

University of Nevada, Reno

# **LDAT: A Web Data Visualization Tool for LiDAR Point Cloud Data Analysis**

A thesis submitted in partial fulfillment of the  
requirements for the degree of Master of Science  
in Computer Science and Engineering

by

Andrew Edward Muñoz

Dr. Frederick C Harris, Jr., Thesis Co-Advisor  
&  
Dr. Sergiu M. Dascalu, Thesis Co-Advisor

August, 2021

© by Andrew E. Muñoz  
All Rights Reserved



THE GRADUATE SCHOOL

We recommend that the thesis  
prepared under our supervision by

**Andrew E. Muñoz**

Entitled

**LDAT: A Web Data Visualization Tool for  
LiDAR Point Cloud Data Analysis**

be accepted in partial fulfillment of the requirements  
for the degree of

MASTER OF SCIENCE

Dr. Frederick C. Harris, Jr., Co-Advisor

Dr. Sergiu M. Dascalu, Co-Advisor

Dr. Scotty Strachan, Graduate School Representative

David W. Zeh, Ph.D., Dean, Graduate School

August, 2021

## **Abstract**

Light Detection and Ranging (LiDAR) sensors have been employed in many different ways over time and continue to be utilized today. These sensors produce point clouds which are large and complex data sets that are a collection of position points across a 3D space. The research presented in this thesis focuses on the analysis and visualization of LiDAR point cloud data. The data obtained for this project is from LiDAR sensors located on street lights on Virginia Street to analyze traffic information. A web tool was developed to analyze and visualize this data, ensuing in an interactive and readable representation of the data. In order to ensure the effectiveness of the tool, a user study was conducted to test the functionality and assess possible improvements. This thesis aims to provide a template for creating an effective and a useful data visualization tool in an increasingly data-driven society.

## Dedication

I dedicate this thesis to my parents, Edward and Lucy Muñoz, for their unconditional love and unwavering support throughout this journey.

## Acknowledgments

I would like to sincerely thank my advisors Dr.Sergiu M. Dascalu and Dr. Frederick C. Harris, Jr. and committee member Dr. Scotty Strachan for all their time and guidance to help me in the completion of this thesis. I'd also like to thank both Chase Carthen and Vinh Le graciously for their assistance on the software development of this thesis.

# Contents

<b>Abstract</b>	<b>i</b>
<b>Dedication</b>	<b>ii</b>
<b>Acknowledgments</b>	<b>iii</b>
<b>List of Tables</b>	<b>vii</b>
<b>List of Figures</b>	<b>viii</b>
<b>1 Introduction</b>	<b>1</b>
<b>2 Background</b>	<b>4</b>
2.1 Introduction . . . . .	4
2.2 Technology . . . . .	4
2.2.1 LiDAR . . . . .	4
2.2.2 Point Clouds . . . . .	13
2.2.3 Data Visualization . . . . .	16
2.3 Data . . . . .	19
2.3.1 Data Collection . . . . .	19
2.3.2 Locations . . . . .	20
2.4 Tools . . . . .	22
2.4.1 Angular . . . . .	22
2.4.2 Flask . . . . .	23
2.4.3 MQTT Networking Protocol . . . . .	23
2.4.4 Docker . . . . .	24
2.4.5 Python . . . . .	24
2.4.6 JavaScript . . . . .	25
2.4.7 WebGL & Three.js . . . . .	25
<b>3 Related Work</b>	<b>26</b>
3.1 Similar Works in Research . . . . .	26
3.1.1 LiDAR Sensor Traffic Detection System . . . . .	27
3.1.2 3DSYSTEK: A Web Based 3D Point Cloud Viewer . . . . .	27
3.1.3 LiDAR Data Management and Storage Studies . . . . .	29
3.2 Additional Applications . . . . .	30

3.2.1	LiDAR Visualization Tool for 3D Annotation . . . . .	30
3.2.2	LidarView: A Data Processing and Visualization Tool . . . . .	31
<b>4</b>	<b>Software Architecture and Implementation</b>	<b>33</b>
4.1	Introduction . . . . .	33
4.2	Software Requirements . . . . .	34
4.2.1	Functional Requirements . . . . .	34
4.2.2	Non-Functional Requirements . . . . .	37
4.3	Use Case Modeling . . . . .	38
4.4	Architecture Design . . . . .	41
4.4.1	System Design . . . . .	41
4.4.2	Application Design . . . . .	43
4.5	Implementation . . . . .	47
<b>5</b>	<b>Prototype in Action</b>	<b>53</b>
5.1	Introduction . . . . .	53
5.2	User Interface . . . . .	53
5.3	Visualization and Analysis . . . . .	57
5.3.1	Dashboard Visualizations . . . . .	58
5.3.2	Configurable Components . . . . .	60
<b>6</b>	<b>User Study</b>	<b>63</b>
6.1	Introduction . . . . .	63
6.2	Participants . . . . .	64
6.3	Apparatus . . . . .	65
6.4	Procedure . . . . .	66
6.5	Tasks . . . . .	67
6.6	Design . . . . .	68
<b>7</b>	<b>Results and Analysis</b>	<b>70</b>
7.1	Data From Pre Questionnaire . . . . .	70
7.2	Data Analysis . . . . .	72
7.3	Data From Post Questionnaire . . . . .	79
7.4	Discussion . . . . .	83
<b>8</b>	<b>Conclusion and Future Work</b>	<b>85</b>
8.1	Conclusions . . . . .	85
8.2	Future Work . . . . .	86
	<b>Bibliography</b>	<b>88</b>
	<b>Appendices</b>	<b>93</b>
	<b>A UNR IRB Application Cover Sheet</b>	<b>93</b>



<b>B UNR IRB Exempt Application</b>	<b>98</b>
<b>C UNR IRB Consent Information Sheet</b>	<b>104</b>
<b>D UNR IRB Pre-Test Survey</b>	<b>108</b>
<b>E UNR IRB Post-Test Survey</b>	<b>111</b>
<b>F UNR IRB Participant Recruitment Email</b>	<b>114</b>

# List of Tables

4.1	List of Functional Requirements . . . . .	35
4.2	List of Non-Functional Requirements . . . . .	38
4.3	Full List of Detailed Use Case Descriptions . . . . .	40
6.1	Overview of tasks participants asked to perform in user study . . . . .	67
6.2	Measurements taken from tasks . . . . .	69
7.1	Task Completion Time for each task . . . . .	73
7.2	Number of Left Clicks for each task . . . . .	75
7.3	Number of Right Clicks for each task . . . . .	76
7.4	Number of Mouse Wheel Scrolls for each task . . . . .	78
7.5	Weighted average of ratings for post-questionnaire ranged-based questions . . . . .	82
7.6	FPS performance of point cloud data stream measured using three different methods. . . . .	84

# List of Figures

2.1	Corner cube reflector placed on the moon in 1969 that contained a high powered ruby laser to allow for ranging purposes [32]. . . . .	6
2.2	An image showing a military soldier using the AN/GVS5 Binoculars [19] in the field. . . . .	7
2.3	Displays early 3D laser radar scans [32] from 1983. . . . .	8
2.4	Diagram that illustrates an airborne LiDAR data capture with the red dots showing where the LiDAR data points are located [28]. . . . .	10
2.5	Example image of LiDAR point cloud data [28] represented by a 3D model where the heights of the objects in the digital environment are indicated by the color displayed. . . . .	11
2.6	Diagrams that show the MEMS-based scanning LiDAR system (left) and Flash-LiDAR system (right) [12]. . . . .	12
2.7	Graph that exemplifies PCL split into different code libraries [41]. . .	15
2.8	Image from [22] that highlights the various types of data visualization methods. . . . .	18
2.9	Screenshot of the initial test environment running in a browser showcasing the WebGL 3D visualization of the LiDAR point cloud data. .	21
2.10	The MQTT publish and subscribe architecture [36]. . . . .	23
4.1	Use case diagram for LDAT. . . . .	39
4.2	System level diagram that showcases the main system components of LDAT. . . . .	42
4.3	Website map for LDAT application. . . . .	43
4.4	Activity diagram for completing the data explorer form. . . . .	44
4.5	Activity diagram to change the point cloud color on the Settings page.	45
4.6	Complete class diagram for the LDAT application. . . . .	46
4.7	Velodyne LiDAR Ultra Puck Sensor [54]. . . . .	48
4.8	Overview image, obtained from Google Maps [15], of the street intersection of East 15th Street and North Virginia Street. The leftmost cross represents the Northwest sensor location and the rightmost cross represents the Southeast sensor location. . . . .	49
5.1	LDAT Home page which greets users when they first open up the application. . . . .	54

5.2	Data Explorer Form which allows users to fill out and specify fields pertaining to the data visualizations. . . . .	55
5.3	Data Visualization Dashboard showcasing the three different data visualizations available. . . . .	56
5.4	Settings page which allows for changes to be made to the 3D point cloud.	56
5.5	About page that details the purpose of the website and provides a general guide on how to use the application. . . . .	57
5.6	Bar chart that displays the current count of each object being detected in the point cloud. The objects that can be detected are vehicles and pedestrians. . . . .	58
5.7	Line graph that presents the total number of objects detected within the point cloud. . . . .	59
5.8	Screenshot of near real time 3D point cloud mesh render which displays the view from the sensor located on the Southeast corner. . . . .	60
5.9	UI layout for the Appearance section found on the Settings page. . .	61
5.10	Color picker UI window. . . . .	61
5.11	Screenshot of 3D point cloud mesh render scene with object colors changed and a point cloud size of 0.1. The sensor view shown is the Northwest corner of the intersection. . . . .	62
6.1	Screenshot of Mousotron Interface. . . . .	65
7.1	Bar chart displaying the range of ages of the participants in the study.	71
7.2	Bar chart showing participant response to their familiarity with LiDAR sensor technology. . . . .	71
7.3	Bar chart that displays how often participants use data visualization.	72
7.4	ANOVA chart for Task Completion Time. Shows the mean average completion time in seconds and indicates the min and max values for each task. . . . .	74
7.5	Analysis chart for number of left mouse clicks per task. . . . .	75
7.6	Analysis chart for number of right mouse clicks per task. . . . .	77
7.7	Analysis chart for number of mouse wheel scrolls per task. . . . .	78
7.8	Results from Question 1 of the post-questionnaire. . . . .	79
7.9	Results from Question 2 of the post-questionnaire. . . . .	80
7.10	Results from Question 3 of the post-questionnaire. . . . .	81
7.11	Results from Question 4 of the post-questionnaire. . . . .	81

# Chapter 1

## Introduction

Sensor technology has been around since the mid 20th century and continues to advance on many fronts. One such piece of sensor technology called LiDAR has become a highly effective tool that can be used to create solutions for a diverse range of problems. The term LiDAR stands for Light Detection and Ranging meaning that it uses a light source to detect both the distance and position of various objects within a real-world environment [16]. More specifically, the light sources used are lasers that are emitted from the sensor which allows for accurate and reliable positional tracking of an object or objects as well as the distance between the objects and the sensor. The positional data gathered from LiDAR sensors are referred to as point clouds. A point cloud data set is a group of points that are a 3D representation of a specific object or location. These complex data sets often can be difficult to analyze in their raw format and therefore need to be cleaned and processed to procure useful information.

Another field that has seen many advances is data analysis, which has grown exponentially in recent years due to the rise of big data. Big data refers to the process of extracting and analyzing useful information from large and often complicated data sets. Data analysis is a useful component of big data as it simplifies the data in an attempt to discover valuable and insightful information. A large part of data analysis is that of data visualization, which aids in communicating information found in the data in a readable and comprehensible manner. As the name states, this is accomplished through a visual means of representing the data. The most common types of visualizations are graphs and charts which are extremely useful in expressing

the patterns and relationships found in data points [18]. There are many different types of visualizations that can be gleaned from LiDAR point cloud data sets on account of the large amount of information accumulated within the point cloud.

The problem being addressed in this thesis is the processing and handling of the LiDAR point cloud data and being able to turn the data into a readable format. The motivation behind this work was based upon existing research being conducted at the University of Nevada, Reno (UNR) in collaboration with the City of Reno. Specifically, this work was developed as a means to track street traffic information to be able to assess traffic patterns at different points in the day. The setup placed two sensors on opposite ends of the intersection of North Virginia Street and 15th street, which is directly adjacent to the UNR campus. These sensors collect the point cloud data which is then stored in a cloud server that is connected to a network protocol called MQTT, that of which will be covered in much more detail in Chapter 2. The point cloud data stream was then outputted on a simple Flask app where the user could visualize the 3D render of the scene. The main issue with this setup is that there is no effective means to access the data to gain insights into the street traffic information other than viewing the near real time data stream.

In order to solve this issue, a more effectual implementation is needed to access the data and extract insightful information from the point clouds. This thesis presents a solution to address this problem through the development of a web application called the LiDAR Data Analysis and Visualization Tool (LDAT). This tool is built upon the Angular and Flask web frameworks to create a robust application that is able to process, clean, compress, and output the near real time point cloud data steam. The output is created through a 3D render of the mesh scene that corresponds with a particular LiDAR sensor. In addition, LDAT utilizes other libraries to create data visualizations that allow for quick understanding of the data being presented. Both the data visualization graphs and charts as well as the 3D render are presented within a single web page as a data dashboard which allows for the data to be easily accessible and readable. The data is updated by the data stream but can also be changed or

customized through a separate form page that allows for selection of available sensors and visualization preferences. To reduce the load on the application, only data from one sensor location can be viewed at any given time. LDAT is created to be a proof of concept and therefore is designed with scalability, portability, and efficiency in mind.

The rest of this thesis is structured as follows: background information on the vital technologies, data, and tools are presented in Chapter 2. This includes a brief overview on LiDAR sensors, point cloud data, and visualization methods as well as the software development tools utilized including Angular, Flask, MQTT, and other additional software technologies used. In Chapter 3, related work in the fields of LiDAR technology and data analysis with an emphasis on analyzing street traffic data are covered. Chapter 4 introduces and describes the architecture design of the application, which includes the requirements and system design, and the overall implementation of both the hardware and software components. Chapter 5 then exhibits the final prototype implemented for this thesis through a series of screenshots that detail the interface and design of LDAT. Chapter 6 describes the user study conducted and details the method, apparatus, tasks, procedure, and the overall design of the study as a whole. This is followed by Chapter 7, which presents the data gathered from the tasks and questionnaires, analysis conducted, and a discussion on the results obtained from the study. Lastly, Chapter 8 wraps up this thesis with concluding thoughts on the application as a whole and details possible future work.

# Chapter 2

## Background

### 2.1 Introduction

Data analytics and visualization are both very important in many modern data solutions. Big data has grown exponentially in recent years as data continues to increase in amount and complexity. One such complex data is point cloud data that is obtained through LiDAR sensors. The application developed and presented in this thesis is a web application called LDAT that of which performs data analysis through data visualization of LiDAR point cloud data collected from LiDAR sensors. Through this chapter, we hope to introduce you to the technologies, data, and tools that this solution utilizes as well as the problem that is being addressed. For a more detailed breakdown, Section 2.2 presents a history and current status of LiDAR technology alongside background of point cloud data as well data visualization techniques. Section 2.3 details the main data source used for this project and Section 2.4 provides a quick overview of the main tools used to develop this solution.

### 2.2 Technology

#### 2.2.1 LiDAR

LiDAR, aka Light Detection and Ranging sensors, is a form of sensor technology that has been around for a long time and is still being utilized in many different applications today. It derives from radar technology as both have similar operating



principles with the exception that LiDAR technology uses lasers instead of radio waves to transmit signals which provides a higher accuracy and sampling rate as well as good low-altitude detection [56]. LiDAR technology itself is used primarily to quickly measure the distance between objects or in other words detect the position of objects in a real space. This is accomplished through the use of sensors that contain many lasers embedded within them. The laser beams are emitted from the mounted sensor across a scene, such as a street or room, and the sensor is able to quickly provide accurate position data over large areas and distances [16]. LiDAR is useful due to its ability to collect large amounts of data points over a large location at a very quick rate. Additional information on LiDAR technology and sensors from Matthew Lato, one of the authors of [16], that may not be covered fully in the following sections can be found in his thesis [24].

