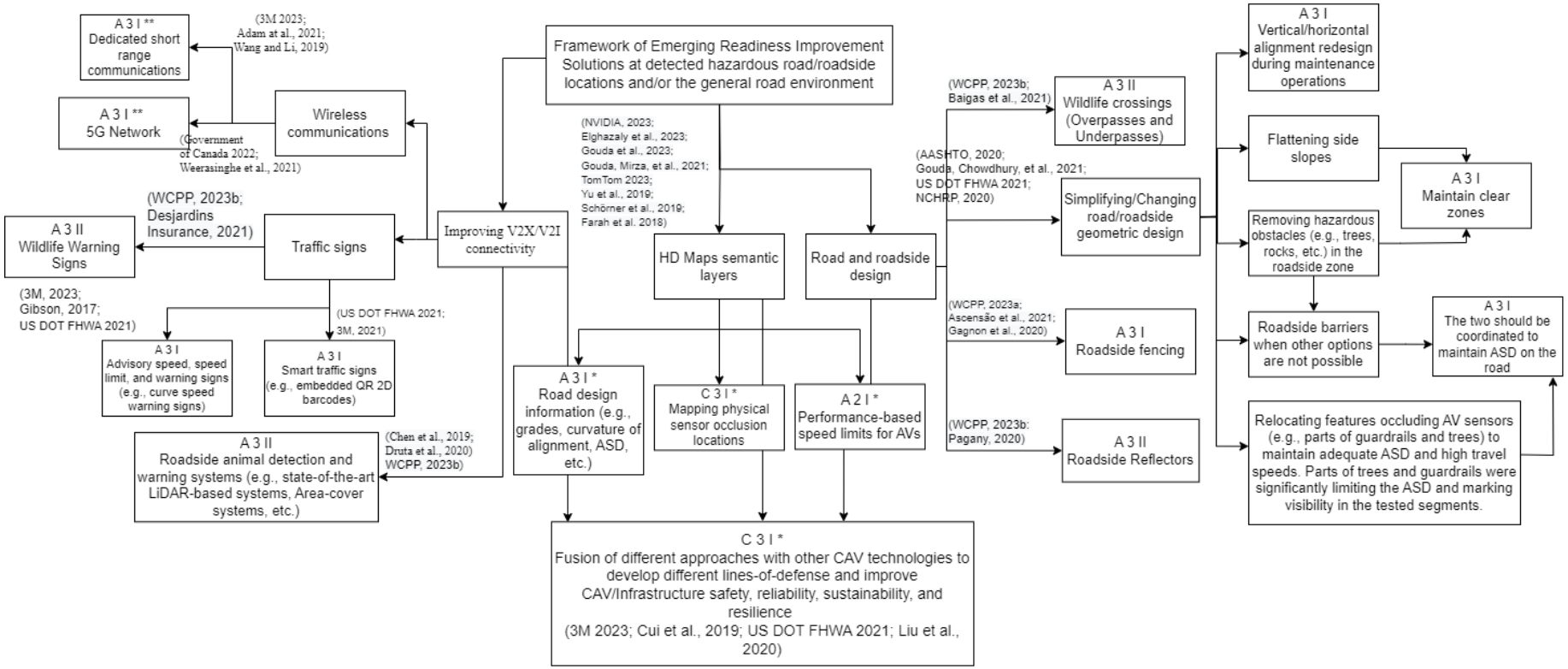


Figure 87 Framework of readiness improvement solution

REFERENCES

3M. (2020). *Solutions for the Road Ahead. Accessed May 7th, 2022.* https://www.3m.com/3M/en_US/road-safety-us/resources/road-transportation-safety-center-blog/full-story/~/what-is-vehicle-to-infrastructure-v2i-communication-and-why-do-we-need-it/?storyid=021748d7-f48c-4cd8-8948-b7707f231795

3M. (2021). *Are you preparing now for the roads of the future? Accessed October 17th, 2023*. https://www.3m.com/3M/en_US/transportation-infrastructure-us/

3M. (2023). *What is Vehicle-to-Infrastructure (V2I) Communication and why do we need it? Accessed October 1, 2023*. https://www.3m.com.au/3M/en_AU/road-safety-au/resources/road-transportation-safety-center-blog/full-story/?storyid=c0ee5878-4977-4ee3-8a09-0ed5ff0a5e33

AASHTO. (2018). *A Policy on Geometric Design of Highways and Streets*. The American Association of State Highway and Transportation Officials.

AASHTO. (2020). *The Case for a National Transportation Automation Vision and Roadmap A Report by the AASHTO National Strategy for Transportation Automation Task Force*.

Adam, A., & Ioannidis, C. (2014). Automatic Road-Sign Detection and Classification based on Support Vector Machines and HOG Descriptors. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*(5), 1–7. https://doi.org/10.5194/isprsannals-II-5-1-2014

Adam, C., Andres, R., Smyth, B., Kleinow, T., Grenn, K., Lakshmanan, S., & Richardson, P. (2021). Performance of DSRC V2V Communication Networks in an Autonomous Semi-Truck Platoon Application. *SAE Technical Papers*, *2021*. https://doi.org/10.4271/2021-01-0156

Agina, S., Shalkamy, A., Gouda, M., & El-Basyouny, K. (2021). Automated assessment of passing sight distance on rural highways using mobile lidar data. In *Transportation Research Record* (Vol. 2675, Issue 12). https://doi.org/10.1177/03611981211031235

Ai, C., & Tsai, Y. C. J. (2014). Critical Assessment of an Enhanced Traffic Sign Detection Method Using Mobile LiDAR and INS Technologies. *Journal of Transportation Engineering*, *141*(5), 04014096. https://doi.org/10.1061/(asce)te.1943-5436.0000760

Ai, C., & Tsai, Y.-C. (2012). Critical Assessment of Automatic Traffic Sign Detection Using 3d Lidar Point Cloud Data. *Transportation Research Board 91st Annual Meeting*.

Alberta Infrastructure and Transportation. (2007). *Roadside Design Guide. Revised Dec. 14, 2018*. http://www.transportation.alberta.ca/Content/docType233/Production/H3-Roadside-Design-Process.pdf

Alberta Infrastructure and Transportation. (2018). *Road geometric design – Design exceptions. Accessed July 8, 2021*. https://www.alberta.ca/design-exceptions

Alberta Infrastructure and Transportation. (2022). *Highway Geometric Design Guide, Chapter A – Basic Design Principles and Chapter G – 3R/4R Geometric Design Guidelines*. https://www.alberta.ca/highway-geometric-design-guide-table-of-contents.aspx

Anastassov, A., Jang, D., & Giurgiu, G. (2017). Driving speed profiles for autonomous vehicles. *IEEE Intelligent Vehicles Symposium, Proceedings*. https://doi.org/10.1109/IVS.2017.7995913

Andrei, M. A., Boiangiu, C. A., Tarbă, N., & Voncilă, M. L. (2022). Robust lane detection and tracking algorithm for steering assist systems. *Machines*, *10*(1). https://doi.org/10.3390/machines10010010

ArcMap. (2019). *What is lidar intensity data?* http://desktop.arcgis.com/en/arcmap/10.3/manage-data/las-dataset/what-is-intensity-data-.htm

