7-7-2020

Road Surface Feature Extraction and Reconstruction of Laser Point Clouds for Urban Environment

Ismail Hamieh
University of Windsor

ROAD SURFACE FEATURE EXTRACTION AND RECONSTRUCTION OF LASER POINT CLOUDS FOR URBAN ENVIRONMENT

By

Ismail Hamieh

A Dissertation
Submitted to the Faculty of Graduate Studies through the Department of Electrical and Computer Engineering in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy at University of Windsor

Windsor, Ontario, Canada

ROAD SURFACE FEATURE EXTRACTION AND RECONSTRUCTION OF LASER POINT CLOUDS FOR URBAN ENVIRONMENT

by

Ismail Hamieh

APPROVED BY:

____
W. Ajib, External Examiner
Université du Québec à Montréal

____
H. Maoh
Department of Civil and Environmental Engineering

____
J. Wu
Department of Electrical and Computer Engineering

____
M. Khalid
Department of Electrical and Computer Engineering

____
K. Tepe, Advisor
Department of Electrical and Computer Engineering

April 20, 2020

DECLARATION OF CO-AUTHORSHIP / PREVIOUS PUBLICATION

I. Co-Authorship

I hereby declare that this thesis incorporates material that is a result of joint research. This thesis contains the outcome of research undertaken by me under the supervision of Dr. K. Tepe. This thesis also reflects the outcomes of joint work realized with Ryan Myers, Taufiq Rahman, and Nikhil Melgiri of National Research Council of Canada. The mathematical analysis, probabilistic model, data collected, results obtained and presented among the thesis were the outcome of this joint research and collaboration. All cases, the key ideas, primary contributions, experimental designs, data collection, analysis and interpretation, were performed by the author of this thesis, and the contribution of co-authors was primarily through the provision of valuable suggestions and help in the comprehensive analysis of the experimental results submitted for the published articles. In addition, Chapter 2 of this thesis was co-authored with Marc-Antoine Drouin. My contribution was focused on LiDAR sensor technology and he was focused on camera technology. The joined work was contributed to Chapter 4 of ”3D Imaging, Analysis and Applications”.

I am aware of the University of Windsor Senate Policy on Authorship and I certify that I have properly acknowledged the contribution of other researchers to my thesis, and have obtained written permission from each of the co-author(s) to include the above material(s) in my thesis.

I certify that, with the above qualification, this thesis, and the research to which it refers, is the product of my own work.

II. Previous Publication

This thesis includes four original papers that have been previously published/submitted for publication in peer reviewed journals, as follows:

Chapter(s)	Publication title/full citation	status
Part of Chapter 2, 3, and 4	Ismail Hamieh, Ryan Myers, and Taufiq Rahman. ”Construction of Autonomous Driving Maps employing LiDAR Odometry.” 2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE). IEEE, 2019.	Published
Part of Chapter 2	Marc-Antoine Drouin, and Ismail Hamieh. ”3D Imaging, Analysis and Applications” Chapter 4 Springer, London 2020	Submitted
Part of Chapter 4, 5, and 6	Ismail Hamieh, Ryan Myers, and Taufiq Rahman. ”LiDAR Based Classification Optimization of Localization Policies of Autonomous Vehicles 2020-01-1028” WCX SAE World Congress Experience). SAE, 2020.	Submitted
Part of Chapter 4, 5, and 6	Ismail Hamieh, Ryan Myers, Hisham Nimri, Taufiq Rahman, Aarron Younan, Brad Sato, Abdul El-Kadri, Selwan Nissan, and Kemal Tepe. ”LiDAR and Camera-based Convolutional Neural Network Detection for Autonomous Driving 2020-01-0136” WCX SAE World Congress Experience). SAE, 2020.	Submitted

I certify that I have obtained a written permission from the copyright owner(s) to include the above published material(s) in my thesis. I certify that the above material describes work completed during my registration as a graduate student at the University of Windsor.

III. General

I declare that, to the best of my knowledge, my thesis does not infringe upon anyone’s copyright nor violate any proprietary rights and that any ideas, techniques, quotations, or any other material from the work of other people included in my thesis, published or otherwise, are fully acknowledged in accordance with the standard referencing practices. Furthermore, to the extent that I have included copyrighted material that surpasses the bounds of fair dealing within the meaning of the Canada Copyright Act, I certify that I have obtained a written permission from the copyright owner(s) to include such material(s) in my thesis.

I declare that this is a true copy of my thesis, including any final revisions,as approved by my thesis committee and the Graduate Studies office, and that this thesis has not been submitted for a higher degree to any other University or Institution.

ABSTRACT

Automakers are developing end-to-end three-dimensional (3D) mapping system for Advanced Driver Assistance Systems (ADAS) and autonomous vehicles (AVs). Using geomatics, artificial intelligence, and SLAM (Simultaneous Localization and Mapping) systems to handle all stages of map creation, sensor calibration and alignment. It is crucial to have a system highly accurate and efficient as it is an essential part of vehicle controls. Such mapping requires significant resources to acquire geographic information (GIS and GPS), optical laser and radar spectroscopy, Lidar, and 3D modeling applications in order to extract roadway features (e.g., lane markings, traffic signs, road-edges) detailed enough to construct a "base map". To keep this map current, it is necessary to update changes due to occurring events such as construction changes, traffic patterns, or growth of vegetation. The information of the road play a very important factor in road traffic safety and it is essential for for guiding autonomous vehicles (AVs), and prediction of upcoming road situations within AVs. The data size of the map is extensive due to the level of information provided with different sensor modalities for that reason a data optimization and extraction from three-dimensional (3D) mobile laser scanning (MLS) point clouds is presented in this thesis. The research shows the proposed hybrid filter configuration together with the dynamic developed mechanism provides significant reduction of the point cloud data with reduced computational or size constraints. The results obtained in this work are proven by a real-world system.

DEDICATION

I dedicate this work to my parents who have supported me all the way since the beginning of my studies. Also, it is dedicated to my beautiful family, my lovely wife Wadad and my kids Aya, Ali, and Adam who have been a source of motivation and inspiration throughout my studies. I like to thanks my friends and family who helped me through this journey.

ACKNOWLEDGEMENTS

I sincerely want to thank all the persons and institutions who were involved in the development of this research project:

To Dr. Kemal Tepe who encouraged me to develop this research, and all the time was aware of my progress and results. His advice and support strengthen my believe to continue and complete this program.

To Dr. Dr. Mohammed Khalid , Dr. Jonathan Wu, and Dr. Hanna Maoh, the Ph.D. Committee Members, for all the support, recommendations and feedback received during my studies.

To Dr. Wessam Ajib, External Committee Member for agreed to be part of this project and his great comments and questions.

To my professors, lecturers, authorities, technical team, and all the administrative personnel of the University of Windsor, especially to Ms. Andria Ballo, for supporting me while the development of this research.

A special thanks to National Research Council of Canada for all the support and experience shared with me during my time as a PhD student. Working with great people, pushed me to do better with my work and I really appreciate all the foundation and contribution in my work.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST of ABBREVIATIONS

AV	Autonomous Vehicle(s)
ADAS	Advanced Driver Assistance Systems
RTK	Real-time kinematic
GNSS	Global Navigation Satellite Systems
BDD	Berkeley Deep Drive
CNN	Convolutional neural network
DNN	Deep neural network
DAQ	Data Acquisition System
GPS	Global Positioning System
IMU	Inertial Measurement Unit
LiDAR	Light Detection and Ranging
ROS	Robot Operating System
SECOND	Sparsely Embedded Convolutional Detection
SLAM	Simultaneous Localization and Mapping
UDP	User Datagram Protocol
ML	Machine Learning
CNN	Convolutional Neural Network
YOLO	You Only Look Once
ToF	Time-of-Flight
TLS	Terrestrial Laser Scanning applications
ALS	Airborne Laser Scanning applications
MLS	Mobile LiDAR Systems
2D	two-dimensional
3D	Three-dimensional
MPE	maximum permissible exposure

CHAPTER 1
INTRODUCTION

1.1 Motivation

Autonomous vehicles (AV) are getting much attention in recent years, due to significant development efforts and dramatic progress made by big companies such as Google, General Motors, Ford, Tesla, and other European manufacturers. This area of research has been growing rapidly and encompasses different domains, including robotics, computer science, cybersecurity, and of course engineering. Furthermore, it should be noted that manufacturers and different automotive suppliers who do not always publicly disclose the details on their approaches or algorithms, owing to commercial sensitivity, have made scientific advances. Although general use of autonomous vehicles for widespread use on public roads is likely years away, these vehicles are already being operated in a limited form on highways or suburban district.

1.2 Problem Description

People through many years of experience, have developed a great intuitive sense for navigation and spatial awareness. With this intuition, people are able to apply a near rules based approach to their driving. With a transition to autonomous driving, these intuitive skills need to be taught to the system that makes perception the most fundamental and critical task. One of the major challenges for autonomous vehicles is accurately knowing the position of the vehicle relative to the world frame. Currently, this is achieved by utilizing expensive sensors such as a differential GPS or Real-Time Kinematic (RTK) that provides centimeter accuracy, or by using computationally taxing algorithms to attempt to match live input data from LiDARs or cameras to previously recorded data or maps. Within this research an algorithm and accompanying hardware stack is proposed to reduce the computational load on the localization of the robot relative to a prior map.

Some of the major difficulties and challenges worked on solving were:

- Noise: All sensors are noisy. There are a few types of noise that include point perturbations and outliers. This means that a point has some probability to be within a sphere of a certain radius around the place it was sampled (perturbations) or it may appear in a random position in space (outliers).
- Rotation: a car turning left and the same car turning right will have different point clouds that represent the same car.
- Computation: 3D scanning representation of the environment required a high performance computing processor.
- Data Size: Point cloud data obtained from LiDAR sensors are large.
- Missing data: The scanned models are usually occluded and parts of the data are missing.

1.3 Novelty of the Research

The novelty of this research came from searching for solutions to solve issues mainly reported in autonomous applications such as complex 3D object interpretation and geometric reconstruction of real-world environments. A novel solution to optimize large dataset without disturbing the quality was presented, as follows:

- Hybrid Filter: A novel filtering solution is proposed that limits the size of the point clouds used for pose initial acquisition and tracking to allow for better autonomous on-board operation.
- Through experimentation, the proposed hybrid filter is demonstrated to exceed available and known filters specifically applied on point cloud data.

- Dynamically Configurable Cascaded Filtering Mechanism: is proposed and implemented.
- Through experimentation it is demonstrated that the system can be applied on different filtering techniques and algorithms and it can be expanded to be used in various applications and different sensors.

A filter that divides scattered point cloud data into multiple grids is the hybrid solution that combines down-sampling and clustering techniques into one based input space sampling using a grid of 3D voxels. A density threshold was implemented for erasing all unnecessary data within a given box with a dynamic ROS environment.

1.4 Thesis Organization

Chapter 2 elaborates the fundamentals of laser scan technology and its used in different industries. At the beginning of the chapter, point-based system was discussed that includes a representation of laser trackers. Then, multi-channel LiDAR technology is discussed. It is important to describe this technology as it is frequently encountered in multiple engineering applications and various disciplines. The chapter then elaborates on area-based (Time of Flight) ToF that is used in consumer grade products followed by ToF system characterization. An experimental section is presented to go over photogrammetry versus ToF for some specific applications. Towards the end of the chapter, data fusion for mobile application is briefly described.

Chapter 3 illustrates the basic concepts and principles of 3D scan system that was developed using a 16 channels LiDAR, GPS, and IMU sensors installed on top of an automobile vehicle. DAQ used in this research and the initial process of sensor fusion specifically for these applications were elucidated to determine the absolute position and orientation of the vehicle sign information provided by GPS, INS and odometer. In addition, SLAM algorithm and understanding all the detail aspect of this model is presented to generate HD

Map for autonomous and active safety applications.

Chapter 4 tackles the main issues of multiple applications in automotive, manufacturing, agriculture, construction, and other industries that require scanning of complex three-dimensional object, interpretation, and geometric 3D reconstruction of real-world environments to incorporate into a computer-based processing. To reach the final result, multiple algorithms and known filters were experimented for the environment. Some of the major filter techniques for point cloud data and reasoning behind the method used are covered. This chapter provides a comprehensive review of each filter used for this research. Note that each filter used is fully integrated within the developed system and it is graphically displayed with each projection process and steps.

Chapter 5 covers the overall process taken to reach the main goal and outcome. Since one voxel can represent numerous points, using a voxel grid for spatial representation can dramatically reduce data size, when compared to representing the same space with points. Depending on accuracy requirement, a voxel grid is a down-sampled version of a point cloud. The point cloud could contain inaccuracies based on how a point cloud is acquired. This is a problem when it comes to mapping, calibration, and sensor alignment for autonomous vehicle. This chapter provides a rundown of the system design and approach taken to development different filtering methods and a comparison analysis with other known filter designs.

Chapter 6 provides a report presentation of each filter output used in this research. A quantitative comparison and analysis is presented between raw data of Rotational Multi-Beam LiDAR Sensor (RMBL) and results after applying the hybrid filter. In addition, an explanation of Design for Six Sigma (DFSS) approach to ensure that all design aspects and selection process are considered, from market research through the design phase, process implementation, and product development.

Chapter 7 summarizes the entire work with an outlook on future work. This chapter contains conclusions, limitations, and investigations that could be led in the future. It discusses where the study may be extended to solve some of the challenges in the autonomous field.

CHAPTER 2

ACTIVE REMOTE SENSING SYSTEM

Active Three-dimensional (3D) imaging systems use artificial illumination in order to capture and record digital representations of objects. The use of artificial illumination allows the acquisition of dense and accurate range images of textureless objects. An active 3D imaging system can be based on different measurement principles that include Time-of-Flight (ToF), triangulation and interferometry. The different time-of-flight technologies allow the development of a plethora of systems that can operate at a range of a few meters to many kilometers (reference chapter 4 from 3D Imaging Analysis And Application). In this chapter we will focus on ToF technologies that operate from a few meters to a few hundred meters. The characterization of this systems is discussed and experimental results related to various construction and engineering industries are presented.

2.1 Background

3D vision systems capture and record a digital representation of the geometry and appearance (e.g. color-texture) information of visible 3D surfaces of people, animals, plants, objects and sites. Active 3D imaging systems use an artificial illumination (visible or infrared) to acquire dense range maps with a minimum of ambiguity. The different ToF technologies allow the development of a plethora of systems that can operate from a range of a few meters to many kilometers. Systems that operate up to a few meters (approx. 5 m) are typically called range cameras or RGB-D cameras and are typically dedicated to indoor application. Systems that operate at greater range are called Lidar (Light Detection and Ranging). Lidar started as a method to directly and accurately capture digital elevation data for Terrestrial (TLS) and Airborne (ALS) Laser Scanning applications. In the last few years, the appealing feature in Lidar attracted automotive industry and it became one of the most important sensors in active safety and autonomous applications where it is call Mobile Laser Scanning (MLS).

Terrestrial, airborne, and mobile laser scanning differ in terms of data capture mode, typical project size, scanning mechanism and obtainable accuracy and resolution; however, they share many features. In this chapter, we will focus on time-of-flight technology that operates from a few meters to a few hundred meters.

2.2 Historical Context

The fundamental work on ToF systems can be traced back to the era of RADAR (Radio Detection And Ranging), which is based on radio waves. With the advent of lasers in the late 1950s, it became possible to image a surface with angular and range resolutions much higher than possible with radio waves. This new technology was named Lidar. One of the initial use of Lidar was for mapping particles in the atmosphere [1]. During the 1980s, the development of the Global Positioning System (GPS) opened up the applications to moving sensors (airborne Lidar). Bathymetric Lidar was actually one of the first uses of airborne Lidar [2]. The early 1990s saw the improvement of the inertial measurement unit (IMU) and the ability to begin achieving decimeter accuracies. Some of the earlier non-bathymetric airborne applications were in the measurement of glaciers and how they were changing [3]. Terrestrial Lidar Systems (TLS) are also beginning to be used as a way to densely map the three-dimensional nature of features and ground surfaces to a high level of accuracy [4]. TLS is now an important tool in the construction and engineering industry.

Many modern Lidar system works in the near and short-wave infrared regions of the electromagnetic spectrum. Some sensors also operate in the green band to penetrate water and detect bottom features. In recent years, the automotive industry was exposed to the usage of Lidar and this remote sensing technology is now an essential technology for Automated and Advanced Driver Assistance Systems. Also in recent years, the entertainment industry was exposed to the usage of Lidar system. The progressive addition of motion sensitive interface to gaming platforms and the desired to personalize the gaming experience led to the development of a short-range time-of-flight cameras targeted at a large demography of cost-sensitive consumers.

2.3 Basic Measurement Principles

Active 3D imaging systems can be based on different measurement principles. The three most used principles of commercially available systems are triangulation, interferometry and time-of-flight. Seitz describes triangulation as a method based on geometry, interferometry as one that uses accurate wavelengths and time-of-flight as based on an accurate clock [5]. Figure 2.1 summarizes the typical accuracy of each type of active 3D imaging system technology found on the market as a function of the operating distance.

It can be observed that each optical technique covers a particular range of operations. Many in-depth classifications of optical distance measurement principles have been published in important references in the field of 3D vision, e.g. [6, 7, 8, 9]. Both active and passive triangulation systems are based on the same geometric principle: intersecting light rays in 3D space. Typically, an active system replaces one camera of a passive stereo system by a projection device. This projection device can be a digital video projector, an analog slide projector or a laser. Interferometry is based on the superposition of two beams of light [8]. Typically, a laser beam is split into two paths. One path is of known length, while the other is of unknown length. The difference in path lengths creates a phase difference between the light beams. The two beams are then combined together before reaching a photo-detector. The interference pattern seen by the detector resulting from the superposition of those two light beams depends on the path difference (i.e. the distance).

2.4 Time-of-Flight Methods

Most Time-Of-Flight (ToF) technologies presented in this chapter are classified as active optical non-contact 3D imaging systems because they emit light into the environment and

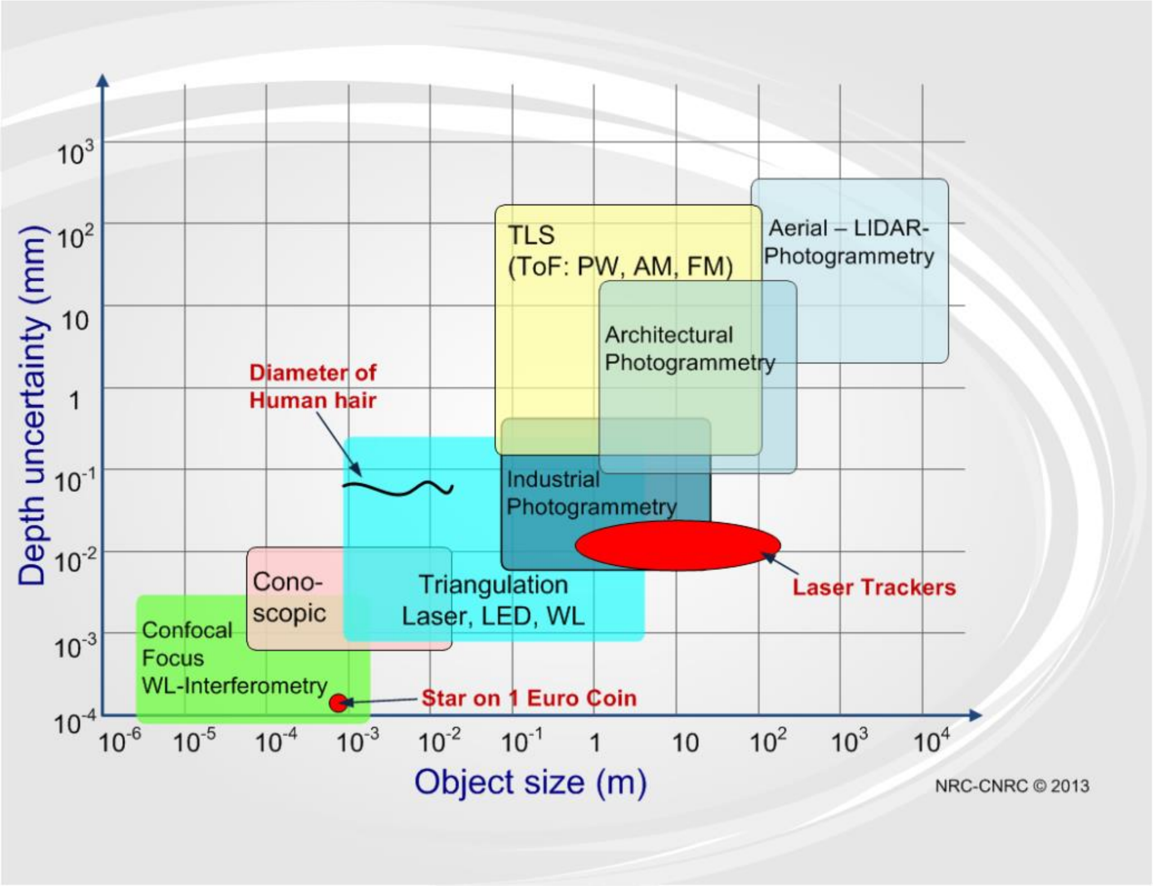


Figure 2.1: Expected depth (range) measurement uncertainty level as a function of depth-of-field/object size for different non-contact 3D imaging methods[10].

use the reflected optical energy to estimate the distance to a surface in the environment. The distance is computed from the round-trip time which may be estimated directly using a high-resolution clock to measure the time between outgoing and incoming pulses (Pulsed-based ToF), or indirectly by measuring the phase shift of an amplitude-modulated optical signal (Phase-based ToF) [11].

There are many ways to classify ToF sensors, according to their components, application fields and performance. One of the key dimensions within this taxonomy is the way in which the active 3D imaging system illuminates the scene. Some measurement systems are points based and need to scan the laser spot along two axes in order to acquire a range image, other systems used multiples laser spot and require the scanning along a single axis to obtain a range image. Finally, some systems illuminated simultaneously the entire scene.

The first family of systems that we present are point-based scanners. A large subset within this family are known as Terrestrial Lidar Systems (TLS) and have multiple applications within the construction and engineering industry. Many TLS contains a biaxial leveling compensator used to align the coordinate system of the generated range image with respect to gravity.

A second type of point-based systems frequently encounter in the construction and engineering industry is the laser tracker (LT) which is the only type of systems presented in this chapter that is classified as a contact ones because the light emitted into the environment is reflected by a retroreflector, which is placed in contact with a surface at the time of measurement.

Systems using multiple laser spots are typically referred as multi-channel Lidar. Those systems are encountered in automotive applications. Multi-channel Lidar is considered by many automakers as a key technology required for autonomous driving. In the automotive industry, the technology is known as Mobile Lidar Systems (MLS).

It emits multiple laser beams that are contained within a plane. Each acquisition generates a profile contained within that plane and by modifying the orientation or position of this plane it is possible to generate a range image.

Systems that illuminate simultaneously the entire scene are now frequently encountered in consumer grade applications. Those systems are referred as an area-based system and the detection of the incoming light is done by a two-dimensional (2D) array of detectors. Typically, because of constraints imposed by eye safety requirements for consumer grade products, their operational range is smaller than the other type of ToF systems.

ToF systems measure the distance to a point by calculating the round-trip time of light reflected from the surface[12, 11], based on an assumption of the speed at which the light is able to travel through the medium (typically air). Factors such as air temperature and pressure[13, 14], relative humidity , CO_2 concentration[12], atmospheric turbulence[15], and the presence of particulate matter[12, 16] or fog[17] can all affect the speed at which light can travel through the medium, further complications by gradients in these factors along the beam path [18].

Moreover, the measurement quality is strongly dependent on the surface being measured due to factors such as reflectance factor[19] , surface orientation[18] , optical penetration[20, 21, 22], and materials like water on the surface[23, 24].

The output of a ToF system is typically referred as a point cloud or range image. A point cloud is an unorganized set of 3D points, while a range image is an organized array of 3D points that implicitly encoded the neighborhood relation between points. This neighborhood structure is related to the physical acquisition process. As an example, for a range image produced by a multi-channel Lidar, one axis represents the laser beam index while the other represents the angular position of the laser beam along the scanning axis.

When working with coherent light sources (lasers) eye-safety is of paramount importance and one should never operate laser-based 3D imaging sensors without appropriate eye-safety training. Many 3D imaging systems use a laser in the visible spectrum where fractions of a milliwatt are sufficient to cause eye damage, since the laser light density entering the pupil is magnified, at the retina, through the lens. For an operator using any laser, an important safety parameter is the maximum permissible exposure (MPE) which is defined as the level of laser radiation to which a person may be exposed without hazardous effect or adverse biological changes in the eye or skin [25]. The MPE varies with wavelength and operating conditions of a system.

One possible mitigation strategy used by scanner manufacturers is to use a laser at 1.55 μm. At this wavelength, the light is absorbed by the eye fluids before being focused on the retina. This tends to increase the maximum permissible exposure to the laser source. Eye safety is extensively referred in *American National Standard for Safe use of Lasers* [25].

2.5 Point-based Systems

Those systems measure the distance of one point at a time and need to be scanned along two axes in order to acquire a range image. A large subset of system within this family is known as TLS. This section will mostly focus on this type of system that is now commonly used as a survey method for monitoring large structures such as bridges and building information modeling. TLS can be used for forensics applications in large environments, and they are regularly used to document cultural heritage sites.

In TLS, the scanning is performed by two rotating components. One control the elevation, while the other determined the azimuth. Many TLS contains a biaxial leveling compensator used to align the coordinate system of the generated range image with respect to gravity.

One configuration that can be encountered uses a galvanometer with a mirror to control the elevation and the azimuth controlled by rotating the complete scanner head. Using this configuration, the elevation can be scanned at a higher frequency than the azimuth. A typical configuration could have a 360-degree field of view in the azimuth and 30 to 120 degrees of field of view in elevation.

In the idealized case, each time a distance measurement is made, the value of the optical encoder of the rotating head and the readout value for the galvanometer are recorded. When the system is calibrated, those values can be converted into angles. Using the distance measurement and the angles it is possible to compute the position of the 3D points.

In the non-idealized case the rotation axis of the scanner head and the one of the galvanometer may not be perpendicular and some small translation offsets can result from the misalignment of the laser and galvanometer with respect to the rotation center of the scanner head.

Pulse-based systems perform a direct measurement of the time required by the light to do round trip between the scanner and the scene. Whereas the phase-based systems perform an indirect measurement of the time by measuring a phase offset.

2.6 Pulsed-based Systems

Pulsed-based system continually pulse a laser, and measures how long it takes for each light pulse to reach a surface within the scene and return to the sensor. Typically, the pulse has a Gaussian shape with a half-beam width of 4 ns to 10 ns. Because the speed of light is known, the range $r(\Delta t)$ of the scene surface is defined as

$$r(\Delta t) = \frac{c\Delta t}{2} \tag{2.1}$$

where c is the speed of light, and Δt is the time between the light being emitting and it being detected. Typically, the detector performed a sampling of the signal every 1 or 2 ns. Different algorithms that perform the detection of pulses in the incoming signal are discussed in Section 2.6.2.

In many applications, the detection of the peak of a pulse with a sub-nanosecond accuracy is critical as the pulse travels approximately 30 cm in one nanosecond.

The simplest implementation assumes that a detected pulse corresponds to the last pulse emitted. In some situation, the ordering of the outgoing pulses may be different from the order of the returning pulses. This can occur in scenes with large depth variations. A pulse can reach a distant surface and by the time the pulse return to the sensor a second pulse is emitted to a close surface and back to the sensor. In order to avoid this situation, the maximum pulsing rate f_p of a pulse-based system is limited by the maximum range R_{max} of the system using

$$f_p \leq \frac{c}{2R_{max}}. \tag{2.2}$$

As an example, a system having a maximum range of 1.5 km is expected to generate at most 100,000 range measurements per second.

2.6.1 Multiple Returning Pulses

A property of pulse-based system is that it may register multiple return signals per emitted pulse. An emitted laser pulse may encounter multiple reflecting surfaces and the sensor may register as many returns as there are reflective surfaces (i.e. the laser beam diameter is not infinitely thin[26]).

This situation is frequently encountered in airborne applications related to forester and archaeology where it may simultaneously register the top of the vegetation and the ground [27, 28, 29]. Note that a pulse can hit a thick branch on its way to the ground and it may not actually reach the ground. For terrestrial applications, the analysis of multiple return signals can sometime allow detecting building behind vegetation or detecting both the position of a building's window and a surface within the building. Finally, some system records the complete return signal which forms a vector of intensity values where each value is associated with a time stamp. Those systems are known as waveform Lidar[30]. Waveform Lidar is capable of measuring the distance of several objects within the laser footprint and this allows characterizing the vegetation structure [28, 31, 32].

Finally, some detected pulses can be the result of an inter-reflection within the scene. This situation is illustrated in Figure 2.2. In this figure, a part of the laser light emitted by the system is first reflected on the ground and then reflected on the road sign before reaching the sensor. Typically, multi-channel and area-based systems are more sensitive to

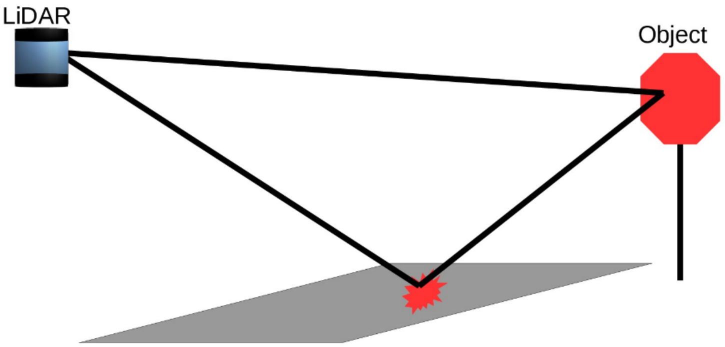


Figure 2.2: Spurious data points induced by inter-reflection

inter-reflection artifacts then point-based systems.