#### **2.2.1.1 History**

The history of LiDAR technology dates back to the mid-to-late 1960's and has continued to evolve into the modern day. LiDAR, originally popularly known as laser radar, started in the 1960's following the invention of the ruby laser by Theodore Maiman at the Hughes Research Laboratory [56]. Laser radar referred specifically to any systems that used high powered lasers for range detection and become more advantageous to use over other methods of measurement due to its high precision and ease of use. NASA was one of the first corporations to work on early laser radar technology through the development of the argon ion wind velocity laser radar. This piece of laser radar was being developed and used specifically to detect wind velocity in wind tunnels [32].

After only a few years, in-house development on a high powered ruby laser was underway in the late 1960's at the U.S. Air Force in their Avionics lab. The main purpose of the high powered laser was that it could be used for laser ranging and distance measuring. Following the development of the in-house Ruby laser, it was used for ranging on the moon in 1969. The very first laser ranging retro reflector was

placed on the moon by Apollo 11 astronauts which then aimed laser pulses directly at the Earth. This feat allowed scientists to be able to determine an accurate round trip travel time between the moon and the Earth [32]. Figure 2.1 shows a corner cube that was placed on the moon by Apollo 11 astronauts in 1969 for ranging and distance measuring.

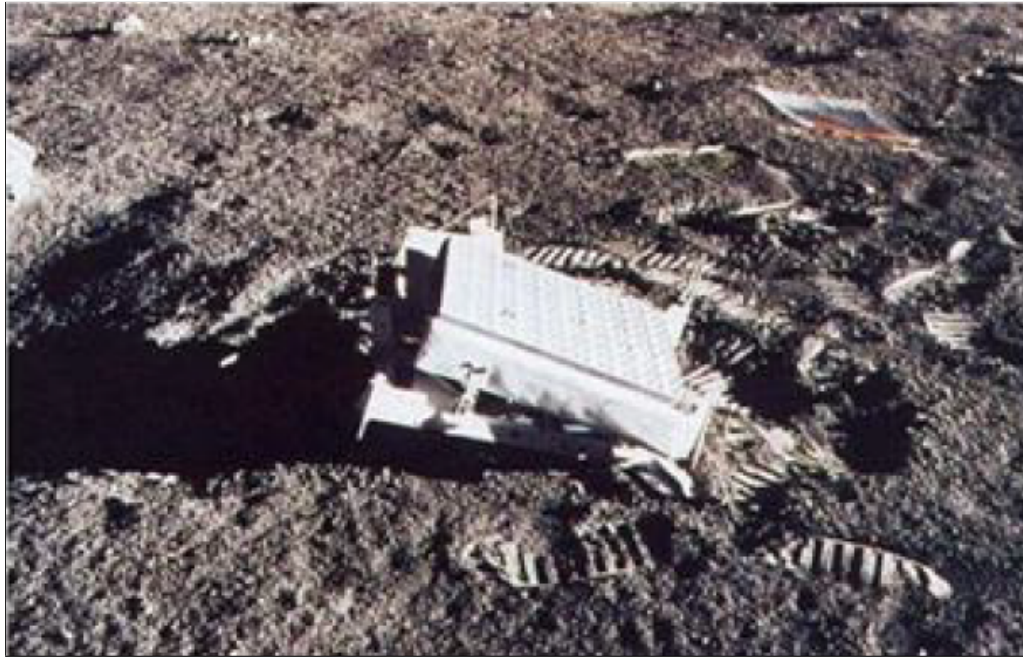


Figure 2.1: Corner cube reflector placed on the moon in 1969 that contained a high powered ruby laser to allow for ranging purposes [32].

Around the same time as the development of the high-powered ruby laser in the late 1960's, the military began using laser radar in the implementation of the Paveway Laser Guided Bomb. The Paveway bomb was a type of weapon that changed tactical warfare drastically as it was the first laser guided bomb ever developed. This meant that due to the use of a reflective laser, these Paveway bombs had increased accuracy and reliability of anything that had come before and therefore also reduced the number of munitions needed to be discharged [32]. The military continued to research laser radar to develop tools that could be utilized in the field. The laser ranging system AN-GVS3 was a first generation rangefinder released in 1971 by the U.S. military [56]. Similar to previous laser radar technology it utilized a ruby laser to

emit laser pulses and had a photomultiplier detector which functioned well, however, it did have the disadvantage of being heavy and had a high power consumption. It was not long before the AN-GVS3 was replaced by the second generation rangefinder system which used a near-infrared neodymium laser otherwise known as the Nd:YAG laser.

By 1977, the first handheld rangefinder using the Nd:YAG laser was successfully developed in the United States. This handheld rangefinder was the size of standard military grade binoculars and had a weight of only 2kg which allowed for it to be easily carried [56]. Figure 2.2 displays the handheld binoculars known as the AN/GVS5 rangefinder being used in the field. Continued development of laser ranging was focused mostly in scientific and military research. This led to the discovery that there were some shortcomings with the second generation laser ranging systems including low accuracy under some weather conditions, poor compatibility, as well as potential damage to eyes after prolonged use [56] which led to the development of a third generation laser ranging system. This unit was lighter in weight, lower in power consumption, higher in accuracy, and safe for eyes.

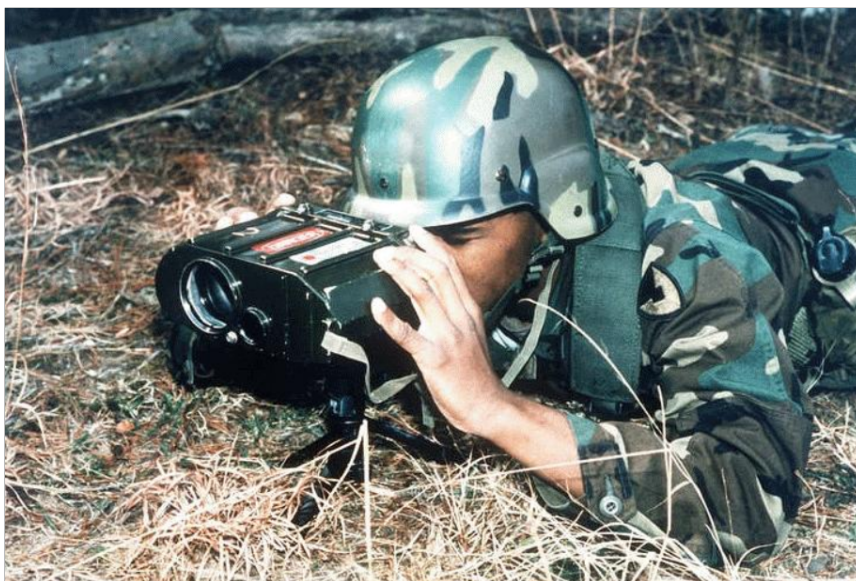


Figure 2.2: An image showing a military soldier using the AN/GVS5 Binoculars [19] in the field.

Jumping forward to the 1980's, early development of 3D laser radar image scanning technology was being conducted. Figure 2.3 highlights some test images collected in 1983 from the ground on the top row of images and long range images on the bottom row. The images were gathered using a short pulsed CO2 heterodyne 3D imaging laser which output the images in a 10-bit gray scale representation [32]. The portions of the images highlighted in white are detected values in the range gate (area scanned or detected by the imaging laser) while everything else in black has no detected value and is therefore out of range of the sensor.

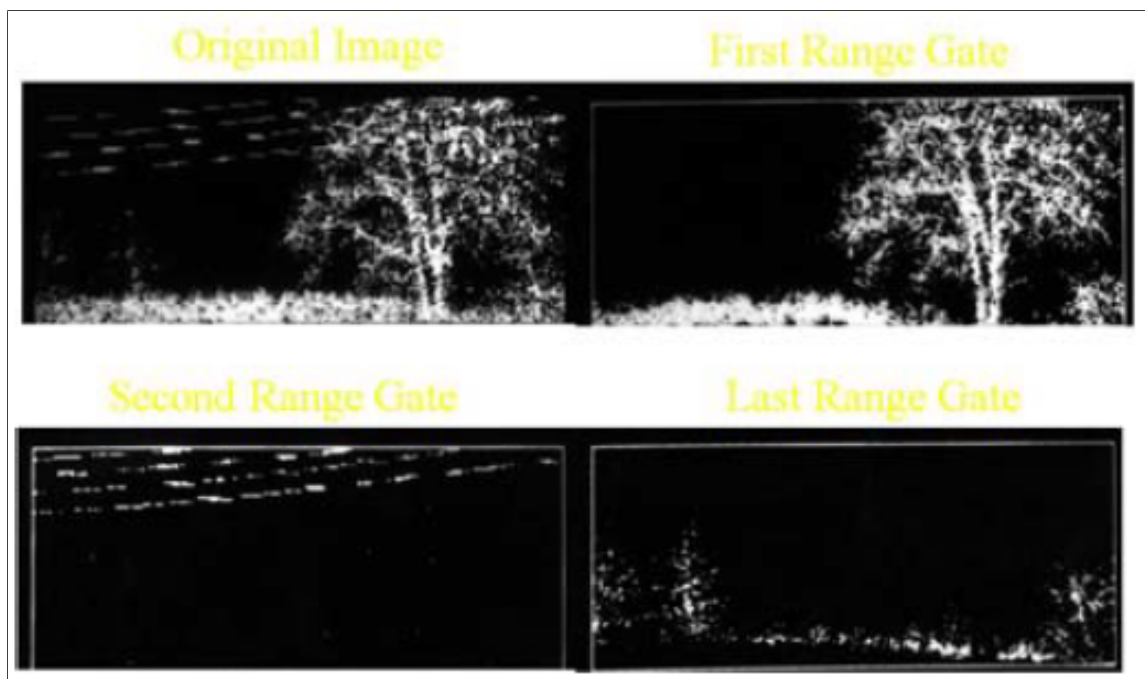


Figure 2.3: Displays early 3D laser radar scans [32] from 1983.

In the 1990s LiDAR technology began to progress due to the improvements made to the optical system and signal processing. It was in 1996 that Bushnell introduced the 400 LD laser rangefinder Yaddaga400 which had a range sensor that could detect up to 400m [56]. This was followed by the creation of the LD laser rangefinder camera type developed by tasco in 1998 which increased the sensor range up to 800m. Airborne laser scanning, which had originally been conceptualized in the 1970s, was also being developed and improved upon during the 1990s [57]. This type of laser

scanning was employed primarily for complex 3D terrain mapping which could then be used for mapping roads, railway tracks, electrical transmission lines, and to obtain accurate measurements of specific geographical areas.

LiDAR sensors has had some rapid development over the last couple of decades. These developments have allowed LiDAR technology to grow past theoretical research and into a dedicated market [56]. LiDAR hardware and data products have grown exponentially with an annual rate of over thirty percent which has allowed the hardware to come down in cost and make it a more viable and accessible solution for many different products. The growing market has led to the emergence of new LiDAR products from companies all over the world such as Velodyne, Trimble, Riegl, Sick, and Quanenergy to name a few. LiDAR technology continues to grow and improve today which has led to many exciting advancements.

#### **2.2.1.2 Current Use Cases**

The use cases for LiDAR sensors continues to expand in the modern era and is being used to solve a host of real-world issues. These issues can range from a variety of areas including but not limited to airborne LiDAR, 3D mapping, robotics, autonomous vehicles, and traffic information. As can be seen from its long history, LiDAR sensors can be utilized in different forms to collect data with the most popular methods being airborne or terrestrial vehicles and other offshore platforms [16]. Each of these forms is used to collect data for different purposes. For instance, airborne LiDAR is used mainly to collect complex models of terrain for land management and mapping while fixed LiDAR is used more in infrastructure mapping to create detailed sites for building or construction as well as city planning.

The first use case that will be discussed in detail is that of airborne LiDAR. While airborne LiDAR has been around for awhile, it is still one of the most common use cases for LiDAR sensors today. The last couple of decades have seen airborne LiDAR become a swift and extremely accurate tool for mapping various types of terrain to create three dimensional representations [28]. Airborne LiDAR works by

having a Global Positioning (GPS) unit on the airborne vehicle linked to satellites to track its position while a LiDAR sensor scans the terrain below the vehicle as can be seen in Figure 2.4. The red dots shown in the diagram represent the LiDAR data points obtained from the airborne scan. In addition to the LiDAR sensor, it is often practiced to use a digital camera to obtain the color information for the terrain.

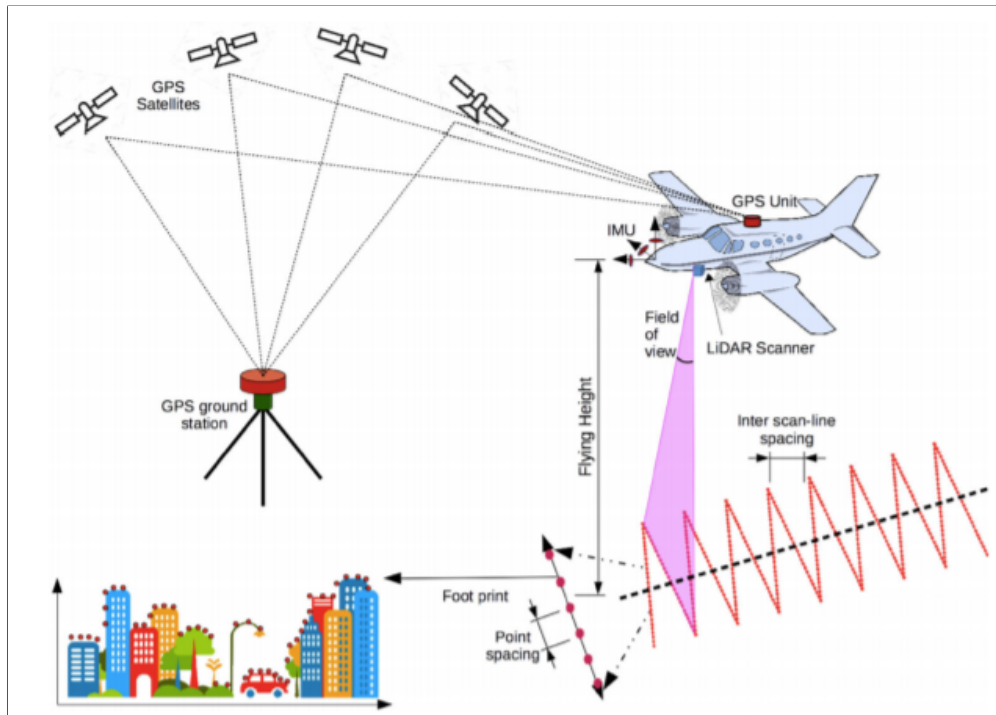


Figure 2.4: Diagram that illustrates an airborne LiDAR data capture with the red dots showing where the LiDAR data points are located [28].