Arcos-García, Á., Soilán, M., Álvarez-García, J. A., & Riveiro, B. (2017). Exploiting synergies of mobile mapping sensors and deep learning for traffic sign recognition systems. *Expert Systems with Applications*, *89*, 286–295. https://doi.org/10.1016/j.eswa.2017.07.042

Ascensão, F., Yogui, D. R., Alves, M. H., Alves, A. C., Abra, F., & Desbiez, A. L. J. (2021). Preventing wildlife roadkill can offset mitigation investments in short-medium term. *Biological Conservation*, *253*. https://doi.org/10.1016/j.biocon.2020.108902

Australian Transport and Infrastructure Council. (2019). *National Land Transport Technology Action Plan (2020-2023)*. https://www.infrastructure.gov.au/transport/land-transport-technology/files/national_land_transport_technology_action_plan_2020-2023.pdf

Baigas, P., & Gunson, K. (2021). *NCHRP Project 25-25: Safe Crossings for Wildlife*. http://www.trb.org/Publications/Blurbs/182193.aspx

Baumgartner, J., Bey, H., Fasbender, D., & Thielecke, J. (2020). Determining potential obstacles in unobservable areas based on current and past perception. *International Conference on Control, Automation and Systems*, *2020-October*. https://doi.org/10.23919/ICCAS50221.2020.9268303

Bremer, M., Wichmann, V., & Rutzinger, M. (2013). Eigenvalue and Graph-based Object Extraction from Mobile Laser Scanning Point Clouds. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, *2*(5), 55–60. https://doi.org/10.5194/isprsannals-II-5-W2-55-2013

Bresenham, J. E. (1965). Algorithm for computer control of a digital plotter - (A variation to the slope algorithm from 1963). *IBM Systems Journal*, *4*(1).

Briechle, S., Krzystek, P., & Vosselman, G. (2019). Semantic labeling of als point clouds for tree species mapping using the deep neural network pointnet++. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences - ISPRS Archives*, *42*(2/W13), 951–955. https://doi.org/10.5194/isprs-archives-XLII-2-W13-951-2019

Bulut, V. (2021). Path planning for autonomous ground vehicles based on quintic trigonometric Bézier curve: Path planning based on quintic trigonometric Bézier curve. *Journal of the Brazilian Society of Mechanical Sciences and Engineering*, *43*(2). https://doi.org/10.1007/s40430-021-02826-8

Burghardt, T. E., Popp, R., Helmreich, B., Reiter, T., Böhm, G., Pitterle, G., & Artmann, M. (2021). Visibility of various road markings for machine vision. *Case Studies in Construction Materials*, *15*. https://doi.org/10.1016/j.cscm.2021.e00579

California DMV. (2023). *Autonomous Vehicle Collision Reports. Department of Motor Vehicles. Accessed October 10, 2023*. https://www.dmv.ca.gov/portal/vehicle-industry-services/autonomous-vehicles/autonomous-vehicle-collision-reports/

Castro, M., Anta, J. A., Iglesias, L., & Sánchez, J. A. (2014). GIS-based system for sight distance analysis of highways. *Journal of Computing in Civil Engineering*, *28*(3), 1–7. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000317

Castro, M., Lopez-Cuervo, S., Paréns-González, M., & de Santos-Berbel, C. (2016). LIDAR-based roadway and roadside modelling for sight distance studies. *Survey Review*, *48*(350), 309–315. https://doi.org/10.1179/1752270615Y.0000000037

Chen, J., Tian, S., Xu, H., Yue, R., Sun, Y., & Cui, Y. (2019). Architecture of vehicle trajectories extraction with roadside lidar serving connected vehicles. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2929795

Chen, L., Li, Q., Li, M., Zhang, L., & Mao, Q. (2012). Design of a multi-sensor cooperation travel environment perception system for autonomous vehicle. *Sensors (Switzerland)*, *12*(9). https://doi.org/10.3390/s120912386

Chen, X., Kohlmeyer, B., Stroila, M., Alwar, N., Wang, R., & Bach, J. (2009). Next Generation Map Making : Geo-Referenced Ground-Level LIDAR Point Clouds for Automatic Retro-Reflective Road Feature Extraction. Presented in the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems, Seattle, WA. *Proceedings of the 17th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, *Ladybug 3*, 488–491. https://doi.org/10.1145/1653771.1653851

Chen, Y. Z., Zhao, H. J., & Shibasaki, R. (2007). A Mobile System Combining Laser Scanners and Cameras for Urban Spatial Objects Extraction. Presented in the Sixth International Conference on Machine Learning and Cybernetics, ICMLC 2007. Hong Kong, China. *Proceedings of the Sixth International Conference on Machine Learning and Cybernetics, ICMLC 2007*, *3*, 1729–1733. https://doi.org/10.1109/ICMLC.2007.4370426

Claussmann, L., Revilloud, M., Gruyer, D., & Glaser, S. (2020). A Review of Motion Planning for Highway Autonomous Driving. *IEEE Transactions on Intelligent Transportation Systems*, *21*(5). https://doi.org/10.1109/TITS.2019.2913998

CloudCompare. (2023). *CloudCompare Documentation.* http://www.cloudcompare.org/doc/wiki/index.php?title=Main_Page

CoE. (2023). *Signs and Signals. The City of Edmonton. Accessed May 5th, 2023*. https://www.edmonton.ca/transportation/report_requests/signs-signals.aspx

Colwell, I., Phan, B., Saleem, S., Salay, R., & Czarnecki, K. (2018). An Automated Vehicle Safety Concept Based on Runtime Restriction of the Operational Design Domain. *IEEE Intelligent Vehicles Symposium, Proceedings*, *2018-June*(Iv), 1910–1917. https://doi.org/10.1109/IVS.2018.8500530

Cui, J., Liew, L. S., Sabaliauskaite, G., & Zhou, F. (2019). A review on safety failures, security attacks, and available countermeasures for autonomous vehicles. *Ad Hoc Networks*, *90*, 101823. https://doi.org/10.1016/j.adhoc.2018.12.006

Dadvar, S., & Ahmed, M. M. (2021). California Autonomous Vehicle Crashes: Explanatory Data Analysis and Classification Tree. *Transportation Research Board 100th Annual Meeting*. https://trid.trb.org/view/1759180

de Santos-Berbel, C., Castro, M., Medina, S. L.-C., & Paréns-González, M. (2014). Sight Distance Studies on Roads: Influence of Digital Elevation Models and Roadside Elements. *Procedia - Social and Behavioral Sciences*, *160*(May 2015), 449–458. https://doi.org/10.1016/j.sbspro.2014.12.157

Desjardins Insurance. (2021). *Wildlife-Vehicle Collisions: What You Need to Know. Accessed May 10th, 2022*. https://www.desjardinsgeneralinsurance.com/blog/-/wildlife-vehicle-collisions-what-you-need-to-know#:~:text=There%20is%20a%20wildlife%2Dmotorabout%20%24800%20million%20in%20Canada