2.6.2 Detecting a Returning Pulse

The detection of the peak of the returning signal with a sub-nanosecond accuracy is critical for many commercially available systems and scanner manufacturers provides few implementation details. Implementation details about the peak detector of experimental systems developed for the landing of spacecraft are discussed in [33, 34]. In [33], the returning signal is convolved with a Gaussian and the peaks are located by examining the derivative. The standard deviation of the Gaussian is derived from the physical characteristic of the system. In [34], a 6 degree polynomial is fitted on the returning signal. Waveform lidar record the complete return signal and make it available to the end user. The end user can implement specialized peak detectors adapted to specific applications. A significant body of knowledge about peak detection for waveform lidar is available [30]. One approach is to model the waveform as a series of Gaussian pulses. The theoretical basis for this modeling is discussed in [35].

2.7 Phase-Based Systems

Phase-based system emits an amplitude-modulated (AM) laser beam [36]. The systems presented in this section are also known as continuous-wave ToF system. Frequency Mod-

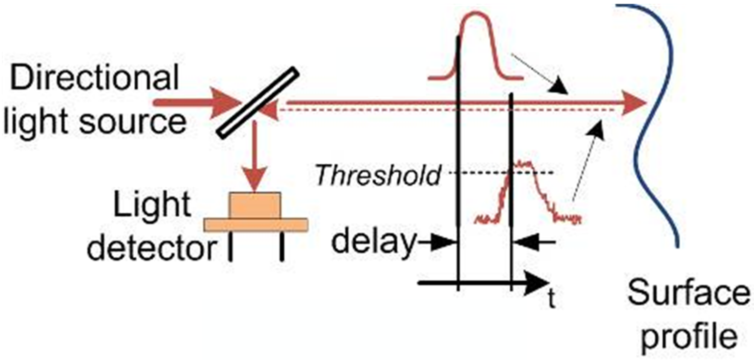


Figure 2.3: Spurious data points induced by inter-reflection

ulation (FM) is rarely used so it is not considered here[1]. The range is deduced from the phase difference between the detected signal and the emitted signal. Fig 2.3 illustrates the principle of the phase-based system.

The temporal intensity profile $I_s(t)$ for the illumination source is

$$I_s(t) = A_s + B_s \cos(\phi(t) + \theta_s) \tag{2.3}$$

where A_s, B_s and θ_s are constant and $\phi(t) = 2\pi t f_{cw}$ where f_{cw} is the modulation frequency. The temporal intensity profile $I_d(t)$ for the detected signal is

$$I_d(t) = A_d + B_d \cos(\phi(t) + \theta_d + \delta\phi) \tag{2.4}$$

where $\delta\phi$ is the phase offset related to the range. Note that A_d, B_d depend on A_s, B_s the scene surface properties and the sensor characteristics. In general, the value of θ_s and θ_d are assumed to be constant but their values are unknown. The conversion from phase offset to range can be achieved using

$$r(\delta\phi) = \frac{c\delta\phi}{4\pi f_{cw}} \tag{2.5}$$

where c is the speed of light in the medium. The range over which the system can per-

[1] FM systems for small measurement volume exist.

form unambiguous measurement can be computed from the modulation frequency. This unambiguous range r_{max} is defined as

$$r_{max} = r(2\pi) = \frac{c}{2f_{cw}}. \tag{2.6}$$

Increasing the modulation frequency f_{cw} will simultaneously reduce the value of r_{max} and the measurement uncertainty.

2.7.1 Measuring Phase Offset

Phase-Based System combined the detected signal with the emitted signal in order to perform the phase offset measurement. At this point, one should realize a similarity with interferometry. A unified presentation of phase-based ToF and interferometry can be found in [37]. Combining the detected signal with the emitted signal is mathematically equivalent to the cross-correlation between eq. 2.3 and eq. 2.4 which results in another sinusoidal function $I_c(t)$ defined as

$$I_c(t) = A_c + B_c \cos(\phi(t) + \theta_c + \delta\phi) \tag{2.7}$$

where A_c, B_c and $\delta\phi$ are unknown and $\phi(t)$ and θ_c are known. Typically, θ_c it is assumed to be zero during the processing of the signal as a non-zero value simply creates a bias in the range measurement that can be compensated by the calibration. By sampling eq. 2.7 three or more times with different values of t_i, it is possible to construct an equation system that allows to solve for A_c, B_c and $\delta\phi$. This is done similarly as described in section 3.4.2 and the details are are not repeated here. When selecting the different sampling time t_i such that $\phi(t_i) = \frac{2\pi i}{N}$ where N is the number of samples,

$$A_c = \frac{1}{N}\sum_{i=0}^{N-1} I(t_i), \tag{2.8}$$

$$B_c = \frac{\sqrt{(\sum_{i=0}^{N-1} I(t_i)\sin(\frac{2i\pi}{N}))^2 + (\sum_{i=0}^{N-1} I(t_i)\cos(\frac{2i\pi}{N}))^2}}{N} \tag{2.9}$$

and

$$\delta\phi = \arctan(\sum_{i=0}^{N-1} I(t_i)\sin(\frac{2i\pi}{N}), \sum_{i=0}^{N-1} I(t_i)\cos(\frac{2i\pi}{N})) \tag{2.10}$$

where $\arctan(n, d)$ represents the usual $\arctan(n/d)$ where the sign of n and d are used to determinate the quadrant.

Removing the Phase Offset Ambiguity

Once the phase difference $\Delta\Phi$ is computed, the range is defined as

$$r(\Delta\Phi) = \frac{c\,(\Delta\Phi + 2k\pi)}{4\pi f_{cw}} \tag{2.11}$$

where k is an unknown integer that represents the phase ambiguity. The value of k must be recovered in order to compute the location of the 3D points. A simple method uses multiple modulation frequencies denote as f_{cw}^i with $i > 0$. When using this scheme, a value of k^i must be recovered for each f_{cw}^i. The lower modulation frequencies are used to remove the range ambiguity while the higher ones are used to improve the accuracy of the measurement. In that scheme, the value of f_{cw}^1 and r_{max} are selected such that k^1 is always zero and the value of $\Delta\Phi^i$ for $i > 1$ can be used to determinate the value of k^{i+1} (assuming that $f_{cw}^i < f_{cw}^{i+1}$).

2.8 Multi-Channel Systems

Typically, a multi-channel LiDAR emits multiple laser beams that are contained within a plane. Each acquisition generates a profile contained within that plane and by modifying the orientation or position of this plane it is possible to generate a range image. Multi-channel LiDAR is now considered by many automakers as a key technology required for autonomous driving and for the remainder , e.g. VLP-16 [2],of this section we will focus on imaging systems typically encounter in autonomous driving vehicles. Two variants of this concept are presented in the remaining of this section. The first one uses physical scanning, while the second one uses a time multiplexing strategy to perform digital scanning. In both cases, the distance measurement is performed using a pulse-based method.

A simple modification to the point-based approach is the integration into a single scanning head of multiple point-based systems having their laser beams contained within a plane. By scanning the head perpendicularly to this plane, it is possible to generate a range image. This configuration limits the lateral resolution and/or field of view along one axis, but allows a higher sampling rate than point-based system. This type of system is well adapted to navigation applications where the horizontal orientation requires a larger field of view and higher resolution than the vertical one.

A typical system for navigation could be generated by rotating 16-channel scanner head, a 360-degree range image that contains an array of 20000×16 3D points . The 3D information along the vertical is used to verify that the proper clearance is available for the vehicle, while the other orientation is used for obstacles avoidance. For some autonomous driving applications, the desired vertical field of view is about 30 degrees.

As shown in Figure 2.4, a 30 degree vertical field of view allows the detection of objects on the road just in front of the autonomous vehicle, approaching hills and it allows the monitoring of the vertical clearance of garage entrances and other structures. For moving vehicle, the limitation of vertical lateral resolution can be compensated by integrating multiple range image into a single point cloud.

2.9 Characterization of ToF Systems

Before presenting the characterization of time-of-flight systems, it is essential to understand the definition of uncertainty, accuracy, precision, repeatable, reproducible and lateral

[2] https://velodynelidar.com/vlp-16.html

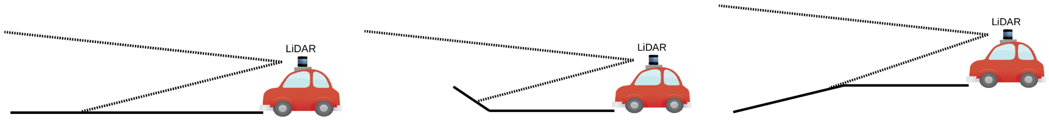


Figure 2.4: Left) A 30-Degree field-of-view collision avoidance system on the top of a vehicle. Middle) Vehicle just before driving uphill. Right) Vehicle just before driving downhill.

resolution. Those terms are not interchangeable as they are not exactly the same.

The uncertainty is the expression of the statistical dispersion of the values associated to a measured quantity. There are two types of uncertainty defined in [38]. Type A uncertainties are obtained using statistical methods (standard deviation), while Type B is obtained by other means than statistical analysis (a frequently encounter example is the maximum permissible error). In metrology, a measured quantity must be reported with an uncertainty.

For a meteorologist, an accuracy is a qualitative description of the measured quantity. However, many manufacturers are referring to quantitative values that they described as the accuracy of theirs systems. Note that often accuracy specifications provided by the manufacturer do not include information about the test procedures used to obtain these values. For many applications the measured quantities need to be geo-referenced and the reader may encounter the terms relative and absolute accuracy. Relative Accuracy is the position of something relative to another landmark. It is how close a measured value to a standard value on relative terms. For example, you can give your location by referencing a known location such as 100KM east of City of Toronto. Absolute Accuracy describes a fixed position that never changes, regardless of your current location. It is identified by specific coordinates, such as latitude and longitude. Manufacturers are usually referring to accuracy specifications of their system as relative and not absolute. Relative measurement is generally better than absolute for a given acquisition.

Intuitively, precision is how close multiple measurements are to each other. Precise measurements are both repeatable and reproducible. You can call it repeatable if you can get the same measurement using the same operator and instrument. It is reproducible if you

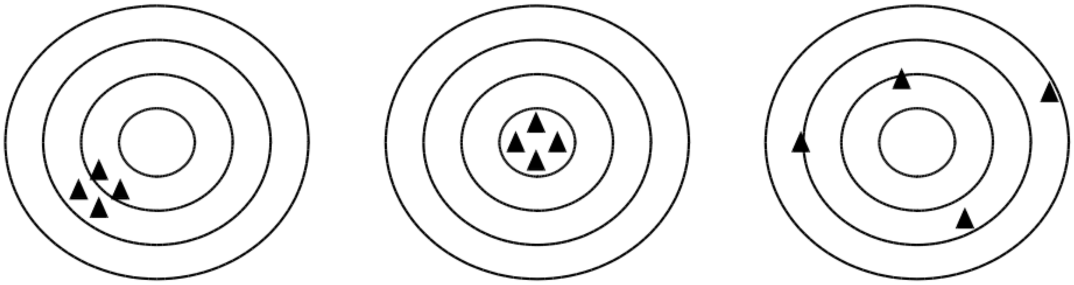


Figure 2.5: The throwing darts represent the measurement process. The position of the throwing darts on the far left board is precise but not accurate, on the middle it is precise and the accurate, and the far right it is imprecise and inaccurate.

can get the same measurement using multiple operators and instruments. In Figure 2.5, the difference between precision and accuracy is shown with a dart board example. Precision is grouping of the shots on average to each other. Whereas accuracy is how close on average to the bullseye are the darts.

Intuitively, the lateral resolution is the capability of a scanner to discriminate two adjacent structures on the surface of a sample. The lateral resolution is limited by two factors: structural resolution and spatial resolution. The knowledge of beam footprint on the scene allows one to determine the structural component of the lateral resolution of the system. The spatial resolution is the smallest possible variation of the scan angle. Increasing the resolution of the scan angle can improve the lateral resolution as long as the spatial resolution does not exceed the structural one.

CHAPTER 3
THEORETICAL FOUNDATIONS

In AV, many important sensors have been utilized, but the two most researched and debated are LiDAR and camera sensors. LiDAR is a light-based RADAR. The sensor sends out short pulses of invisible laser light, and times how long it takes to see the reflection. From this, both the brightness of the target, and how far away is it, with good accuracy can be determined. Some LiDARs installed on mobile platforms are used for acquiring high-resolution mapping of urban environment, while others are used for the navigating purpose of AV.

This chapter summarizes the basic concepts and principles of 3D scans that were obtained using a 16-channel LiDAR, GPS, and IMU sensors installed on top of a automobile vehicle. Simultaneous Localization and Mapping (SLAM) algorithm and all the details of this model needed to generate High Definition (HD) maps for autonomous and active safety applications will be explained.

3.1 Mobile LiDAR Systems (MLS)

The automatic extraction of road layouts from MLS makes possible the availability of detailed digital road maps that contain precise information on widely used geometric parameters, such as horizontal (straight lines, circular curves and clothoids) and vertical (slope, vertical curves and super-elevations). A number of authors recently employed mobile mapping systems (MMS) to derive precise information on horizontal and vertical road parameters.

MLS allows for a wide range of possible applications and features. A surface model can be obtained for planning and design of roads that can be captured using Grafe model [39]. Ai and Tsai presented a method that can correctly detect 94.0% and 91.4% of the traffic signs on both roadways, respectively, with less than 7 false positive cases. Ai and Tsai methods demonstrate that the presented solution using 3D LiDAR point cloud data is promising for traffic sign inventory [40]. In 2011, El-Halawany and Lichti presented a research to automatically determine the position and the dimension of the poles. The data are 3D point cloud collected by a vehicle-based laser scanning system named TITAN. This method is based on the eigenvalue analysis of the covariance matrix in a local neighbourhood [41]. In 2012, Yang et al. presented a novel approach to extract automatically road markings from mobile LiDAR point clouds. The method generates georeferenced feature image of the point cloud that isolate the points of road surfaces. An algorithm is then used to separate these laser beams within a range according to their strength of reflection [42]. In 2013, Wang et al. is focused on obtaining point cloud data on excavation volume extraction for road construction through MLS data processing [43].

3.2 Static Scan

In Figure 3.1, a 360-degree range and a front facing camera image are presented. These were obtained while the automobile vehicle was parked. Generally, cameras must deal with lighting variation and objects are moving shadows. This is a significant disadvantage of cameras. LiDAR can provide a 3D representation of the surrounding environment independent of ambient light.

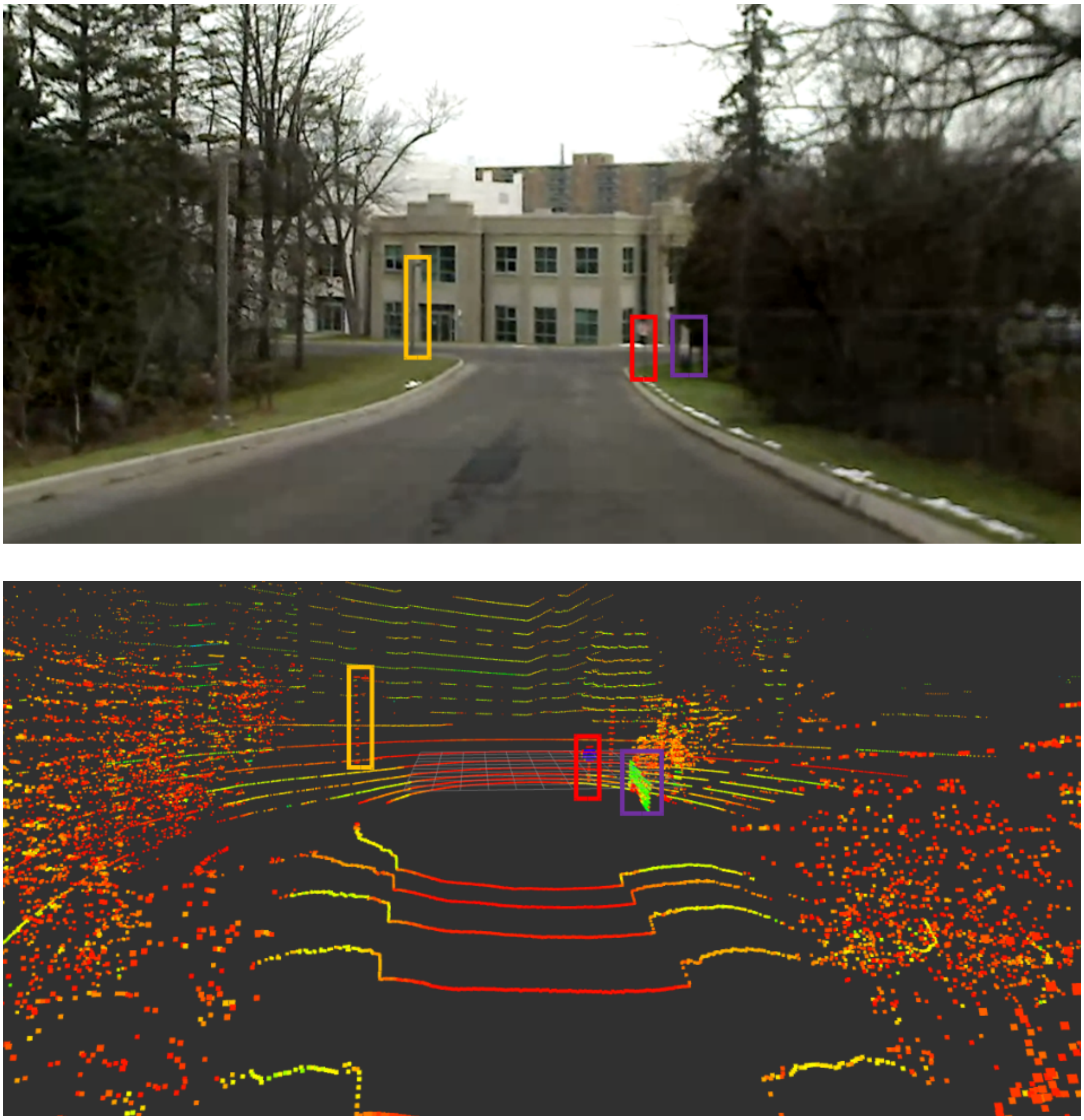


Figure 3.1: Snapshot of an area using a camera (top) and a view of the same area using VLP-16 LiDAR (bottom). Note the stop sign, the Street Sign the street light pole contained respectively in the red, purple and yellow rectangles.

Another example of a 360-degree scan is presented in Figure 3.2. The accompanying image was generated using the panoramic feature on a smart phone.

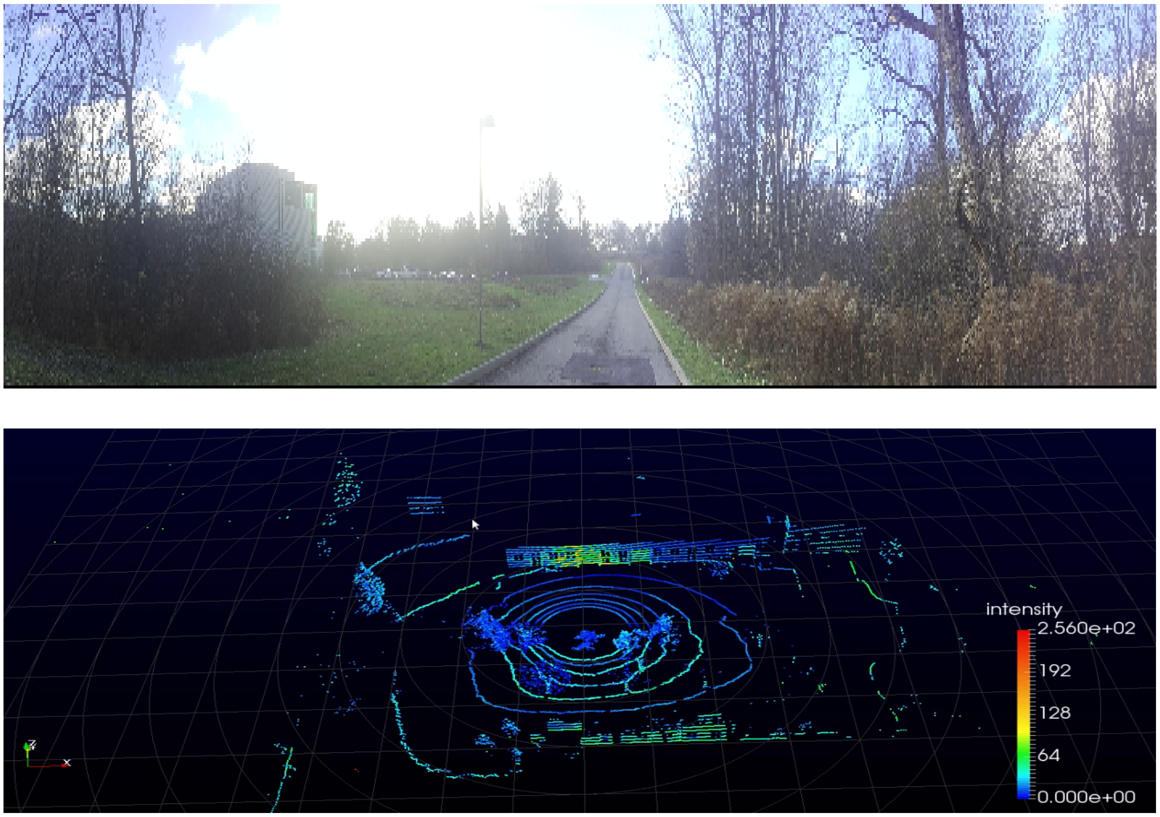


Figure 3.2: Top) a panoramic picture taken by a smartphone. Bottom) 3D data collected by a Velodyne VLP-16 LiDAR from the same location.

3.2.1 Simultaneous Localization and Mapping (SLAM)

A typical robot integrated with a SLAM system will build a model of the surrounding environment and estimate its trajectory simultaneously. SLAM system relies on several key algorithms, like feature extraction, registration and loop closure detection. SLAM is one of the main challenges in robotics is navigating autonomously through large, unknown, and unstructured environments. In recent years, SLAM became an important research topics that has been investigated heavily [44]. In autonomous vehicles and robotics applications, both cameras and the LiDAR can be used for localization [45]. Laser scanners have the advantage of being independent of external lighting and making use of full 3D representation. Figure 3.2 shows the result of a building a map of the environments using Velodyne VLP-16 LiDAR, which was previously used. Note that the SLAM method that produced this result exploited the 360-degree field of view of the VLP-16 LiDAR and the availability odometry data.

Currently the most robust implementations of SLAM leverages LiDAR systems to relate a 360-degree field of view 3D point clouds to 6D spatial points (x,y,z-position, and the roll, yaw, and pitch angles) with the assistance of auxiliary odometry sensors. Figure 3.3 explains the concept of SLAM [46].

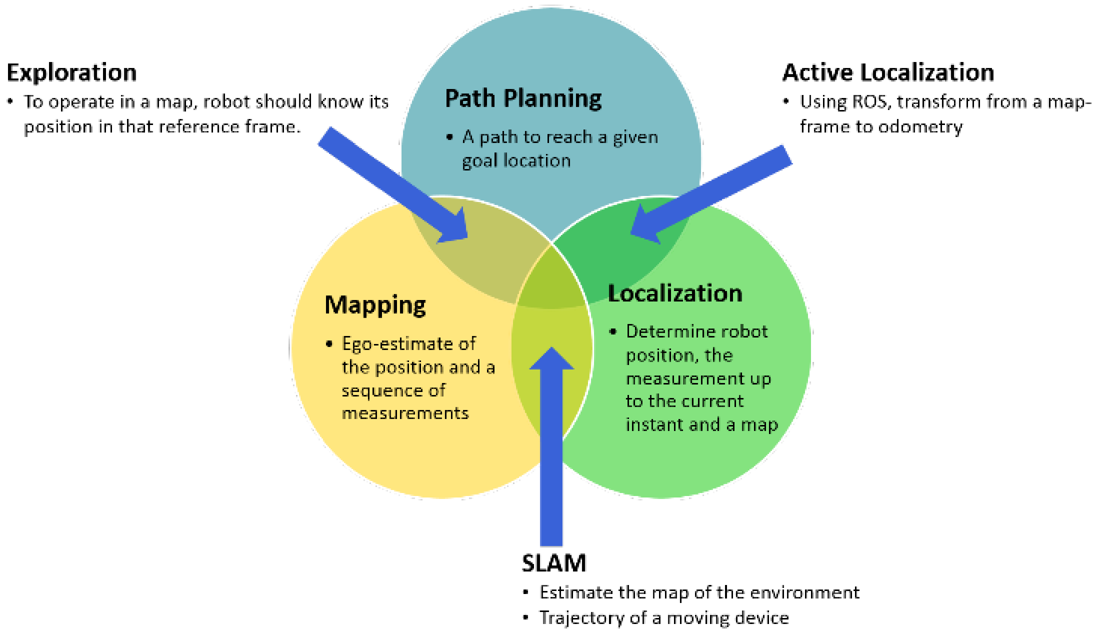

Exploration
• To operate in a map, robot should know its position in that reference frame.
Path Planning
• A path to reach a given goal location
Active Localization
• Using ROS, transform from a map-frame to odometry
Mapping
• Ego-estimate of the position and a sequence of measurements
Localization
• Determine robot position, the measurement up to the current instant and a map
SLAM
• Estimate the map of the environment
• Trajectory of a moving device

Figure 3.3: SLAM is the process by which a robot builds a map of the environment and, at the same time, uses this map to compute its location.Typical, the robot reports its (x,y) position in some Cartesian coordinate system and also reports the current bearing/heading.

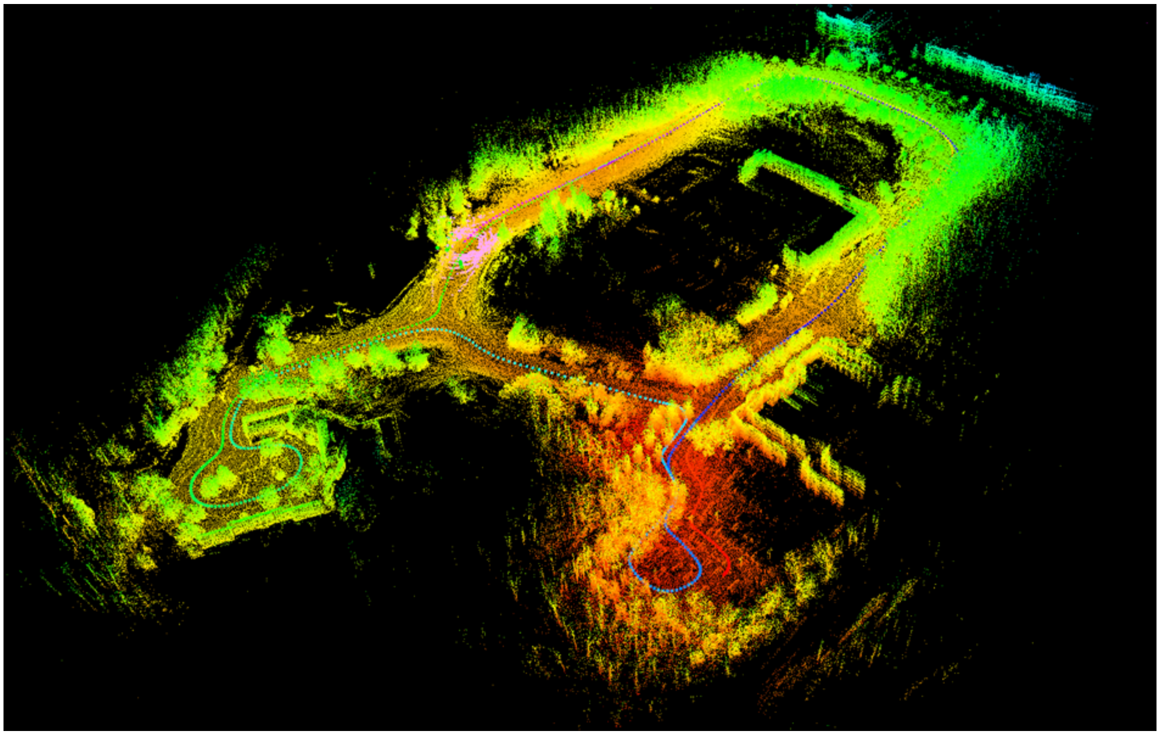


Figure 3.4: 3D Map of the Collip Circle road located in London (Canada) generated using a SLAM Algorithm. The trajectory of the vehicle is shown using a dotted line.

3.3 Sensors Fusion and Navigation

Sensor fusion is the process of integrating data from different sensors in order to construct a more accurate perception of the environment than the ones obtained using the independent sensors. In recent year, progress has been made in the development of autonomous driving vehicle and Advanced Driver-Assistance System (ADAS). LiDAR is an important sensor that made this progress possible. Nevertheless, LiDAR data must be fused with other sensors' data in order to improve the situational awareness and the overall reliability and security of autonomous vehicles. see figure 3.5.