An important aspect of airborne LiDAR data capture is flight planning. Typically, an airborne vehicle covers the area of interest (AOI) in parallel strips [28] which benefits from a flight plan to reduce flight hours, reduce cost, and ensure proper data acquisition. The data obtained from an airborne LiDAR scan is usually a representation of a specified physical environment. Figure 2.5 shows an image of three dimensional LiDAR point cloud data with colors used to represent the different heights of objects. A comprehensive explanation of point cloud data is provided in the following section. Airborne LiDAR is best used for applications that rely on large scale terrain mapping such as flood modelling, electromagnetic wave modelling, and

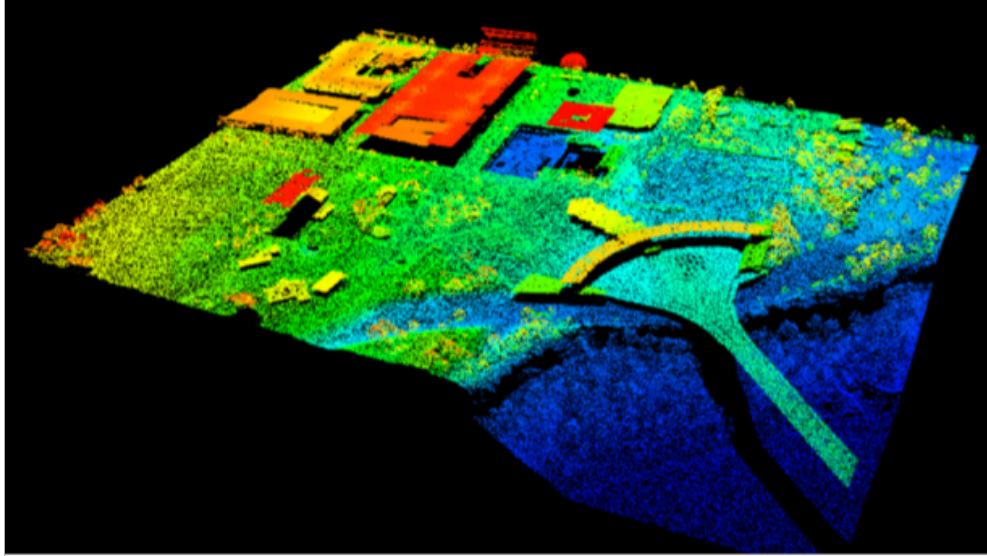


Figure 2.5: Example image of LiDAR point cloud data [28] represented by a 3D model where the heights of the objects in the digital environment are indicated by the color displayed.

topographic modelling. In addition, systems can be developed [13] to further improve airborne LiDAR by filling any holes within a visualization thereby creating a more robust rendering of the data.

A second primary use case of LiDAR sensors is in the field of robotics. One way LiDAR has been used in robotics research is to solve problems in human-robot interactions. In a research study conducted in Spain [46], the researchers presented a data set that was focused on robots properly detecting people within an indoor environment. This assessment of data helped to determine how the robot should act in certain situations to improve interactions with people while performing various tasks. Furthermore, LiDAR has also been used in research to develop autonomous robots which can perform difficult or dangerous tasks. One recent study [37], explored the use of Simultaneous Localization and Mapping (SLAM) in mobile robots. The use of SLAM was shown to be able to execute mapping in an unknown environment and improve navigation of the robot to avoid various obstacles.

Another method in which LiDAR is currently being utilized is in the development of autonomous vehicles. A recent experiment [33] was conducted in which airborne

LiDAR was used in an unmanned aerial vehicle (UAV) to improve autonomy. An algorithm named as ALORID was designed using scan data from the LiDAR sensor and created specifically for obstacle recognition and intrusion detection. The algorithm proved effective in the experiments conducted but there is still more extensive research needed to determine its viability. On the other hand, autonomous driving cars are much farther along in research and development thus much closer to being fully realized.

Autonomous driving cars is an advanced technology that is likely paving the way towards the future of driving. This technology needs to be highly reliable to protect not only the driver and passengers of the vehicle but also pedestrians and other drivers on the road. Research on autonomous driving conducted in [12] shows that both LiDAR and Radar systems are needed in autonomous driving cars to be able to observe obstacles in the surrounding environment. As can be seen in Figure 2.6, there are two main LiDAR approaches used for environment detection in autonomous vehicles those of which are Flash-LiDAR and MEMS-based scanning LiDAR. Each of these approaches have different strengths and therefore are used for different purposes. MEMS-based scanning LiDAR is primarily used for distance and velocity measurements of objects outside the vehicle while Flash-LiDAR is used to detect nearby objects on the sides of the vehicle due to a more limited range. Research on fully autonomous driving vehicles has been conducted [26] but they are still

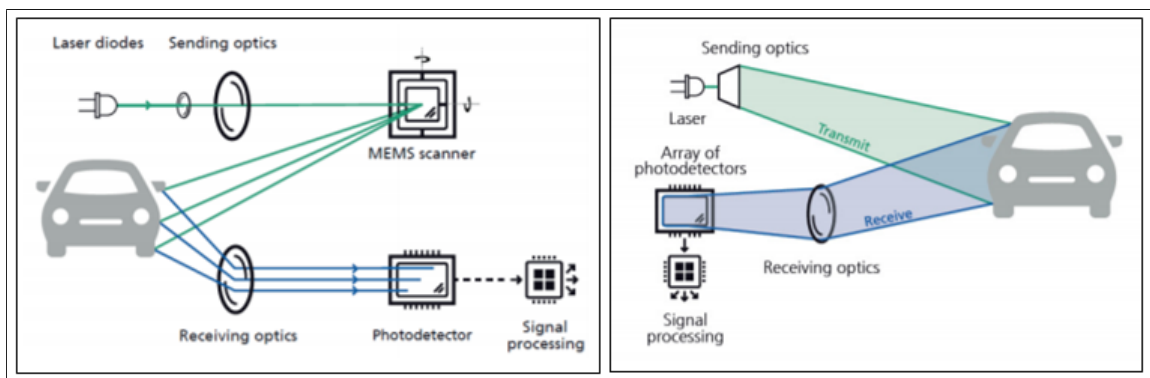


Figure 2.6: Diagrams that show the MEMS-based scanning LiDAR system (left) and Flash-LiDAR system (right) [12].



not currently available; however, many modern vehicles today utilize advanced driver assistance systems (ADAS) such as adaptive cruise control, forward collision warning, and lane departure warning [6]. In addition, Tesla uses their own internally developed autopilot system [51] in their vehicles which continues to push the technology further.

The last use case discussed in this section in which LiDAR is being employed is to collect traffic information. More specifically, LiDAR can be utilized for traffic surveillance tasks such as vehicle speed, trajectory, counting, and tracking. One such experiment conducted in [49] analyzed a data set obtained from a monocular camera and LiDAR sensors to evaluate vehicle speed measurements. The LiDAR sensors used in this study linked to reference GPS tracks which allowed for a more precise measurement of the vehicle speed. This study looks to push this research towards the development of a fully automatic system to mine traffic surveillance data. This could be further improved through the use of point clouds and data visualization techniques to allow for a better understanding of the surveillance data.

## **2.2.2 Point Clouds**

### **2.2.2.1 Introduction**

A point cloud is a type of data structure that plots collections of points across a multi-dimensional space and it is most frequently used as a representation of three dimensional data [41] such as an object or a specific geographical environment. Point clouds are most commonly produced by LiDAR sensors, structure from motion (SFM) methods, as well as 3D scanners and 3D sensors such as the Microsoft Kinect and Asus Xtion [27]. SFM methods tend to generate point clouds with a low and sparse point density whereas LiDAR and 3D sensors can generate point clouds that contain a much higher number of points. A LiDAR sensor can generate a point cloud that consists of approximately up to 2.2 million points per second. Point clouds are most commonly represented by a 3D model therefore each point in the point cloud has its own X, Y, Z coordinate indicated as a vector to provide a unique position within the model. The vector is also used to indicate different attributes as well such as

reflection, colors, and normals. These attributes will be discussed in more detail later in this section.

While LiDAR sensors can be used to generate point clouds, they depend upon multiple sets of measurements to generate a precise point cloud. First, the position and location of the sensor must be known for each scan [16]. This is necessary as the points could be unevenly distributed across the target if the exact location is unknown. The second piece of information needed is the exact distance which can be measured using one of two methods being either time-of-flight or phase-based. Time-of-flight achieves this by sending out a laser pulse and waiting for the return pulse. The return pulse also contains a time measurement which allows for an accurate measurement of the distance, however, this method is limited to due its need for a return signal. The second method, phase-based, uses an amplitude modulated continuous waveform laser (AMCW) that resets after interacting with a target thereby initiating a new phase. The return signal that the sensor receives obtains the distance and time span it took to interact with a target. Phase-based is the faster and more precise method between the two but it is limited in its range.

#### **2.2.2.2 Point Cloud Library (PCL)**

Point cloud data is complex and can be difficult to process. This is due to data density as LiDAR scanners can accumulate a large amount of data points per second as stated previously. A commonly used tool for 2D/3D image and point cloud processing is the Point Cloud Library, which is also referred to as the PCL. The PCL is a large scale standalone open source project released under the BSD license meaning that it is free to use for commercial and research projects [41]. The framework for PCL contains a variety of algorithms that are used for 3D processing on point cloud data [47] and cover processes such as data filtering, feature estimation, reconstruction, registration, model fitting and segmentation. These algorithms allow for PCL to be utilized in many key ways including filtering outliers that can be found in noisy data, stitching together separate point clouds, segment parts of a scene, and assess descriptors to be able to

recognize specific objects in the world based on the geometric appearance of the object.

Another key aspect of the PCL is that it is a cross platform tool that can be installed on used on Linux, Windows, Android, and MacOS. The PCL is also extremely adaptable to the users needs as it can be split into a series of smaller libraries that can be compiled individually [41] allowing for simplified development and increased modularity. Figure 2.7 exhibits a graph with PCL split into a many different code libraries. This is important as the increased modularity provides users a customized experience to run only what is required and decrease the load on their machine. Lastly, PCL includes its own visualization library as can be seen in Figure 2.7. This library is based on VTK which provides support for multiple platforms in rendering 3D point cloud data [47].

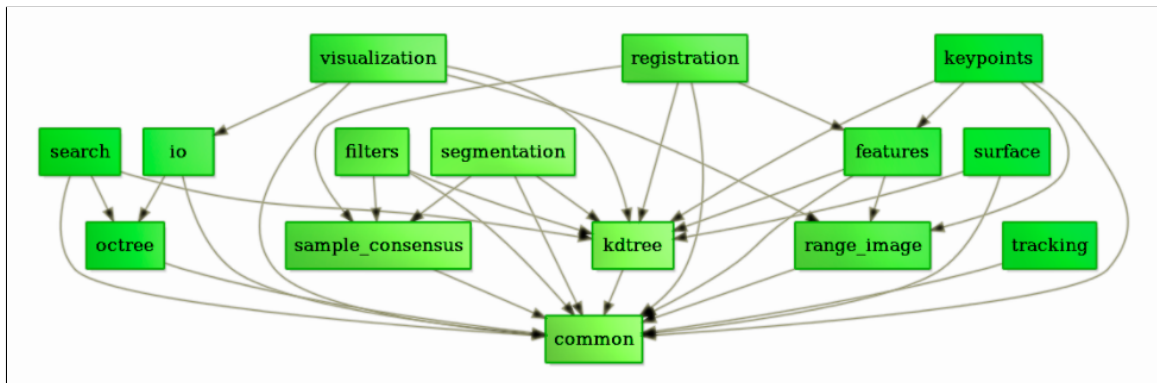


Figure 2.7: Graph that exemplifies PCL split into different code libraries [41].

### 2.2.2.3 Point Cloud Data (.pcd) Files

The Point Cloud Data (.pcd) file format is a data file that supports 3D point data [42]. While it is not the only file format to support 3D data, it has been configured to work well with PCL as of version 0.7 (PCD\_0.7). The files each include a header which covers specific properties and is encoded in ASCII values. These properties include file version, fields (name of each dimension/field), size (specifies number of bytes of each field), type (specifies type of each field i.e. char, int, float), count (number

of elements in a field), viewpoint (sensor orientation), points, and data (data type specified).

### 2.2.3 Data Visualization

Data visualization is a necessary and important component of data analysis that helps to convey the information collected in an easy to interpret manner. Data visualization is a visual representation of data most commonly in a graph or chart [18]. The use of visual images through graphs and charts convey the relationship of data points in a readable format which makes patterns and trends easily discoverable. Data visualization and analysis have become increasingly important as large data sets have become commonplace with the advent of big data.

Visualizations are essential because it allows the reader to obtain quick insight into important findings presented in the data. While visualizations are useful, they can also be ineffective if used wrongfully. In her book entitled *Effective Data Visualization: The Right Data for the Right Chart* [10], author Stephanie Evergreen is able to effectively present what to look for when creating data visualizations and how visualizations can be harmful or misleading when implemented incorrectly. It is stated that it is important to first consider what the point of the data is before creating a visualization as this gives the data a clear cut purpose. Once a purpose has been established then it is much easier to choose the best visualization method. It is also noted that some charts can be deceptive which occurs when a visual is distorted or purposefully missing data.

LiDAR point cloud data can be visualized in many different ways due to the large amount of information that is collected in the point cloud. As seen previously, it can be represented visually as a 3D model of a terrain or object with colors used to distinguish different variables. These visualizations are very effective but a simpler graph or chart can possibly yield greater insight depending on what the purpose of the data is and what specific information the reader is seeking. In the case of this project, the data is streamed in near real-time allowing the reader to see and create

visualizations based on what is happening on the street close to when it occurs.

### **2.2.3.1 Use Cases**

In a previous work [7], a recommender system called a Containerized Amazon Recommender System (CARS) was developed to process and analyze Amazon review data and provide users with accurate recommendations and insightful statistics. The statistics which covered items over time, summary statistics, and prediction performance were displayed through the use of data visualizations. While this application dealt with large and complex data, the visualizations were all variations of common data visualization types with the purpose of making the insights clear to the reader.

There are many different situations in which data visualizations prove to be useful. Some of the most common ways data visualization is used involve changes over time, determining frequency or recurrence over time, determining correlations between points, examining a network, scheduling using a gantt chart, and analyzing values which can be effectively done through color coding [18]. Some of these use cases are strongly related to time while others focus on the relationship between certain variables. These can all be accomplished using various types of data visualizations and sometimes can even be fulfilled using a combination of types. There is no one type of visualization that must be used for any specific use case but there are some that work better for certain scenarios.

### **2.2.3.2 Visualization Types and Challenges**

There are many different types of data visualizations that can be used in different situations. Some of the most basic graphs and charts that are repeatedly used include line charts, bar charts, histograms, scatter plots, pie charts, and heat maps [18]. This only briefly touches upon all the visualization types as there are many that cover a wide range of use cases. For instance, a line chart is useful for displaying changes over time as the x or y axis could be used to represent a period of time to correlation with a specific value such as quantity or sales.

The various types of data visualizations are detailed extensively by author Cole Nussbaumer Knaflic in her book entitled *Storytelling With Data: A Data Visualization Guide for Business Professional* [22]. As seen in Figure 2.8, Knaflic highlights an assortment of data visualization types and explains the best practices for each. One of the most common types of visualizations highlighted is the scatter plot, which is explained to be used to exhibit a relationship between two things that are clearly indicated by the x and y axis. Unlike the line chart, scatter plots often work best when the data in question is not measured in time. Another frequently used visualization type is the bar chart. This type of visualization is most effective when displaying categorical data. Figure 2.8 features variations of the bar chart that can be used to make the data more presentable in certain cases. For instance, the horizontal bar chart can be used over the vertical bar chart when the text is a bit long. This makes the text more readable and provides a better overall presentation.

There are many challenges and issues that can emerge during the creation of



Figure 2.8: Image from [22] that highlights the various types of data visualization methods.

data visualizations. As stated previously, an issue with data visualization is that they can be harmful and misleading in some cases due to misrepresentation of data [10]; however, there are some additional challenges that can effect the readability and basic understanding of a visualization. One of the most prevalent challenges is clutter as this creates a messy and difficult to read visualization. This most often occurs when there is no clear main point to the data being presented creating an unrefined graph or chart. Another similar challenge is a lack of structure and this often happens when the data is not organized in a logical manner. A good example of organizing data effectively is a bar graph that has categorical data presented from the highest to lowest number to display a clear range and purpose.

## **2.3 Data**

Data is a collection of information that can be used for analysis to provide clear and effective insight that may otherwise be missed or overlooked. Raw data files can be very difficult to read and oftentimes do not glean any valuable insight and, therefore, need to be processed and cleaned to be effective. This project gathers complex point cloud data directly from LiDAR sensors to analyze and visualize street traffic information. This section goes over the collection of the data and how the data is sent over the network. The different file types will also be discussed as well the data cleaning process. Additionally, the locations of the sensors will be briefly reviewed.

### **2.3.1 Data Collection**

Data collection and ingestion is a key factor when assessing what specific information is being analyzed. There are four main types of data collection techniques that can be used depending on the type of information that is being explored. These techniques include observation, questionnaires, interviews, and group sessions [17]. In this project, the technique of observation is used to gather data. This is accomplished through the use of LiDAR sensors which are placed on street lights and pointed down

towards the street view. The point cloud data collected from the sensors can be seen visualized in Figure 2.9. The colors in the image highlight the intensity of the data points which specifies the strength of the return laser pulse.