Druta, C., & Alden, A. S. (2020). Preventing Animal-Vehicle Crashes using a Smart Detection Technology and Warning System. *Transportation Research Record*, *2674*(10). https://doi.org/10.1177/0361198120936651

E. Welzl, & B. Gärtner. (2013). *Convex Hull, Computational Geometry (252-1425-OOL) HS13, Institute of Theoretical Computer Science – Swiss Federal Institute of Technology (ETH) Zurich. Class Lecture Notes – Chapter 3*. https://www.ti.inf.ethz.ch/ew/courses/CG13/lecture/Chapter%203.pdf

Elbanhawi, M., Simic, M., & Jazar, R. (2015). In the Passenger Seat: Investigating Ride Comfort Measures in Autonomous Cars. *IEEE Intelligent Transportation Systems Magazine*, *7*(3). https://doi.org/10.1109/MITS.2015.2405571

Elghazaly, G., Frank, R., Harvey, S., & Safko, S. (2023). High-Definition Maps: Comprehensive Survey, Challenges, and Future Perspectives. *IEEE Open Journal of Intelligent Transportation Systems*, *4*. https://doi.org/10.1109/OJITS.2023.3295502

Elliott, D., Keen, W., & Miao, L. (2019). Recent advances in connected and automated vehicles. *Journal of Traffic and Transportation Engineering (English Edition)*, *6*(2), 109–131. https://doi.org/10.1016/j.jtte.2018.09.005

European Commission. (2019). *Horizon 2020: Action Plan for the future of Mobility in Europe*. https://cordis.europa.eu/project/id/690732

European Commission CORDIS. (2019). *Horizon 2020: Action Plan for the Future Mobility in Europe*.

Fakhfakh, M., Chaari, L., & Fakhfakh, N. (2020). Bayesian curved lane estimation for autonomous driving. *Journal of Ambient Intelligence and Humanized Computing*, *11*(10). https://doi.org/10.1007/s12652-020-01688-7

Farah, H., Erkens, S. M. J. G., Alkim, T., & van Arem, B. (2018). Infrastructure for Automated and Connected Driving: State of the Art and Future Research Directions. In *Lecture Notes in Mobility*. https://doi.org/10.1007/978-3-319-60934-8_16

Favarò, F. M., Eurich, S. O., & Rizvi, S. S. (2019). "Human" Problems in Semi-Autonomous Vehicles: Understanding Drivers' Reactions to Off-Nominal Scenarios. *International*

Journal of Human-Computer Interaction, *35*(11). https://doi.org/10.1080/10447318.2018.1561784

FHWA. (2009). *Manual on Uniform Traffic Control Devices for Streets and Highways. Federal Highway Administration.* (Issue May).

F.P. Preparata, & M.I. Shamos. (1985). *Computational Geometry: An Introduction*. Springer-Verlag. https://link.springer.com/book/10.1007/978-1-4612-1098-6

Freeman, P., Wagner, J., & Alexander, K. (2016). Run-off-road and recovery – state estimation and vehicle control strategies. *Vehicle System Dynamics*, *54*(9). https://doi.org/10.1080/00423114.2016.1200728

FTVG. (2021). *Connected and Autonomous Vehicle Road Scoring Index. Future Transport Visions Group*. https://autonomousvehicleindex.com/

Gábor Orosz, Jin I. Ge, Chaozhe R. He, Sergei S. Avedisov, Wubing B. Qin, & Linjun Zhang. (2017). Seeing Beyond the Line of Site – Controlling Connected Automated Vehicles. *Mechanical Engineering*, *139*(12).

Gargoum, S. A., & El Basyouny, K. (2019). A Literature Synthesis of LiDAR Applications in Transportation: Feature Extraction and Geometric Assessments of Highways. *GIScience & Remote Sensing*, *56*(6), 864–893. https://doi.org/10.1080/15481603.2019.1581475

Gargoum, S. A., El Basyouny, K., Shalkamy, A., & Gouda, M. (2018). Feasibility of Extracting Highway Vertical Profiles from Lidar Data. *Canadian Journal of Civil Engineering*, *421*(November 2017), 418–421. https://doi.org/10.1139/cjce-2017-0620

Gargoum, S. A., El-Basyouny, K., & Sabbagh, J. (2018). Assessing Stopping and Passing Sight Distance on Highways Using Mobile LiDAR Data. *Journal of Computing in Civil Engineering*, *32*(4), 1–13. https://doi.org/10.1061/(ASCE)CP.1943-5487.0000753

Gargoum, S. A., & Karsten, L. (2021). Virtual assessment of sight distance limitations using LiDAR technology: Automated obstruction detection and classification. *Automation in Construction*, *125*. https://doi.org/10.1016/j.autcon.2021.103579

Gargoum, S., & El-Basyouny, K. (2017). Automated extraction of road features using LiDAR data: A review of LiDAR applications in transportation. *2017 4th International Conference on Transportation Information and Safety, ICTIS 2017 - Proceedings*, 563–574. https://doi.org/10.1109/ICTIS.2017.8047822

Gargoum, S., El-Basyouny, K., Sabbagh, J., & Froese, K. (2017). Automated Highway Sign Extraction Using Lidar Data. *Transportation Research Record: Journal of the Transportation Research Board*, *2643*(1), 1–8. https://doi.org/10.3141/2643-01

Ge, J. I., Avedisov, S. S., He, C. R., Qin, W. B., Sadeghpour, M., & Orosz, G. (2018). Experimental validation of connected automated vehicle design among human-driven vehicles. *Transportation Research Part C: Emerging Technologies*, *91*. https://doi.org/10.1016/j.trc.2018.04.005

Geese, M., Seger, U., & Paolillo, A. (2018). Detection probabilities: Performance prediction for sensors of autonomous vehicles. *IS and T International Symposium on Electronic Imaging Science and Technology*. https://doi.org/10.2352/ISSN.2470-1173.2018.17.AVM-148

Ghallabi, F., El-haj-shhade, G., & Mittet, M. (2019). LIDAR-Based Road Signs Detection For Vehicle Localization in an HD Map. Presented in IEEE Intelligent Vehicles Symposium. Paris, France. 2019. *IEEE Intelligent Vehicles Symposium*.

Ghandour, R., Victorino, A., Doumiati, M., & Charara, A. (2010). Tire/road friction coefficient estimation applied to road safety. *18th Mediterranean Conference on Control and Automation, MED'10 - Conference Proceedings*. https://doi.org/10.1109/MED.2010.5547840

Gibson, B. (2017). *Analysis of autonomous vehicle policies. Kentucky Transportation Center Research Report. 1568.* https://doi.org/10.13023/KTC.RR.2016.25

González-Gómez, K., & Castro, M. (2020). Analysis of sight distances at urban intersections from a vulnerable users' approach: A case study. *Transportation Research Procedia*, *45*(2020), 226–233. https://doi.org/10.1016/j.trpro.2020.03.011

Google Maps. (2023). *What is Street View?* https://www.google.com/streetview/

Gopalakrishna, D., Carlson, P., Sweatmen, P., Raghunathan, D., & Urena Serulle, N. (2019). Impacts of Automated Vehicles on Highway Infrastructure. *FHWA Publication*.