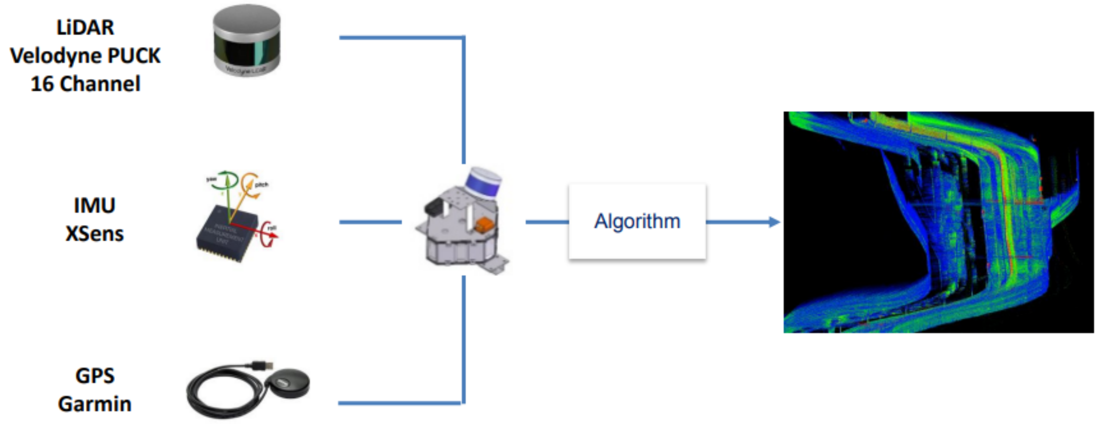


Figure 3.5: Combination of multiple sensors to generate a HD map to be used in AV applications. The sensors are not accurate so the robot will not exactly know its location. nevertheless, the fusion of those sensors estimate better than relying on odometry alone.

For autonomous driving, ADAS and navigation application sensors may include stereo camera, camera,radar, sonar, LiDAR, GPS, Inertia Navigation System (INS), *etc.* Radar is one of the most reliable sensor. It can operates through various conditions such as fog, snow, rain, and dust when most optical sensors fail. However, radar has a limited lateral resolution. Cameras are cheap and have high resolution, but are affected by ambient illumination. Stereo camera are inexpensive to purchase, but are computational expensive and do not cope well with low-texture areas. Sonar is useful for parking assistance, but has a limited range. The main disadvantage of sensor fusion is that different sensors can have incompatible perceptions of the environment: some may detect an obstacle while others may not.

In autonomous systems and ADAS, all sensors data are feed to the Data Acquisition System (DAQ). The gathering of the data by the DAQ can also be used for testing, developing, improving efficiency, ensuring reliability and safely. Figure 3.6 explains the collaboration of the main sensors in autonomous or ADAS applications. The sensor fusion process starts by using information provided by GPS, INS, and the odometer in order to determine the absolute position and orientation of the vehicle. Then sensors information from cameras and LiDARs are compared to known maps that are downloaded from the cloud (Web Security Services). The cloud-based information is typically refereed as High Definition (HD) map and it allows the driving to adapt to potential obstacles, including traffic, in order to avoid potential accidents. An HD map consists of geo-spatial coordinates of the static elements that describe roadway features such as lane markings, traffic signs, landmarks, road-edges, etc. In order to construct these maps, special mapping vehicles are augmented with a number of high-accuracy perception sensors [47]. This allows for acquisition of a geo-spatial representation of the roadways that can be used by autonomous vehicles or ADAS that can use cheaper perception sensors. However, this HD-map approach creates significant data storage, computation, data delivery, and cyber-security issues that fall outside the scope of this chapter.

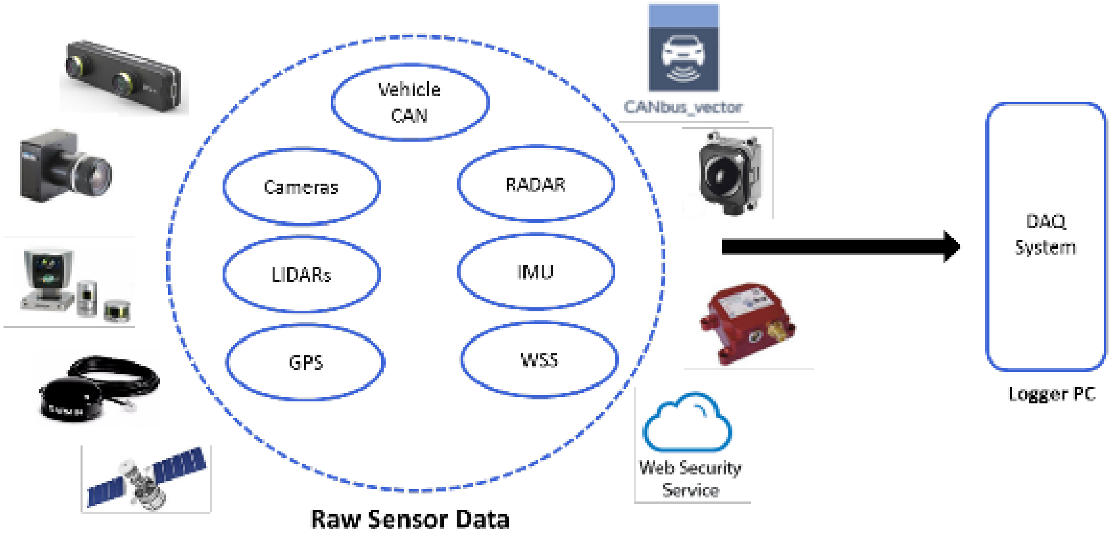


Figure 3.6: Data acquisition is a sampling process that measures real world physical conditions and converting the resulting samples into a digital representation recorded on the Data Acquisition System (DAQ). In automotive applications, the data from the different sensors are typically transferred on a data bus known as a Controller Area Network or CANbus.

The Global Positioning System is not as accurate as the Global Navigation Satellite System/Inertia Navigation System (GNSS/INS) that is heavily used in various autonomous applications. A GNSS receiver provides accurate position and time to the navigation system. When the GNSS signal deteriorates, the INS sensor provides the position and navigation until the GNSS conditions improve. A GNSS/INS is expensive and, for some applications, GPS is the only economically viable geo-location solution. To reach a high accuracy potential, LiDAR system must be well calibrated and equipped with a high-end GNSS/INS navigation unit. Typical LiDAR sensor range accuracy is 1-5 cm; GPS accuracy is 2-5 cm; INS accuracy for pitch/roll is 0.005 degree and for heading is 0.008 degree with the laser beam divergence being 0.25 to 5 milliradian. However, the final vertical and horizontal accuracies are in an order of 5 to 15 cm and 15-50 cm (one sigma). Note that; some systematic errors may be due to miss-alignment of the laser with respect to the vehicle roll, pitch and yaw axis or a measurement error of the relative position of the GPS antenna with respect to the INS reference system.

CHAPTER 4
POINT CLOUD FILTERING

4.1 Introduction

This chapter will tackle an issue that faces multiple applications in automotive, manufacturing, agriculture, construction, and other industries that require scanning of complex three-dimensional object, interpretation, and geometric 3D reconstruction of real-world environments to incorporate them into a computer-based processing. A large amount of data laying on objects' surfaces is called point cloud set representation of the boundary of 3D objects and generates a large population of data points. In addition, handling a large number of points produces error distortion that causes error-estimation. The developed and implemented approach for this issue is based on a point cloud filtering techniques that reduce the amount of noise and outliers in large-scale urban point cloud data sets derived from remote sensing.

4.2 Related Work

Point cloud data obtained through a computer vision is unbalanced in density and it includes a lot of noise and outliers. This issue will greatly reduce point cloud search efficiency and affect the surface reconstruction. Filtering routine is one of the key aspects that needs to be considered when working with point cloud data.

The development of a laser-scanning device and enhancement of computer vision technology and point cloud technology has been widely used in surface reconstruction and 3D simulation. Massive cloud data collected are very dense, and usually not uniform because of the interference factors with the superposition of many outliers and noises, which will seriously affect the subsequent work such as point cloud data search or 3D reconstruction process. Filtering for point cloud data mainly includes the algorithms based on mathematical morphology, the triangulation, wavelet transform etc. Structured light vision systems have been successfully used for accurate measurement of 3D surfaces in computer vision. However, their applications are mainly limited to scanning stationary objects [48, 49]. Chen et al. (2008) presented an idea for real-time acquisition of 3D surface data by a specially coded vision system. To achieve 3D measurement for a dynamic scene, the data acquisition must be performed with only a single image.

Light detection and ranging (LiDAR) technology allow rapid and inexpensive measurements of topography over large areas. This technology is becoming a primary method for generating high-resolution digital terrain models (DTMs) that are essential to numerous applications such as flood modeling and landslide prediction. Airborne LiDAR systems usually return a three-dimensional cloud of point measurements from reflective objects scanned by the laser beneath the flight path. In order to generate a DTM, measurements from nonground features such as buildings, vehicles, and vegetation, they must be classified and removed. In 2003, Zhang et al.published a progressive morphological filter that was developed to detect nonground LiDAR measurements. By gradually increasing the window size of the filter and using elevation difference thresholds, the measurements of vehicles, vegetation, and buildings are removed, while ground data are preserved. Datasets from mountainous and flat urbanized areas were selected to test the progressive morphological filter. The results show that the filter can remove most of the nonground points effectively [50].

A point obtained by laser altimetry represents points from not only the ground surface but also objects found on it. For civil works applications points representing the surface of non-ground objects must be removed from the point set in a filtering process. In 2001, Sithole described modifications made to an existing “slope based” filtering algorithm, and presents some results obtained from the use of the filter. Sithole resampled the discrete point cloud to generate regular grid data and then filtered the data using the method of image processing. This method is superior in terms of processing speed, and can make use of mature image processing theory. The shortcomings are yet the need of interpolation and resampling for the point cloud data, which will lose some accuracy [51].

Many filtering techniques have been presented, but as technology evolves, more research is needed. As the outlier detection is an integral part of data, it has attracted great attention. [52] proposed a new method for evaluating outlierness called the Local Correlation Integral (LOCI). It offers the following advantages and novelties: (a) It provides an automatic, data-dictated cutoff to determine whether a point is an outlier in contrast. (b) It can provide a LOCI plot for each point; the plot summarizes many information on the data, determining clusters, micro-clusters, diameters and inter-cluster distances. None of the existing outlier-detection methods can match this specific feature, because they results with a single number for each point that is outlierness, (c) The LOCI method can be computed as quickly as the best previous methods, (d) Moreover, LOCI leads to a practically linear approximate method, and aLOCI (for approximate LOCI) provides fast highly-accurate outlier detection. Domestic and foreign scholars have studied the identification and filtering of outliers of the discrete point cloud. Recognition methods of outlier are mainly based on depth, distribution, distance, and density [53].

Similar to this current situation, Breunig [54] put forward Local Outlier Factor (LOF) that can deal with general scattered point cloud data and must first estimate outlier density according to the point cloud density change. It is local in that the degree depends on how isolated the object is with respect to the surrounding neighborhood. These references about filtering algorithm are based mostly on the point cloud data acquired from LiDAR. Using real-world datasets demonstrates that LOF can be used to find outliers that appear to be meaningful, but can otherwise not be identified with existing approaches. However, domestic and foreign research on he point cloud data based on computer vision is still in its infancy stage both domestically and internationally. [55]

Due to the various questions and the problematic noise of 3D point cloud data, a novel method is proposed to accurately filter series of 3D scans. It is developed specifically to create high definition maps for autonomous and active safety applications but it is not limited to be used on other applications such as 3D imaging models. This model presents a method of filtering point cloud data to reduce noise and unwanted elements while maintaining desirable geometric features. The data, as it is presented in Euclidean Velodyne coordinate, includes intensity and channel number and uses Cartesian coordinates of X, Y and Z in the domain of real numbers. To be able to deal with large sparse data, a voxelization method was created that can store necessary point cloud data with calculated ”density” of each box; depending on the threshold, can either keep or delete the points inside the box.

4.3 Major Filters

To reach the final result, multiple algorithms were tested and known filters for our environment. The work done is mainly to contain the typical errors and the natural variations or unexpected changes in the behaviour of the system. Some of the major filters for point cloud data and the reasons for a new solution are provided.

4.4 Parametric Model

Projecting point cloud data on a parametric model have been used with increasing frequency in many computer vision tasks such as optical flow calculation [56, 57], range image segmentation [58, 59], estimating the fundamental matrix [60, 61], and tracking [62]. Figure 4.1 displays the flowchart of this model that uses a projection model called ”ProjectInliers” and a set of inlier indices from a point cloud to project them into a separate point cloud. A basic filter method that implements the need to set the output such as data, row step, point step, width, height, and density is used. Then, the ProjectInliers object and Model Coefficients defines projection onto the model to project. The final step is to show the content of the projected cloud.

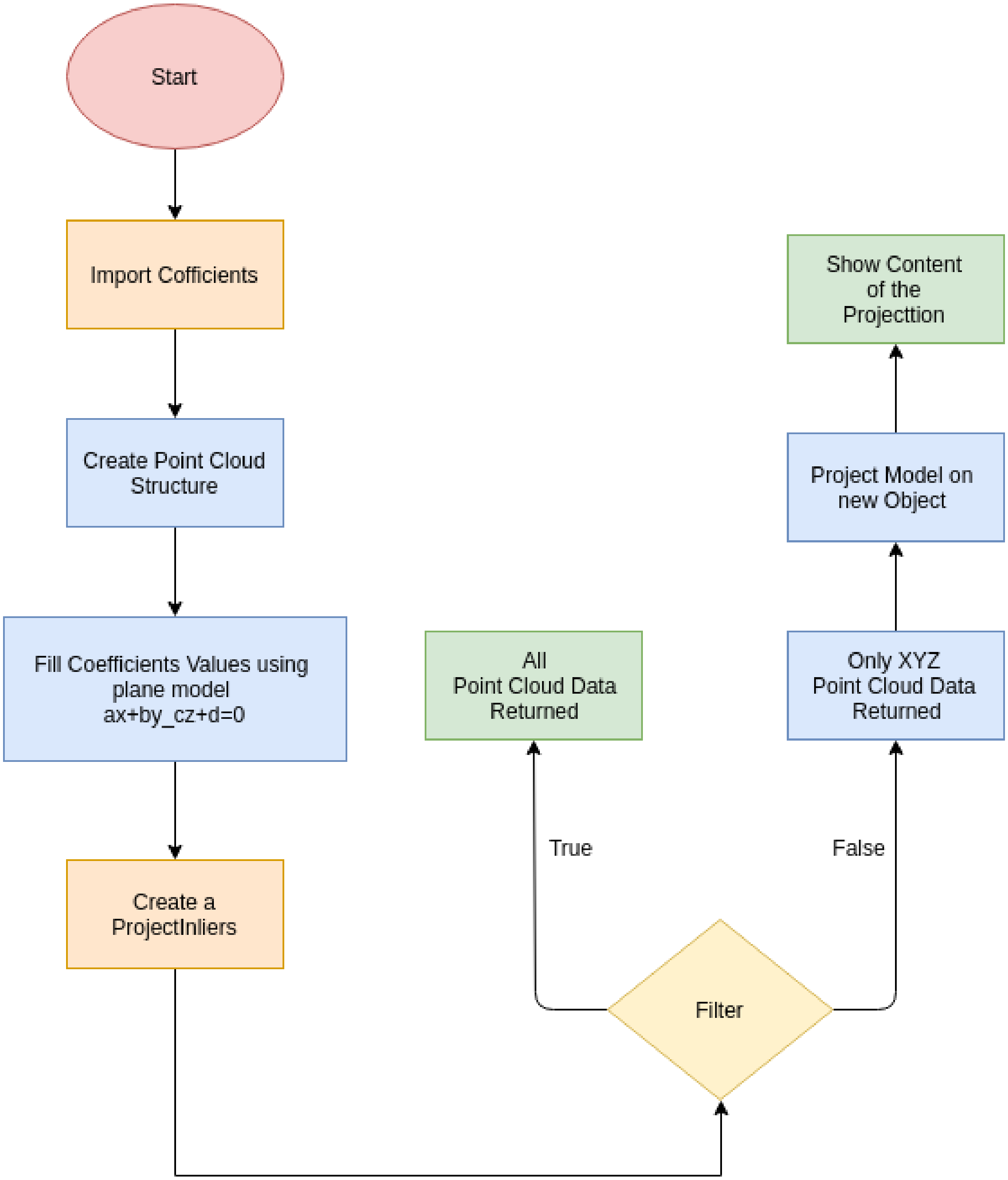

Start
Import Cofficients
Create Point Cloud Structure
Fill Coefficients Values using plane model ax+by_cz+d=0
Create a ProjectInliers
Filter
True
False
All Point Cloud Data Returned
Only XYZ Point Cloud Data Returned
Project Model on new Object
Show Content of the Projecttion

Figure 4.1: A Step-by-Step Flowchart of the Parametric Model on Point Cloud Data

In other words, Parametric Model Filter projects all the point clouds data onto a surface. This filter is currently compatible with only flat planes, but future updates may include parabolic, hyperbolic, spherical, and any other figure that can be written as a standard multi-variable equation.

A graphical display of the projection process is shown in details in Figure 4.2. The first step is to import point cloud data location as a parameter and the project onto a given surface. After generating the surface, then all points above or below a given plane are eliminated. This will provide a filtered point cloud that matches only what is selected.

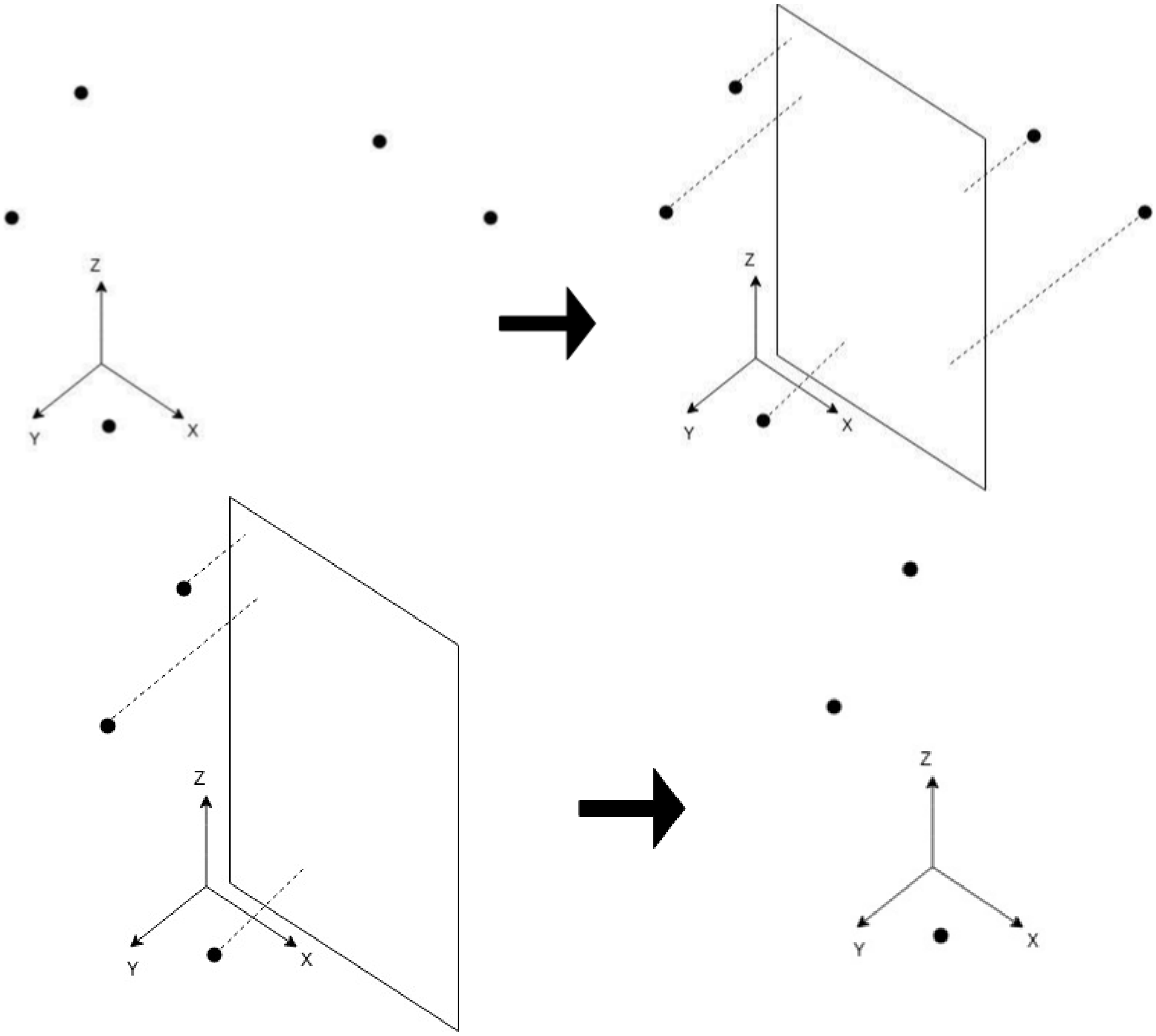


Figure 4.2: An example of a Parametric Model Filter that uses a surface plane to filter certain points

The following parameters were used in the example:

1. Name: “Parametric Filter”

2. Input Vector

    - Minimum Number of points (4)

        - ax + by + cz = d

        - (a, b, c, d) are the parameters

3. Display summary

4. Display PCL

Listing 4.1: Parametric Model Example Command and output

```
#include "PCL_Filters.h"
int main(int argc, char **argv)
{
    std::vector<PCL_Filter> input = {
    PCL_Filter("Parametric Filter", {0, 0, 1, 3}, true, false),
    // 0x + 0y + z = 3 };
    filters = PCL_Filters(input, "/home/ismail/summary.csv");
   ROSRUN(argc, argv, "/home/ismail/bagfiles/filtered_PCL.bag");
return EXIT_SUCCESS;
}
```

4.5 Passthrough Filter

In automotive and road information field, the precision is limited due to 1) a lack of direct high-density three-dimensional (3D) coordinate information, 2) shadows caused by trees and moving vehicles, and 3) visibility of road surfaces and subsidiaries. [63]

Passthrough filter input points through certain constraints are based on a particular field. It iterates through the entire point cloud once, performing two operations. First, it removes non-finite points. Second, it removes any points that lie outside the specified interval for the specified field. For example, a programmer is able to set the field so it refers to the z-dimension (depth) and set the limit so that the filter removes any points that are half a meter away from the sensor. [64]

The aforementioned studies mentioned used feasible methods to extract road characterizations and to investigate the potential of MLS in road applications. Wang et al. (2018) findings have contributed to more comprehensive approaches for describing and checking road networks. Inspired by these investigations and suggestions, estimated road curbs using GPS trajectory-based segmentation and extracted road markings with a passthrough-statistical-radius-filter (multi-filter) in the expressway point clouds. [63]

In [50], a study was done on airborne LiDAR technology that allowed rapid and inexpensive measurements of topography over large areas. It provided a primary method for generating high-resolution digital terrain models (DTMs) that are essential to numerous applications such as flood modeling and landslide prediction. A complete description of the algorithm can be found in the article. For this purpose a modified algorithm was used to apply it on road information from vehicle level. Figure 4.3 presents the modified version of the passthough filter to extract road curbs and road markings from mobile laser scanning (MLS) point clouds.

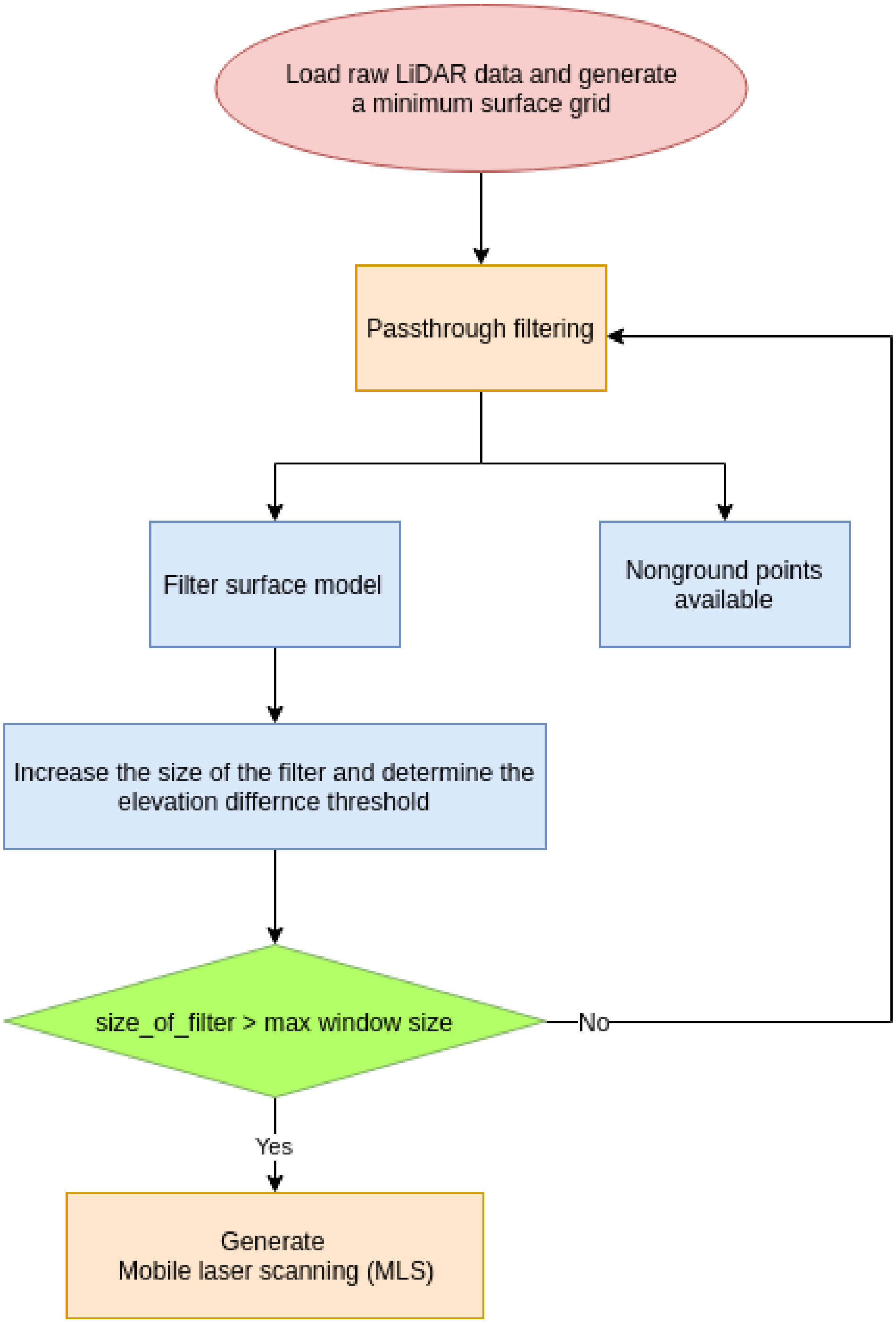

Load raw LiDAR data and generate a minimum surface grid
Passthrough filtering
Filter surface model
Nonground points available
Increase the size of the filter and determine the elevation differnce threshold
size_of_filter > max window size
No
Yes
Generate Mobile laser scanning (MLS)

Figure 4.3: A Step-by-Step Flowchart of the Passthrough Filter for MLS Point Cloud Data

In general, a passthrough filter is used to eliminate all points above or below a set multi-variable threshold. This filter was used only on filtering below the XY plane as shown Figure 4.4.

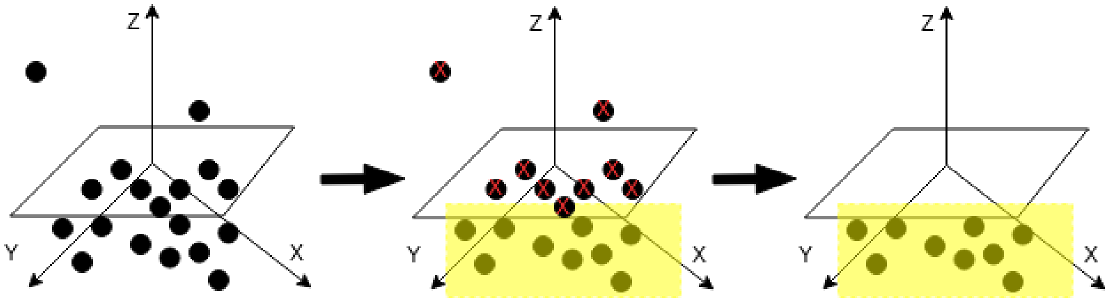


Figure 4.4: Passthough filter eliminates all points in selected plane

The following parameters are used in the example. Please note that this filter only works on filtering the XY plane:

1. Name: “Passthrough Filter”
2. Input Vector
    - Z1, Z2
3. Display summary
4. Display PCL

Listing 4.2: Passthrough Filter Example Command and output

```
#include "PCL_Filters.h"
int main(int argc, char **argv)
{
      std::vector<PCL_Filter> input = {
      PCL_Filter("Passthrough Filter", {-10, 0}, true, false)
          ,};
      filters = PCL_Filters(input, "/home/ismail/bagfiles/
          summary_1.csv");
```

```
    ROSRUN(argc, argv, "/home/ismail/bagfiles/filtered_PCL_1.bag");
return EXIT_SUCCESS;
}
```

4.6 K-means Clustering

A method based on an algorithm able to classify or group a set of (3D) points into K groups using attributes/features. The grouping is done by minimizing the sum of squares of distances between a point and the corresponding cluster centroid. The original K-means algorithm presented by MacQueen et al. (1967) [65] was then exploited for point clouds by various researchers (Comaniciu and Meer, 2002 [66]; Lavoue et al., 2005 [67]; Yamauchi et al., 2005 [68]; Zhang et al., 2008 [69]; LeCun et al., 2015 [70].

K-means Clustering is the number one filtering option for compact clusters. The K-mean clustering algorithm is used for partitioning the high-dimensional data, as each partitioning is considered a cluster. It is sensitive to outliers and noise and uses only numerical attributes. The input of K-means algorithm is a set of feature vectors which can be defined by X as a set of $x_1, x_2, ..., x_N$ and another input is the number of clusters to be detected the algorithm which is defined by the variable K and also the convergence threshold ς.

$$X = x_1, x_2, ..., x_N \tag{4.1}$$

The k-means cluster algorithm works in the following 5 steps:

Step 1: Define an initial (random) solution as vectors of means. This solution is defined randomly and it can be defined by the following vector of means using the first iteration at $t = 0$ which is composed by K, per this equation:

$$m(t = 0) = [m_1, m_2, ...m_K]^T \tag{4.2}$$

Step 2: This step is used to created a solution to classify each input data according to $m(t)$. In classifying all input data, there were going to be too many input data with a class that is associated to one of the K classes.