Once the data is obtained from the sensor, it is stored as a point cloud data file (.pcd) and sent over the network. As discussed previously, the .pcd file format is a file type that supports 3D point cloud data and each file contains a header that defines properties of the data that are stored in the file [42]. These files contain properties such as the time, topic, intensity, object position, height, and viewpoint. An important characteristic to note is that the time for each file is based on the number of minutes since the top of the hour and the topic relates specifically to the location of an individual sensor in which the data is being obtained. As the files are gathered, they are sent over the network by the MQTT broker network protocol where they are parsed and cleaned through a back end data server before being sent to the main web server to be used for analysis and visualization. The MQTT broker and other various development tools will be outlined further in Section 2.4 while the implementation of the system will be explained in Chapter 4.

### **2.3.2 Locations**

At the time of the development of this thesis, there are two LiDAR sensors that are active and operational. These sensors are located at an intersection adjacent to the University of Nevada, Reno (North Virginia Street and 15th Street). The data obtained from these sensors is the main data source used to test the visualization functionality of this tool. There are more sensors that are planned to be setup and operational in the near future which will then be accessible on the web page to connect and subscribe through the MQTT broker. Seeing as this thesis is a proof of concept, the limited number of locations is to test the functionality and build out a practical web page as the network of sensors expands.



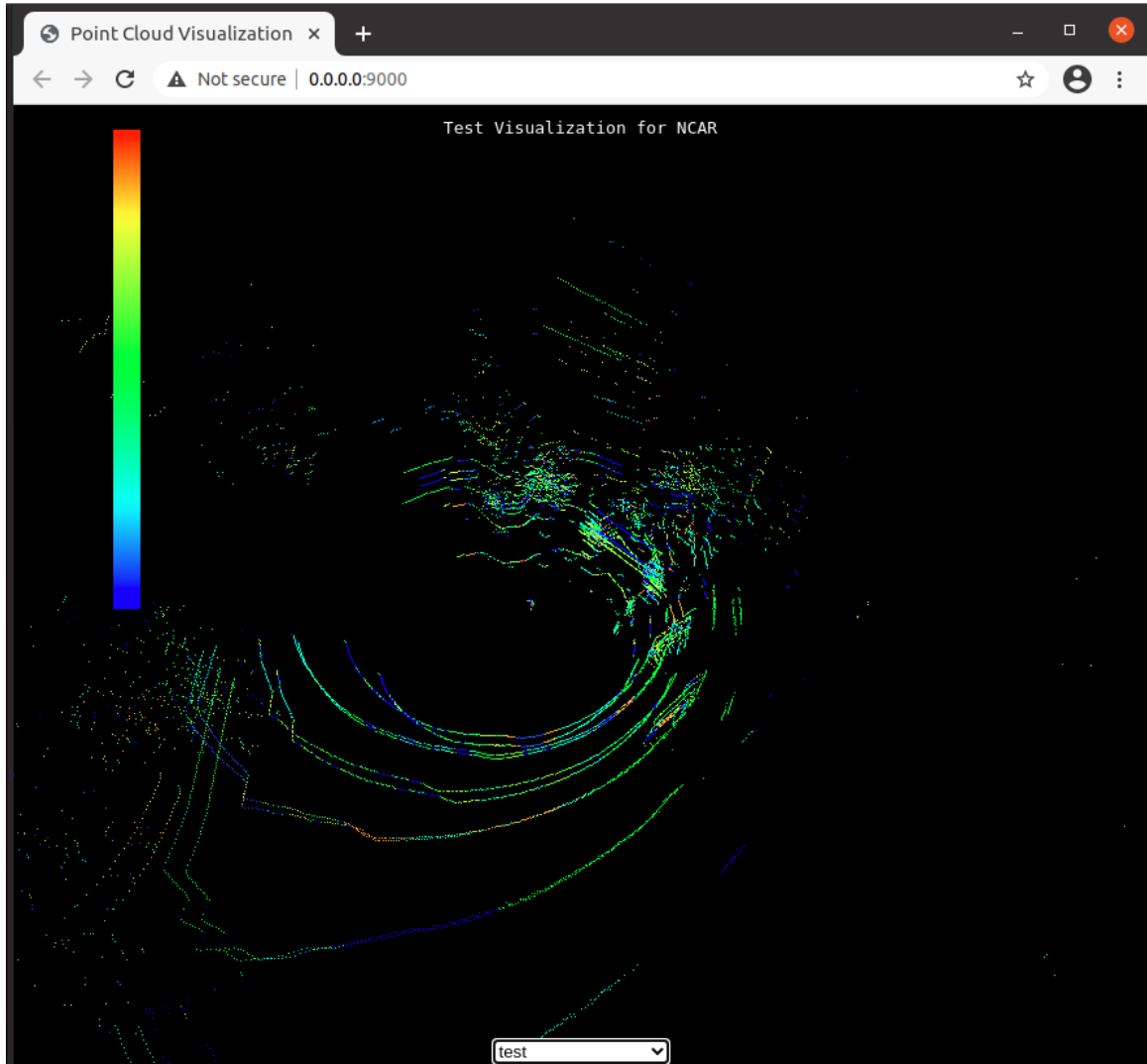


Figure 2.9: Screenshot of the initial test environment running in a browser showcasing the WebGL 3D visualization of the LiDAR point cloud data.

## 2.4 Tools

The development and implementation of this thesis required the right software tools to create a clean and interactive experience for the end user. In this last section, a brief overview will be given of the tools that were used to build the data visualization web page. The first set of tools that will be discussed are web frameworks that were used to build up the web page. This is then followed by the network protocols that were used to obtain the data and send to the back end server. The next tool that will be covered is Docker which was used to make the development process more efficient. The programming languages and libraries used to develop the application will be presented in conclusion.

### 2.4.1 Angular

Angular is a web development platform that is used to build web and mobile applications [2]. The platform is built on the programming language Typescript, which will be detailed more in Section 2.4.6. The platform is capable of building out scalable applications and has wide range of libraries that include features such as routing, forms, user interface, and client-server communication to name a few. Additionally, there are also many development tools that make building, testing, and updating code quick and efficient. Through these libraries and tools, Angular provides the necessary components to build and create simple applications that can be simply updated and expanded. Development is also made easy with Angular through the Angular Command Line Interface (CLI). The CLI allows a developer to quickly build, run, test, and generate files for a project. For this application, Angular was used to build the main web server and front end web application to display and visualize the data to the end user.

## 2.4.2 Flask

Flask is a microframework that is used primarily to develop web applications [11]. This development tool is based in the python programming language meaning that applications built using Flask are written using at least some python code. Being a microframework, Flask is a very simple framework that delivers a core foundation for web development while also providing an extensible environment that is easy to update and build upon. Flask is highly customizable as it includes quite a few extensions and libraries that can be incorporated into an application which can provide features such as database integration, upload handling, network calls, and data transfer among others. Customization is highly important as it allows Flask to be easy to use with other tools and frameworks such as Angular to develop robust and effective solutions with simplicity and efficiency.

## 2.4.3 MQTT Networking Protocol

Networking is a key component when transferring data among servers. Mosquitto, also known as MQTT, is a network messaging protocol for Internet of Things (IoT) messaging [36]. The design of the protocol is a publish and subscribe system that is ideal for connecting remote devices that require a small number of resources. It allows for a device to connect to a cloud and cloud to connect to device making it simple to send messages to groups of different devices. Figure 2.10 shows the publish and subscribe architecture for MQTT which highlights the connection between clients and

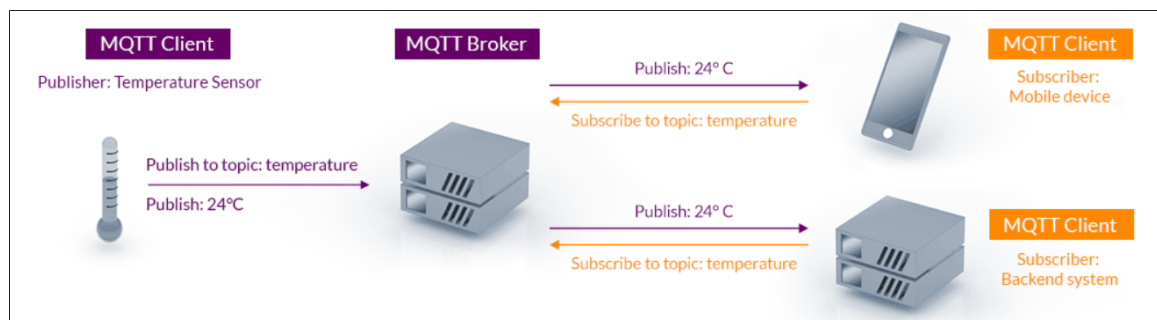


Figure 2.10: The MQTT publish and subscribe architecture [36].

brokers and how messages are exchanged between the two. MQTT is very secure as it can easily encrypt messages through TLS as well as authenticate clients through the authentication protocol OAuth. Additionally, MQTT can be used to connect over a network through the use of websockets. This application utilizes the websockets through the socket.io library which enables real-time communication between clients and servers and makes real-time data analytics possible [3].

#### **2.4.4 Docker**

Docker is a software development tool that enables the packaging of software applications into a container or otherwise referred to as a containerized environment [9]. A container is a unit of packaged software that contains the application code and all its dependencies. It acts as a wrapper to nicely package a piece of software and, in conjunction, makes the software portable. A container is similar to a virtual machine in the sense that they both have resource isolation and allocation; however, the main difference is that containers virtualize the operating system and not the hardware, which also makes containers much smaller in size. Docker is a very useful tool to ensure that an application can run quickly and uniformly on different machines regardless of any differences between the computing environments. It is a tool that is used across multiple operating systems including Linux and Windows-based machines as well as MacOS through the use of the Docker Desktop application.

#### **2.4.5 Python**

Python is a highly efficient, interpretive, object-oriented programming language that can be used to build a wide variety of applications [44]. It allows for smooth integration of modules, exceptions, dynamic data types, and classes as well as supports procedural and functional programming. It works well with many different system calls and libraries that allow for features such as data analytics, data visualization, internet protocols, and string processing. Python is a very useful and effective language that has a clear and easy to read syntax making it a very commonly used language in

both research and industry. The application discussed in this thesis utilized python as part of the Flask backend service, which processed and cleaned the incoming data from the LiDAR sensors to prepare it for analysis and visualization.

### 2.4.6 JavaScript

JavaScript (also referred to as JS) is a compiled programming language that is very lightweight and is most commonly used in the development and scripting of web pages [35]. More specifically, it is a scripting language that supports different programming styles such as object oriented programming and is often used to design web pages and control their behavior. It is important to note that JavaScript is not the same as Java as both languages have different syntax and use cases. Additionally, JavaScript is also used in some form in many other environments and frameworks, notably Node.js, Adobe Acrobat, and Angular. In Angular, a version of JavaScript is used in TypeScript, which is an open-source language that builds upon JavaScript through the addition of static type definitions [2]. This project uses TypeScript as the base language for the front end Angular application.

### 2.4.7 WebGL & Three.js

WebGL is an application programming interface (API) that is used to create 3D graphics inside a web application [20]. It is a cross-platform API that is based on OpenGL and uses the OpenGL shading language known as GLSL. In addition, WebGL has full integration capabilities with Document Object Model (DOM) interfaces meaning that it can be used by any language that is DOM compatible such as JavaScript. WebGL is also compatible with libraries in JavaScript, notably the Three.js library. Three.js is a 3D library that draws 3D content to a web page using WebGL [52]. This library is extremely useful as it can greatly reduce the amount of code needed to create 3D content for the web page. It does this by directly handling scenes, lights, materials, and textures among others through built in functions which do not exist when using base WebGL.

# Chapter 3

## Related Work

This chapter covers related work inside the fields of LiDAR technology and data analysis. More specifically, the works introduced in this chapter relate to LiDAR technology being used as a means to analyze street traffic data through the use of data analysis and visualization methods. In addition, some data management and storage techniques will also be covered as well as different methods of visualizing data. Section 3.1 details some similar work covered in research that uses similar design principles and technologies to conduct data analysis from LiDAR sensor data. Section 3.2 covers two different data visualization tools that each have a focus in LiDAR data analysis but use different technology design patterns to achieve their visualization output.

### 3.1 Similar Works in Research

The works presented in this section cover research based applications or studies that share similarities to the data visualization tool built for this thesis. Each of these research based tools uses LiDAR point cloud data as the main data source to either conduct data analysis or visualize the 3D point clouds within a web application. The areas of research discussed in this section touch upon different focal points of an application designed around LiDAR point cloud data those being traffic detection, 3D rendering of point clouds, and data management.

### 3.1.1 LiDAR Sensor Traffic Detection System

The study presented in [34] is an examination of three low cost and scalable sensor technologies that could be mounted onto streetlights to create a street traffic detection system. The three sensor types used in this study included passive infrared motion detectors (PIR), thermographic cameras, and LiDAR sensors. Each of the sensors were fixed and secured to a pedestrian overpass located six meters directly above the road. The researchers conducted tests to assess each sensors ability to detect vehicles of varying speeds and detect count of vehicles. Following the experimental tests, it was found that out of the three sensor types both LiDAR and thermographic cameras had the best results with a rate of about 91% true positive detection. The authors concluded that the research presented indicates that sensors could be combined to create an inexpensive and reliable solution to traffic detection.

This work relates strongly to the LDAT due to the use of the LiDAR sensor to assess traffic data. In this study, the LiDAR sensor is mounted onto a streetlight to detect and monitor traffic data, specifically speed and count variables. This is very similar to the sensor setup for this application as the LiDAR sensors are mounted on streetlight poles that look over the street view. While the setup is very similar, the goals of each ultimately makes them different from one another. One difference between the two is that the study uses a combination of sensors to assess street traffic information whereas the LDAT strictly uses LiDAR sensors to obtain the data. Additionally, the study is entirely focused on the comparison between the different sensors and their effectiveness in tracking vehicles. This is another big difference as the LDAT also includes a data processing and visualization element that allows for improved readability and analysis of the data collected from the sensors.

### 3.1.2 3DSYSTEK: A Web Based 3D Point Cloud Viewer

The 3DSYSTEK viewer was designed by a group of researchers to be a web-based application that makes viewing of detailed 3D point cloud models easy and efficient [31]. It was made to address mobility and portability problems that are typically associ-

ated with remote field applications. The researchers used Terrestrial Laser Scanner (TLS) technologies, otherwise known as Terrestrial LiDAR, due to its ability to swiftly gather 3D point cloud data that included coordinate, color, and orientation information. In order to create and develop the 3DSYSTEK viewer, the research team used the open source tool WebGL in correlation with the Three.js library which allowed for the rendering of the 3D point clouds within the web application. The 3DSYSTEK viewer also used a particle system technology that made it possible and easy to handle an entire point cloud object. In addition, the Three.js library also permitted for the inclusion of features such as camera control which enables the user to control the movement and position of a perspective camera in the 3D scene. The final product allowed users to view many different types of point cloud objects that were stored in an accessible repository thereby making the web application highly portable in terms of usability.

This application bears notable similarities to the LDAT in that it is a web-based tool that is heavily focused on visualizing 3D point cloud data obtained from a LiDAR sensor. It also follows a similar design as it was developed with some of the same tools such as WebGL, Three.js, and the JavaScript programming language. This is not to say, however, that these web applications are completely alike. There are a few key differences that separate these applications and make them distinct from one another. The first difference being that the 3DSYTEK viewer brings in 3D point clouds of varying objects which cover museum exhibits, trees, and terrains whereas the LDAT is designed toward processing street view point clouds for specific analysis of street traffic. Another difference is the inclusion of a 3D point cloud repository that makes the point cloud files accessible directly from the 3DSYSTEK web page. LDAT is different in this regard in that it obtains the data in near real-time from the LiDAR sensor via a network protocol. Lastly, the 3DSYSTEK viewer only permits viewing of the 3D rendering of the model and does not contain any additional visualizations or data analysis.