Gouda, M., Chowdhury, I., Weiß, J., Epp, A., & El-Basyouny, K. (2021). Automated assessment of infrastructure preparedness for autonomous vehicles. *Automation in Construction*, *129*. https://doi.org/10.1016/j.autcon.2021.103820

Gouda, M., El-Basyouny, K., & Epp, A. (2020). A Deep Learning Algorithm to Extract Traffic Signs from Point Cloud Data. *Transportation Research Board 99th Annual Meeting*.

Gouda, M., Epp, A., Tilroe, R., & El-Basyouny, K. (2022). Traffic sign extraction using deep hierarchical feature learning and mobile light detection and ranging (LiDAR) data on rural highways. *Journal of Intelligent Transportation Systems: Technology, Planning, and Operations*. https://doi.org/10.1080/15472450.2022.2074792

Gouda, M., Mello, B. A. de A., & El-Basyouny, K. (2021). Automated object detection, mapping, and assessment of roadside clear zones using lidar data. In *Transportation Research Record* (Vol. 2675, Issue 12). https://doi.org/10.1177/03611981211029934

Gouda, M., Mirza, J., Weiß, J., Ribeiro Castro, A., & El-Basyouny, K. (2021). Octree-based point cloud simulation to assess the readiness of highway infrastructure for autonomous vehicles. *Computer-Aided Civil and Infrastructure Engineering*, *36*(7). https://doi.org/10.1111/mice.12643

Gouda, M., Pawliuk, Z., Mirza, J., & El-Basyouny, K. (2023). Using convex hulls with octree/voxel representations of point clouds to assess road and roadside geometric design for automated vehicles. *Automation in Construction*, *154*, 104967. https://doi.org/10.1016/j.autcon.2023.104967

Gouda, M., Shalkamy, A., Li, X., & El-Basyouny, K. (2022). Fully Automated Algorithm for Light Pole Detection and Mapping in Rural Highway Environment Using Mobile Light Detection and Ranging Point Clouds. In *Transportation Research Record* (Vol. 2676, Issue 7). https://doi.org/10.1177/03611981221082531

Government of Alberta. (2013). *Drivers Reminded to be Aware of Wildlife on Roads.* https://www.alberta.ca/release.cfm?xID=35284CD81198C-031F-D478-65CE4D0DE75DDA4F

Government of Canada. (2022). *Communications Research Centre Canada*. https://www.ic.gc.ca/eic/site/069.nsf/eng/h_00050.html

Griffiths, D., & Boehm, J. (2019). A Review on deep learning techniques for 3D sensed data classification. *Remote Sensing*, *11*(12), 1499. https://doi.org/10.3390/rs11121499

Guan, H., Yan, W., Yu, Y., Zhong, L., & Li, D. (2018). Robust Traffic-Sign Detection and Classification Using Mobile LiDAR Data with Digital Images. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *11*(5), 1715–1724. https://doi.org/10.1109/JSTARS.2018.2810143

Gudigar, A., Chokkadi, S., & Raghavendra, U. (2016). A review on Automatic Detection and Recognition of Traffic Sign. *Multimedia Tools and Applications*, *75*(1), 333–364. https://doi.org/10.1007/s11042-014-2293-7

Guo, J., Cheng, X., Chen, Q., & Yang, Q. (2019). Detection of occluded road signs on autonomous driving vehicles. *Proceedings - IEEE International Conference on Multimedia and Expo*, *2019-July*. https://doi.org/10.1109/ICME.2019.00152

Habib, K., Gouda, M., & El-Basyouny, K. (2020). Calibrating Design Guidelines using Mental Workload and Reliability Analysis. *Transportation Research Record*, *2674*(8). https://doi.org/10.1177/0361198120928075

Habib, K., Shalkamy, A., & El-Basyouny, K. (2019). Investigating the Effects of Mental Workload on Highway Safety. *Transportation Research Record*, *2673*(7), 619–629. https://doi.org/10.1177/0361198119846474

Hackel, T., Wegner, J. D., & Schindler, K. (2016). Contour Detection in Unstructured 3D Point Clouds. Presented at the 29th IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, Nevada. *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*. https://doi.org/10.1109/CVPR.2016.178

He, J., Radford, A., Li, L., Xiong, Z., Tang, Z., Fu, X., Leng, S., Wu, F., Huang, K., Huang, J., Zhang, J., & Zhang, Y. (2019). Cooperative connected autonomous vehicles (CAV): Research, applications and challenges. *Proceedings - International Conference on Network Protocols, ICNP*, *2019-October*. https://doi.org/10.1109/ICNP.2019.8888126

Here. (2023). *Maps for ADAS and HAD. Accessed October 10th, 2023*. https://www.here.com/platform/adas-had

Hinks, T., Carr, H., Gharibi, H., & Laefer, D. F. (2015). Visualisation of urban airborne laser scanning data with occlusion images. *ISPRS Journal of Photogrammetry and Remote Sensing*, *104*, 77–87. https://doi.org/10.1016/j.isprsjprs.2015.01.014

Hirt, P. R., Holtkamp, J., Hoegner, L., Xu, Y., & Stilla, U. (2022). Occlusion detection of traffic signs by voxel-based ray tracing using highly detailed models and MLS point clouds of vegetation. In *International Journal of Applied Earth Observation and Geoinformation* (Vol. 114). https://doi.org/10.1016/j.jag.2022.103017

Howe, S. J. (2006). Assessment of Road Signs for Retroreflectivity. Presented in Engineering Asset Management. Springer, London. *Engineering Asset Management*, 1080–1089. https://doi.org/10.1007/978-1-84628-814-2_118

Huang, A. S., Moore, D., Antone, M., Olson, E., & Teller, S. (2009). Finding multiple lanes in urban road networks with vision and lidar. *Autonomous Robots*, *26*(2–3). https://doi.org/10.1007/s10514-009-9113-3

Huang, P., Cheng, M., Chen, Y., Luo, H., Wang, C., & Li, J. (2017). Traffic Sign Occlusion Detection Using Mobile Laser Scanning Point Clouds. *IEEE Transactions on Intelligent Transportation Systems*, *18*(9), 2364–2376. https://doi.org/10.1109/TITS.2016.2639582

Hubmann, C., Quetschlich, N., Schulz, J., Bernhard, J., Althoff, D., & Stiller, C. (2019). A POMDP maneuver planner for occlusions in urban scenarios. *IEEE Intelligent Vehicles Symposium, Proceedings*, *2019-June*. https://doi.org/10.1109/IVS.2019.8814179

IIHS. (2018). *IIHS Examines Driver Assistance Features in Road, Track Tests. Accessed May 5, 2022. Insurance Institute for Highway Safety*. https://www.iihs.org/news/detail/iihs-examines-driver-assistance-features-in-road-track-tests

Ilci, V., & Toth, C. (2020). High definition 3D map creation using GNSS/IMU/LiDAR sensor integration to support autonomous vehicle navigation. *Sensors (Switzerland)*, *20*(3). https://doi.org/10.3390/s20030899

International Telecommunication Union. (2011). T.871 : Information technology - Digital compression and coding of continuous-tone still images: JPEG File Interchange Format (JFIF). In *ITU-T T.871*.