Step 3: The classification obtained in Step 2 is used to re-compute the vectors of means in order to compute the vector at 4.3

$$m(t + 1) \tag{4.3}$$

Step 4: Updates the time stamp to 4.3 and then create a test to check the convergence.

$$t = t + 1 \tag{4.4}$$

Step 5: In the last step, a test was made to check the convergence or continue to recompute the means up to conversions using the following code:

Algorithm 1 K-means Clustering Algorithm

```
if $||m(t) - m(t-1)|| < \varsigma$ (convergence) then
    $m(t)$ as a solution
else
    Go back to Step 2
```

In other words, K-means filter takes the point cloud and searches for all the neighbors to every point and if there is a certain number of points which in total is below the threshold, that point is filtered out.

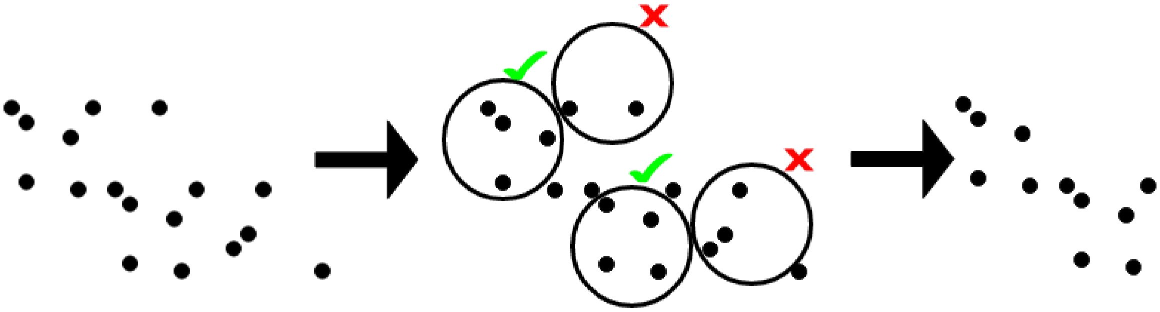


Figure 4.5: K-means filter takes the point cloud and searches for all the neighbors to every point

Figure 4.5 is a simplified version of the algorithm that was run for the test. First, define the collected point cloud data using circular shape and the threshold of number of points. Note that in 3D space, the circle refers to that of a sphere. Then, check the number of neighbors for every point, and if they do not have at least the minimum number of points as in the threshold, then delete that point. The following parameters were used in this project:

1. Name: “K Means Filter”
2. Input Vector
    - Minimum Number of points
    - Radius of Search
3. Display summary
4. Display PCL

Listing 4.3: K-means Clustering Example Command and output

```
#include ”PCL_Filters.h”
int main(int argc, char **argv)
{
    std::vector<PCL_Filter> input = {
        PCL_Filter(”K Means Filter”, {25, 0.05}, true, false),};
        filters = PCL_Filters(input,”/home/ismail/summary.csv”);
        ROSRUN(argc, argv, ”/home/ismail/bagfiles/filtered_PCL.bag”);
return EXIT_SUCCESS;
}
```

4.7 Voxel Grid Filter

Standard Voxel Grid filter assembles a 3D box over the entire input data. This means that a set of cubes is placed over the entire point cloud. For each individual voxel, the points that lie within are down-sampled with respect to their centroid. This approach has a few drawbacks: (i) it requires a slightly longer processing time as opposed to using the voxel center, (ii) it is sensitive to noisy input spaces, and (iii) it does not represent the underlying surface accurately [71]. This filter is good for rapid non-specific filtering technique of large point cloud data set.

In this example, a cuboid was used for symmetry reasons and to avoids fitting problems while grouping and also minimized the effect of voxel shape during feature extraction. Multiple test were run for the voxel size, but the actual voxel sizes varied according to the maximum and minimum values of the neighboring points found along each axis to ensure the profile of the structure. Figure 4.6 shows the 2D voxel grid that average point cloud data in each generated box.

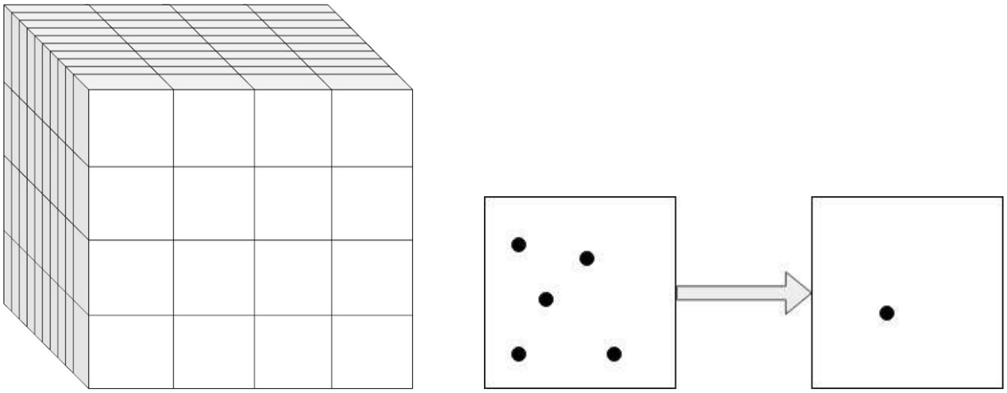


Figure 4.6: Converting the point cloud into a 3D grid. (left) This is the first step that divide the point cloud data into a grid with boxes of a given dimensions. (right) Then, average all the points within each individual box for a singular output.

Figure 4.7 represents a 2D voxel filter over a 2D point cloud containing uniformly dispersed points. Each axis was divided into 5 regions resulting in 25 voxels. The points are down-sampled by taking the centroid of the points within each voxel. The centroids are shown in red asterisks. The centers of the voxels are shown in green squares [72].

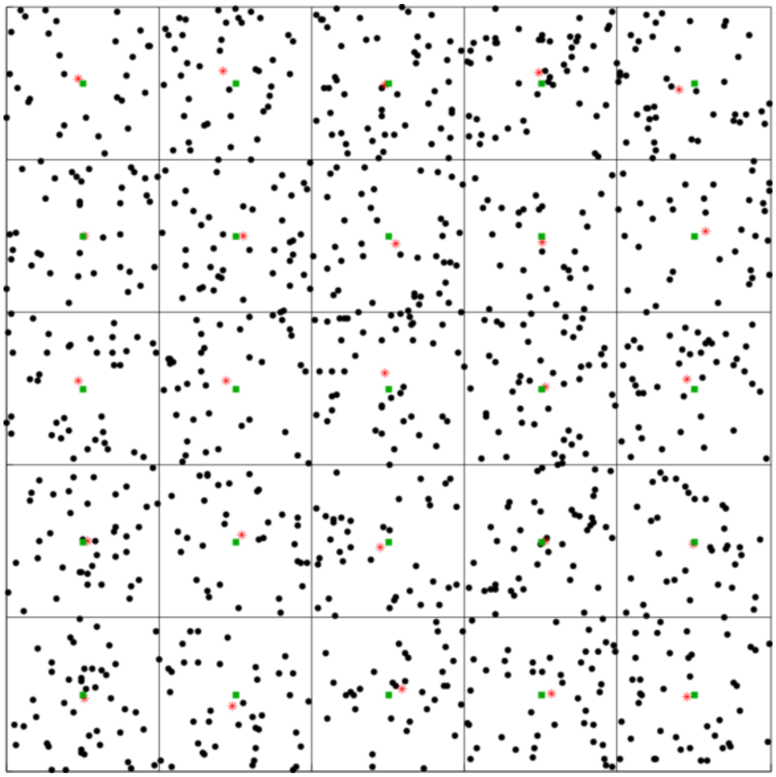


Figure 4.7: A 2D Voxel Grid with red stars representing the voxel centroids

Figure 4.8 explains the general process of such filter. The voxel grid filters downsamples the point cloud data by taking a spatial average of the points in the cloud. This method can be applied in 2D dataset.

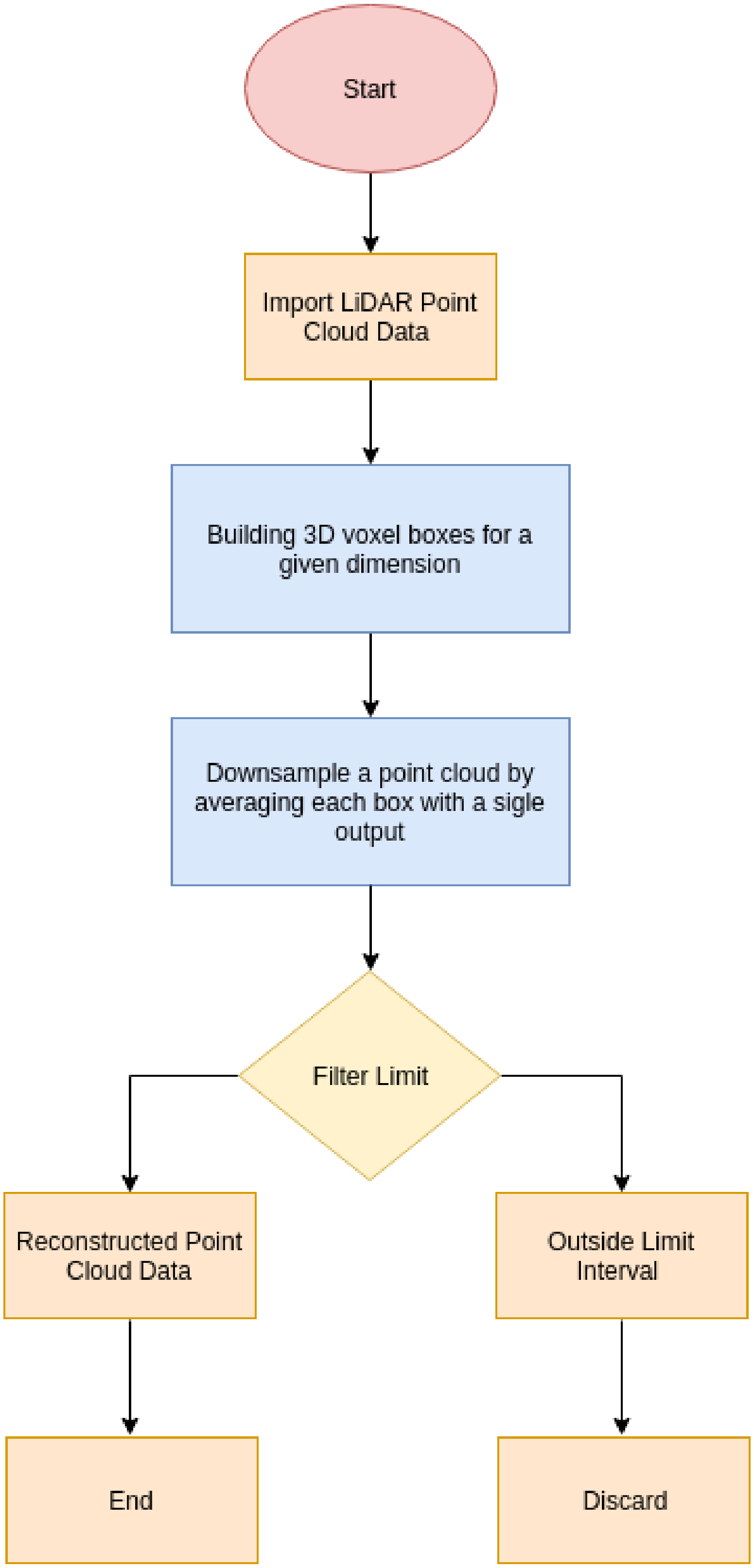

Start
Import LiDAR Point Cloud Data
Building 3D voxel boxes for a given dimension
Downsample a point cloud by averaging each box with a sigle output
Filter Limit
Reconstructed Point Cloud Data
Outside Limit Interval
End
Discard

Figure 4.8: Voxel grid filter process associated filter based limit

The parameters used in this voxel filter are specifically for point cloud data. Using cubical representation for further processing has distinct advantages in data size, processing time, noise suppression, and easier volumetric manipulations. Using a voxel grid for spatial representation can dramatically reduce data size, when compared to representing the same space with points, since one voxel can represent numerous points. Depending on accuracy requirement, a voxel grid is a down-sampled version of a point cloud. The following parameters were used in this example:

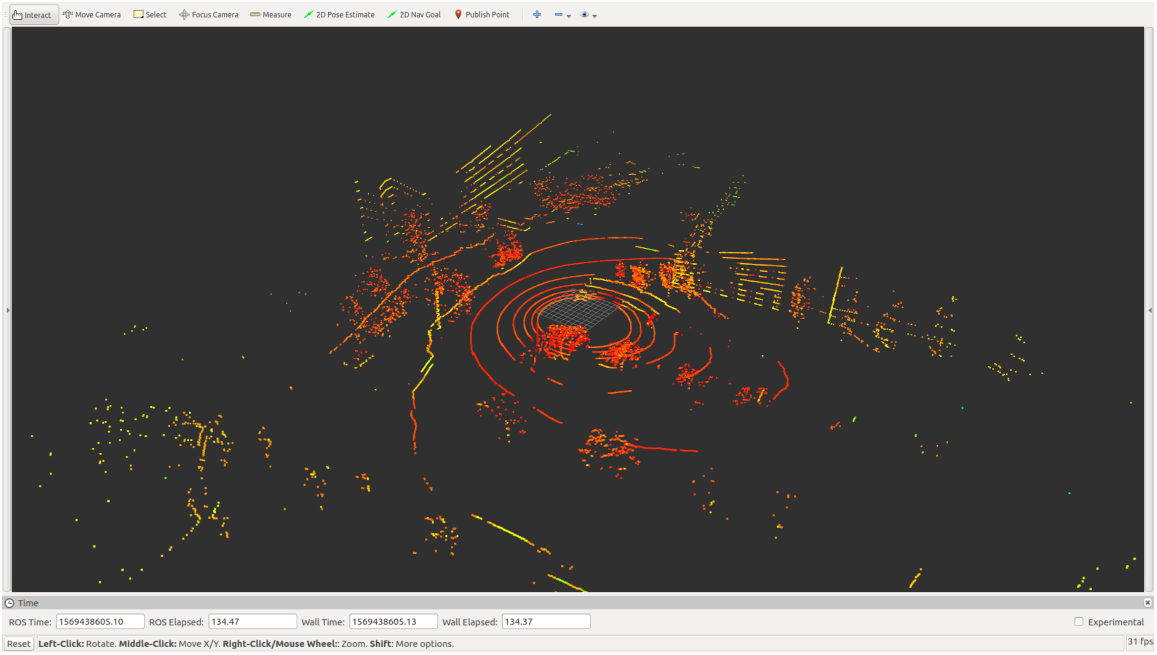
1. Name: "Voxel Grid Filter"
2. Input Vector
   - Length
   - Width
   - Height
3. Display summary
4. Display PCL

Listing 4.4: Standard Voxel Example Command and output

```
#include "PCL_Filters.h"
int main(int argc, char **argv)
{
    std::vector<PCL_Filter> input = {
    PCL_Filter("Voxel Grid Filter", {0.75, 0.75, 0.75}, true
        , false),};
    filters = PCL_Filters(input,"/home/ismail/summary.csv");
    ROSRUN(argc, argv, "/home/ismail/bagfiles/filtered_PCL.
        bag");
return EXIT_SUCCESS;
}
```

4.8 Hybrid Filter

In automotive industry and point cloud engineering design, one of the commonly used filters for point cloud data is quantized in a process known as voxelization. While results have been rather impressive, a 3D representation is inherently cubic and can quickly become unmanageable as the point cloud grows, even with optimizations. Further-more, most of the computations are wasted as the 3D grid is very sparse, i.e., most of the volume is empty. Figure 4.9 shows a raw representation of a road in London, ON. Please note that this sensor's motor is set to rotate 1200 RPM. It can be changed but the data collection was run with this setting. One rotation, in Velodyne's sensors, can be referred to as a single "frame" of data, beginning and ending at approximately 0° azimuth. The number of frames per second of data generated depends entirely on the RPM setting, e.g. 600 RPM / 60 s/min = 10 frames per second.



ROS Time: 1569438605.10 ROS Elapsed: 134.47 Wall Time: 1569438605.13 Wall Elapsed: 134.37 □ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options.

Figure 4.9: Raw point cloud data presented using 16 channel Velodyne LiDAR

The hybrid solution is utilizing Voxel Grid down-sampling and K-means techniques combined into one filter based on the input space sampling using a grid of 3D voxels. A threshold for erasing all unnecessary data within a give box was implemented. In other words, this filter divides the point cloud into multiple grids. Every box's 'density' will be calculated [number of points within box / volume of box] and will only be kept if the box's density is above a given threshold. The following figures display a simplified version of the process for general understanding. This study utilized and combined multiple solutions to provide the best optimal solution without losing importance of the data. Figure 4.10 shows the first step to convert the point cloud data into boxes. In Figure 4.11 a threshold of the density required was set for the data. Figure 4.12 shows the application of the threshold and then reconstruction of the boxes back together in Figure 4.13.

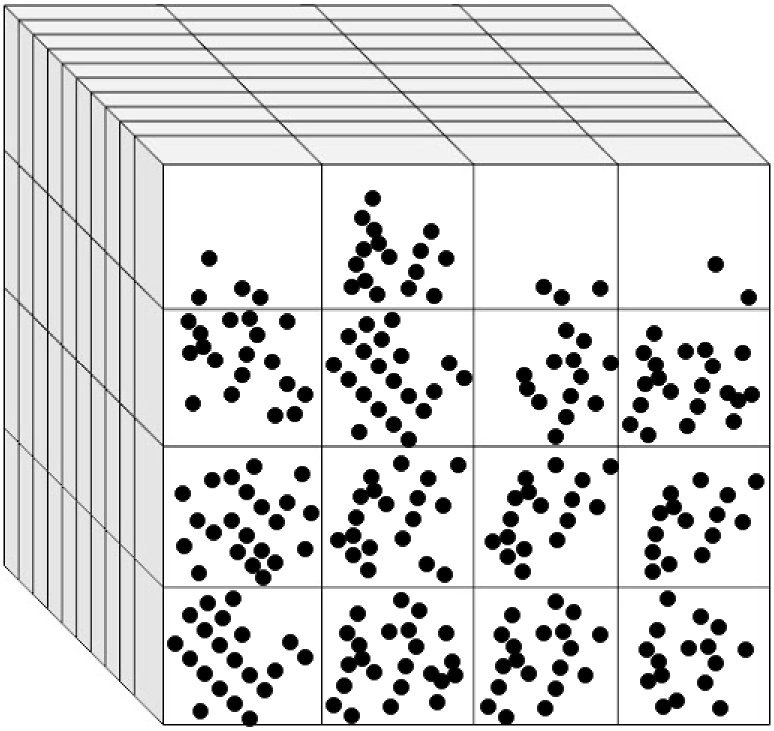


Figure 4.10: Turn point cloud data into a 3D grid of boxes

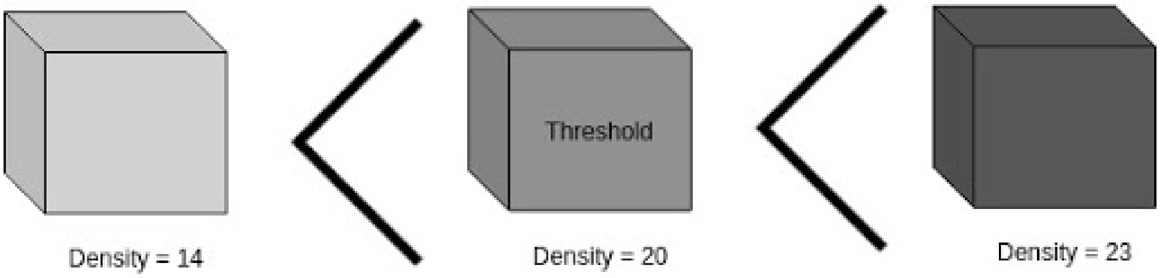


Figure 4.11: Check the density of each box (number of points / box volume)

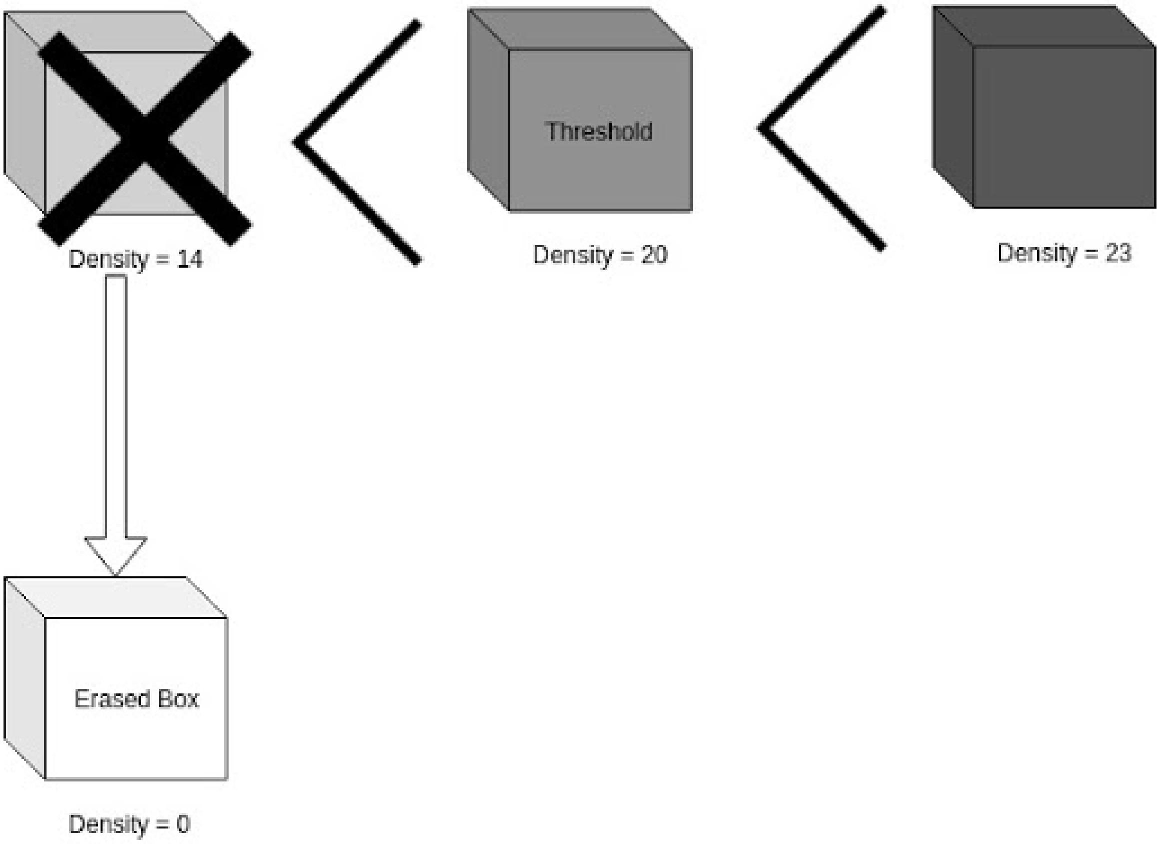


Figure 4.12: Erase all points in boxes with a “density” less than the set threshold

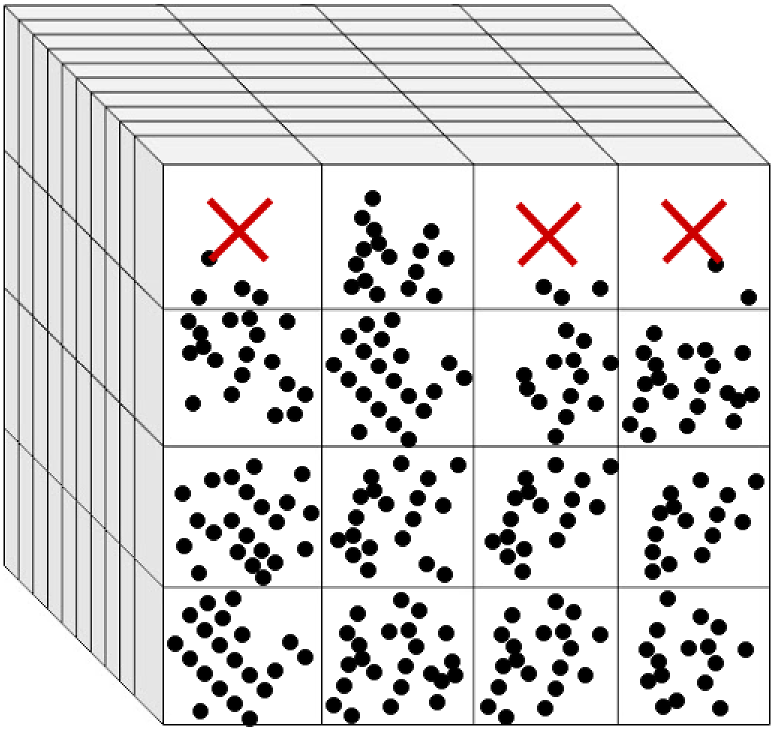


Figure 4.13: Put the boxes back together and remove all boxes less than threshold

It is important to understand the problem of consistently aligning various 3D point cloud data views into a complete model. This is an issue known as point cloud registration. It aligns various points to find the relative positions and orientations of the separately acquired views in a global coordinate framework, such that the intersecting areas between them overlap perfectly. In other words, registration is the process of aligning 3D point clouds on each other to give a complete model. Therefore, the idea came from here and a software was implemented based on that idea. In the following, Figure 4.14 shows the main idea and the steps taken for the results:

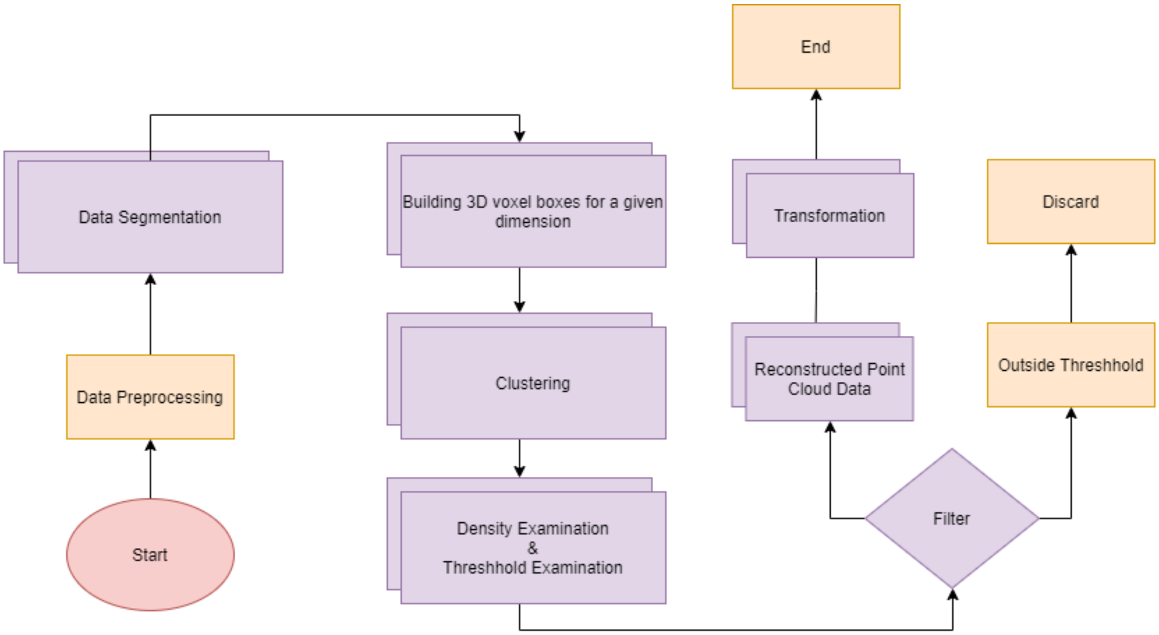

End
Data Segmentation
Building 3D voxel boxes for a given dimension
Transformation
Discard
Data Preprocessing
Clustering
Reconstructed Point Cloud Data
Outside Threshhold
Start
Density Examination & Threshhold Examination
Filter

Figure 4.14: Flowchart Hybrid Filer

The following parameters are used in our example:

1. Name: "Hybrid Filter"

2. Input Vector

   - Length of an individual box
   - Width of an individual box
   - Height of an individual box
   - Threshold for erasing all PCL data within any given box

3. Display summary

4. Display PCL

Listing 4.5: Hybrid Filter Example Command and output

```
#include "PCL_Filters.h"
```

```
int main(int argc, char **argv)
{
    std::vector<PCL_Filter> input = {
    PCL_Filter(”Hybrid Filter”, {5, 10, 5, 20}, true, false),};
    filters = PCL_Filters(input, ”/home/ismail/summary.csv”);
    ROSRUN(argc, argv, ”/home/ismail/bagfiles/filtered_PCL.bag”);
return EXIT_SUCCESS;
}
```

CHAPTER 5
PROPOSED APPROACH

5.1 Introduction

Using cubical representation for further processing has distinct advantages in data size, processing time, noise suppression, and easier volumetric manipulations. Since one voxel can represent numerous points, using a voxel grid for spatial representation can dramatically reduce data size when compared to representing the same space with points. Depending on accuracy requirement, a voxel grid is a down-sampled version of a point cloud. The point cloud could contain inaccuracies based on how the point cloud is acquired. This is a problem when it comes to mapping for autonomous vehicle. For example, using an open source Robotic Operating System (ROS) and RVIZ software for real-time visualizing and processing of 3D data from Velodyne 16 Channel LiDAR (VLP-16) sensors for processing are assumed to have accuracy of +/- 3cm, which is typical. It is deducible that noise should become a reasonable concern when one attempts to obtain results from the point cloud.