### 3.1.3 LiDAR Data Management and Storage Studies

Data management is an essential part of software development when working with large and complex data sets. One team of researchers implemented a LiDAR data-framework pipeline that utilized a spatial database to store raw LiDAR data [25]. The data management system created in this research contained three main functional aspects which covered storing the 3D LiDAR point cloud data in an accessible form, visualizing the processed data in a browser window, and retrieving the data either by user or system input. The final product was a web application that made the data easy to access and allowed for dynamic visualization through a 3D web viewer built in WebGL. The last key component was the integration with a PostgreSQL object-relational database that handled and managed the LiDAR point cloud data making it easily accessible through queries on the web server.

Another research study [8] considered the integration of big data storage technologies within LiDAR visualization web applications. The primary purpose of this research was to explore the advantages and disadvantages of big data solutions when compared to more traditional server based solutions for handling large sets of LiDAR data in web-based applications. In the study, the researchers compared four different big data technologies against an Apache HTTP server to determine if there was any significant differences in the management of large LiDAR point cloud data sets. It was found that two of the technologies, Cassandra and MongoDB, performed optimally and therefore were considered viable options for storage of LiDAR point cloud data. The main improvements were lower latency, higher data throughput, increased storage capacity, and versatility in data modeling.

Each of these studies relates to the LDAT in the fact that they are each aimed at web-based applications that are used to process and visualize LiDAR point cloud data. The design of the first application also shared similarities as it was developed with WebGL to render the 3D view of the LiDAR point cloud data. These applications do also differ from LDAT with the main distinction being the emphasis each has on data

storage and management using big data methods. Both of these applications center on how the data is managed or stored as opposed to the visualization and analysis of the data through a web application.

## **3.2 Additional Applications**

This section looks into two different data visualization applications that each use various technologies to achieve their output. These are akin to LDAT in that they are all grounded in data visualization as well as their use of LiDAR point cloud data; however, they are also each different in their goals and overall application designs. More specifically, each of these applications are not web-based applications but rather applications that run locally on a machine. As data visualization has become increasingly important due to the onset of big data, the applications presented in this section are fairly recent works.

### **3.2.1 LiDAR Visualization Tool for 3D Annotation**

As stated previously, data visualization is used to represent data in a readable and graphical format and is often necessary when dealing with complex data sets such as point clouds. In the study [23], a 3D annotation tool was designed to visualize point cloud scenes. The main purpose of the tool was to be able to annotate objects such as vehicles and pedestrians within a 3D point cloud render. This was accomplished using bounding boxes to annotate the objects and make them easily distinguishable in the scene. The LiDAR sensor used for this tool captured point clouds by rotating around a test vehicle to capture data from every direction. In order to visualize and annotate the scene, the researchers developed a GUI where they could load the point cloud and related image to visualize the data and add in annotations via bounding boxes. A machine learning algorithm would run after the point cloud file and RGB image file were selected and loaded into the tool to search for the specified object in the files.

The 3D annotation tool bears some resemblance to LDAT in that each uses

LiDAR point cloud data for visualization purposes. Additionally, this tool was also used to differentiate between vehicles and pedestrians within a point cloud scene. This correlation is also a differing factor as the annotation tool can identify objects in a point cloud through a machine learning algorithm, which can be limiting depending on what the algorithm is trained to identify. LDAT on the other hand is able to not only distinguish between vehicles and pedestrians in a point cloud but also collects the count of each to display in graph for more detailed data analysis. Another difference between the two is that the annotation tool needs to take in a point cloud file and corresponding image file to detect objects and annotate the scene. These files are loaded into the tool manually whereas LDAT obtains data through a network protocol in which it is streamed directly to the website.

### **3.2.2 LidarView: A Data Processing and Visualization Tool**

LidarView is an application that conducts processing and visualization of 3D point cloud data captured from LiDAR sensors [40]. The LidarView application is built on top of the Paraview open-source code base which allows for loading, interacting, and displaying of the data. Some features that are added onto the Paraview code base include post processing of the data and data handling over a network which manages multiple LiDAR brands in real-time with configurable listening ports. On the end-user side of the application, it has a user interface that enables real-time viewing and analysis of the data. LidarView is also capable of both playing pre recorded data from .pcap files and recording a live data stream that is saved as a .pcap file.

This application is very much comparable to LDAT in many aspects. For instance, each application is able to stream live captured data from LiDAR sensors with the purpose of visualizing the data that is coming through. Moreover, they are both able to return specific properties from the point cloud data set with those being intensity, time, and distance. The main differentiating factors is in the design and some of the key features of each application. LidarView was designed to be a visualization and processing application for LiDAR point cloud data that serves as

a customizable code base that can be used in other applications. The LiDAR data visualization tool performs both visualization and analysis of the LiDAR data being streamed by taking input from the user to find and view the desired data stream. Therefore, LidarView has a more broad goal whereas LDAT is built as a full-fledged application to visualize data.

# Chapter 4

## Software Architecture and Implementation

### 4.1 Introduction

This chapter outlines the application LDAT which is a web application that provides users with the ability to visualize near real-time street traffic data obtained from LiDAR sensors located and placed near street lights. As stated previously, LiDAR data is a complex set of data that is encapsulated within a point cloud. There is a lot of information that can be obtained from point cloud data sets. This application takes in the data from the LiDAR point cloud, cleans the data, and outputs it to the user. Through the web page, the user is able to select different types of visualizations such as the near real-time 3D mesh render, charts, and graphs each of which will highlight different pieces of information found in the data set.

The LDAT architecture and implementation is detailed thoroughly in this chapter to provide a high and low level description of the development process. Section 4.2 provides an overview of both the functional and non-functional requirements placed on this application. Section 4.3 then covers the use cases that are utilized by the application through a use case diagram and detailed description. This is followed by Section 4.4 which goes over the architecture design of the application via a system level design diagram, web site map, and activity diagram. Lastly, Section 4.5 describes the implementation of the application which includes a breakdown of the components that make up the application as well as a brief overview of the tools used during the

implementation process.

## 4.2 Software Requirements

Software design begins with the assessment of the requirements for the application or software system being developed. According to Sommerville [50], software requirements are simply short descriptions of the what the application should do and how it should operate. Requirements provide a detailed breakdown of the applications abilities in addition to some of the operational constraints that are placed on the application. Sommerville further breaks down requirements into distinct types, specifically functional and non-functional requirements, that generally cover user and system based functionality and constraints. This section is split into two parts and offers an exhaustive look at both the functional and non-functional requirements that were developed for this application.

### 4.2.1 Functional Requirements

Functional requirements are defined as descriptive requirements that specify what the application should do [50]. The data visualization web application was developed with key functionality in mind, and these requirements were further broken down into three separate priority categories. These are as follows: base functionality, secondary functionality, and future functionalities. The first category represents the functionalities that are considered pivotal for the application to operate at a basic and fundamental level. The second requirement category depicts the functionalities that are not necessary to the base operation of the application but incorporate additional functionality that improves both the overall usability and user experience. The third and final requirement category are the low priority requirements or future work that could be added to further improve the application.

In Table 4.1, all of the functional requirements for this application and associated priority category are listed along with a brief description. There are a total of eight category one functional requirements, labeled FR1-FR8, that encompass the

Table 4.1: List of Functional Requirements

<b>Functional Requirements</b>		
Name	Priority	Description
FR1	1	The application will enable the user to visualize 3D LiDAR point cloud data.
FR2	1	The application will be near real-time.
FR3	1	The application will be able to display and visualize data from individual sensors through point clouds.
FR4	1	The application will display count for vehicles.
FR5	1	The application will display count for pedestrians.
FR5	1	The application will be able to inform the user if there is no data being received.
FR7	1	The application will convert point cloud data from .pcd file format.
FR8	1	The application will generate a friendly format from the LiDAR data.
FR9	2	The application will be able to visualize classifications which include vehicles, pedestrians, and cyclists through LiDAR sensors.
FR10	2	The application will allow the user to select and visualize data from a specific location.
FR11	2	The application will include different visualization methods that the user will be able to select in the GUI.
FR12	2	The application will be able to handle or add multiple data formats.
FR13	2	The application will be able to visualize data in high quality while handling large amounts of data in the background.
FR14	3	The application will have a login page.
FR15	3	The application will display statistics such as the speed and trajectory for any given vehicle or pedestrian.
FR16	3	The application will be able to visualize data from multiple sensors in one scene.
FR17	3	The application will be able to add new locations and sensors once they become available or active.
FR18	3	The application will include ability to change the color palette of the point cloud.
FR19	3	The application will handle low quality data streams or missing data.
FR20	3	The application will be able to record the data coming over the data pipe.

base functionality of the application. The first and second requirements focus on the applications ability to enable users to visualize a 3D mesh render of the point cloud data in near real-time to what is happening at the sensor site. The third requirement highlights that the data will be streamed and visualized from individual LiDAR sensors at any given time. The fourth and fifth requirements center on data visualization, specifically by emphasizing that additional data will be displayed such as the current count value for both vehicles and pedestrians. The sixth requirement showcases the applications ability to notify the user when no data is being received from the sensor. Finally, the seventh and eighth requirements cover the systems ability to convert point cloud data from the .pcd format and ability to generate cleaned up data that is sent to the front end.

The category two requirements present the secondary functionality or the additional functionality that add to the overall usability of the application. There are a total of five category two requirements designated by labels FR9-FR13. The first requirement (FR9) establishes the applications ability to visualize classifications of variables detected from the LiDAR sensor, which is especially important as it adds to the usefulness of the visualization. The second requirement (FR10) covers the selection of different locations. This is necessary as there are multiple sensors that user will be able to choose from and therefore different data sets that the user can glean different information. The third requirement (FR11) is the inclusion of different visualization methods that the user can choose from which includes various graphs and charts that each highlight different pieces of information. The fourth requirement (FR12) involves the applications capability to handle multiple data formats such as .pcd and .pcap files. The fifth requirement (FR13) involves the applications capacity to handle large amounts of data in the background while a visualization is being presented on the front end. This is essential as the application needs to be able to function efficiently so as to provide a high quality experience to the user.

Lastly, the category three requirements go over functionality that can be developed in future iterations. Similarly to the previous category, this category also has



a sum of seven requirements with these being identified as FR14-FR20. The first requirement (FR14) is a login page where users can create an account to access their own personal dashboard to review the data. The second requirement (FR15) covers the ability to display statistics such as speed and trajectory of any vehicle or pedestrian. This functionality is not included in this thesis as it will require complex machine learning techniques to run properly. The third and fourth requirements (FR16 and FR17) focus on sensor locations, specifically visualizing data from multiple sensors at once and adding new sensor locations to the system as they become available. The fifth requirement (FR18) involves the customization of the color palette to change the color of the point cloud based on user preference. The sixth and seventh requirements (FR19 and FR20) cover additional data management. This includes being able to handle low quality or missing data and therefore notifying the user as well as being able to record data in the data pipeline. This additional functionality was not implemented as the sensor network was limited to two sensors and the data management was not required as this application was developed as a proof of concept. Future development and features will be discussed further in Chapter 8.

## 4.2.2 Non-Functional Requirements

As explained by Sommerville [50], non-functional requirements are not directly related with the services and use cases an application provides to the user but rather relate to the implementation, performance, and reliability of a system. There are far fewer non-functional requirements than functional requirements, numbering less than half of the latter, as can be seen in Table 4.2. For this application, the non-functional requirements act as constraints on the overall system implementation. These constraints specify certain technologies, programming languages, and hardware that were used to implement and build the system.

This application has eight non-functional requirements in total. The first and second requirements state that the system will use data obtained from LiDAR sensors and that the data will be in the form of a point cloud. In addition, the third explains

Table 4.2: List of Non-Functional Requirements

<b>Non-Functional Requirements</b>	
Name	Description
NFR1	The system will use data from LiDAR sensors.
NFR2	The system will obtain data in the form of a point cloud data stream that will come from the LiDAR sensors.
NFR3	The system point cloud data will be in .pcd file format.
NFR4	The system will be developed using web technologies Angular 11 and the Flask Microframework.
NFR5	The system will be built using Python, JavaScript, HTML, and CSS.
NFR6	The system will utilize WebGL.
NFR7	The system will have version control through a Git repository.
NFR8	The system will be ADA compliant.

that the point cloud data will be given to the system in the .pcd file format. The fourth requirement details Angular 11 and Flask framework as the primary development technologies. Furthermore, the fifth requirement specifies the programming languages used to build the application, those specifically being Python, JavaScript, HTML, and CSS. The sixth covers the utilization of WebGL to aid in graphical processing of 2D/3D graphics in the application. The seventh requirement delineates the use of a version control system through the OIT department on the University of Nevada, Reno campus. The eighth and final requirement points out that the system will be ADA compliant meaning that it will be made accessible to all users.

### 4.3 Use Case Modeling

Use cases expand on the requirements discovery process by focusing on the perspective of the end user and effectively communicating the systems behavior and response to user input [55]. This is typically accomplished through a high level diagram which indicates the actors involved in a specific interaction and the relationships between the actors and use cases [50]. In this section, we will discuss the use cases for this application by analyzing the use case diagram along with full-fledged descriptions of each use case.

The LDAT use case diagram is illustrated in Figure 4.1. The application is

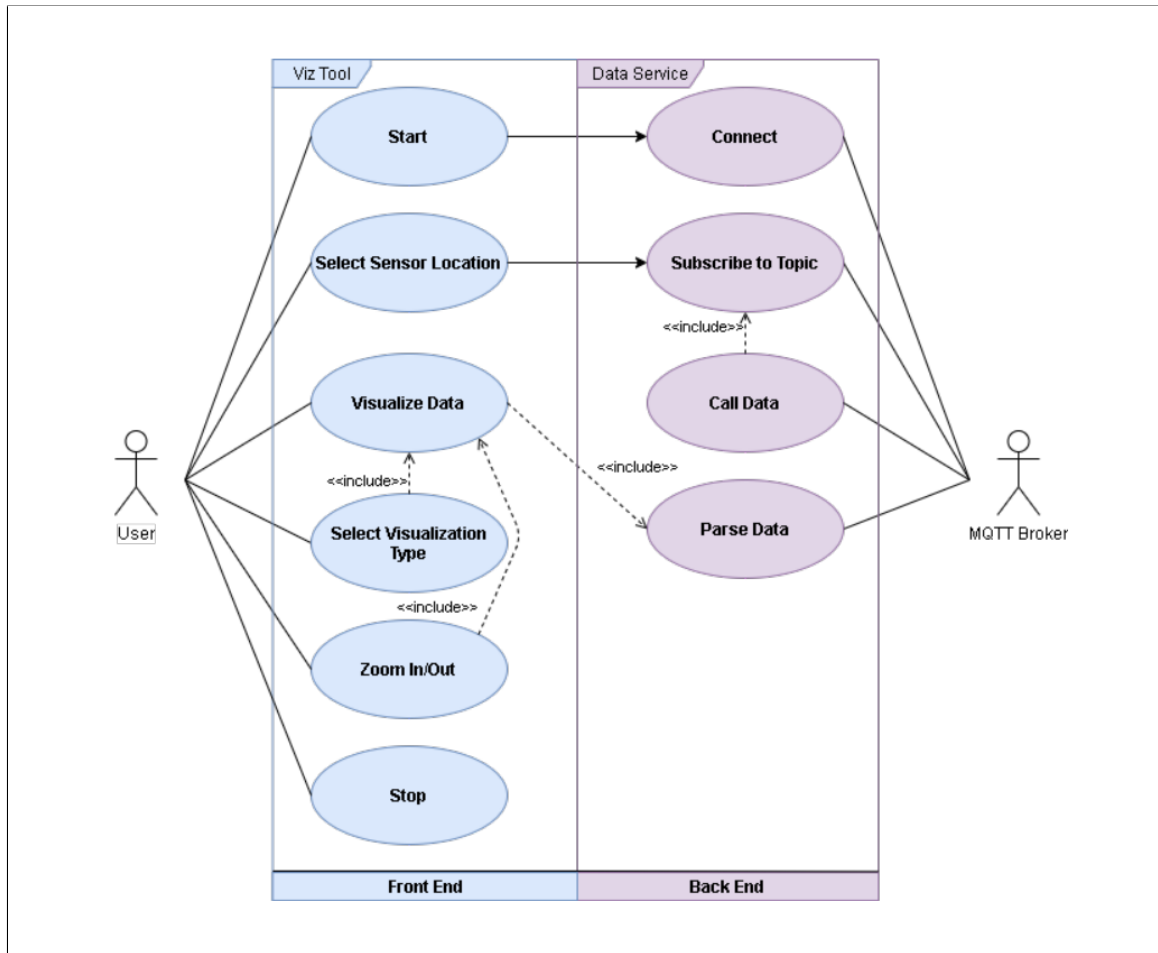


Figure 4.1: Use case diagram for LDAT.

comprised of ten use cases with those use cases being divided between two separate actors. The two actors on the system are the User and the MQTT Broker. The User is the one who would be accessing the website to analyze and visualize the data that is being streamed from the LiDAR sensors. They would have different options available through the web application that allow for different visualization types and location selection. The MQTT Broker operates on and connects to the back end of the system and ensures that the system is connected to the data stream and therefore receiving data from the sensors. Table 4.3 provides a list containing brief descriptions of each use case in the system.