Janai, J., Güney, F., Behl, A., & Geiger, A. (2020). Computer Vision for Autonomous Vehicles: Problems, Datasets and State of the Art. *Foundations and Trends® in Computer Graphics and Vision*, *12*(1–3). https://doi.org/10.1561/0600000079

Javanmardi, M., Song, Z., & Qi, X. (2018). Automated Traffic sign and Light Pole Detection in Mobile LiDAR Scanning Data. *IET Intelligent Transport Systems*, *13*(5), 803–815. https://doi.org/10.1049/iet-its.2018.5360

Jung, J., & Bae, S. H. (2018). Real-time road lane detection in Urban areas using LiDAR data. *Electronics (Switzerland)*, *7*(11). https://doi.org/10.3390/electronics7110276

Jung, J., Olsen, M. J., Hurwitz, D. S., Kashani, A. G., & Buker, K. (2018). 3D virtual intersection sight distance analysis using lidar data. *Transportation Research Part C: Emerging Technologies*, *86*(August 2017), 563–579. https://doi.org/10.1016/j.trc.2017.12.004

J.W. Gagnon, C.D. Loberger, K.S. Ogren, C.A. Beach, H.P. Nelson, & S.C. Sprague. (2020). *Evaluation of the Effectiveness of Wildlife Guards and Right of Way Escape Mechanisms for Large Ungulates in Arizona (No. FHWA-AZ-20-729). Department of Transportation, Arizona.*

Karsten, L. (2019). *A Data-Driven Approach to Creating a Traffic Sign Asset Inventory using Remote Sensing Technology* [University of Alberta]. https://doi.org/https://doi.org/10.7939/r3-7c0y-k709

Kaur, G., & Kumar, D. (2015). Lane Detection Techniques: A Review. *International Journal of Computer Applications*, *112*(10).

Khattak, A. J., & Shamayleh, H. (2005). Highway Safety Assessment through Geographic Information System-Based Data Visualization. *Journal of Computing in Civil Engineering*, *19*(4), 407–411. https://doi.org/10.1061/(asce)0887-3801(2005)19:4(407)

Khoury, J., & Amine, K. (2019). Highway Geometric Design Changes in Response to a Fully Autonomous Vehicle Fleet. *Transportation Research Board 98th Annual Meeting*.

Khoury, J., Amine, K., & Saad, R. A. (2019). An Initial Investigation of the Effects of a Fully Automated Vehicle Fleet on Geometric Design. *Journal of Advanced Transportation*, *2019*. https://doi.org/10.1155/2019/6126408%0AResearch

Kilani, O., Gouda, M., Weiß, J., & El-Basyouny, K. (2021). Safety assessment of urban intersection sight distance using mobile lidar data. *Sustainability (Switzerland)*, *13*(16). https://doi.org/10.3390/su13169259

Kingma, D. P., & Ba, J. (2014). Adam: A Method for Stochastic Optimization. *ArXiv Preprint ArXiv:1412.6980*, 1–15. http://arxiv.org/abs/1412.6980

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

Kuo, W.-J., & Lin, C.-C. (2007). *Two-Stage Road Sign Detection and Recognition*. 1427–1430. https://doi.org/10.1109/icme.2007.4284928

Laconte, J., Kasmi, A., Aufrère, R., Vaidis, M., & Chapuis, R. (2022). A survey of localization methods for autonomous vehicles in highway scenarios. In *Sensors* (Vol. 22, Issue 1). https://doi.org/10.3390/s22010247

Laefer, D. F. (2020). Harnessing Remote Sensing for Civil Engineering: Then, Now, and Tomorrow. In *Lecture Notes in Civil Engineering* (Vol. 33). https://doi.org/10.1007/978-981-13-7067-0_1

Laine, S. (2010). Efficient Sparse Voxel Octrees – Analysis , Extensions , and Implementation. *IEEE Transactions on Visualization and Computer Graphics*, *d*(February).

Lawson, S. (2018). *Roads that Cars Can Read: Report III - Tackling the Transition to Automated Vehicles*. http://www.eurorap.org/new-report-tackles-the-transition-to-automated-vehicles-on-roads-that-cars-can-read/%0Ahttps://trid.trb.org/view/1527921

Lee, H. S., & Kim, K. (2018). Simultaneous Traffic Sign Detection and Boundary Estimation Using Convolutional Neural Network. *IEEE Transactions on Intelligent Transportation Systems*, *19*(5), 1652–1663. https://doi.org/10.1109/TITS.2018.2801560

Li, B., Song, D., Li, H., Pike, A., & Carlson, P. (2018). Lane Marking Quality Assessment for Autonomous Driving. *IEEE International Conference on Intelligent Robots and Systems*. https://doi.org/10.1109/IROS.2018.8593855

Li, H., Luo, Y., & Wu, J. (2019). Collision-free path planning for intelligent vehicles based on bézier curve. *IEEE Access*, *7*. https://doi.org/10.1109/ACCESS.2019.2938179

Liu, H., Yang, R., Wang, L., & Liu, P. (2019). Evaluating Initial Public Acceptance of Highly and Fully Autonomous Vehicles. *International Journal of Human-Computer Interaction*, *35*(11). https://doi.org/10.1080/10447318.2018.1561791

Liu, J., Jayakumar, P., Stein, J. L., & Ersal, T. (2017). Combined Speed and Steering Control in High-Speed Autonomous Ground Vehicles for Obstacle Avoidance Using Model Predictive Control. *IEEE Transactions on Vehicular Technology*, *66*(10), 8746–8763. https://doi.org/10.1109/TVT.2017.2707076

Liu, L., Lu, S., Zhong, R., Wu, B., Yao, Y., Zhang, Q., & Shi, W. (2021). Computing Systems for Autonomous Driving: State of the Art and Challenges. *IEEE Internet of Things Journal*, *8*(8). https://doi.org/10.1109/JIOT.2020.3043716

Liu, Z., Jiang, H., Tan, H., & Zhao, F. (2020). An Overview of the Latest Progress and Core Challenge of Autonomous Vehicle Technologies. *MATEC Web of Conferences*, *308*, 06002. https://doi.org/10.1051/matecconf/202030806002

Ma, L., Li, Y., Li, J., Wang, C., Wang, R., & Chapman, M. A. (2018). Mobile laser scanned point-clouds for road object detection and extraction: A review. *Remote Sensing*, *10*(10), 1531. https://doi.org/10.3390/rs10101531

Ma, Y., Zheng, Y., Wang, S., Wong, Y. D., & Easa, S. M. (2022). Point cloud-based optimization of roadside LiDAR placement at constructed highways. *Automation in Construction*, *144*. https://doi.org/10.1016/j.autcon.2022.104629

Malik, R. Q., Ramli, K. N., Kareem, Z. H., Habelalmatee, M. I., & Abbas, H. (2020). A review on vehicle-to-infrastructure communication system: Requirement and applications. *2020 3rd International Conference on Engineering Technology and Its Applications, IICETA 2020*. https://doi.org/10.1109/IICETA50496.2020.9318825

Matrone, F., Paolanti, M., Felicetti, A., Martini, M., & Pierdicca, R. (2022). BubblEX: An Explainable Deep Learning Framework for Point-Cloud Classification. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *15*. https://doi.org/10.1109/JSTARS.2022.3195200

Mcdonald, D. R. (2018). How Might Connected Vehicles and Autonomous Vehicles Influence Geometric Design ? How Might Connected Vehicles and Autonomous Vehicles Influence Geometric Design ? *Transportation Research Board 97th Annual Meeting*, 4–5.