5.2 Methodology

The systematic analysis of the methods applied to this project is represented in Figure 5.1. It outlines the direction this research used and undertook to achieve the final results. The design process describes the general methodology starting with the collection process of raw data passing through the filter design to obtain specific results. The process started with designing the DAQ system that combines three major sensors. Then, a filter system was designed to reduce point cloud data based on specific parameters. The last phased of this design was to extract the clusters generated by the previous step.

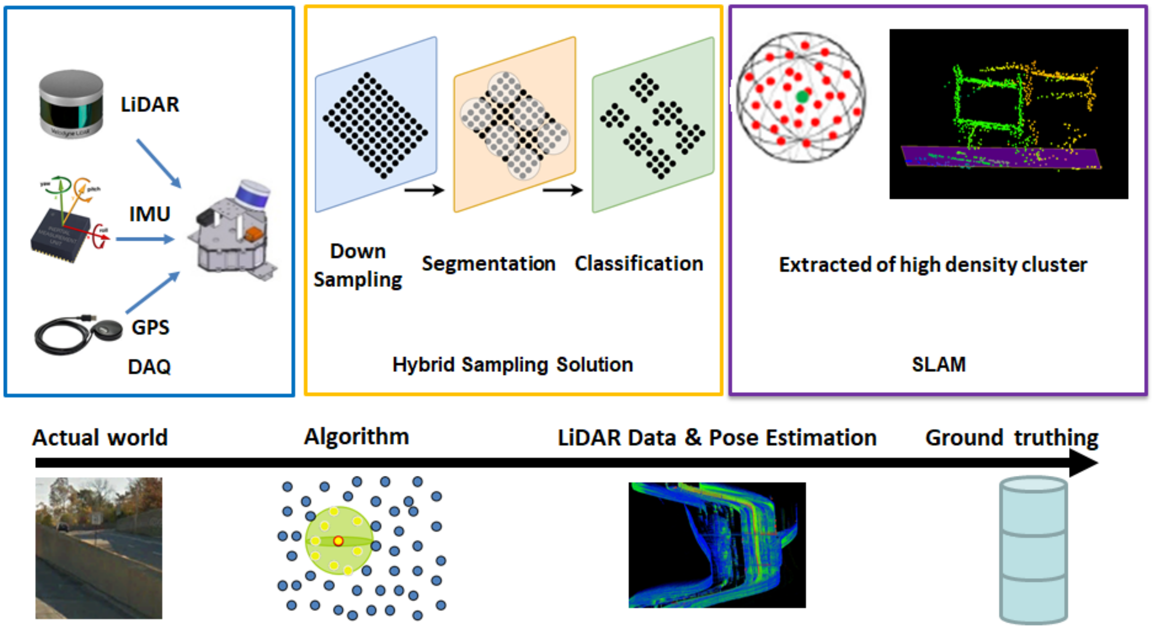


Figure 5.1: The design process for carrying out this research

To understand the whole process in details, a methodology process is presented in a modeling system architecture approach in Figure 5.2. The key architectural decision of the system design is based on the combination of volume representations of multiple filters. In this research, the concept is illustrated in prepossessing model, voxelization, system input in different forms, and possible applications.

The implementation model of commonly-used functionality, message-passing between processes, and package management were all done in ROS environment. This design, from the file-system level to the processing level, was brought together with the ROS infrastructure tools. As is can be implemented in any modern programming language, ROS framework was developed. In this case, C++ programming language was used for designing a point cloud filter. In addition to utilizing Point Cloud Library (PCL), C++ is a procedural programming language intended for intensive functions of CPU and control over hardware.

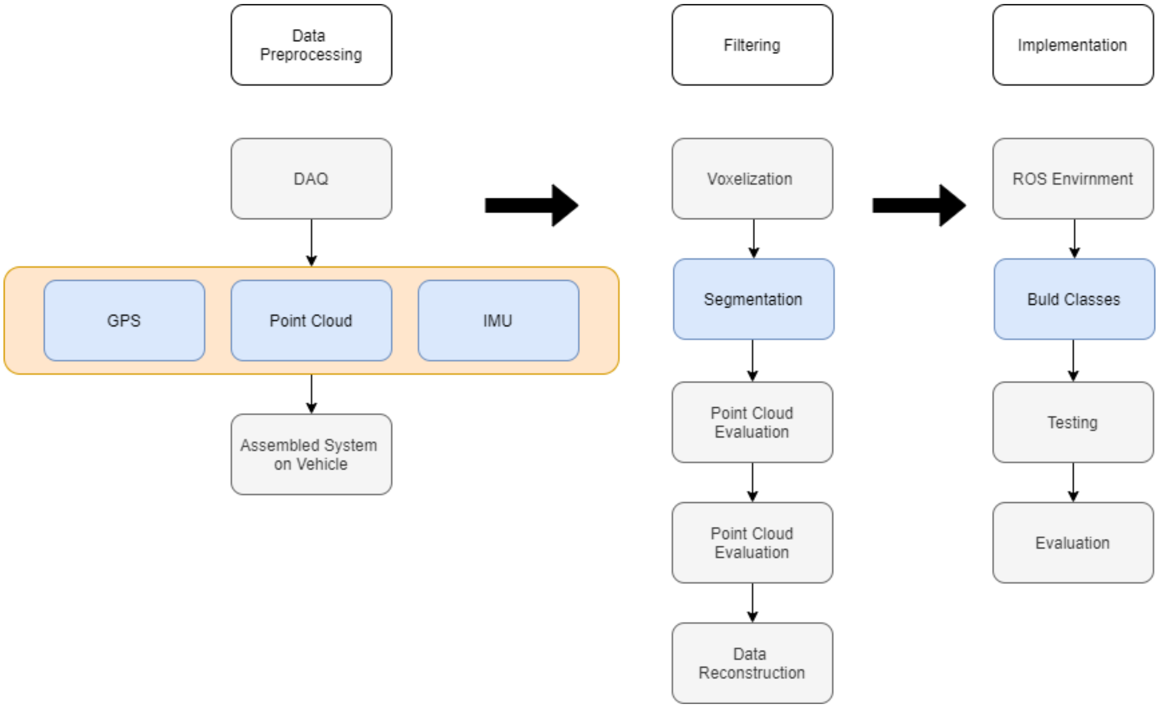


Figure 5.2: System architecture is divided to three major models

5.3 Performance Evaluation

In order to conduct a survey of different filtering methods and compare the performances, an experiment was developed to measure certain characteristics for comparison. Since this research is toward autonomous vehicle applications, a lightweight and ground-optimized lidar odometry and mapping method, LeGO-LOAM, was used for real-time six degree-of-freedom pose estimation with ground vehicles [73]. LeGO-LOAM can achieve real-time pose estimation on a low-power embedded system. LeGO-LOAM leverages the presence of a ground plane in segmentation and optimization steps. Initially, point cloud segmentation was applied to filter out noise, and then feature extraction to obtain distinctive planar and edge features. Then, a two-step Levenberg-Marquardt method was implemented to solve different components of the six degree-of-freedom transformation across consecutive scans. At the last stages, a comparison of the results was performed on the recorded data, gathered from Canadian roads using a ground-truth vehicle system. In addition, an experimental analysis of several filter algorithms was carried out to determine which accurately represents an object reducing the noise influence in the detection procedure. An observation of similar or better accuracy with reduced computational expense when applying hybrid filter solution.

5.3.1 Mapping Formulation

The fundamental components of mobile terrestrial mapping, GPS/INS-based direct georeferencing and digital imaging sensor(s), are usually modeled separately while further differentiating the laser and optical imaging sensors. Figure, 5.3 shows the coordinate system definitions; note that the sensor frame is defined according to the conventional LiDAR notation, which is rotated 180° around the X-axis compared to the standard image sensor frame. Also, a polar coordinate system definition (α , β) is used to allow for a common treatment of the laser and image sensors.

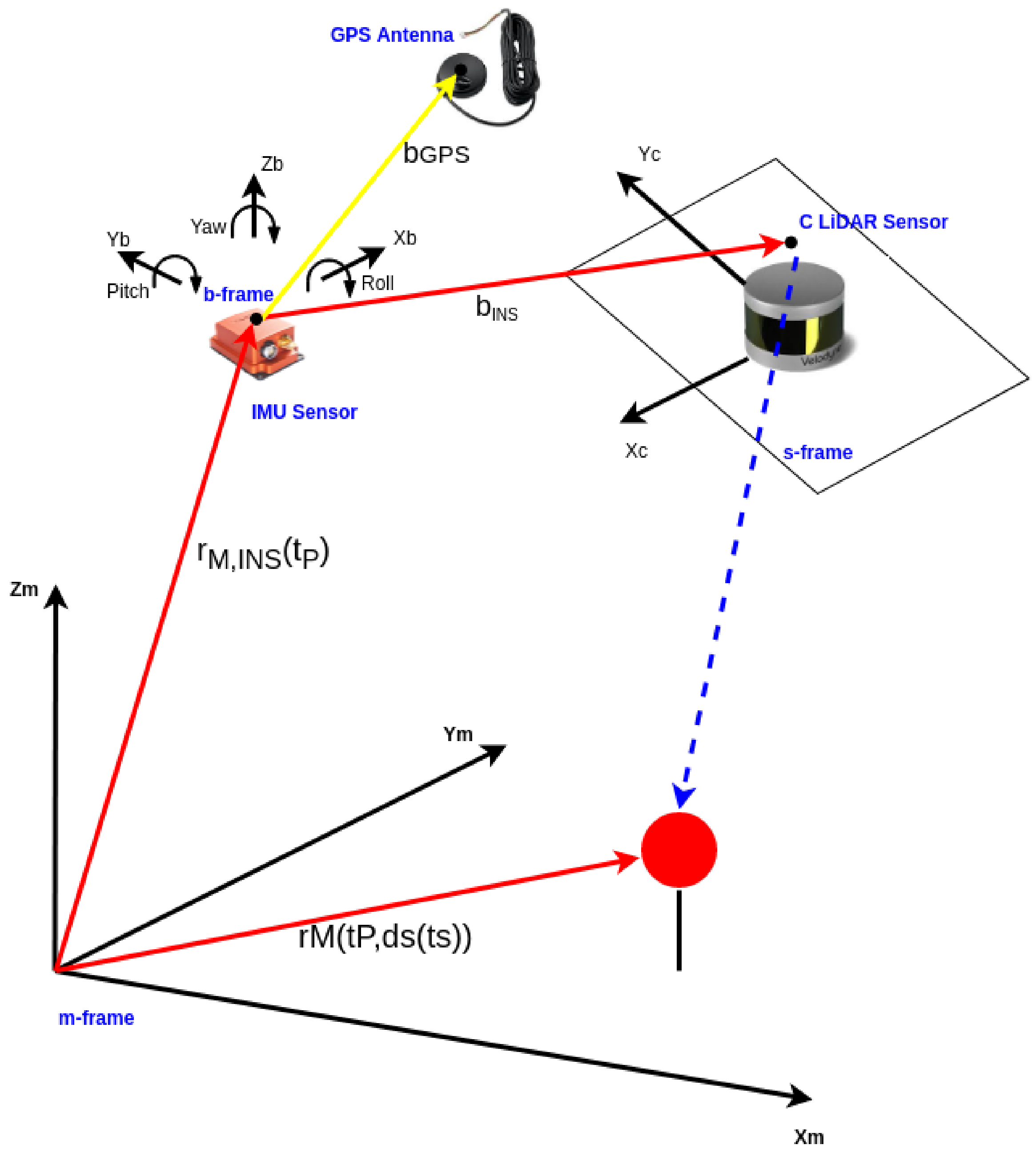


Figure 5.3: Here a combined representation is provided for the sensor model that can be equally applied to laser and optical camera systems, including both frame and line camera models

The mapped object point coordinates are computed according to the sensor equation:

$$r_M(t_p, d_S(t_p)) = r_{M,INS}(t_p) + R_{INS}^{M}(t_p)(R_S^{INS} R_Y(\alpha(t_p)) R_X(\beta(t_p)) d_S(t_p) + b_{INS}) \tag{5.1}$$

where:

$r_M(t_p)$ = 3D coordinates of the sensed object point in the mapping frame

$r_{M,INS}(t_p)$ = 3D INS coordinates in the mapping frame, provided by GPS/INS

$R_{INS}^{M}(t_p)$ = Rotation matrix between the INS body and mapping frame

R_S^{INS} = Boresight matrix between the sensor frame and INS body frame

R_X^S, R_Y^S = Rotation matrix of the sensor frame

$\alpha(t_p)$ = Sensing direction angle from the Y axis of the sensor frame

$\beta(t_p)$ = Sensing direction angle from the X axis of the sensor frame

$d_S(t_p)$ = Distance from sensor reference point to object point

b_{INS} = Boresight offset vector

By defining an error boundary envelope, this formulation provides the first estimate for the geospatial positioning accuracy of a system. For LiDAR systems, all the parameters and measurements are known, so except for the effect of the footprint size and various object-space specific conditions, the accuracy estimates can be directly computed. For optical imagery, the scale (object distance) is unknown, so only a line with an error can be computed the line on which the object point should lie. Then depending on the method, such as intersection with another line (stereo technique) or surface (DEM), additional computations are needed to determine the accuracy estimate of an object point; more details can be found in [74].

5.3.2 System Architecture

To move towards autonomous driving, a 2018 Ford Edge was retrofitted with a data acquisition suite to capture normal Canadian operating conditions and environments. For map creations, a 3D LiDAR and an IMU were used running on a laptop with Robot Operating System (ROS) [75], and a global position system (GPS). However, the system also includes other sensors that were utilized for other activities such as testing and evaluation of active safety and autonomous vehicles features. Figure 5.4 showcases the vehicle and sensor suite used within this study. Figure 5.5 presents an overview picture of the system used to construct the HD Map; however, this was not the full DAQ system initially developed. The LiDAR, IMU, and the GPS were utilized to generate a map but also other sensors were used for different applications [76].

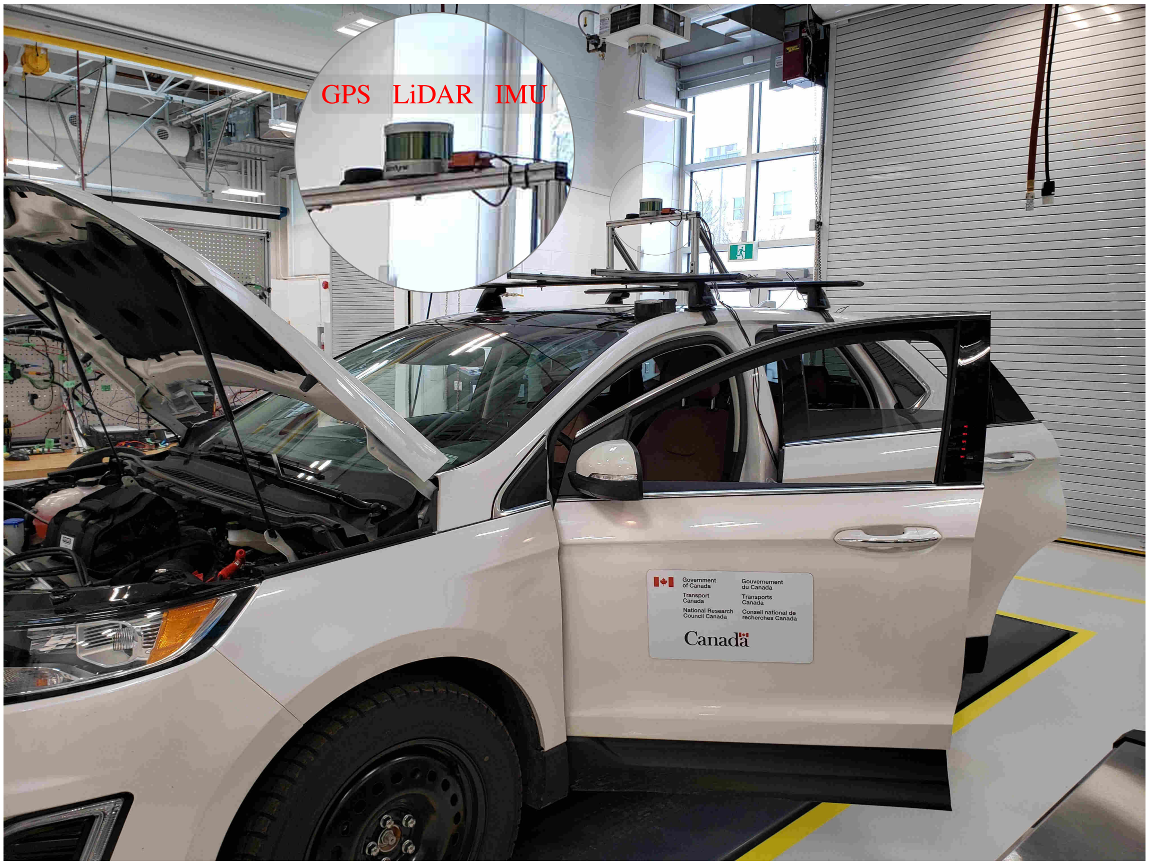


Figure 5.4: Data Acquisition Hardware Stack

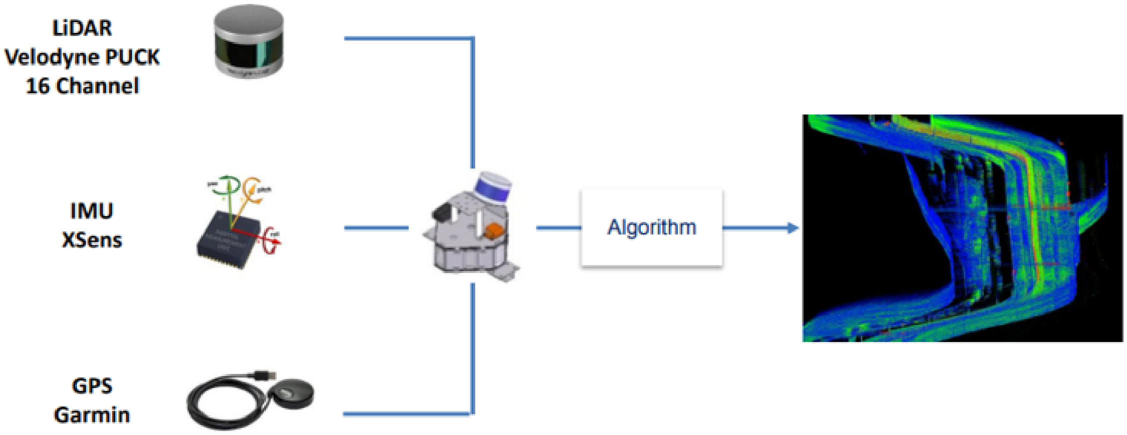


Figure 5.5: A highlight of the Sensors Used for This Experiment

5.3.3 Data Acquisition System

DAQ system was developed to test, evaluate, and compare different methodologies in terms of the survey devices, filtering algorithms, and data sources that can be used to quantify certain part of the roads such as center of curvature, radius of existing roads, as well as road edges. It was pursued to help investigate and analyze vehicle accidents and crashes on Canadian roads.

DAQ system is a process that measures real world physical conditions and converts resulting samples into digital numeric values. This system typically converts analog waveforms into digital values for processing purposes. The main components of data acquisition systems include the following:

- Sensors, to convert physical parameters to electrical signals.
- Actuators, to convert electrical control signals into physical actions.
- Signal conditioning circuitry, to convert sensor signals into a form that can be converted to digital values.

- Analog-to-digital converters (ADC), to convert conditioned sensor signals to digital values.
- A computer running DAQ software [76, 77, 78].

A DAQ system, along with a sensor package, was designed and installed on 2018 Ford Edge to gather data of the surrounding environment. The data collected through the developed system include GPS route, full vehicle CAN Bus, video recording, point cloud data, angular rate, and magnetic field. Figure 5.6 shows the full block diagram of the DAQ system that was used and currently working on integrating additional sensors such as a thermal camera for night vision applications and a long range 77GHz radar to test object detection performance in critical mission day and night, in sunny or rainy weather, for short and long-range, and other weather conditions. The system is used to test precision of autonomous vehicles and active safety applications. As autonomous driving on public roads requires high precise localization with the range of few centimeters [79], a precision system to test the accuracy of the generated map for all road elements relative to ground truth is being built. This system can be coupled with precise positioning (localization) software, which can be enhanced with additional sensor data (e.g., radar, GNSS/INS, or a higher level LiDAR with more laser channels) for a more robust performance.

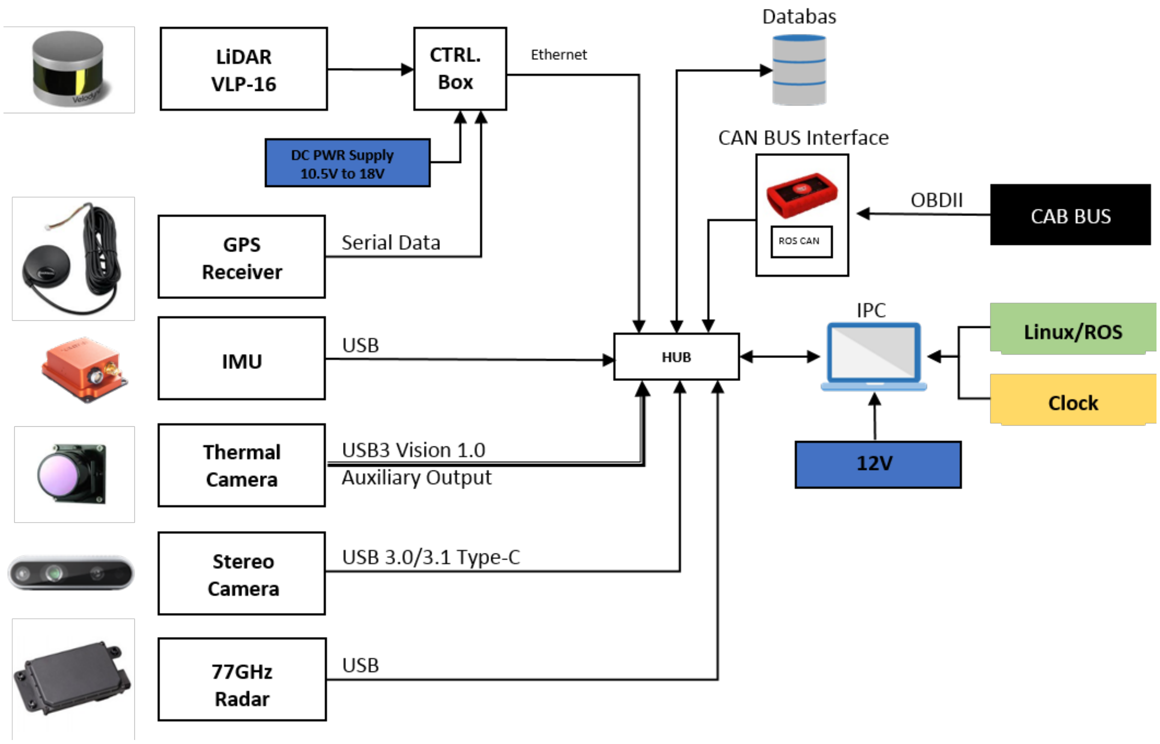


Figure 5.6: DAQ Block Diagram

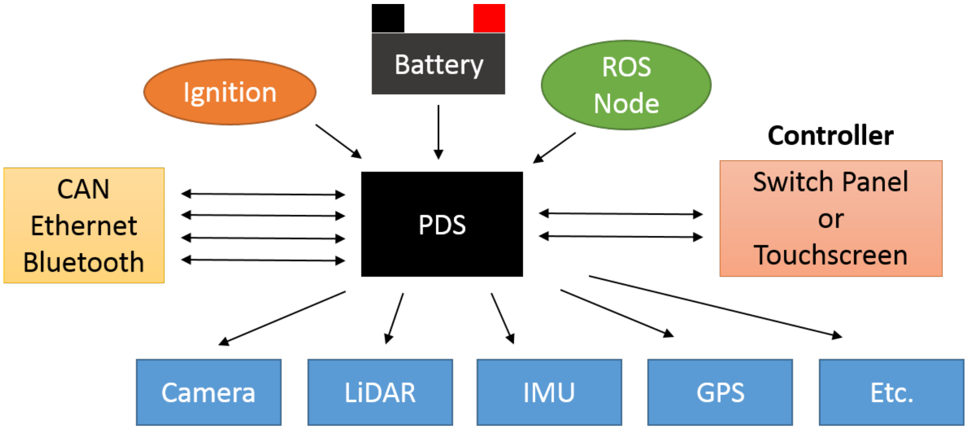


Figure 5.7: Power Distribution System

For this application, it was necessary to build a power distribution system (PDS), Figure 5.7, which is flexible and easy-to-use in order to power the subsystem on a test vehicle. The PDS can support multiple independent DC relay channels (10-12 channels) that can each draw a continuous ∼20 amps and an onboard customizable computer to monitor each channel, report its electrical current and health status, along with a diagnostic software to checks for open and short circuit conditions. The PDS system is setup in a way to avoid power outage especially while running a test or using the DAQ system. It has the capability to regulate power flow at a desired voltage and different frequency level, meeting load demands, and is adaptable to changes in situation.

5.3.4 Sensor and Computing Suite

A 3D Velodyne VLP-16 LiDAR unit was mounted on the top of the vehicle. The LiDAR utilizes a 903 nm wavelength laser to capture range data up to 100m and provides 16 unique channels acting between ± 15.0 degrees with a vertical angular resolution of 2.0 degrees. The laser scanner provides a 360-degree field of view with a horizontal angular resolution ranging between 0.1 to 0.4 degrees and a scanning rate ranging between 5 and 20 Hz.

The Velodyne LiDAR was provided with a Garmin 18x LVC GPS Receiver that was pre-configured for optimized operation. The receiver plugs directly into the Interface port and is used to synchronize the sensor's timestamp with precision GPS time. This generation of GPS sensors includes the capability of FAA Wide Area Augmentation System (WAAS) differential GPS. Even though the accuracy of this sensor is $<$3m, it is not enough for autonomous and active safety applications. This is one of the areas were future development will be required to integrate and use high precision Global Navigation Satellite System (GNSS) technology that provides the accuracy, availability and reliability that a vehicle requires to be self-driving.[80]

One of the features that was developed was a tracking system using a python library to integrate high-level functions at a certain timestamp in the application. It combined OSM (Open Street Map) layer and QT Creator to generate tracking GPS point application within the DAQ system as shown in Figure 5.8. A fully autonomous vehicle needs an accurate localization solution paired with the confidence that the localization solution is correct. This GNSS technology is capable of providing decimeter-level accuracy to ensure a vehicle stays in its lane, or a safe distance from other vehicles.

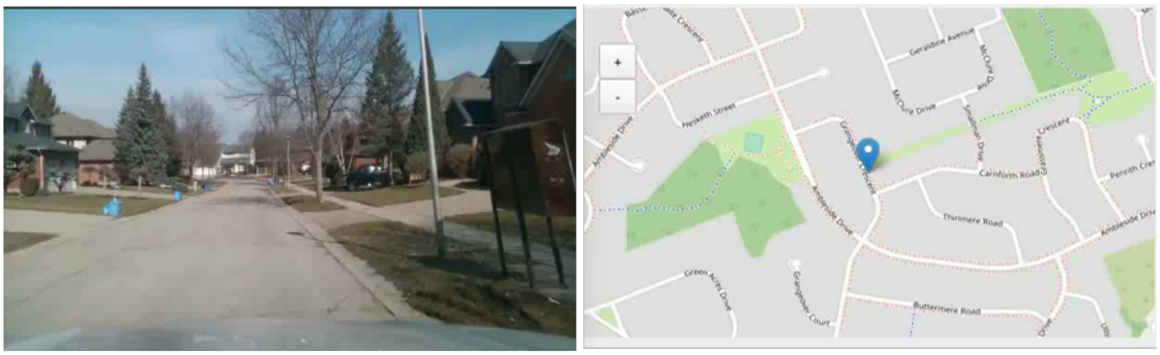


Figure 5.8: GPS Tracker on OSM Layer

For the purpose of external odomoetry measurement, an MTi-G-710 IMU was integrated into the system. This IMU contains a set of accelerometers (±20 g), gyroscopes (±450°/s), magnetometer (±8 G), barometer (300-1100 hPa), and a GNSS receiver (2.5m horizontal accuracy). For a computing platform, ROS Kinetic distribution was selected for this study and was running on a laptop utilizing an i7-6820HQ CPU running at 2.70GHz. The vendor-provided ROS packages were used for transferring data from the sensor interfaces into ROS topics. The IMU was configured to broadcast the Global Navigation Satellite System (GNSS) signals, orientation, position, linear and angular accelerations, and velocities. In addition, the LiDAR package was configured to match what was reported in [73].

5.3.5 Software Architecture

The software architecture that was used in this study is presented in Figure 5.9. The LeGO-LOAM algorithm as depicted within the block diagram has been reported in [73]. This algorithm produces maps and LiDAR odometry with assistance from an IMU. The block diagram of the LeGO-LOAM algorithm is presented in Figure 5.10.