Table 4.3: Full List of Detailed Use Case Descriptions

Use Case Descriptions		
#	Name	Description
UC1	Start	Users will be able to start the application when they connect via their browser. Once the application is started it will connect to the back end to start processing data to prepare for the user.
UC2	Select Sensor Location	Allows the user to select a specific sensor location in which to obtain and visualize data. The user will be able to open a drop-down menu and select the location from the list of available sensors.
UC3	Visualize Data	Users will be able to visualize data through graphs and charts in addition to a 3D mesh render of the near real-time data stream. These options will be accessible to the user through the web page dashboard.
UC4	Select Visualization Type	Allows the user to select a visualization type through a drop down menu. This includes different types of graphs and charts that highlight different pieces of data.
UC5	Zoom In/Out	Users will have the ability to zoom in and out of visualization graphs and charts. This will enable the user to have an interactive experience with the data and possibly gain increased comprehension of the data being presented.
UC6	Stop	Users will be able to stop the application when they disconnect via their browser. When the application stops it will disconnect from the back end server and stop making websocket calls.
UC7	Connect	The MQTT Broker activates over the network to obtain the near real time data being streamed from the LiDAR sensors.
UC8	Subscribe to Topic	Once the MQTT broker is connected and running, the broker then subscribes to the topic that the user selects on the front end. The topics relate to locations of the active sensors.
UC9	Call Data	Following the subscription process, the data is then called to be processed. The data is received as a raw .pcd file that then must be cleaned and prepped for visualization.
UC10	Parse Data	As the data is obtained, it is sent to be parsed and processed to ensure that it is in a clean and readable format for the system. This cleaned data is then used for end user visualization and analysis of the data.

## 4.4 Architecture Design

Architecture design is a key component of software development and is the first piece of the design process. It specifies how the system should be structured based on the requirements and primary use cases of the system [50]. In this section, the architecture and design of the system will be explained through two subdivisions. The first subdivision, Section 4.4.1, will assess the system architecture, otherwise referred to as the high level design, through a system level diagram. The second subdivision, Section 4.4.2, will cover the medium level design of the application. This will be accomplished through various diagrams including a web map, activity diagram, and class diagram. These diagrams will show the layout of the web application, the general perspective of the end user, and relationships between various components within the system.

### 4.4.1 System Design

The system architecture was designed primarily with the idea that the application be able to run as smoothly as possible to create an optimal user experience. In order to fulfill this requirement, the system would need to be designed with multiple servers in order to properly handle, manage, and process the large amounts of data that would be coming from the LiDAR sensors. Figure 4.2 presents the system level diagram for this application that provides a broad overview of the system and indicates the different components in the system along with the relationships between them.

LDAT is comprised of four main components that make up the system. The first of these components is the LiDAR Sensors. This is where the data is gathered in its raw format and pushed to the network through the second main component, the MQTT Broker. The MQTT Broker is a network publish/subscribe protocol that allows the system to subscribe and receive messages directly from the LiDAR sensors. This network protocol not only receives the messages directly from the sensors, but also securely transports the data to other servers that are subscribed to the topic that

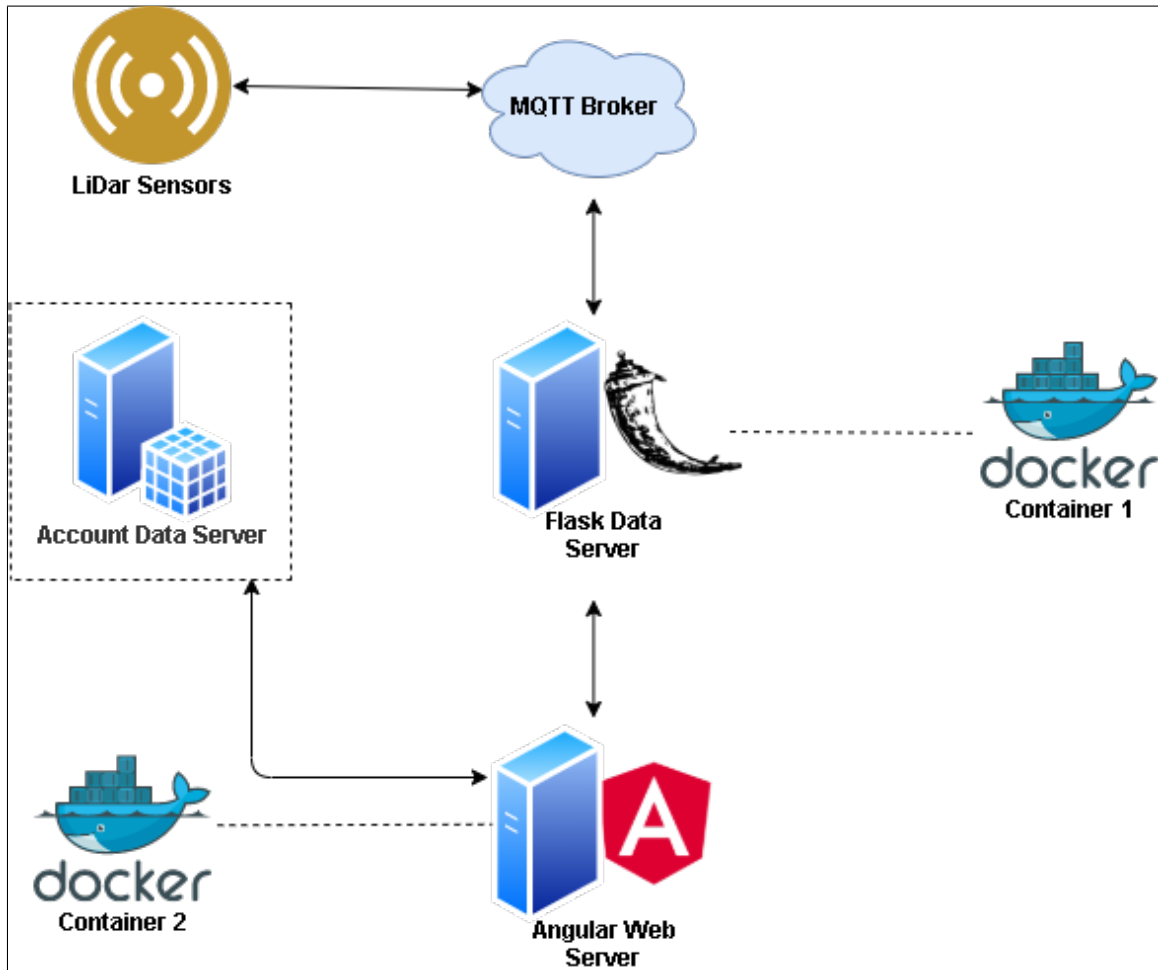


Figure 4.2: System level diagram that showcases the main system components of LDAT.

is linked directly to the sensor.

The third component in the system architecture is the Flask Data Server which connects directly to the MQTT Broker. This data server subscribes to the topics on the MQTT Broker and the broker then pushes or publishes the data onto the server. Once on the server, the Flask app parses through the data and preps it for visualization on the fourth component, otherwise referred to as the Angular Web Server. The Angular Web Server acts as the main web page that is accessible to users and it is where data visualization and analysis occur. An important note is that both the Flask Data Server and Angular Web Server are both placed in separate Docker containers to increase development efficiency and portability. Additionally, there is

an Account Data Server which will serve as a database server for user accounts such as log in information. This server will be covered in more detail in Section 8.2 Future Work.

## 4.4.2 Application Design

For the application design of the LDAT software we have created diagrams to exemplify the front end of the application. Shown in Figure 4.3 is a website map of the LDAT application. This exemplifies the main UI layout of the application and some of the different features that can be found on each page of the website. The user will be welcomed to the application on the home page and from there will be able to go to the data explorer form to create a data visualization dashboard. Once the form is submitted, the user will be navigated to the data visualization dashboard which currently contains three different visualizations that provide the user with differing insights into the point cloud data. The user can then go to the settings page from the dashboard to make some adjustments to the visualizations including changing the

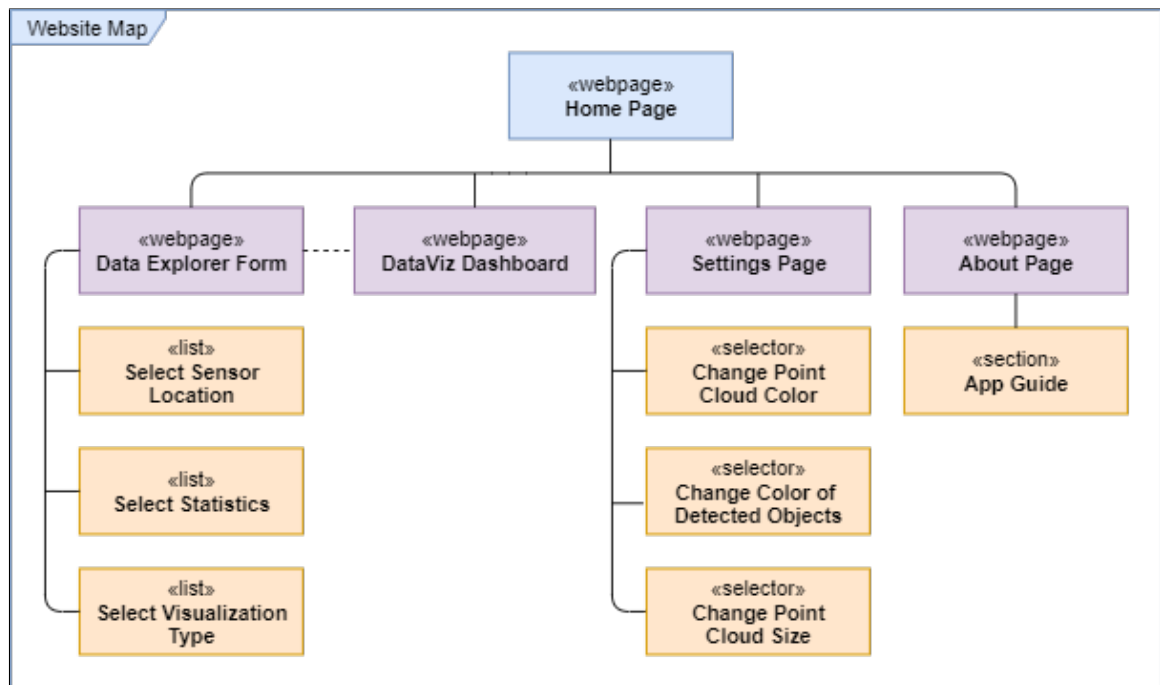


Figure 4.3: Website map for LDAT application.

color and size of the points. Lastly, the user can go to the about page which provides a brief description of the application and a short guide to instruct the user how to use LDAT. A more detailed breakdown along with screenshots of the UI will be covered in Chapter 5.

The next set of design diagrams are activity diagrams which each provide a generalized look at the navigational pipeline for LDAT. The first activity diagram, pictured in Figure 4.4, depicts the general flow to fill out the data explorer form. In order to complete the form, the user must enter all the required fields otherwise they will receive a message to fill in the field before submission. Once all the required fields are entered the user can then successfully submit the form. Figure 4.5 showcases the second activity diagram which illustrates the path and options users have when changing the point cloud color within the settings page. This diagram demonstrates the two options users have to change the color which are either clicking one of the

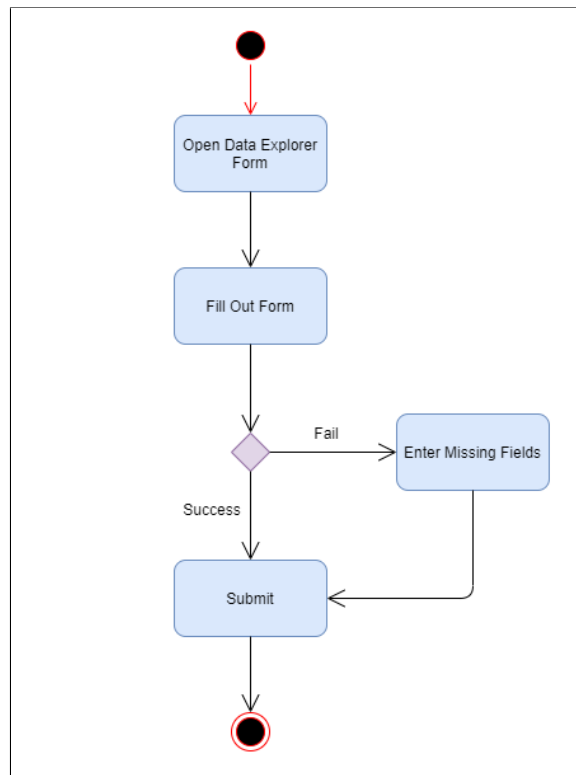


Figure 4.4: Activity diagram for completing the data explorer form.



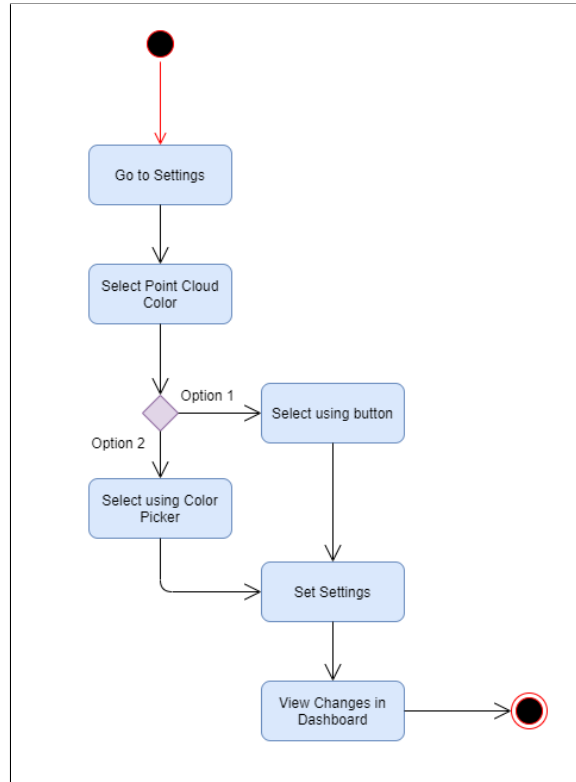


Figure 4.5: Activity diagram to change the point cloud color on the Settings page.

three buttons or clicking the color picker which allows the user to select from the entire color spectrum and therefore have much more customization.

The last design diagram is that of a comprehensive class diagram as depicted in Figure 4.6. This diagram outlines the class hierarchy of the software system behind the LDAT application. The hierarchy of classes can be broken down into three different sections: back end/networking, data service, and front end/configuration. The back end and networking classes include the MQTT Broker, Point Cloud Data Parser, and Websockets. The data service classes will contain the PCD and Data Service classes which read in and push the data to the front end dashboard. Finally, the front end and configuration classes contain classes that directly affect the client side and output of the application. The following list provides a brief description of each class in the diagram.