Mobileye. (2021). *The Crucial Differences Between AV Maps and HD Maps. Accessed May 19, 2022, .* https://www.mobileye.com/blog/av-maps-vs-hd-maps/

Mohammed, A. S., Amamou, A., Ayevide, F. K., Kelouwani, S., Agbossou, K., & Zioui, N. (2020). The perception system of intelligent ground vehicles in all weather conditions: A systematic literature review. In *Sensors (Switzerland)* (Vol. 20, Issue 22). https://doi.org/10.3390/s20226532

M.P. Huijser, A.V. Kociolek, T.D.H. Allen, & P. McGowen. (2015). *Construction Guidelines for Wildlife Fencing and Associated Escape and Lateral Access Control Measures, Western Transportation Institute – Montana State University. Requested by AASHTO as Part of NCHRP Project 25-25, Task 84*. http://onlinepubs.trb.org/onlinepubs/nchrp/docs/NCHRP25-25%2884%29_FR.pdf

NACTO. (2019). *Blueprint for Autonomous Urbanism. National Association of City Transportation Officials.* www.nacto.org

NCHRP. (2009). *Guide for Pavement Friction. National Cooperative Highway Research Program*. Transportation Research Board. https://doi.org/10.17226/23038

NCHRP. (2020). *Connected Road Classification System (CRCS) Development. National Cooperative Highway Research Program.*

NVIDIA. (2023). *DRIVE Map for Self-Driving Vehicles. Accessed September 3rd, 2023*. DRIVE Map for Self-Driving Vehicles

Orzechowski, P. F., Meyer, A., & Lauer, M. (2018). Tackling Occlusions Limited Sensor Range with Set-based Safety Verification. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, *2018-November*. https://doi.org/10.1109/ITSC.2018.8569332

Pagany, R. (2020). Wildlife-vehicle collisions - Influencing factors, data collection and research methods. In *Biological Conservation* (Vol. 251). https://doi.org/10.1016/j.biocon.2020.108758

Patole, S. M., Torlak, M., Wang, D., & Ali, M. (2017). Automotive Radars: A review of signal processing techniques. *IEEE Signal Processing Magazine*, *34*(2). https://doi.org/10.1109/MSP.2016.2628914

Pei, D., Sun, F., & Liu, H. (2013). Supervised Low-Rank Matrix Recovery for Traffic Sign Recognition in Image Sequences. *IEEE Signal Processing Letters*, *20*(3), 241–244. https://doi.org/10.1109/LSP.2013.2241760

Pike, A. M., Barrette, T. P., & Carlson, P. J. (2018). *Evaluation of the effects of pavement marking characteristics on detectability by ADAS Machine Vision. National Cooperative Highway Research Program (NCHRP), Washington, DC.* http://onlinepubs.trb.org/onlinepubs/nchrp/docs/NCHRP20-102-06finalreport.pdf

Qi, C. R., Su, H., Mo, K., & Guibas, L. J. (2017). PointNet: Deep Learning on Point Sets for 3D Classification and Segmentation. Presented in the IEEE Conference on Computer Vision and Pattern Recognition, Honolulu, Hawaii. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 652–660. https://doi.org/10.1109/CVPR.2017.16

Qi, C. R., Su, H., Niebner, M., Dai, A., Yan, M., & Guibas, L. J. (2016). *Volumetric and multi-view CNNs for object classification on 3D data. Presented at the IEEE Computer Society*

Conference on Computer Vision and Pattern Recognition. San Francisco, CA, USA. https://doi.org/10.1109/CVPR.2016.609

Qi, C. R., Yi, L., Su, H., & Guibas, L. J. (2017). PointNet++: Deep Hierarchical Feature Learning on Point Sets in a Metric Space. Presented in Advances in Neural Information Processing Systems, Long Beach, CA. *Advances in Neural Information Processing Systems*, 5099–5108. https://doi.org/10.1016/S0039-9140(96)02179-0

Ray, B. L. (2017). Impacts of Connected Vehicles on multi-modal Geometric Design. *Transportation Research Board 96th Annual Meeting*.

Razali, N. A. M., Shamsaimon, N., Wook, M., & Ishak, K. K. (2020). Conceptual Model for Connected Vehicles Safety and Security using Big Data Analytics. *International Journal of Advanced Computer Science and Applications*, *11*(11). https://doi.org/10.14569/IJACSA.2020.0111136

Riveiro, B., Diaz-Vilarino, L., Conde-Carnero, B., Soilan, M., & Arias, P. (2015). Automatic Segmentation and Shape-Based Classification of Retro-Reflective Traffic Signs from Mobile LiDAR Data. *IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing*, *9*(1), 295–303. https://doi.org/10.1109/JSTARS.2015.2461680

Rodrigues, M., Gest, G., Mcgordon, A., & Marco, J. (2018). Adaptive behaviour selection for autonomous vehicle through naturalistic speed planning. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, *2018-March*. https://doi.org/10.1109/ITSC.2017.8317907

Roynard, X., Deschaud, J. E., & Goulette, F. (2018). Paris-Lille-3D: A large and High-Quality Ground-Truth Urban Point Cloud Dataset for Automatic Segmentation and Classification. *International Journal of Robotics Research*, *37*(6), 545–557. https://doi.org/10.1177/0278364918767506

SAE International. (2019). *SAE: Levels of Driving Automation*. https://www.sae.org/news/press-room/2018/12/sae-international-releases-updated-visual-chart-for-its-“levels-of-driving-automation”-standard-for-self-driving-vehicles

Saeed, T. U. (2019). *Road Infrastructure Readiness for Autonomous Vehicles* (Issue August).

Saeed, T. U., Alabi, B. N. T., & Labi, S. (2021). Preparing Road Infrastructure to Accommodate Connected and Automated Vehicles: System-Level Perspective. *Journal of Infrastructure Systems*, *27*(1). https://doi.org/10.1061/(asce)is.1943-555x.0000593

Sakai, A., Dinius, J., Chawla, K., Raffin, A., & Paques, A. (2018). *PythonRobotics: a Python code collection of robotics algorithms. arXiv preprint arXiv:1808.10703*.