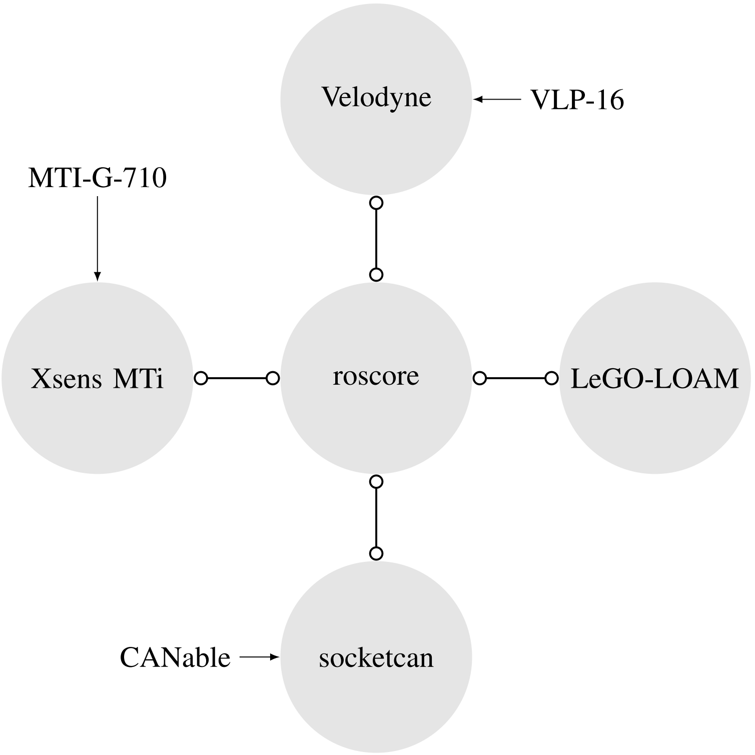


Figure 5.9: ROS Software Architecture

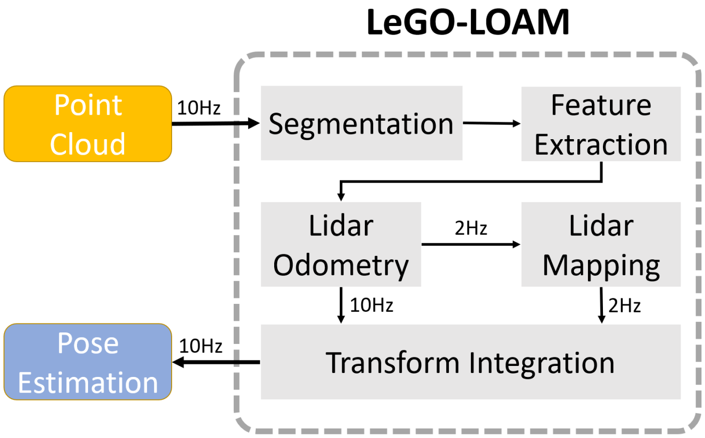


Figure 5.10: LeGO-LOAM software stack [73]

An (over)simplified model of the system is presented in 5.11. The system was designed with an option for the user to select different filters in order to sample point cloud data. In addition, the filter designs and algorithms were integrated into Robotic Operation System (ROS) in order to test the performance of the selected programs. From a visual perspective, this system works as shown in Figure 5.12. It is up to the user to input the parameters and determine which filter to apply on the point cloud data. It can combined multiple filters at once with different inputs:

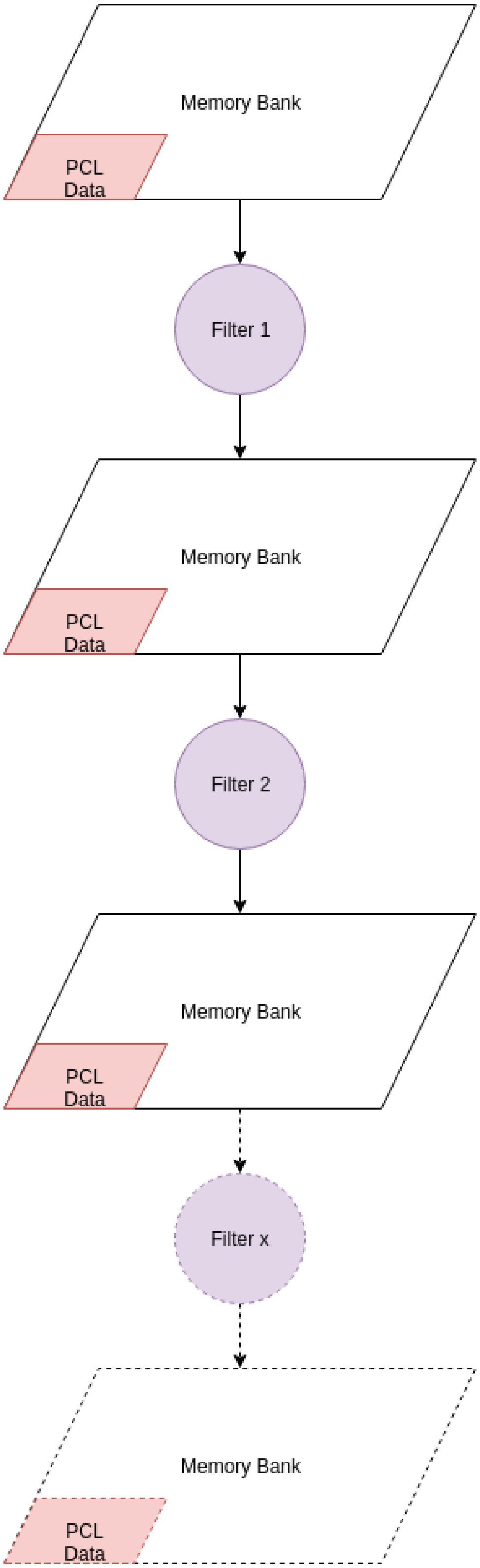


Figure 5.11: (Over)simplified visual model. User can use multiple filters to sample point cloud data.

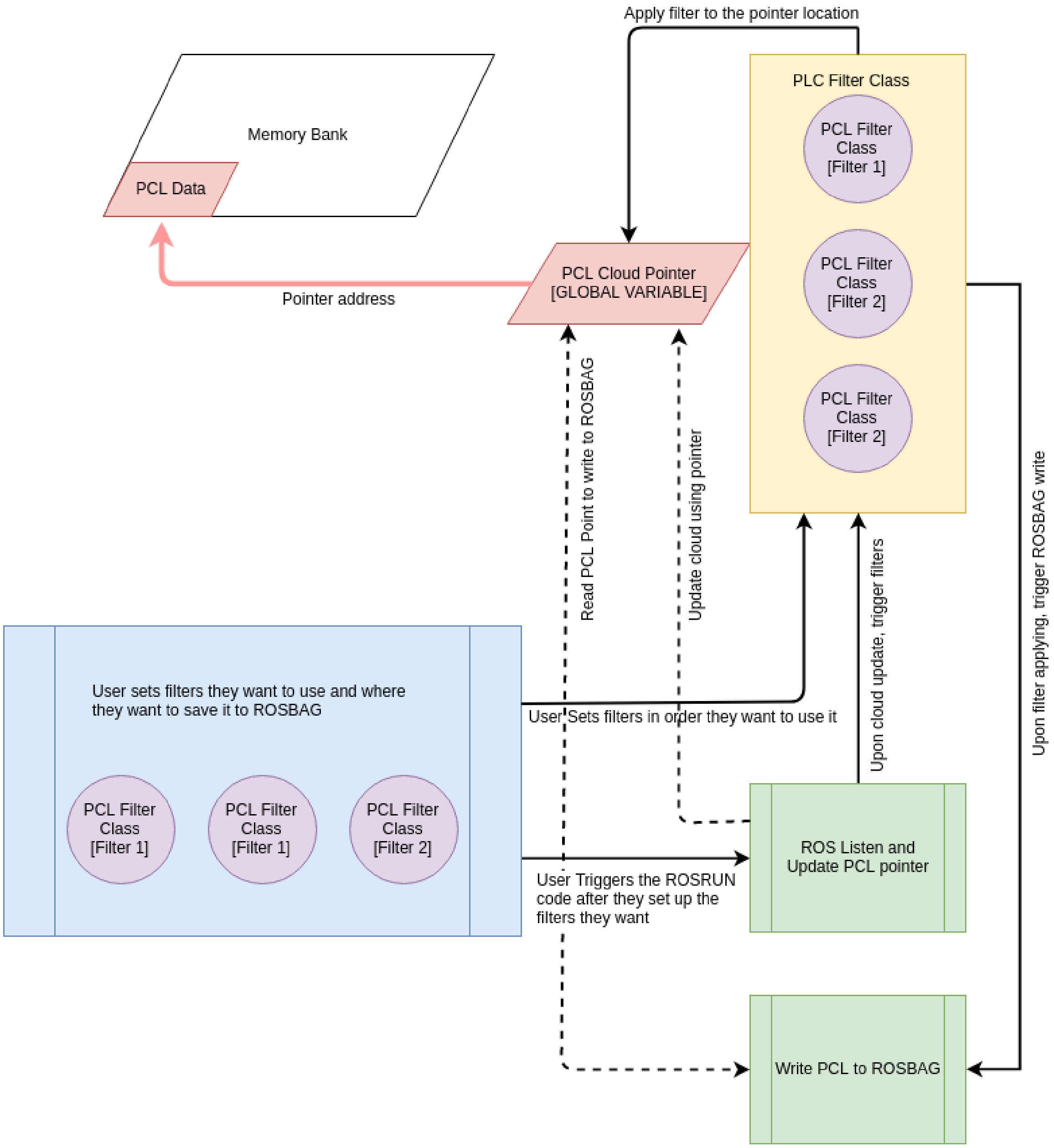

Apply filter to the pointer location
Memory Bank
PCL Data
PLC Filter Class
PCL Filter Class [Filter 1]
PCL Filter Class [Filter 2]
PCL Filter Class [Filter 2]
PCL Cloud Pointer [GLOBAL VARIABLE]
Pointer address
Read PCL Point to write to ROSBAG
Update cloud using pointer
Upon cloud update, trigger filters
Upon filter applying, trigger ROSBAG write
User sets filters they want to use and where they want to save it to ROSBAG
PCL Filter Class [Filter 1]
PCL Filter Class [Filter 1]
PCL Filter Class [Filter 2]
User Sets filters in order they want to use it
User Triggers the ROSRUN code after they set up the filters they want
ROS Listen and Update PCL pointer
Write PCL to ROSBAG

Figure 5.12: From a classes and visual perspective from start to finish

Color	Function
Blue	User Functions
Green	ROS Functions
Purple	Individual PCL Filter
Yellow	Class of PCL Filters
Red	Pointer to and from Cloud

Table 5.1: Color code for the function used in the flowchart 5.12

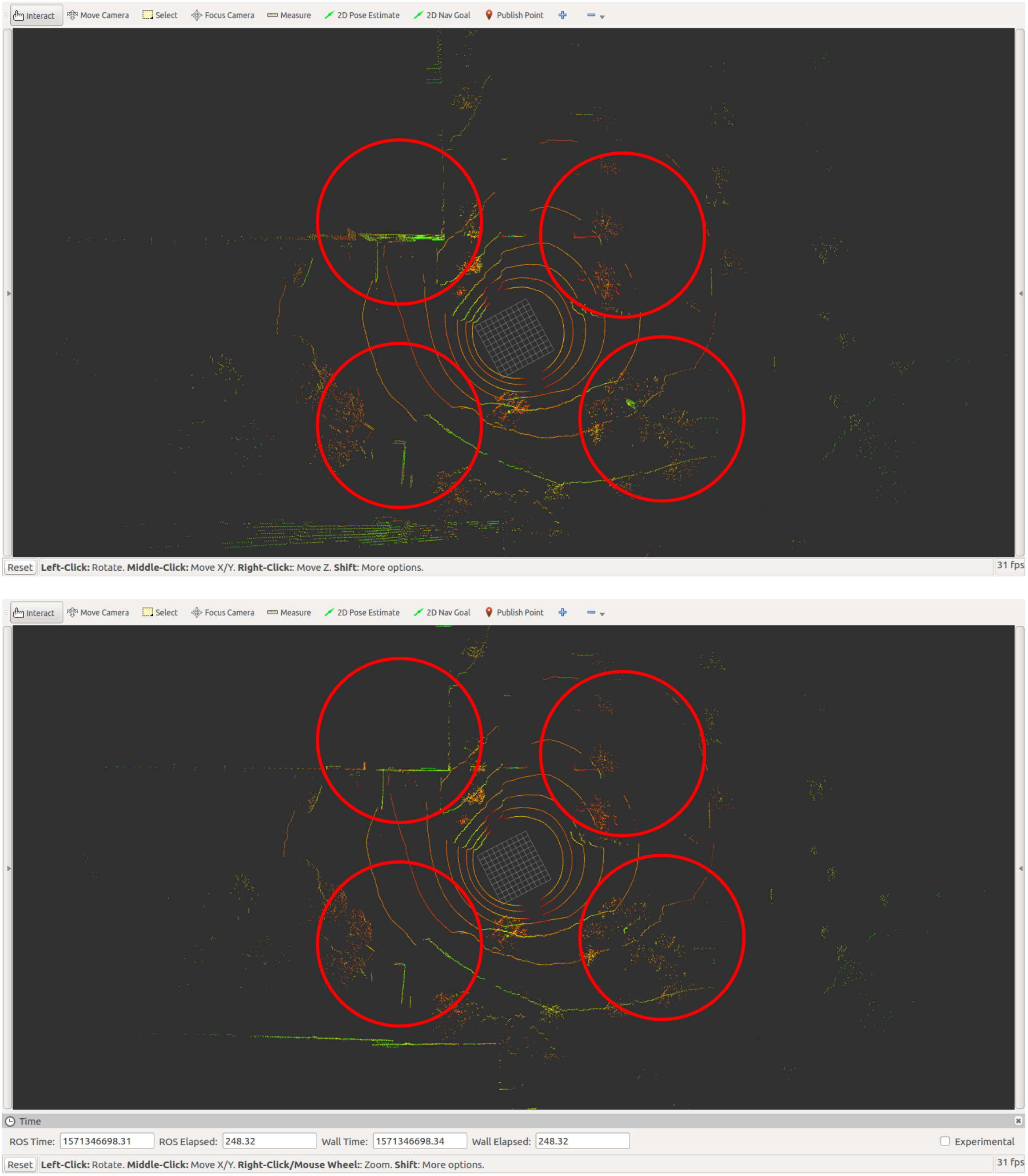
CHAPTER 6
EXPERIMENTAL RESULTS AND PERFORMANCE STUDY

6.1 Introduction

This chapter presents the results of each filter used in this research for comparison and analysis. It provides a general quantitative analysis between the raw data of a Rotational Multi-Beam LiDAR Sensor (RMBL) whose rotation axis is parallel to one of the frame axes and the results after applying Hybrid filter. As stated in previous chapters and without loss of generality, VLP-16 was considered for this work for the addition of a rotation mechanism. This Multi-Beam LiDAR (MBL) sensor is especially suitable for its cost, lighter weight, symmetric FOV, and compact size.

6.2 Parametric Model

Figures 6.1 and 6.2 represent a snapshot of the point cloud data before and after Parametric filter was applied. The red circles are only to identify and examine the difference before raw data and filtered data.



Reset **Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click:**: Move Z. **Shift:** More options.

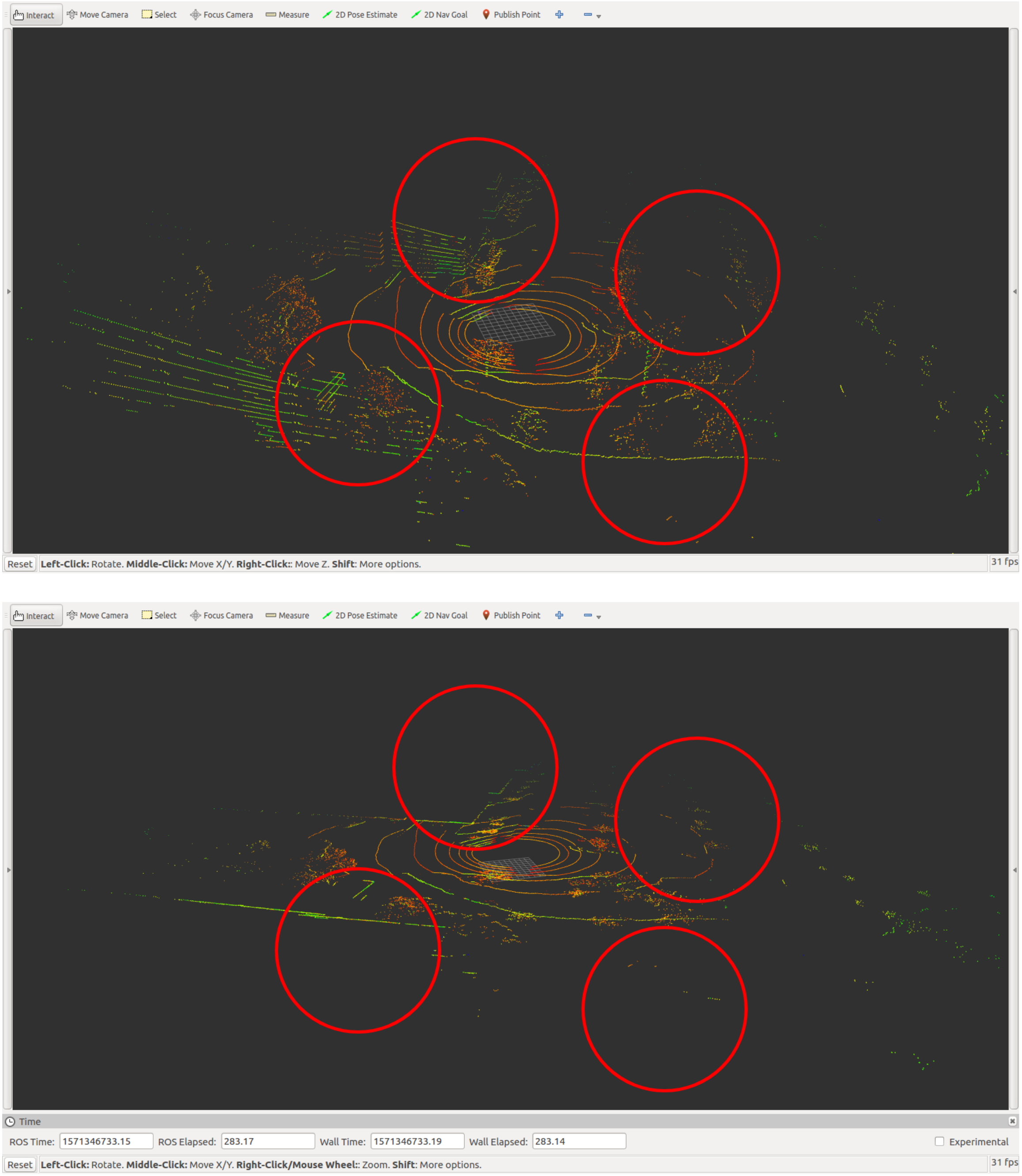
31 fps

ROS Time: 1571346698.31 ROS Elapsed: 248.32 Wall Time: 1571346698.34 Wall Elapsed: 248.32 □ Experimental

Reset **Left-Click:** Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options.

31 fps

Figure 6.1: Parametric filter applied on point cloud data from a bird-view



Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click::** Move Z. **Shift:** More options.

ROS Time: 1571346733.15 ROS Elapsed: 283.17 Wall Time: 1571346733.19 Wall Elapsed: 283.14

□ Experimental

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel::** Zoom. **Shift:** More options.

Figure 6.2: Parametric filter applied on point cloud data from a side-view

A general comparison between raw data of Rotational Multi-Beam LiDAR Sensor (RMBL) where the rotation axis is parallel to one of the frame axes and the results after applying Parametric filtered. Figure 6.3 presents the comparison.

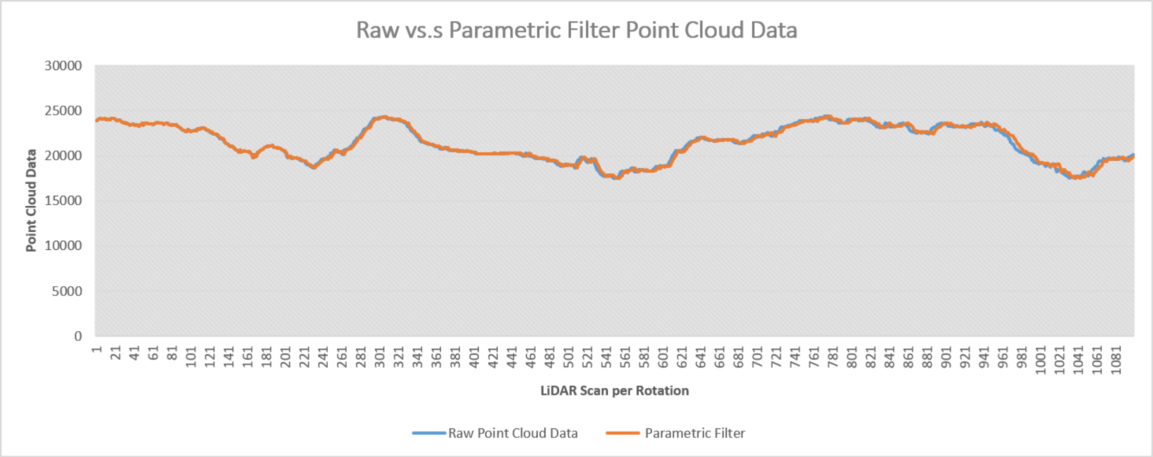


Figure 6.3: Raw point cloud data versus Parametric Filter. The blue colored legend is the raw data without any filter and the orange colored legend is after applying the Parametric filter. The Y-axis is the number of point clouds collected per rotation (X-axis).

Table 6.1 displays the data averaging results before and after applying Parametric filter for a frame.

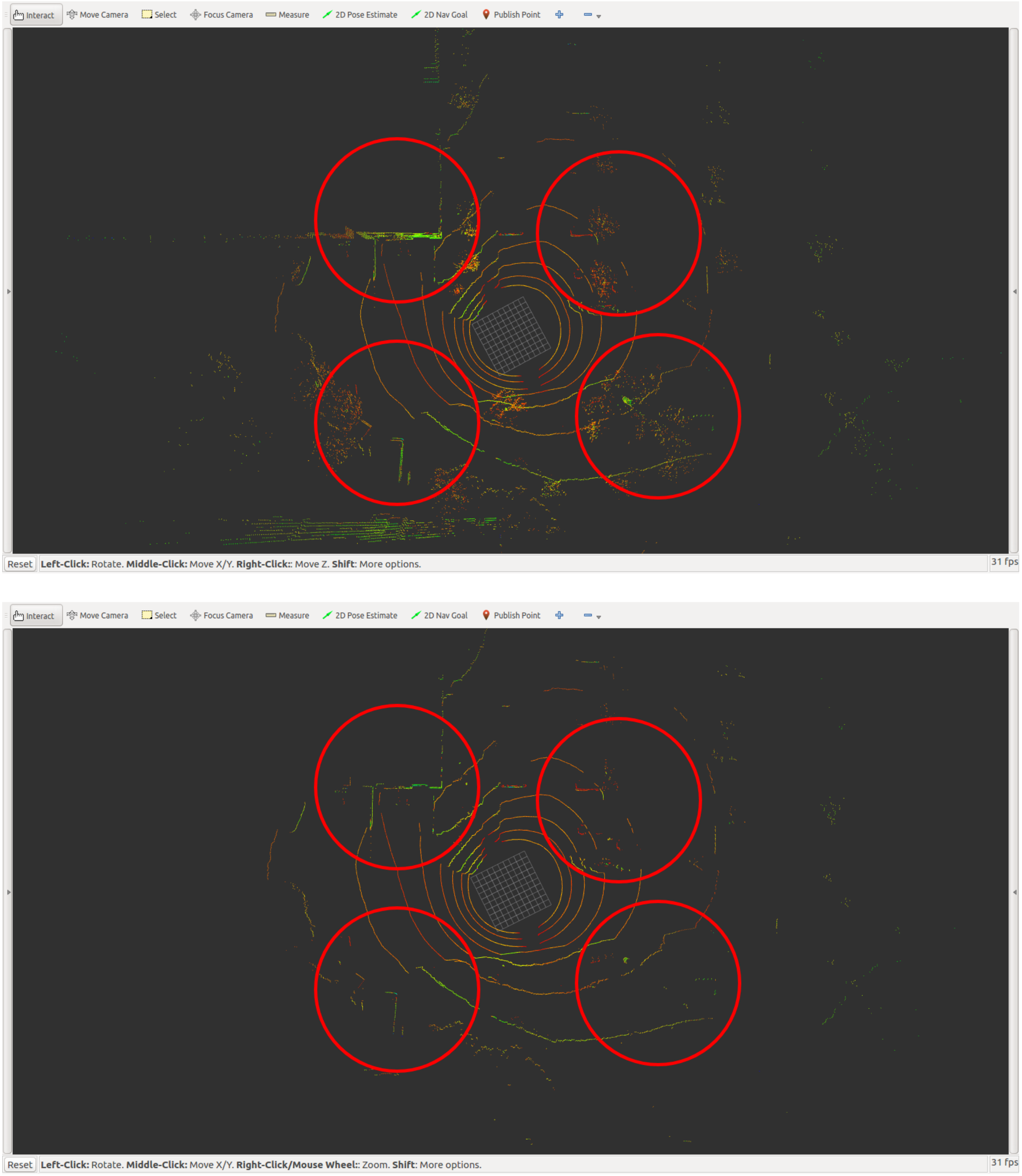
	Raw Data	Parametric Filter
Number of Points (millions)	23.5	23.5

Table 6.1: Summary Table of raw point cloud versus Parametric filter data

After applying the Parametric filter to the data, a major reduction of data was noticed. The Parametric filter projects all collected points on plan selected by the user. In this example, the ground plane is selected for this study and to visualize the data. There are many used cases for this filter but does not provide a solution to the main complications.

6.3 Passthrough Filter

Figures 6.4 and 6.5 represent a snapshot of the point cloud data before and after Passthrough filter was applied. The red circles are only to identify and examine the difference before raw data and filtered data.



Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click:**: Move Z. **Shift:** More options.

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options.

Figure 6.4: Passthrough filter applied on point cloud data from a bird-view

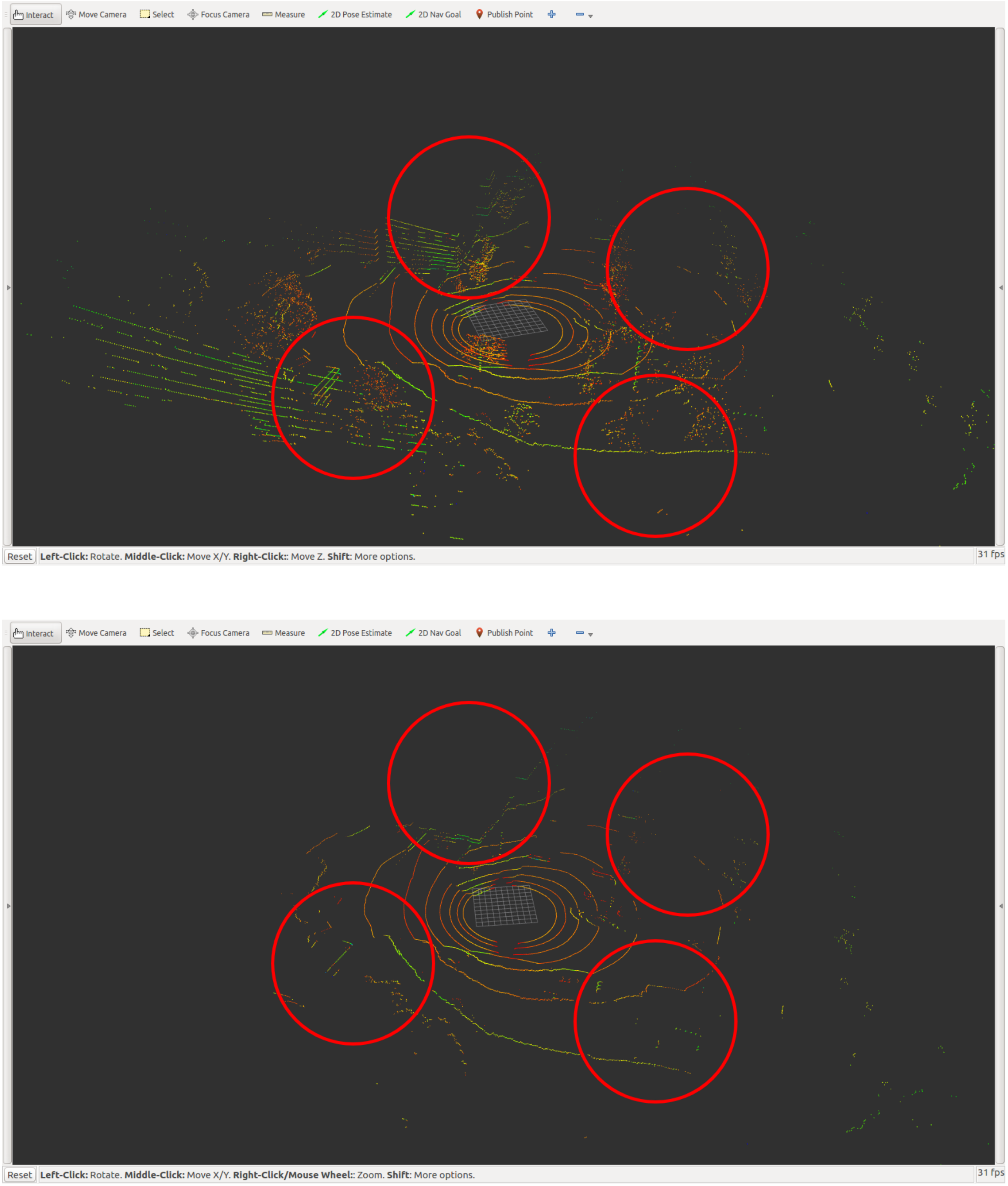


Figure 6.5: Passthrough filter applied on point cloud data from a side-view

A general comparison between raw data of Rotational Multi-Beam LiDAR Sensor (RMBL) where the rotation axis is parallel to one of the frame axes and the results after applying Passthrough filter. Figure 6.6 presents the comparison.