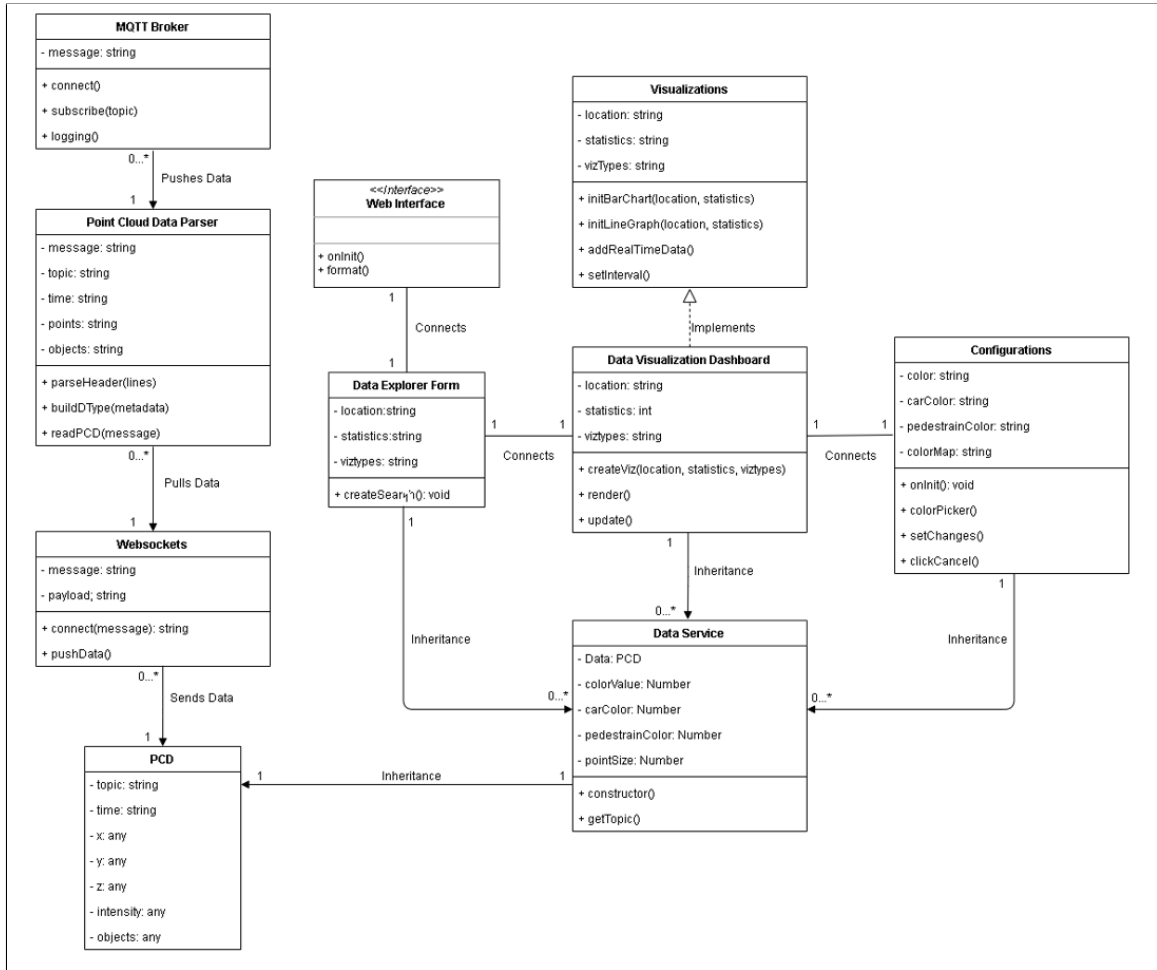


Figure 4.6: Complete class diagram for the LDAT application.

**MQTT Broker:** Class designed to connect to MQTT broker and subscribe to the different topics. This class obtains the initial packets of data being sent from the LiDAR sensors.

**Point Cloud Data Parser:** Parses and cleans the raw data files that come through from the MQTT broker. Class is responsible for obtaining the metadata and object information to output in a readable format for the front end.

**Websockets:** Class responsible for connecting the Flask back end to the Angular front end to pass up the cleaned data.

**PCD:** Point Cloud Data class which stores the cleaned data and breaks down the points based on xyz coordinates, intensity value, and detected objects.

**Data Service:** Main data class that is responsible for sending the data to different modules on the front end and updating variables such as selected location and point color.

**Web Interface:** Main web interface of the LDAT application. Initializes and formats the web application on start up.

**Data Explorer Form:** Class that covers the form submission and selection of different locations, statistics presented, and visualization types. This class module connects to the Data Service to view the selected sensor location.

**Data Visualization Dashboard:** Class responsible for creating visualizations and rendering the 3D mesh of the selected sensor location.

**Configurations:** Allows users to make changes to the 3D point cloud mesh including changing color of the environment and detected objects in addition to updating the size of the points in the point cloud.

**Visualizations:** This class initializes the bar chart and line graph and adds in the near real time data stream. In order to update, an interval needs to be set to refresh each graph with the most recent information every few seconds.

## 4.5 Implementation

The implementation of LDAT included the setup of hardware to collect live street traffic data and software that was developed to allow for visualization and analysis of the data collected. The hardware used for this application was the Velodyne LiDAR Ultra Puck Sensor, also known as the VLP-32C. The Ultra Puck is a LiDAR sensor

with a high density, 200 meter range with a full 360° view of the surrounding area thereby providing high performance in a very compact form [54]. A snapshot of the sensor itself can be seen in Figure 4.7 showcasing its compact design. Due to its high performance and thoughtful construction it has become a fairly popular sensor in many different fields including terrain mapping, robotics, and autonomous vehicles.

For this application, the sensors were set up near street lights in the city of Reno, Nevada specifically along the corner of East 15th Street and North Virginia Street. This is located adjacent to the UNR campus on opposite corners of the intersection and is represented in Figure 4.8. The first sensor was stationed on the Southeast corner of the intersection right outside of the UNR E.L. Wiegand Fitness Center while the second sensor was stationed along the Northwest corner directly across the street from the Lawlor Events Center. Once activated, the sensors were able to generate a steady stream of live data from the street view. The data collected by the sensors was then passed to the MQTT Broker which published the data to the subscribed services [36]. It should also be noted that the sensors can be given



Figure 4.7: Velodyne LiDAR Ultra Puck Sensor [54].



Figure 4.8: Overview image, obtained from Google Maps [15], of the street intersection of East 15th Street and North Virginia Street. The leftmost cross represents the Northwest sensor location and the rightmost cross represents the Southeast sensor location.

a set rotations per minute (rpm) or spin rate. A higher rpm means that there will be more frames but overall less quality due to there being less points in the scene while a lower rpm means less frames but higher number of points. Therefore, lower rpm provides a higher object detection and classification accuracy due to there being more points within the frame. The sensors for this application were set with a rpm value of about 600 which resulted in approximately 10 frames per second (fps). This was done to ensure that the object detection was as accurate as possible while also providing enough detail in the point cloud scene.

Moving forward towards the application software, LDAT is currently comprised of two main software systems that work together to create a fully functional web service that can host large amounts of complex point cloud data. The first of these is a back end web application built using the Flask microframework and Python programming language. This application uses the MQTT and threading [43] libraries to pull in and process the raw data being streamed from the LiDAR sensors. Once

pulled in, the data is then run through an additional Python script named `parser.py` which handles the parsing of the `.pcd` file. The parser first runs through header of the file which is exemplified in Listing 4.1 in order to obtain the metadata. The metadata in header contains the `.pcd` version, the fields a point can have, the size of each field, the number of points in the file, the topic or location of the sensor, and the time measured in minutes since the top of the hour. The header is then followed by the point values which map out the environment and track the objects in the scene. The point values are parsed out and passed into a pandas [39] data frame as a cleaned data set before being stored in dictionary to be passed back up to the main Flask application. After the file has been parsed and cleaned it is then compressed using the `zstd` library [48] and passed to the front end web client via websockets.

Listing 4.1: Example of a Point Cloud Data File (.PCD) Header

```
# .PCD v0.7
VERSION 0.7
FIELDS x y z intensity
SIZE 4 4 4 4
TYPE F F F F
COUNT 1 1 1 1
WIDTH 42660
HEIGHT 1
VIEWPOINT 0 0 0 1 0 0 0
POINTS 42660
DATA binary
Objects Bounding Boxes
minx maxx miny maxy minz maxz
5.50767 8.2054 -9.48886 -7.57338 -1.52405 -0.622798
1.87516 3.27128 12.3199 14.0863 -0.6699 0.22545
0.917429 1.31955 -9.86547 9.10885 -1.54158 0.864566
-6.5374 -5.81841 27.8998 29.7645 -0.331794 0.515684
9.33511 9.75947 -8.03737 -7.9267 -1.01152 -0.572798
end objects
Topic test15thVirginiaSE
Time 51071920
```

The second main software component is the front end web service which was built using the Angular 11 web framework. The webpage was designed to act as a visualization tool for users to be able to access, visualize, and analyze the near real

time LiDAR data stream that was coming over the network. As stated previously, the data was transferred from the back end to the front end using web sockets. This was accomplished through the use of the SocketIO library [3] which allows for real-time communication between the client and server, which in this case is the front end and the back end. Once the data was obtained on the front end it was stored in a singleton service [14] which is a service in an Angular application in which there is only one instance. This makes the data set easily accessible for other components within the application. In order to output a 3D render of the point cloud mesh the three.js library [52] was used to create the scene, set the camera, set the points, and animate. Additionally, there was an update function which would update the position of the points and the color of the points therefore allowing for changes in the scene to be observed. Lastly, ngx-charts [38] was the chart library used to create additional data visualizations such as bar charts and line graphs. The visualization charts and UI components built on the Angular front end will be described in detail in Chapter 5.

Another aspect of the software implementation was the use of containers. For both the front and back end, Docker containers were used to package each server in its own containerized environment. Additionally, a docker compose configuration file was created to be able to run both containers simultaneously through a simple terminal command. Instructions on how to run the program can be found in the GitHub repository where the code for this application is stored [1]. The file structure for the application is separated based on the front end and back end with one main folder dedicated to each. The Flask back end is fairly simple and straightforward containing a Dockerfile to install dependencies within the Docker container and a source folder named src which contains the two main python scripts used to connect to the MQTT broker and parse the data.

The Angular front end on the other hand had a lot more components and therefore was divided into many sub folders for better organization. The root directory contained many Angular created files as well as the application Dockerfile, application source folder, and package.json file which contains all the package libraries. The

source folder contains the application code folder and assets folder which stores images used for some of the main background in the UI. The application code folder is broken down based on the different services created in the application such as an MQTT service, renderer service, and UI components or, more specifically, the different web pages found when browsing the application. The main application files and singleton data service are stored in root application code folder.

There were also some challenges and limitations during the implementation that should be noted. The first challenge was within the renderer, which as stated previously utilized the three.js library to create and animate the point cloud scene. While the library was fairly complex, one of the biggest challenges was configuring controls to be able to interact with the scene on the web server. The main challenge here was finding an additional library that worked with Angular Typescript as the main three.js library did not have a controls class built in to it. The controls library that was ultimately used was the Orbit Controls library [4]. The second challenge was managing the point cloud data set. The primary issue here was latency and slow performance of the website due to heavy data throughput. This was solved through the use of the zstd library to compress the data while being transferred from the back end to front end. This does solve the issue and greatly increased the performance of LDAT, however, a more optimal solution has been discussed and will be presented in Chapter 8



# Chapter 5

## Prototype in Action

### 5.1 Introduction

User Interface (UI) design is a highly crucial element of software development as it directly affects the end user experience. The goal of a UI should be to make an application intuitive and easy to use thereby increasing both the usability and general memorability of the application. This Chapter provides an outline of the LDAT application prototype and showcases the final UI design as well as some of the applications key functionality. Section 5.2 presents each web page in the application and details the general navigation flow that the end user will be guided through. This is then followed by Section 5.3, which lists the data visualization charts and presents their design and general purpose. In addition, some configurable components that allow for customization of the appearance of the 3D point cloud mesh within the LDAT application prototype will be highlighted and discussed.

### 5.2 User Interface

LDAT was developed with simplicity in mind. It was designed to be a fairly easy to use website with a focal point on the visualization and analysis of LiDAR data. The application itself is comprised of five separate web pages that are geared towards providing the end user with a clear and straightforward experience. These pages include the application home, explorer form, visualization dashboard, settings, and about pages. Also included in the application to make navigating the website quick

and easy is a navigation bar which includes tabs for each page. The general navigation path begins on the home page by welcoming the user and guiding them towards the next step in accessing the data.

Currently, the application runs on a local host server and is accessible through a web browser. Upon starting LDAT, the user is greeted by the Home Page which can be seen in Figure 5.1. The Home page contains a brief welcoming message that is overlaying an expansive background image of the city of Reno. This page was created to quickly introduce users to the website and provide a preface explaining the purpose behind LDAT. Underneath the welcoming preface is a button labeled ‘Get Started!’ which starts users on the path towards accessing the data. When this button is clicked, users will be taken to the Data Explorer form which is presented in Figure 5.2. The Data Explorer form contains three fields which cover the selection of the sensor location, analysis variables, and visualization type. The sensor location is a list of the active sensors that can be selected and visualized while the analysis variables are the different types of data statistics that users can include for analysis. Lastly, the visualization type includes a list of different types of visualization charts

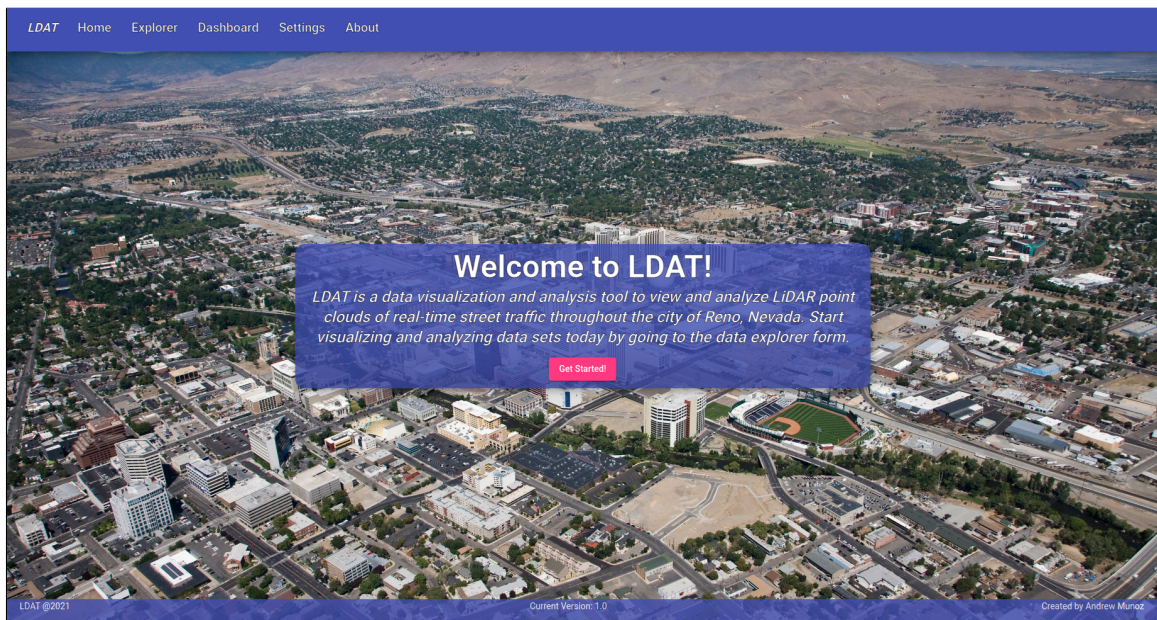


Figure 5.1: LDAT Home page which greets users when they first open up the application.