Schorner, P., Tottel, L., Doll, J., & Zollner, J. M. (2019). Predictive trajectory planning in situations with hidden road users using partially observable markov decision processes. *IEEE Intelligent Vehicles Symposium, Proceedings*, *2019-June*. https://doi.org/10.1109/IVS.2019.8814022

Seif, H. G., & Hu, X. (2016). Autonomous Driving in the iCity—HD Maps as a Key Challenge of the Automotive Industry. *Engineering*, *2*(2), 159–162. https://doi.org/10.1016/J.ENG.2016.02.010

Senate Canada. (2018). *Driving Change: Technology and the future of the automated vehicle. Report of the Standing Senate Committee on Transport and Communications*. https://sencanada.ca/content/sen/committee/421/TRCM/Reports/COM_RPT_TRCM_AutomatedVehicles_e.pdf

Sensible4. (2020). *Autonomous Vehicles and the Problem with Speed. Accessed May 5, 2022*. https://sensible4.fi/2020/09/29/autonomous-vehicles-and-the-problem-with-speed/#:~:text=Sensible%204%20autonomous%20vehicles%20have,110%20to%20130%20km%2Fh

Shetty, A., Yu, M., Kurzhanskiy, A., Grembek, O., Tavafoghi, H., & Varaiya, P. (2021). Safety challenges for autonomous vehicles in the absence of connectivity. *Transportation Research Part C: Emerging Technologies*, *128*. https://doi.org/10.1016/j.trc.2021.103133

Shladover, S. E., & Bishop, R. (2015). Road Transport Automation as a Public–Private Enterprise. *Transportation Research Board Conference Proceedings*.

Soilán, M., Riveiro, B., Martínez-Sánchez, J., & Arias, P. (2016). Traffic sign detection in MLS acquired point clouds for geometric and image-based semantic inventory. *ISPRS Journal of*

Photogrammetry and Remote Sensing, *114*, 92–101. https://doi.org/10.1016/j.isprsjprs.2016.01.019

Spanowicz, A. G., Teixeira, F. Z., & Jaeger, J. A. G. (2020). An adaptive plan for prioritizing road sections for fencing to reduce animal mortality. *Conservation Biology*, *34*(5). https://doi.org/10.1111/cobi.13502

Su, H., Maji, S., Kalogerakis, E., & Learned-Miller, E. (2015). Multi-View Convolutional Neural Networks for 3D Shape Recognition. *Proceedings of the IEEE Computer Society Conference on Computer Vision*, 945–953. https://doi.org/10.1109/CVPR.2018.00035

Tak, S., Kim, S., Yu, H., & Lee, D. (2022). Analysis of Relationship between Road Geometry and Automated Driving Safety for Automated Vehicle-Based Mobility Service. *Sustainability (Switzerland)*, *14*(4). https://doi.org/10.3390/su14042336

Tak, S., Yoon, J., Woo, S., & Yeo, H. (2020). Sectional Information-Based Collision Warning System Using Roadside Unit Aggregated Connected-Vehicle Information for a Cooperative Intelligent Transport System. *Journal of Advanced Transportation*, *2020*. https://doi.org/10.1155/2020/1528028

Tchapmi, L., Choy, C., Armeni, I., Gwak, J., & Savarese, S. (2017). SEGCloud: Semantic segmentation of 3D point clouds. *Proceedings - 2017 International Conference on 3D Vision, 3DV 2017*, 537–547. https://doi.org/10.1109/3DV.2017.00067

Tesla. (2019). *Tesla Autonomy Day. Accessed July 8, 2021* . https://www.youtube.com/watch?v=Ucp0TTmvqOE&t=8201s&ab_channel=Tesla

Tesla. (2023). *Tesla Future of Driving. Accessed September 2nd, 2023*. https://www.tesla.com/en_CA/autopilot?redirect=no

TomTom. (2023). *ADAS Maps. Accessed October 10th, 2023*. https://www.tomtom.com/products/adas-map/

Transport Canada. (2019). *Canada's Safety Framework for Automated and Connected Vehicles*. https://tc.canada.ca/sites/default/files/migrated/tc_safety_framework_for_acv_s.pdf

Transport Canada. (2023). *Transport Canada Innovative Technologies: Automated and Connected Vehicles*. https://www.tc.gc.ca/en/services/road/innovative-technologies/automated-connected-vehicles.html

Tudose, S. (2019). *New Tesla Model X Crashes Going Up Mountain, Was Autopilot System Slow To React? Carscoops. Accessed May 5, 2022*. from https://www.carscoops.com/2019/04/tesla-model-xs-autopilot-system-slow-to-react-to-s-bend/

US DOT FHWA. (2020). *Federal Highway Administration. National Dialogue on Highway Automation.* https://ops.fhwa.dot.gov/automationdialogue/

US DOT FHWA. (2021). *Impacts of Automated Vehicles on Highway Infrastructure. PUBLICATION NO. FHWA-HRT-21-015*.

US DOT FHWA. (2023). *Clear Zones. Accessed July 8, 2021* . https://safety.fhwa.dot.gov/roadway_dept/countermeasures/safe_recovery/clear_zones/#zones

US DOT National Science and Technology Council. (2020). *Ensuring American Leadership in Automated Vehicle Technologies Automated Vehicles 4.0*. *January*, 56. https://www.transportation.gov/sites/dot.gov/files/docs/policy-initiatives/automated-vehicles/360956/ensuringamericanleadershipav4.pdf

Vo, A. V., Laefer, D. F., Smolic, A., & Zolanvari, S. M. I. (2019). Per-point processing for detailed urban solar estimation with aerial laser scanning and distributed computing. *ISPRS Journal of Photogrammetry and Remote Sensing*, *155*(June), 119–135. https://doi.org/10.1016/j.isprsjprs.2019.06.009

Vu, A., Yang, Q., Farrell, J. A., & Barth, M. (2013). Traffic sign Detection, State Estimation, and Identification using Onboard Sensors. Presented in IEEE Conference on Intelligent Transportation Systems, The Hague, Netherlands. *IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 875–880. https://doi.org/10.1109/ITSC.2013.6728342

Wahyono, Kurnianggoro, L., Hariyono, J., & Jo, K. H. (2014). Traffic sign recognition system for autonomous vehicle using cascade SVM classifier. *IECON Proceedings (Industrial Electronics Conference)*. https://doi.org/10.1109/IECON.2014.7049114

Wali, S. B., Abdullah, M. A., Hannan, M. A., Hussain, A., Samad, S. A., Ker, P. J., & Mansor, M. Bin. (2019). Vision-based traffic sign detection and recognition systems: Current trends and challenges. In *Sensors (Switzerland)* (Vol. 19, Issue 9). https://doi.org/10.3390/s19092093

Walters, J. G., Meng, X., Xu, C., Jing, H., & Marsh, S. (2019). Rural positioning challenges for connected and autonomous vehicles. *ION 2019 International Technical Meeting Proceedings*. https://doi.org/10.33012/2019.16727

Wang, S., & Li, Z. (2019). Roadside Sensing Information Enabled Horizontal Curve Crash Avoidance System Based on Connected and Autonomous Vehicle Technology. *Transportation Research Record*, *2673*(5). https://doi.org/10.1177/0361198119837957

Wang, S., & Yu, B. (2019). Impacts on the Geometric Design of Highway from Non-Autonomous To Autonomous Vehicles (L1 ~ L4). *Transportation Research Board 98th Annual Meeting*, 1–6.

Wang, Y., & Tsai, Y. (2018). A lane detection method based on 3D-LiDAR. *Automotive Research & Testing Center (ARTC)*.