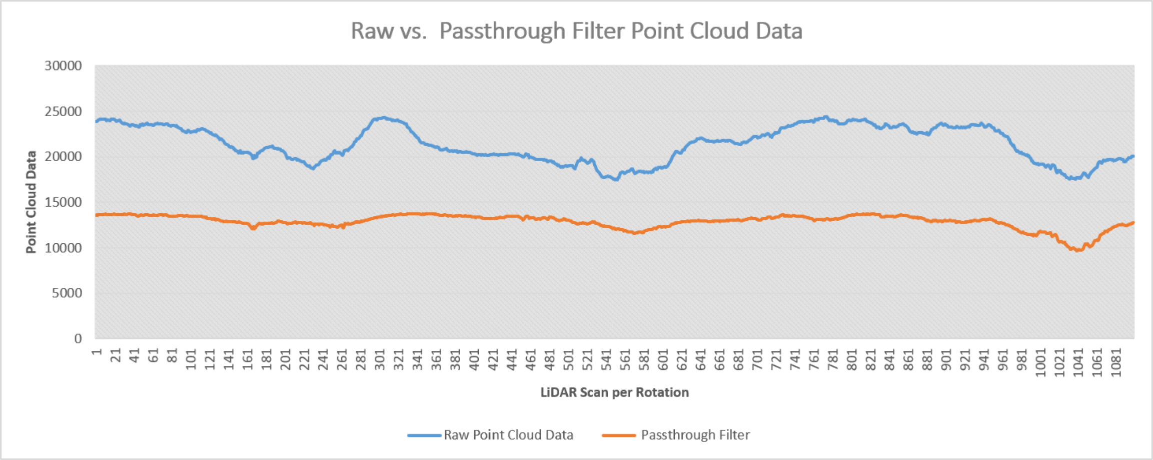


Figure 6.6: Raw point cloud data versus Passthrough Filter. The blue colored legend is the raw data without any filter and the orange colored legend is after applying the Passthrough filter. The Y-axis is the number of point clouds collected per rotation (X-axis).

Table 6.2 displays the data averaging results before and after applying Passthrough filter for a frame.

	Raw Data	Passthrough Filter
Number of Points (millions)	23.5	14.1

Table 6.2: Summary Table of raw point cloud versus Passthrough filter data

6.4 K-means Clustering

Figures 6.7 and 6.8 represent a snapshot of the point cloud data before and after K-means filter was applied. The red circles are only to identify and examine the difference before raw data and filtered data.

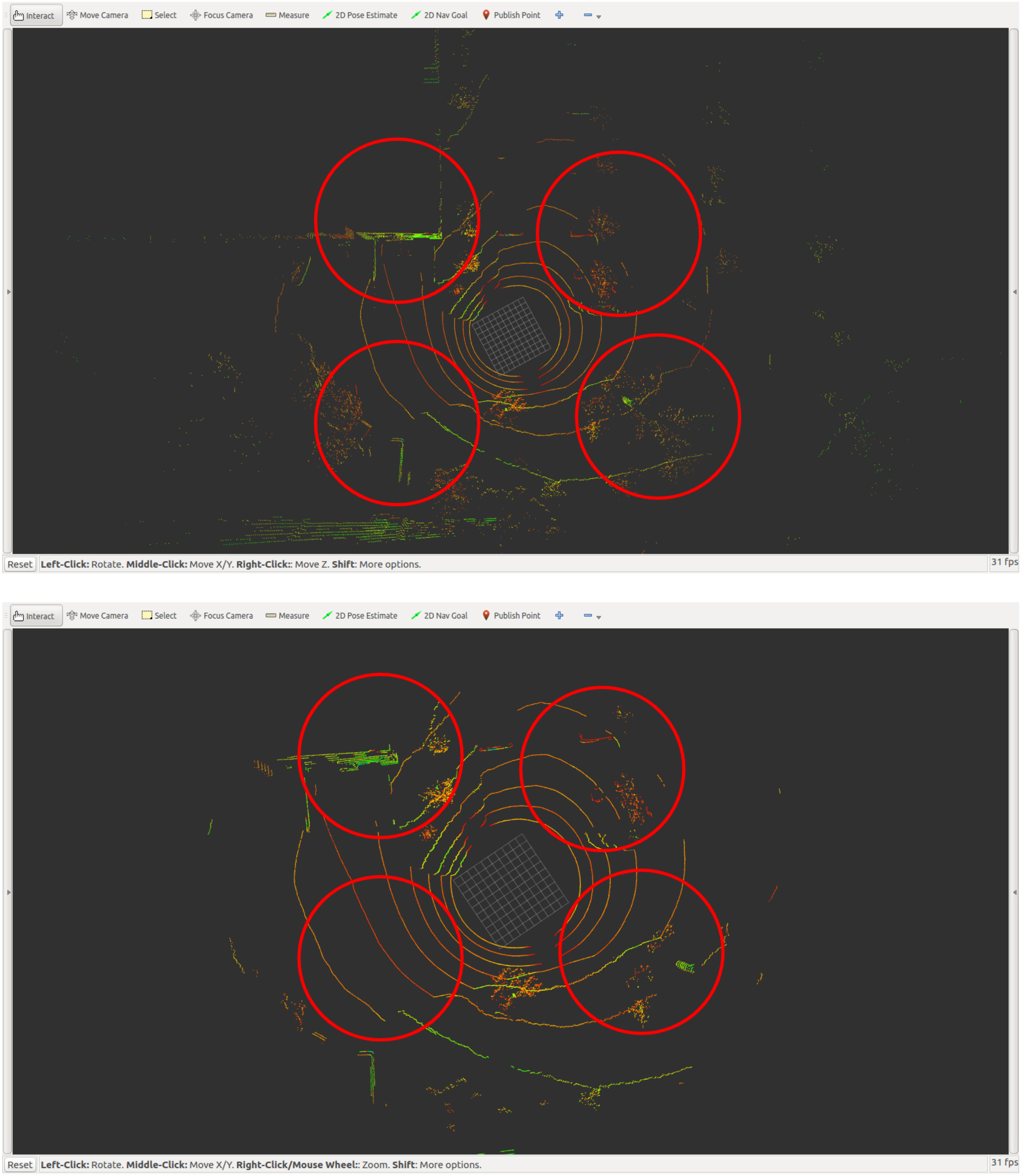


Figure 6.7: K-means Clustering filter applied on point cloud data from a bird-view

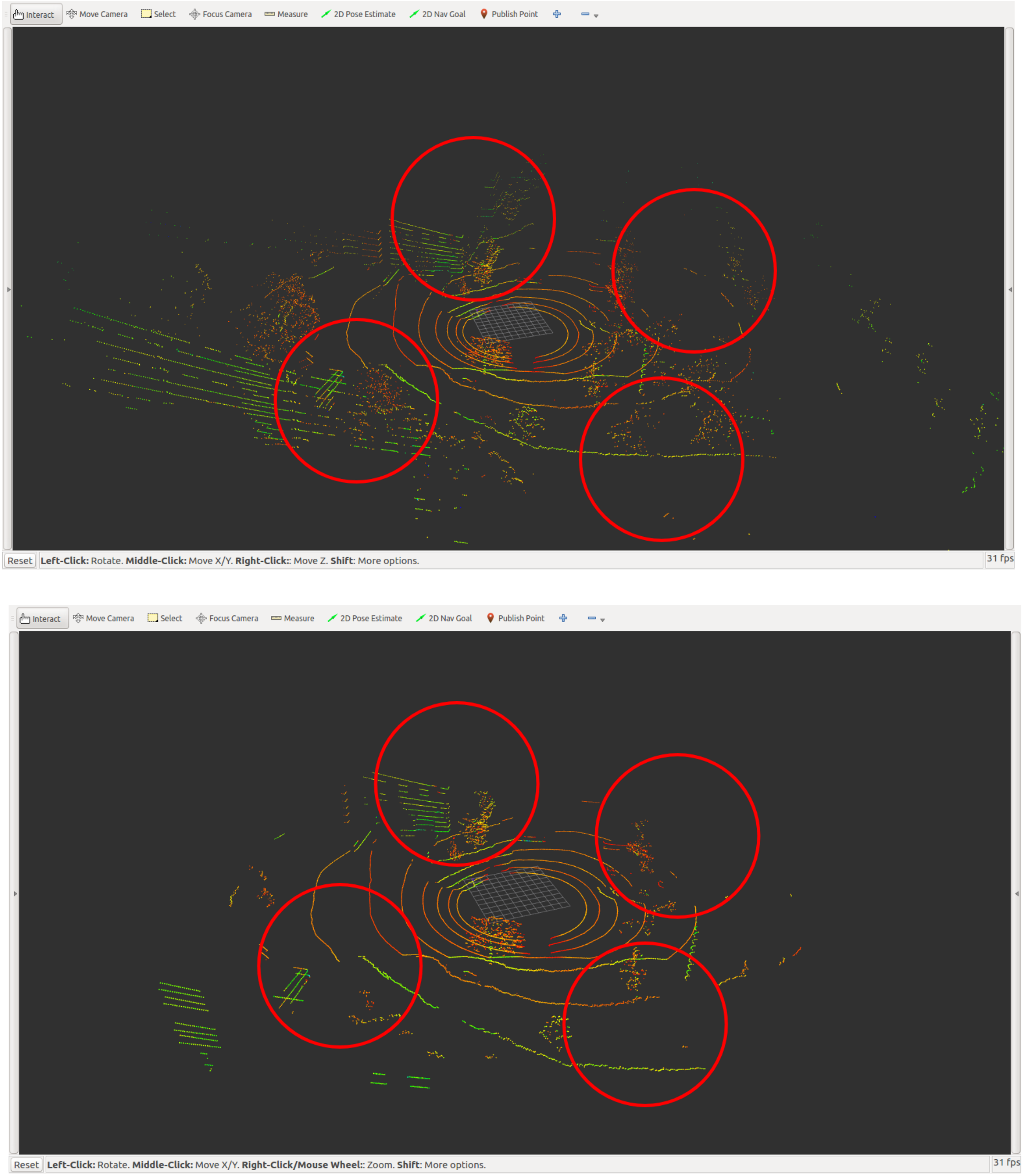


Figure 6.8: K-means Clustering filter applied on point cloud data from a side-view

A general comparison between raw data of Rotational Multi-Beam LiDAR Sensor (RMBL) where the rotation axis is parallel to one of the frame axes and the results after applying K-means filter. Figure 6.9 presents the comparison.

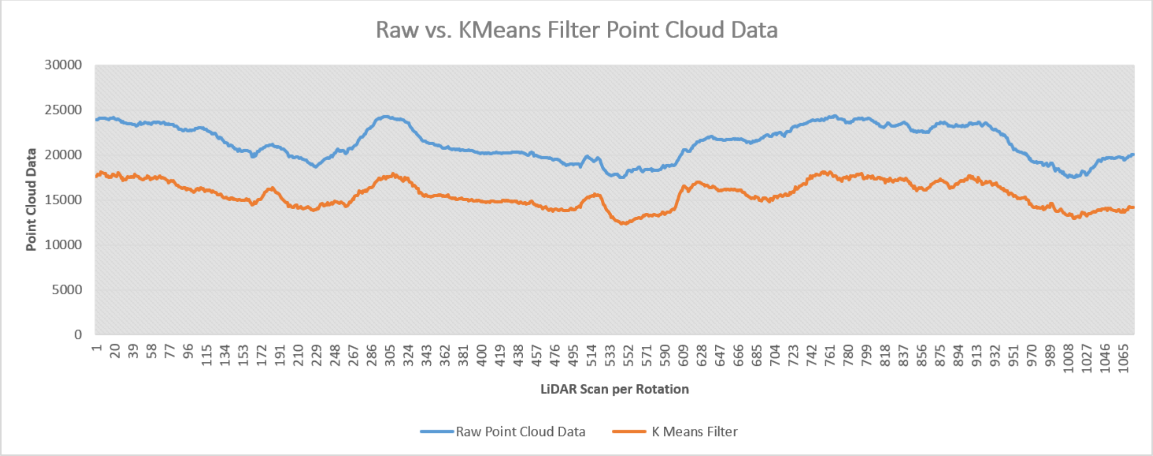


Figure 6.9: Raw point cloud data versus K-means Filter. The blue colored legend is the raw data without any filter and the orange colored legend is after applying the K-means filter. The Y-axis is the number of point clouds collected per rotation (X-axis).

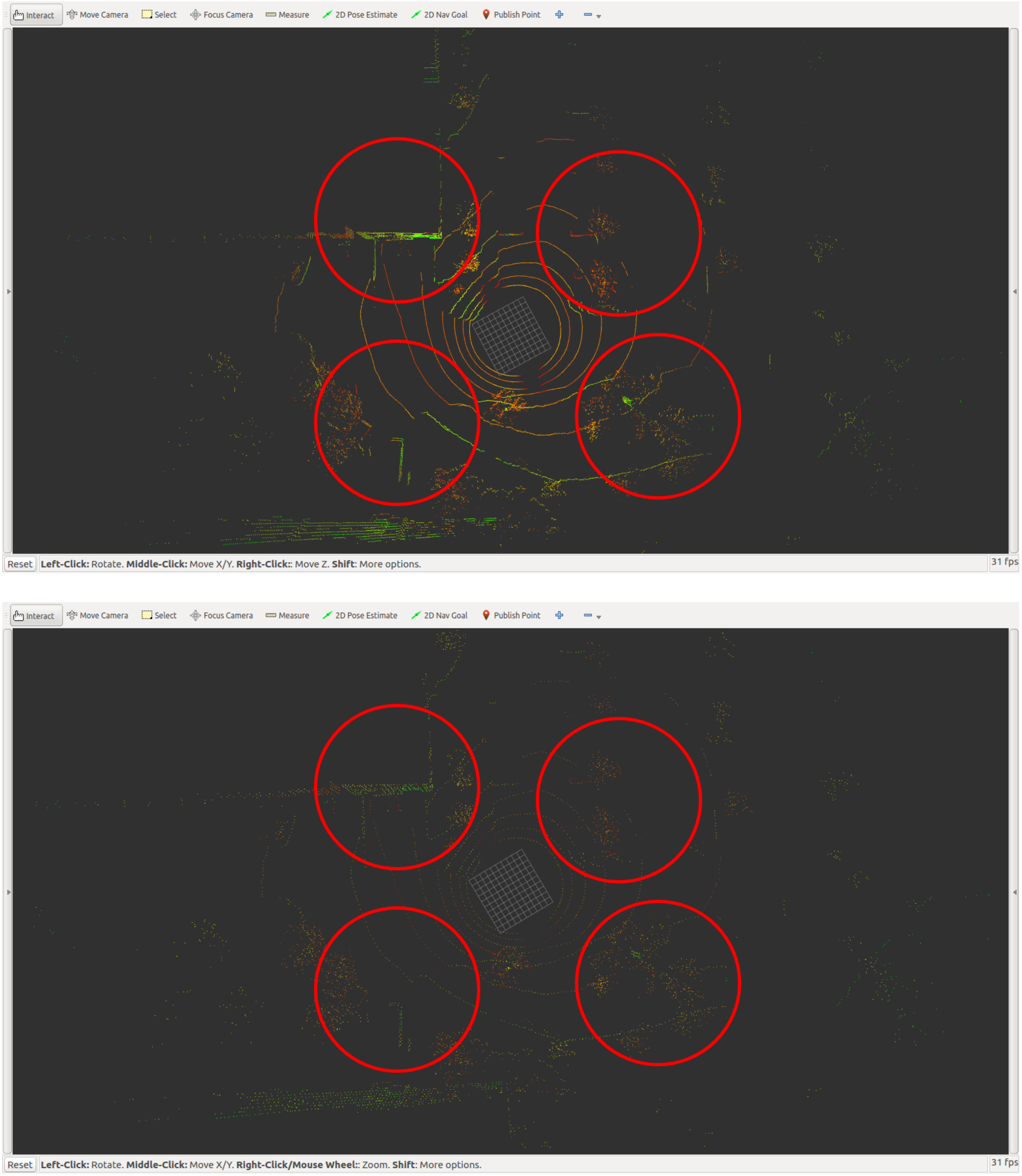
Table 6.3 displays the data averaging results before and after applying K-means filter for a frame.

	Raw Data	K-means Filter
Number of Points (millions)	23.5	16.8

Table 6.3: Summary Table of raw point cloud versus K-means filter data

6.5 Voxel Grid Filter

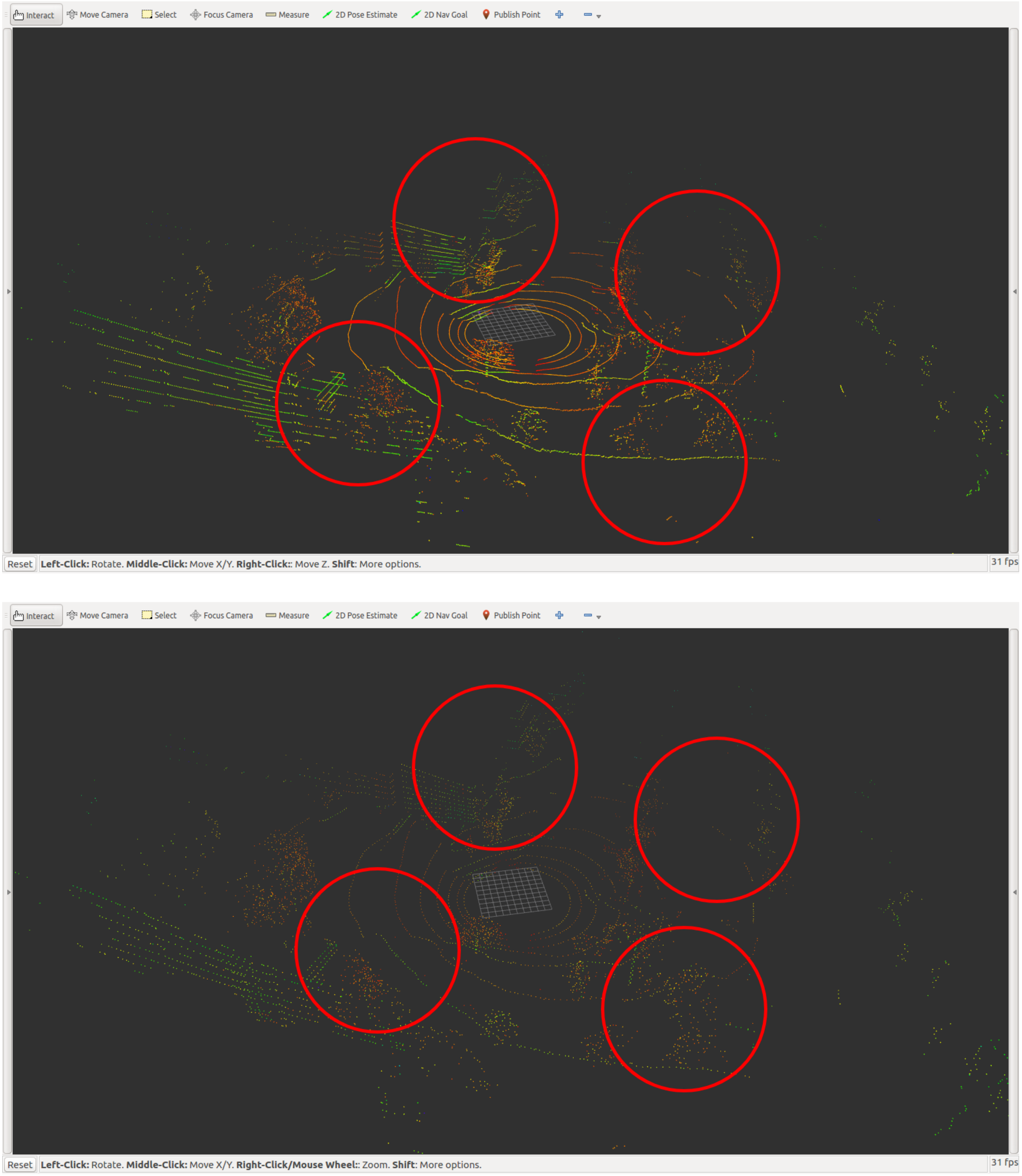
One of the commonly used filters for point cloud data is quantized in a process known as voxelization. While some positive results have been presented using 3D, it is fundamentally cubic, and can immediately become troublesome dealing with large point cloud data. In addition, computation is misdirected as the 3D grid is extremely scattered, i.e., most of the volume is empty. Figures 6.10 and 6.11 shows the before and after applying the filter.



Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click:**: Move Z. **Shift:** More options.

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options.

Figure 6.10: Voxel Grid filter applied on point cloud data from a bird-view



Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click:**: Move Z. **Shift:** More options.

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options.

Figure 6.11: Voxel Grid filter applied on point cloud data from a side-view

A general comparison between raw data of Rotational Multi-Beam LiDAR Sensor (RMBL) where the rotation axis is parallel to one of the frame axes and the results after applying a Voxel Grid filter. Figure 6.12 presents the comparison.

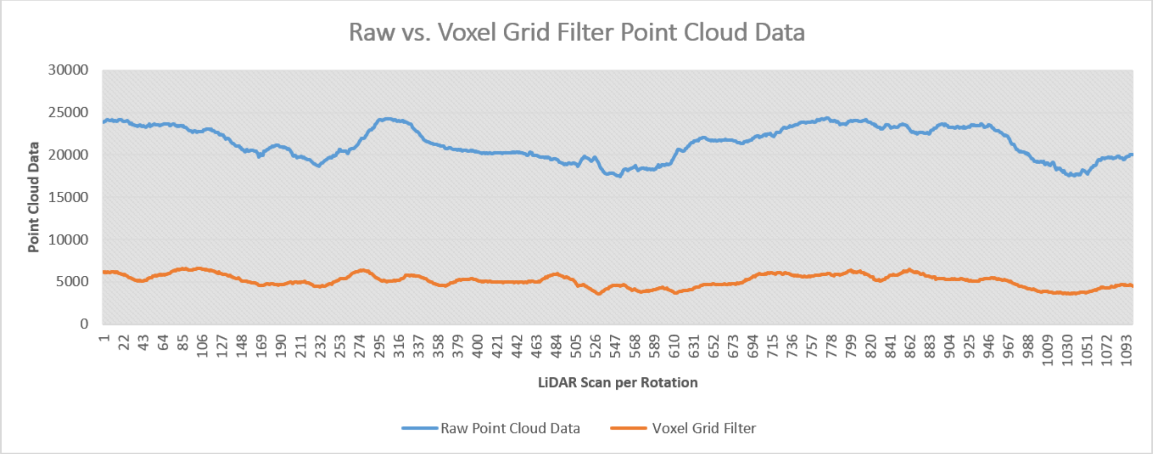


Figure 6.12: Raw point cloud data versus Voxel Grid Filter. The blue colored legend is the raw data without any filter and the orange colored legend is after applying the Voxel Grid filter. The Y-axis is the number of point clouds collected per rotation (X-axis).

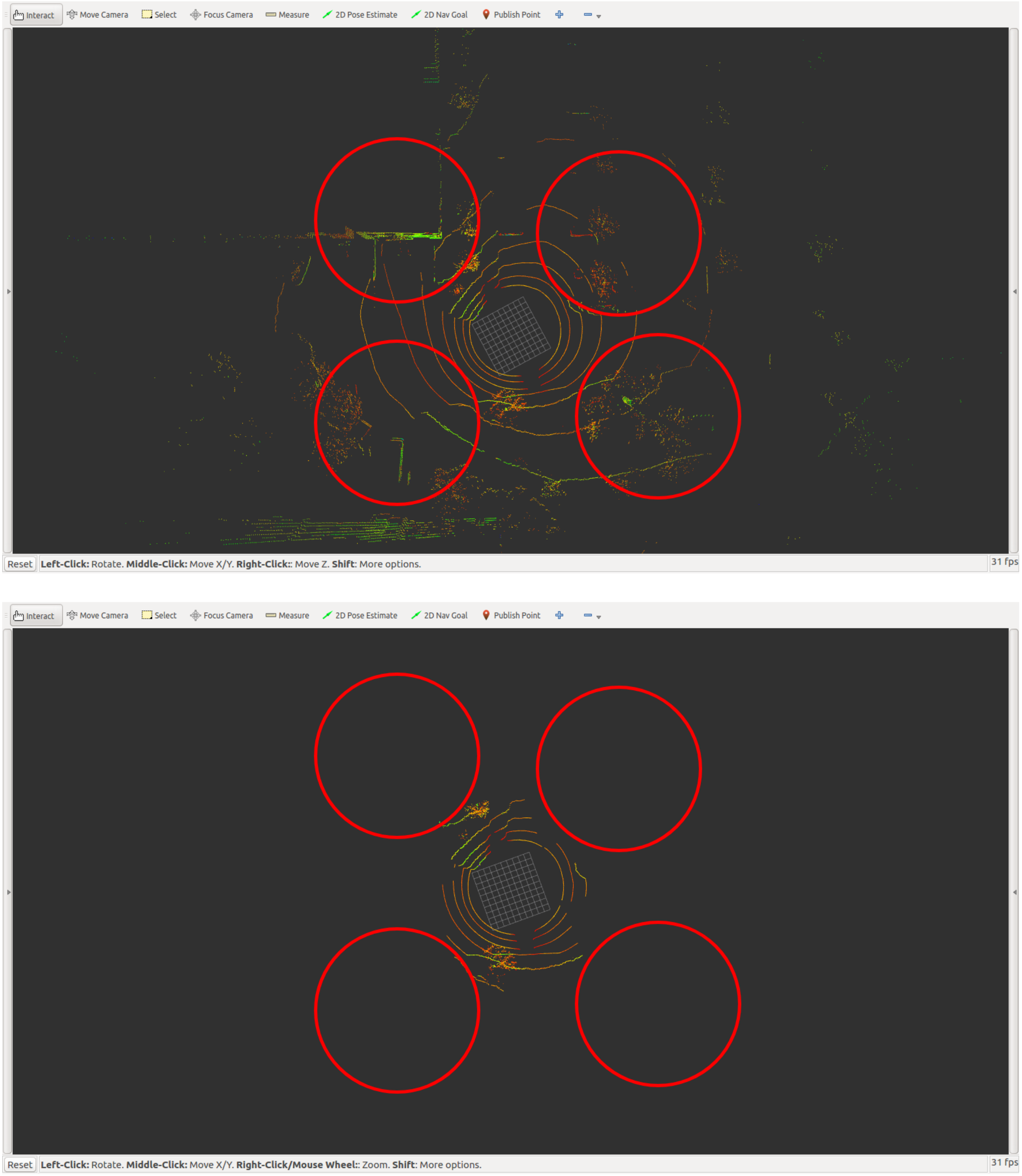
Table 6.4 displays the data averaging results before and after applying Voxel Grid filter for a frame.

	Raw Data	Voxel Grid Filter
Number of Points (millions)	23.5	5.6

Table 6.4: Summary Table of raw point cloud versus Voxel Grid filter data

6.6 Hybrid Filter

Figures 6.13 and 6.14 below; represent the results after going through a hybrid-designed filter. The highlighted red circles clearly present differences that are unique to this filter. There are various point cloud methods and algorithm discussing how data can be optimized but the decision was made to develop a filter that it flexible enough to be adjusted to the environment and to the user selection. As stated earlier, this filter can be modified to be used on other applications such as 3D imaging models.



Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click:**: Move Z. **Shift:** More options.

Left-Click: Rotate. **Middle-Click:** Move X/Y. **Right-Click/Mouse Wheel:**: Zoom. **Shift:** More options.

Figure 6.13: Hybrid filter applied on point cloud data from a bird-view

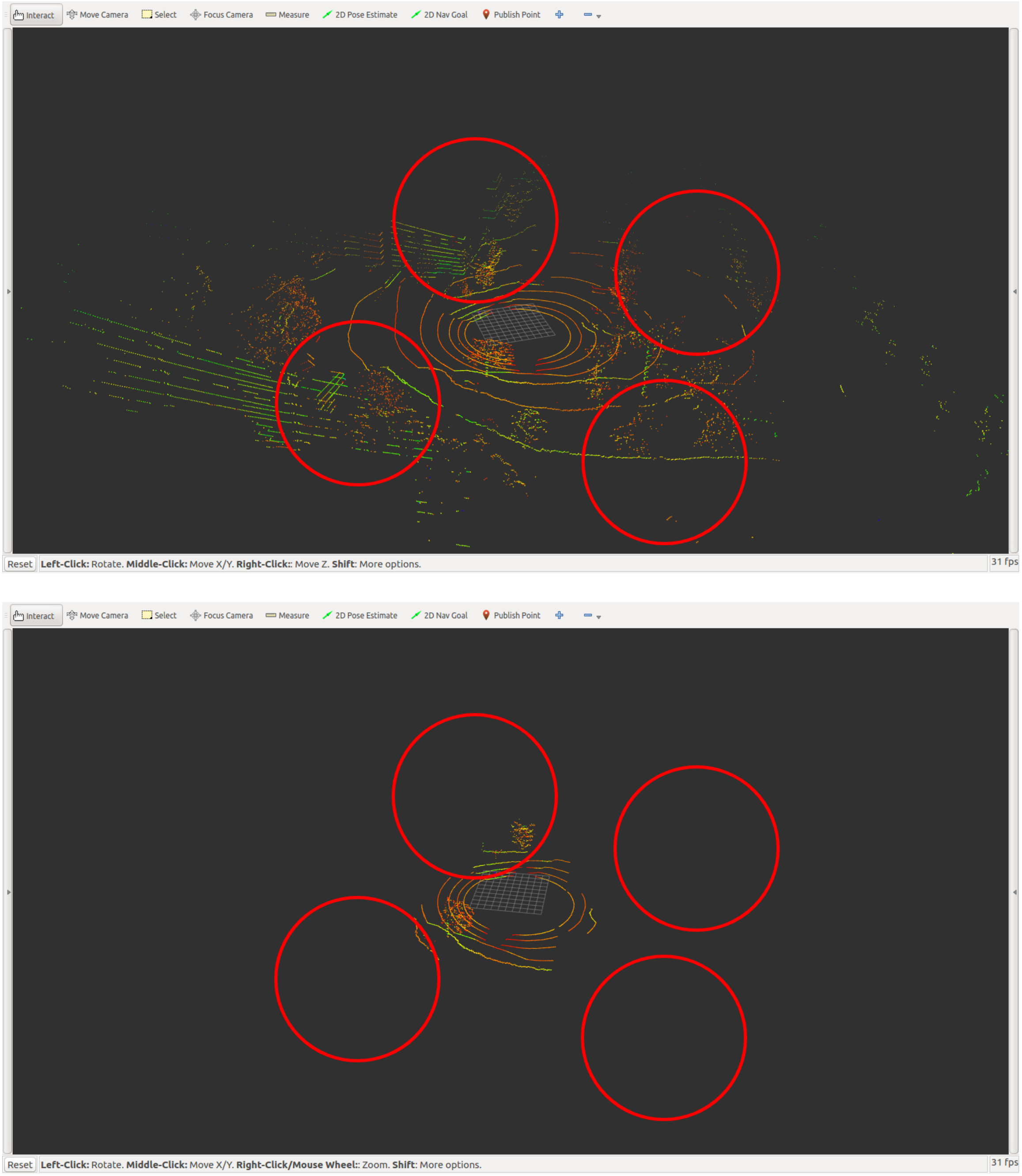


Figure 6.14: Hybrid filter applied on point cloud data from a side-view

Figure 6.15 provides a general comparison between a raw data of a Rotational Multi-Beam LiDAR Sensor (RMBL) where the rotation axis is parallel to one of the frame axes and the results after applying a Hybrid filter.

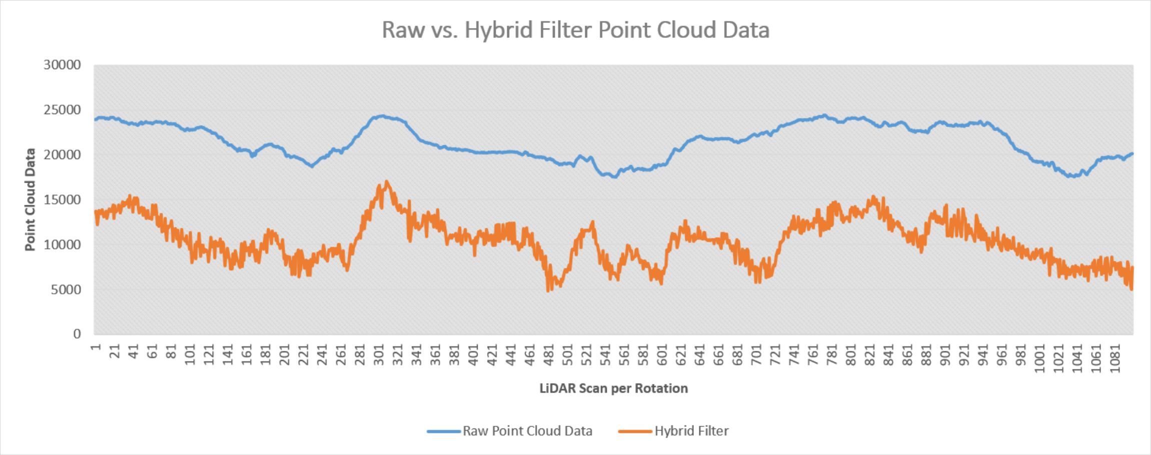


Figure 6.15: Raw point cloud data versus Hybrid Filter. The blue colored legend is the raw data without any filter and the orange colored legend is after applying the Hybrid filter. The Y-axis is the number of point clouds collected per rotation (X-axis).

Additionally, a measuring results of data with other filtering techniques is presented. Those results can be found in 7.1, Table 7.1. Below table, Table 6.5 showcase the data averaging results before and after applying Hybrid filter for a frame.

	Raw Data	Hybrid Filter
Number of Points (millions)	23.5	11.5

Table 6.5: Summary Table of raw point cloud versus Hybrid filter data

6.7 Six Sigma Implementation Process

The analysis used in this research employs Design for Six Sigma (DFSS) process. DFSS is an enhancement to an existing new product development (NPD) tool. It provides a structural way to manage the deliverables, resources, and trade-offs. It employs strategic and tactical method to enhance existing design to achieve entitlement performance. Since there are several competing concepts, a measuring system is essential to determine the amplification that will allow for testing the performance of the system for important design attributes [81].

In this research, Six Sigma technique is used to measure and compare key performance attributes of each filter impeded in this design configuration. This involves ranking each item of importance with a quantified value of intended importance for each item discussed in the evaluation matrix.

6.7.1 Pugh Matrix

A Pugh matrix is a decision-matrix method or concept selection that was invented by Stuart Pugh [82]. It is a qualitative technique used to rank the multi-dimensional options of an option set. It provides a basic decision matrix consisting of establishing a set of criteria options that are scored and summed to gain a total score which can then be ranked.

For this work, different filters to measure their importance based on specific criteria believed to be important to the design aspects. Table 6.6 present the first round of a decision matrix that consists set of a criteria options that are scored and summed for a total score.

Concept Criteria	Raw Data	Parametric Filter	Passthrough Filter	K-means Filter	Voxel Grid Filter	Hybrid Filter
Data Size		+	+	+	+	+
Flexibility	D	+	+	+	+	+
Computation	A	+	+	+	+	+
Accuracy	T	+	-	S	S	+
Ease of Control	U	S	S	+	+	+
Noise Monitoring	M	S	+	+	+	+
$\sum +$ (Positives)		4	4	5	5	6
$\sum -$ (Negatives)		0	1	0	0	0
$\sum S$ (”Sames”)		2	1	1	1	0

Table 6.6: Pugh Matrix of Raw point cloud data versus all other filters worked on in this research. ”+”, ”-”, and ”S” represent if concept is SIGNIFICANTLY better, worse, or same as the datum concept.

Table 6.6 provides the criteria chosen for this research. It is intended to compare the size, flexibility, computation, accuracy, easiness, and noise. The accuracy of this table is related to the positional accuracy of the point cloud. The less noise and data there is to analyze, the higher the positional accuracy. Laser scans typically generate point cloud datasets of varying point densities. Additionally, measurement errors lead to sparse outliers that corrupt the results even more. In the table, a dataset measurement of the noise level and outliers is mentioned.

By improving the data size in a more controllable and manageable situation, the scoring for using new filtering technique becomes higher. To have better results, a comparison of the highest scores is proceeded. Table 6.7 compares K-means and Voxel Grid filtering to the newly designed Hybrid filter using the same criteria assigned in the previous table.

Concept / Criteria	Hybrid Filter	K-means Filter	Voxel Grid Filter
Data Size		-	+
Flexibility	D	-	-
Computation	A	-	+
Accuracy	T	-	-
Ease of Control	U	-	-
Noise Monitoring	M	-	-
$\sum +$ (Positives)		0	2
$\sum -$ (Negatives)		6	4
$\sum S$ (”Sames”)		0	0

Table 6.7: Pugh Matrix of Hybrid Filter versus. K-means clustering and Voxel Grid filtering methods. ”+”, ”-”, and ”S” represent if concept is SIGNIFICANTLY better, worse, or same as the datum concept.

Based on the results presented, the Hybrid solution has the best performance when it comes to data size, flexibility, computation, accuracy, ease of control, and noise. It is clear that the number of negatives compared with K-means and Voxel Grid filters are more than the positives. This is one of the main reasons the Hybrid filter solution exceed the performance of other filers. It combines k-means and voxel techniques and adds an essential step that examine the density of the point cloud data. The research shows that proposed filter configuration together with the developed control algorithm provide significant reduction of the point cloud data. The results obtained in simulations are proven by a laboratory analysis on a real system.

CHAPTER 7
CONCLUSION AND FUTURE WORK

7.1 Conclusion

This chapter covers the conclusion and contribution derived from this research. Followed by future work recommendation that can complement this work. In this dissertation, a novel way of optimizing raw point cloud data within ROS environment was presented. Since it has both a theoretical and practical component, a discussion of the main conclusions, the bigger picture, and future directions.

In Chapter 3, a real-time data acquisition system with LiDAR perception system was developed and assembled on a vehicle for this application. The LiDAR plays an important role in decision making for autonomous vehicle implementation. Keeping with the goal to achieve an optimized, robust, and reliable solution of point cloud data, this approach has been presented, validated, and demonstrated in a real-time implementation.

The effects of the sparse outlier analysis and removal of point cloud data have been presented to include a comparison, in Chapter 4 and 5, between original dataset, and before and after filtering. As stated in previous chapters and without loss of generality, VLP-16 was considered for this work for the addition of a rotation mechanism. This Multi-Beam LiDAR (MBL) sensor was especially suitable for its cost, lighter weight, symmetric FOV, and compact size. Chapter 2 provided deep understanding of LiDAR sensor and point cloud data.

In this dissertation, a combined of multiple solutions and techniques provided to ensure the optimal results without losing important point cloud data. As presented in Chapter 4 and 5, a flexible system based on ROS was used as a tool to add/remove filters/programs as required. During this work great understanding of point cloud data usage, structure, and data-mining was learned and discovered that provided the opportunity to control the dataset by reducing noise and size.

To move towards robust fully autonomous driving systems, high definition static maps need to be constructed and shared between vehicles and supporting infrastructure. From experience relating to mapping and from the literature reviewed, it is apparent that an element of data reduction must be considered to handle the volume of data produced by the point clouds generated by technologies such as LiDARs and Stereo-cameras. In addition to the data sizes, localization is often handled through highly computationally expensive algorithms focused on matching the current perception frames to past frames. When dealing with high quantities of points, these algorithms are forced to operate at much lower frequencies, potentially limiting the top speed of the vehicle. This research presents a fusion solution of Voxel and K-Means that ensures only highly dense areas are recorded and the remainder of the points is eliminated. As showcased in Table 6.5 in Chapter 6, total point count was reduced from 23.5 to 11.5 million. Within the same frame, critical features were maintained after the hybrid-filtering approach. This filtering mythology provides a basis for lightweight maps in which landmarks are emphasized for efficient localization, and updates to the static layer of the vehicles maps.

Table 7.1 summarizes the work and compares it with other known point cloud filter configurations. The Voxel Grid filter stand-alone provides the lowest number of point cloud data but based on the analysis and research it reduces the size of the data but does not eliminate the outliers.

	Raw	Parametric	Passthrough	K-Means	Voxel	Hybrid
Data Points	23.5	23.5	14.1	16.8	5.6	11.5

Table 7.1: Summary Table of raw point cloud data versus multiple filtered data in millions

7.2 Future Work

The performance of the proposed point cloud filtering approach is primarily dependent on the environment used to collect the data. The approach was not evaluated in harsh weather conditions that definitely would affect the collected data. The system might reduce the data size and eliminate noise but the original data quality might be poor.

Director of Mcity – University of Michigan, Dr. Huei Peng, said that although automated driving features are coming soon, “it may be decades before a vehicle can drive itself safely at any speed on any road in any weather.” Gill Pratt, CEO of the Toyota Research Institute (TRI) and former MIT professor stated, “Level 5 autonomy—when a car can drive completely autonomously in any traffic or weather condition is a wonderful goal but none of us in the automobile or IT industries are close to achieving true Level 5 autonomy.”

Additional testing is necessary on different road types such as country, urban, and suburban roads. The system is flexible enough to allow the user to choose certain parameters such as length, width, height of an individual box, and density threshold for erasing point clouds data within a given box. However, this has not been tested to suggest what parameters to use in different environment.

The mapping technology has moved from print to digital and autonomous applications are moving to high definition (HD) maps that will provide sub-meter accuracy. The more accurate it is, the more data will be used for autonomous driving. Pushing this data up to the cloud to maintain its freshness will require a fast internet service. This area of work needs to be evaluated to understand what level of filtering is required or acceptable.

REFERENCES

[1] G Fiocco and L. Smullin, “Detection of scattering layers in the upper atmosphere (60–140 km) by optical radar,” *Nature*, vol. 199, no. 4900, p. 1275, 1963.

[2] R. Thomas and G. Guenther, “Theoretical characterization of bottom returns for bathymetric lidar,” in *International Conference on Lasers*, 1979, pp. 48–59.

[3] H. J. Zwally, R. Bindschadler, and R. Thomas, “Ice-sheet dynamics by satellite laser altimetry,” in *International Geoscience and Remote Sensing Symposium*, 1981, pp. 1012–1022.

[4] J. C. K. S. K. W. L. B. B. H. R. Mataosky and J. Halleran, *Lidar 101: And Introduction to Lidar Technology, Data, and Applications*. National Oceanic and Atmospheric Administration (NOAA) Coastal Services Center, 2012.

[5] P Seitz, “Photon-noise limited distance resolution of optical metrology methods,” in *Proceedings of SPIE Volume 6616 Optical Measurement Systems for Industrial Inspection V*, 2007.

[6] P. J. Besl, “Active, optical range imaging sensors,” *Mach. Vision Appl.*, vol. 1, no. 2, pp. 127–152, 1988.

[7] D. Nitzan, “Three-dimensional vision structure for robot applications,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 10, no. 3, pp. 291–309, 1988.

[8] B. Jahne, H. W. Haussecker, and P. Geissler, *Handbook of Computer Vision and Applications. 1. Sensors and Imaging*. Academic Press, 1999.

[9] F. Blais, “Review of 20 years of range sensor development,” *J. Electronic Imaging*, vol. 13, no. 1, pp. 231–243, 2004.

[10] S. Robson, J.-A. Beraldin, A. Brownhill, and L. MacDonald, “Artefacts for optical surface measurement,” in *Proceedings of the SPIE-IS&T in Videometrics, Range Imaging, and Applications XI*, 2011.

[11] A. Wehr and U. Lohr, “Airborne laser scanning—an introduction and overview,” *ISPRS Journal of photogrammetry and remote sensing*, vol. 54, no. 2, pp. 68–82, 1999.

[12] D. C. Carmer and L. M. Peterson, “Laser radar in robotics,” *Proceedings of the IEEE*, vol. 84, no. 2, pp. 299–320, 1996.

[13] W. Boehler and A. Marbs, “Investigating scanner accuracy,” German University FH Mainz, Tech. Rep., 2003.

[14] *ASME B89.4.19 performance evaluation tests and geometric misalignments in laser trackers*, 2009.

[15] H. T. Yura, “Atmospheric turbulence induced laser beam spread,” *Applied optics*, vol. 10, no. 12, pp. 2771–2773, 1971.

[16] E. P. Baltsavias, “Airborne laser scanning: Basic relations and formulas,” *ISPRS Journal of photogrammetry and remote sensing*, vol. 54, no. 2, pp. 199–214, 1999.

[17] J. Ryan and A. Carswell, “Laser beam broadening and depolarization in dense fogs,” *Journal of the Optical Society of America A*, vol. 68, no. 7, pp. 900–908, 1978.

[18] C. Grönwall, O. Steinvall, F. Gustafsson, and T. Chevalier, “Influence of laser radar sensor parameters on range-measurement and shape-fitting uncertainties,” *Optical Engineering*, vol. 46, no. 10, pp. 106 201–106 201, 2007.

[19] J. Hancock, D. Langer, M. Hebert, R. Sullivan, D. Ingimarson, E. Hoffman, M. Mettenleiter, and C. Froehlich, “Active laser radar for high-performance measurements,” in *Robotics and Automation, 1998. Proceedings. 1998 IEEE International Conference on*, vol. 2, 1998, 1465–1470 vol.2.

[20] S. El-Hakim, J.-A. Beraldin, M. Picard, and L. Cournoyer, *Surface reconstruction of large complex structures from mixed range data-The Erechtheion experience*. Proceedings of the XXI Congress of the International Society for Photogrammetry and Remote Sensing, 2008.

[21] G. Godin, M. Rioux, J.-A. Beraldin, M. Levoy, L. Cournoyer, and F. Blais, “An assessment of laser range measurement on marble surfaces,” in *Proceedings of the 5th Conference on Optical 3D Measurement Techniques Vienna Austria*, 2001.

[22] G. Guidi, F. Remondino, M. Russo, and A. Spinetti, “Range sensors on marble surfaces: Quantitative evaluation of artifacts,” in *SPIE Optical Engineering+ Applications*, International Society for Optics and Photonics, 2009, pp. 744 703–744 703.

[23] N. Mak, J.-A. Beraldin, L. Cournoyer, M. Picard, *et al.*, “A distance protocol for mid-range TLS in support of ASTM-E57 standards activities,” in *Proceedings of the ISPRS Commission V Mid-Term Symposium’Close Range Image Measurement Techniques’*, Citeseer, vol. 38, 2010, pp. 428–433.

[24] D. D. Lichti and B. Harvey, “The effects of reflecting surface material properties on time-of-flight laser scanner measurements,” in *Proceedings of the Symposium on Geospatial Theory, Processing and Applications. ISPRS Commission IV*, 2002.

[25] A. N. S. Institute and L. I. of America, *American National Standard for safe use of lasers*. Laser Institute of America, 2007.

[26] W. J. Smith, *Modern Optical Engineering*, third. McGraw-Hill, 2000.

[27] G. Petrie and C. K. Toth, *Topographic Laser Ranging and Scanning*. CRC Press, 2009.

[28] M. A. Lefsky, W. B. Cohen, G. G. Parker, and D. J. Harding, "Lidar remote sensing for ecosystem studies: Lidar, an emerging remote sensing technology that directly measures the three-dimensional distribution of plant canopies, can accurately estimate vegetation structural attributes and should be of particular interest to forest, landscape, and global ecologists," *AIBS Bulletin*, vol. 52, no. 1, pp. 19–30, 2002.

[29] A. F. Chase, D. Z. Chase, J. F. Weishampel, J. B. Drake, R. L. Shrestha, K. C. Slatton, J. J. Awe, and W. E. Carter, "Airborne lidar, archaeology, and the ancient maya landscape at caracol, belize," *Journal of Archaeological Science*, vol. 38, no. 2, pp. 387–398, 2011.

[30] C. Mallet and F. Bretar, "Full-waveform topographic lidar: State-of-the-art," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 64, no. 1, pp. 1 –16, 2009.

[31] A. S. Whitehurst, A. Swatantran, J. B. Blair, M. A. Hofton, and R. Dubayah, "Characterization of canopy layering in forested ecosystems using full waveform lidar," *Remote Sensing*, vol. 5, no. 4, pp. 2014–2036, 2013.

[32] T. Adams, P. Beets, and C. Parrish, "Extracting more data from lidar in forested areas by analyzing waveform shape," *Remote Sensing*, vol. 4, no. 3, pp. 682–702, 2012.

[33] A. Gelbart, S. Bybee-Driscoll, J. Freeman, G. J. Fetzer, D. Wasson, K. Hanna, and W.-Y. Zhao, "Signal processing, image registration, and visualization of flash lidar data," vol. 5086, 2003, pp. 5086 –5086 –12.

[34] A. Bulyshev, D. P. F. Amzajerdian, G. Busch, M. Vanek, and R. Reisse, "Processing of three-dimensional flash lidar terrain images generating from an airborne platform," vol. 7329, 2009, pp. 7329 –7329 –9.

[35] W. Wagner, A. Ullrich, V. Ducic, T. Melzer, and N. Studnicka, "Gaussian decomposition and calibration of a novel small-footprint full-waveform digitising airborne laser scanner," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 60, no. 2, pp. 100 –112, 2006.

[36] M. C. Amann, T. Bosch, M. Lescure, R. Myllyla, and M. Rioux, "Laser ranging: A critical review of usual techniques for distance measurement," *Optical Engineering*, vol. 40, no. 1, 2001.

[37] M. H. Conde, *Compressive Sensing for the Photonic Mixer Device - Fundamentals, Methods and Results*. Springer Vieweg, 2017.

[38] I. BIPM, I. IFCC, I. IUPAC, and O. ISO, "Evaluation of measurement data—guide for the expression of uncertainty in measurement. jcgm 100: 2008," *Citado en las*, vol. 167, 2008.

[39] G. Gräfe, "Kinematic 3d laser scanning for road or railway construction surveys," in *Proceedings of the International Conference on Machine Control & Guidance*, Citeseer, 2008, pp. 24–26.

[40] C. Ai and Y. J. Tsai, "Critical assessment of automatic traffic sign detection using three-dimensional lidar point cloud data," Tech. Rep., 2012.

[41] S. I. El-Halawany and D. D. Lichti, "Detection of road poles from mobile terrestrial laser scanner point cloud," in *2011 International Workshop on Multi-Platform/Multi-Sensor Remote Sensing and Mapping*, IEEE, 2011, pp. 1–6.

[42] B. Yang, L. Fang, Q. Li, and J. Li, "Automated extraction of road markings from mobile lidar point clouds," *Photogrammetric Engineering & Remote Sensing*, vol. 78, no. 4, pp. 331–338, 2012.

[43] J. Wang, H. González-Jorge, R. Lindenbergh, P. Arias-Sánchez, and M. Menenti, "Automatic estimation of excavation volume from laser mobile mapping data for mountain road widening," *Remote Sensing*, vol. 5, no. 9, pp. 4629–4651, 2013.

[44] J. Aulinas, Y. R. Petillot, J. Salvi, and X. Lladó, "The slam problem: A survey.," *CCIA*, vol. 184, no. 1, pp. 363–371, 2008.

[45] T. Taketomi, H. Uchiyama, and S. Ikeda, "Visual slam algorithms: A survey from 2010 to 2016," *IPSJ Transactions on Computer Vision and Applications*, vol. 9, no. 1, p. 16, 2017.

[46] A. A. Makarenko, S. B. Williams, F. Bourgault, and H. F. Durrant-Whyte, "An experiment in integrated exploration," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, IEEE, vol. 1, 2002, pp. 534–539.

[47] H. G. Seif and X. Hu, "Autonomous driving in the icity—hd maps as a key challenge of the automotive industry," *Engineering*, vol. 2, no. 2, pp. 159 –162, 2016.

[48] S. Chen, Y. F. Li, and J. Zhang, "Vision processing for realtime 3-d data acquisition based on coded structured light," *IEEE Transactions on Image Processing*, vol. 17, no. 2, pp. 167–176, 2008.

[49] S. Chen and Y. Li, "Automatic sensor placement for model-based robot vision," *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, vol. 34, no. 1, pp. 393–408, 2004.

[50] K. Zhang, S.-C. Chen, D. Whitman, M.-L. Shyu, J. Yan, and C. Zhang, "A progressive morphological filter for removing nonground measurements from airborne lidar data," *IEEE transactions on geoscience and remote sensing*, vol. 41, no. 4, pp. 872–882, 2003.

[51] G. Sithole and G. Vosselman, "Filtering of laser altimetry data using a slope adaptive filter," *International Archives of Photogrammetry Remote Sensing and Spatial Information Sciences*, vol. 34, no. 3/W4, pp. 203–210, 2001.

[52] S. Papadimitriou, H. Kitagawa, P. B. Gibbons, and C. Faloutsos, "Loci: Fast outlier detection using the local correlation integral," in *Proceedings 19th International Conference on Data Engineering (Cat. No. 03CH37405)*, IEEE, 2003, pp. 315–326.

[53] X. Xu, D. Song, X. Zhang, *et al.*, "Research of detection of outliers based on locally linear weighted value," *Computer Science*, vol. 35, no. 5, pp. 154–157, 2008.

[54] M. M. Breunig, H.-P. Kriegel, R. T. Ng, and J. Sander, "Lof: Identifying density-based local outliers," in *ACM sigmod record*, ACM, vol. 29, 2000, pp. 93–104.

[55] S. Chen, J. Zhang, Y. Li, and J. Zhang, "A hierarchical model incorporating segmented regions and pixel descriptors for video background subtraction," *IEEE Transactions on Industrial Informatics*, vol. 8, no. 1, pp. 118–127, 2011.

[56] A. Bab-Hadiashar and D. Suter, "Robust optic flow computation," *International Journal of Computer Vision*, vol. 29, no. 1, pp. 59–77, 1998.

[57] E. P. Ong and M. Spann, "Robust optical flow computation based on least-median-of-squares regression," *International Journal of Computer Vision*, vol. 31, no. 1, pp. 51–82, 1999.

[58] P. F. Gotardo, O. R. P. Bellon, and L. Silva, "Range image segmentation by surface extraction using an improved robust estimator," in *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings.*, IEEE, vol. 2, 2003, pp. II–33.

[59] K. Koster and M. Spann, "Mir: An approach to robust clustering-application to range image segmentation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 5, pp. 430–444, 2000.

[60] P. H. Torr and D. W. Murray, “The development and comparison of robust methods for estimating the fundamental matrix,” *International journal of computer vision*, vol. 24, no. 3, pp. 271–300, 1997.

[61] P. H. Torr and A. Zisserman, “Mlesac: A new robust estimator with application to estimating image geometry,” *Computer vision and image understanding*, vol. 78, no. 1, pp. 138–156, 2000.

[62] M. J. Black and A. D. Jepson, “Eigentracking: Robust matching and tracking of articulated objects using a view-based representation,” *International Journal of Computer Vision*, vol. 26, no. 1, pp. 63–84, 1998.

[63] J. Wang, H. Zhao, D. Wang, Y. Chen, Z. Zhang, and H. Liu, “Gps trajectory-based segmentation and multi-filter-based extraction of expressway curbs and markings from mobile laser scanning data,” *European Journal of Remote Sensing*, vol. 51, no. 1, pp. 1022–1035, 2018.

[64] C. Moreno and M. Li, “A comparative study of filtering methods for point clouds in real-time video streaming,” in *Internet and Multimedia Technologies (ICIMT), 2016 International Conference on*, 2016.

[65] J. MacQueen *et al.*, “Some methods for classification and analysis of multivariate observations,” in *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, Oakland, CA, USA, vol. 1, 1967, pp. 281–297.

[66] D. Comaniciu and P. Meer, “Mean shift: A robust approach toward feature space analysis,” *IEEE Transactions on Pattern Analysis & Machine Intelligence*, no. 5, pp. 603–619, 2002.

[67] G. Lavoué, F. Dupont, and A. Baskurt, “A new cad mesh segmentation method, based on curvature tensor analysis,” *Computer-Aided Design*, vol. 37, no. 10, pp. 975–987, 2005.

[68] H. Yamauchi, S. Lee, Y. Lee, Y. Ohtake, A. Belyaev, and H.-P. Seidel, “Feature sensitive mesh segmentation with mean shift,” in *International Conference on Shape Modeling and Applications 2005 (SMI’05)*, IEEE, 2005, pp. 236–243.

[69] X. Zhang, G. Li, Y. Xiong, and F. He, “3d mesh segmentation using mean-shifted curvature,” in *International Conference on Geometric Modeling and Processing*, Springer, 2008, pp. 465–474.

[70] Y. LeCun, Y. Bengio, and G. Hinton, “Deep learning,” *nature*, vol. 521, no. 7553, p. 436, 2015.

[71] S. Orts-Escolano, V. Morell, J. García-Rodríguez, and M. Cazorla, “Point cloud data filtering and downsampling using growing neural gas,” in *The 2013 International Joint Conference on Neural Networks (IJCNN)*, IEEE, 2013, pp. 1–8.

[72] *MS Windows NT kernel description*, `https://libpointmatcher.readthedocs.io/en/latest/DataPointsFilterDev/`, Accessed: 2010-09-30.

[73] T. Shan and B. Englot, “Lego-loam: Lightweight and ground-optimized lidar odometry and mapping on variable terrain,” in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, IEEE, 2018, pp. 4758–4765.

[74] C. K. Toth, “R&d of mobile lidar mapping and future trends,” in *Proc. ASPRS Annu. Conf*, 2009, pp. 9–13.

[75] `http://wiki.ros.org/kinetic/Installation/Ubuntu`.

[76] I. Hamieh, R. Myers, and T. Rahman, “Construction of autonomous driving maps employing lidar odometry,” in *2019 IEEE Canadian Conference of Electrical and Computer Engineering (CCECE)*, IEEE, 2019, pp. 1–4.

[77] G Marinelli, M Bassani, M Piras, and A. Lingua, “Mobile mapping systems and spatial data collection strategies assessment in the identification of horizontal alignment of highways,” *Transportation Research Part C: Emerging Technologies*, vol. 79, pp. 257–273, 2017.

[78] S. Y. Choi and J. M. Lee, “Applications of moving windows technique to autonomous vehicle navigation,” *Image and Vision Computing*, vol. 24, no. 2, pp. 120–130, 2006.

[79] R. P. D. Vivacqua, M. Bertozzi, P. Cerri, F. N. Martins, and R. F. Vassallo, “Self-localization based on visual lane marking maps: An accurate low-cost approach for autonomous driving,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 2, pp. 582–597, 2017.

[80] P. Mucka, G. J. Stein, and P. Tobolka, “Passsenger ride comfort and international roughness index specifications in the slovak republic,” *Communications-Scientific letters of the University of Zilina*, vol. 21, no. 1, pp. 14–21, 2019.

[81] D. P. Mader, “Design for six sigma,” *Quality progress*, vol. 35, no. 7, pp. 82–86, 2002.

[82] S. Pugh, “Concept selection: A method that works,” in *Proceedings of the International conference on Engineering Design*, 1981, pp. 497–506.

VITA AUCTORIS

Ismail Hamieh was born in Kuwait City, Kuwait and migrated to North America when he was young. He started his journey at University of Michigan and earned his B.Sc. in Computer Engineering in 2003. While working full-time, we pursued his Master's degree in Electrical Engineering in 2007. He worked for multiple automotive companies such as Chrysler and General Motors. His experience and interested in automotive industry promoted his research in his PhD at University of Windsor.