Waymo. (2023). *Waymo Technology. Accessed August 13th, 2023*. https://waymo.com/tech/

WCPP. (2023a). *Collision Facts. BC Wildlife Collision Prevention Program. Accessed October 10, 2023*. https://www.wildlifecollisions.ca/collision/collision-facts.htm

WCPP. (2023b). *Mitigation. BC Wildlife Collision Prevention Program. Accessed October 10, 2023*. https://www.wildlifecollisions.ca/prevention/mitigation.htm

Weerasinghe, N., Hewa, T., Liyanage, M., Kanhere, S. S., & Ylianttila, M. (2021). A Novel Blockchain-as-a-Service (BaaS) Platform for Local 5G Operators. *IEEE Open Journal of the Communications Society*, *2*. https://doi.org/10.1109/OJCOMS.2021.3066284

Wen, C., Li, J., Luo, H., Yu, Y., Cai, Z., Wang, H., & Wang, C. (2016). Spatial-related traffic sign inspection for inventory purposes using mobile laser scanning data. *IEEE Transactions on Intelligent Transportation Systems*, *17*(1), 27–37. https://doi.org/10.1109/TITS.2015.2418214

Williams, K., Olsen, M. J., Roe, G. V., & Glennie, C. (2013). Synthesis of transportation applications of mobile LIDAR. *Remote Sensing*, *5*(9), 4652–4692. https://doi.org/10.3390/rs5094652

Wong, K., Gu, Y., & Kamijo, S. (2021). Mapping for autonomous driving: Opportunities and challenges. *IEEE Intelligent Transportation Systems Magazine*, *13*(1). https://doi.org/10.1109/MITS.2020.3014152

Wsp. (2020). *FIVE KEY STEPS TO BECOMING READY FOR CONNECTED AND AUTOMATED VEHICLES (CAVS)*. https://www.wsp.com/en-CA/insights/ca-five-key-steps-to-becoming-cav-ready

Wu, B., Wan, A., Yue, X., & Keutzer, K. (2018). SqueezeSeg: Convolutional Neural Nets with Recurrent CRF for Real-Time Road-Object Segmentation from 3D LiDAR Point Cloud. *Proceedings - IEEE International Conference on Robotics and Automation*. https://doi.org/10.1109/ICRA.2018.8462926

Wu, B., Zhou, X., Zhao, S., Yue, X., & Keutzer, K. (2019). SqueezeSegV2: Improved model structure and unsupervised domain adaptation for road-object segmentation from a LiDAR point cloud. *Proceedings - IEEE International Conference on Robotics and Automation*, *2019-May*. https://doi.org/10.1109/ICRA.2019.8793495

Wu, S., Wen, C., Luo, H., Chen, Y., Wang, C., & Li, J. (2015). Using mobile LiDAR point clouds for traffic sign detection and sign visibility estimation. *International Geoscience and Remote Sensing Symposium (IGARSS)*, *2015-Novem*, 565–568. https://doi.org/10.1109/IGARSS.2015.7325826

Xinli, G., Huawei, L., Hao, X., Biao, Y., & Maofei, Z. (2016). Human-driver speed profile modeling for autonomous vehicle's velocity strategy on curvy paths. *IEEE Intelligent Vehicles Symposium, Proceedings*, *2016-August*. https://doi.org/10.1109/IVS.2016.7535472

Xu, J., Yang, K., & Shao, Y. M. (2018). Ride Comfort of Passenger Cars on Two-Lane Mountain Highways Based on Tri-axial Acceleration from Field Driving Tests. *International Journal of Civil Engineering*, *16*(3). https://doi.org/10.1007/s40999-016-0132-0

Yang, Y., Chen, F., Yiru, S., & Tian, D. (2018). FoldingNet : Point Cloud Auto-encoder via Deep Grid Deformation. Presented in 2018 IEEE Conference on Computer Vision and Pattern Recognition, Salt Lake City, Utah. *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 206–215.

Yu, M. Y., Vasudevan, R., & Johnson-Roberson, M. (2019). Occlusion-aware risk assessment for autonomous driving in urban environments. *IEEE Robotics and Automation Letters*, *4*(2). https://doi.org/10.1109/LRA.2019.2900453

Yu, Y., Li, J., Wen, C., Guan, H., Luo, H., & Wang, C. (2016). Bag-of-visual-phrases and hierarchical deep models for traffic sign detection and recognition in mobile laser scanning data. *ISPRS Journal of Photogrammetry and Remote Sensing*, *113*, 106–123. https://doi.org/10.1016/j.isprsjprs.2016.01.005

Yuan, W., Khot, T., Held, D., Mertz, C., & Hebert, M. (2018). PCN : Point Completion Network . Presented in 2018 International Conference on 3D Vision (3DV), Verona, Italy. *2018 International Conference on 3D Vision (3DV)*.

Yuan, Y., Xiong, Z., & Wang, Q. (2019). VSSA-NET: Vertical Spatial Sequence Attention Network for Traffic Sign Detection. *IEEE Transactions on Image Processing*, *28*(7), 3423–3434. https://doi.org/10.1109/TIP.2019.2896952

Zaklouta, F., & Stanciulescu, B. (2012). Real-time Traffic-Sign Recognition using Tree Classifiers. *IEEE Transactions on Intelligent Transportation Systems*, *13*(4), 1507–1514. https://doi.org/10.1109/TITS.2012.2225618

Zenzic. (2019). *UK Connected and Automated Mobility Roadmap to 2030*. https://zenzic.io/content/uploads/2019/09/Zenzic_Roadmap_Report_2019.pdf

Zhang, P., Vo, A., Laefer, D., & Porfiri, M. (2019). Optimization of a highway infrastructure design through a combined computational fluid dynamics and evolutionary algorithm framework. *Bulletin of the American Physical Society*, 64.

Zheng, L., Li, B., Yang, B., Song, H., & Lu, Z. (2019). Lane-level road network generation techniques for lane-level maps of autonomous vehicles: A survey. In *Sustainability (Switzerland)* (Vol. 11, Issue 16). https://doi.org/10.3390/su11164511

Zheng, L., Zeng, P., Yang, W., Li, Y., & Zhan, Z. (2020). Bézier curve-based trajectory planning for autonomous vehicles with collision avoidance. *IET Intelligent Transport Systems*, *14*(13). https://doi.org/10.1049/iet-its.2020.0355

Zhou, L., & Deng, Z. (2014). LIDAR and Vision-Based Real-Time Traffic Sign Detection and Recognition Algorithm for Intelligent Vehicle. *Presented at IEEE Conference on Intelligent Transportation Systems, Proceedings, ITSC*, 578–583. https://doi.org/10.1109/ITSC.2014.6957752

Zhou, Y., & Tuzel, O. (2018). VoxelNet: End-to-End Learning for Point Cloud Based 3D Object Detection. *2018 IEEE Conference on Computer Vision and Pattern Recognition*, 4490–4499.

Zhu, Y., Zhang, C., Zhou, D., Wang, X., Bai, X., & Liu, W. (2016). Traffic Sign Detection and Recognition using Fully Convolutional Network Guided Proposals. *Neurocomputing*, *214*, 758–766. https://doi.org/10.1016/j.neucom.2016.07.009