Figure 5.2: Data Explorer Form which allows users to fill out and specify fields pertaining to the data visualizations.

and graphs that the user can select to view. In this current iteration, LDAT is limited to live data charts that are updated based on live point cloud data stream. Once the user is satisfied with the selections the next step is to click on the 'Submit' button at the bottom of the form.

After submitting the form, the user is routed to the Data Visualization Dashboard. The current iteration of the dashboard, which is exemplified in Figure 5.3, contains three different types of visualization charts and graphs. These include a bar chart, single line chart, and near real time 3D mesh render. The charts are layed out so that the user first sees the live data coming through in the bar and line charts and can then scroll to see the 3D mesh render of the selected sensor location. The data from these charts as well as more in depth explanations will be found in Section 5.3. The second to last page in the website is the settings page as shown in Figure 5.4. The settings page is a very simple page which allows for the user to configure specific elements within the 3D point cloud. These configurations change the appearance of the point cloud by changing the color of the point cloud environment or detected objects as well as the size of each individual point found in the scene.

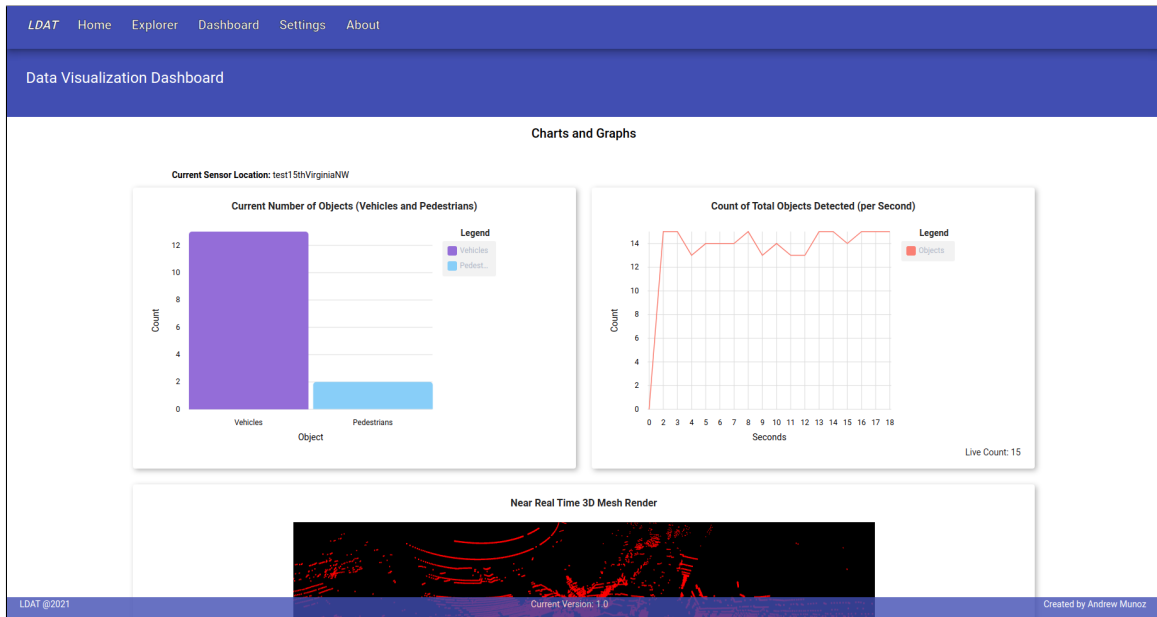


Figure 5.3: Data Visualization Dashboard showcasing the three different data visualizations available.

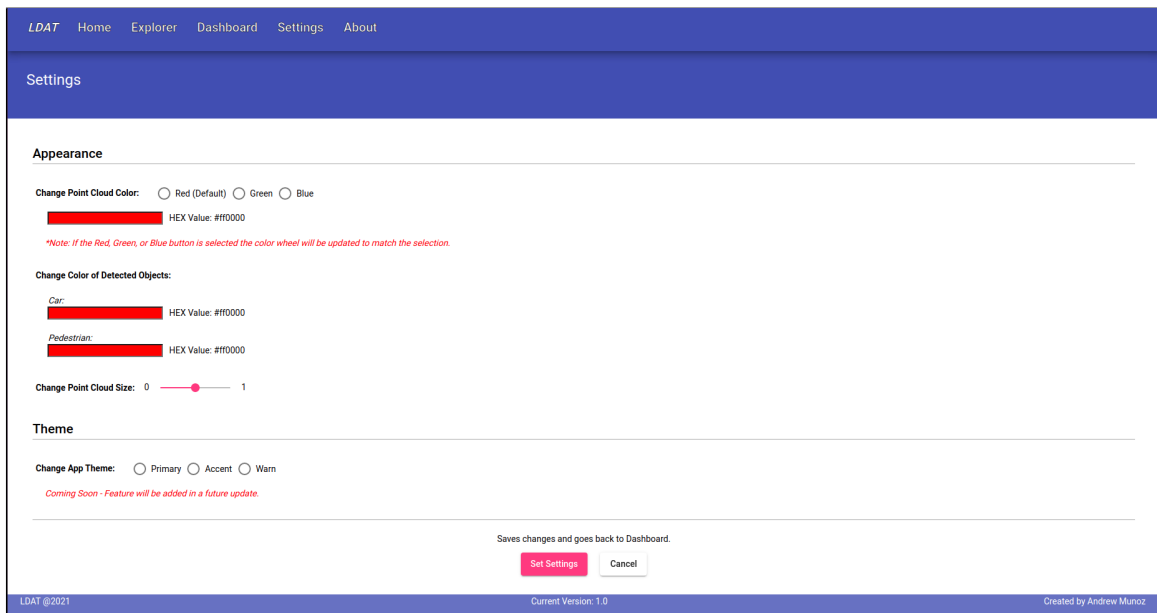


Figure 5.4: Settings page which allows for changes to be made to the 3D point cloud.

The final page of LDAT is the about page exhibited in Figure 5.5. The about page provides some background on the application and details the purpose of LDAT and a simple guide to use the application. The guide outlines the process in which a user would be able to create visualizations and interact with the data. It also details how to interact with the 3D point cloud scene as well as the customizable features located on the settings page. Lastly, there is also a brief section labeled ‘Documentation/Resources’ which includes a link to the GitHub repository that hosts the application code for LDAT.

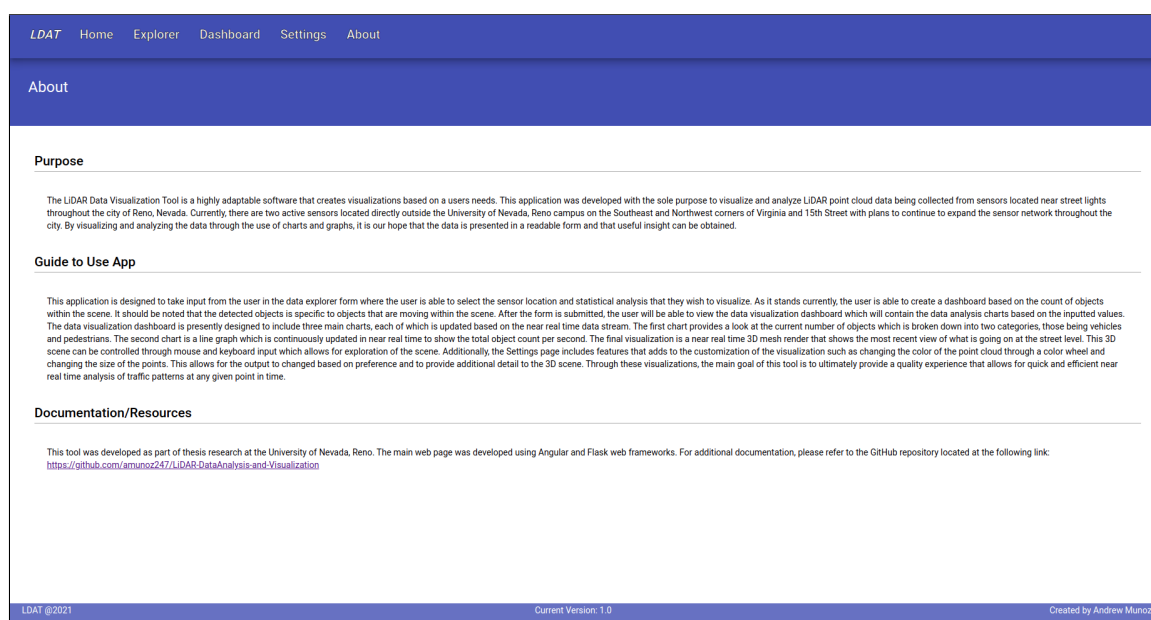


Figure 5.5: About page that details the purpose of the website and provides a general guide on how to use the application.

## 5.3 Visualization and Analysis

Currently, the data that is being visualized within the LDAT application is limited to the near real time LiDAR sensor data stream. This means that the analysis that can be conducted is focused on the live data stream as opposed to a comparative analysis between different data sets. The data from each of the two sensors can be visualized one at a time and can only be selected through the data explorer form. The goal of

the data in its current state is to assess the traffic at any given time and be able to display to the user whether there is a high density of traffic whether it be vehicles, pedestrians, or both.

### 5.3.1 Dashboard Visualizations

The three main data visualizations that are on display on the data visualization dashboard are all live data charts that each provide the users with a different key insight into the data. The first visualization is a bar chart which can be seen in Figure 5.6. The bar chart highlights the current number of objects that are detected in the scene at any given time. In the context of the data, the objects detected in the scene are specifically either vehicles or pedestrians. Additionally, the objects must be moving in order to be detected by the sensor. Therefore, a parked vehicle on the street is typically not detected or listed in the count shown on the chart.

The second visualization is a line graph that displays the count of total objects

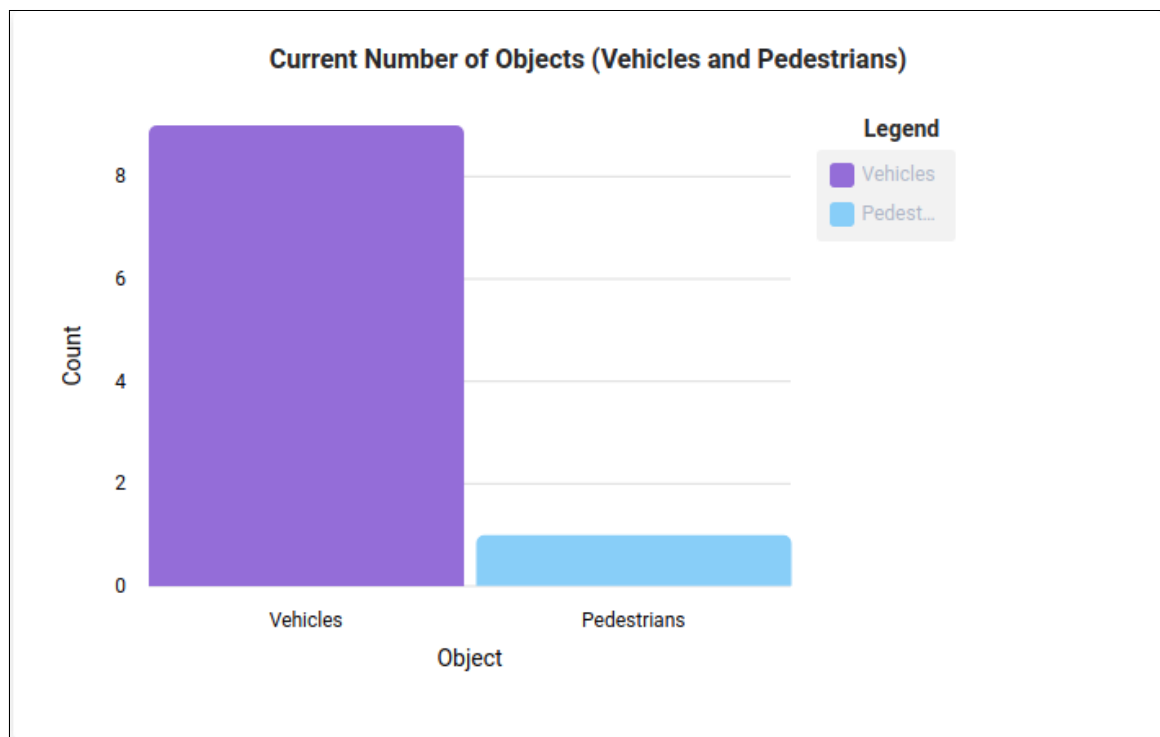


Figure 5.6: Bar chart that displays the current count of each object being detected in the point cloud. The objects that can be detected are vehicles and pedestrians.

detected in the scene. Figure 5.7 showcases the line graph which underlines large shifts in the number of objects in a short range of time. This is due primarily to a large amount of vehicles going through the intersection quickly at any given moment thereby causing fluctuations that is reflected in the data set. The third and final visualization is that of the near real time 3D point cloud mesh shown in Figure 5.8. The render of the 3D point cloud mesh exhibits what is happening at the street level in near real time and allows the user a quick insight into the ongoing street traffic. The scene is also able to be manipulated through mouse and keyboard input. The controls of the scene are as follows: WASD keys move the scene camera, left mouse click rotates the scene, right click pans the scene, and the scroll wheel permits zooming in/out.

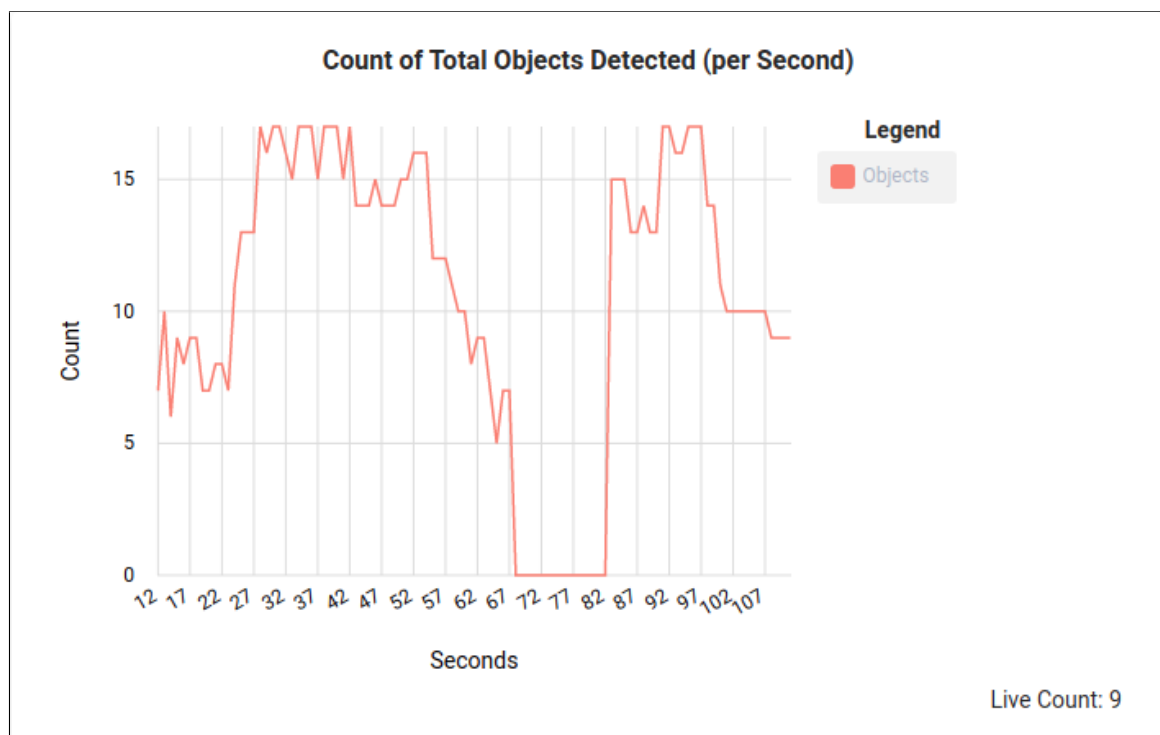


Figure 5.7: Line graph that presents the total number of objects detected within the point cloud.

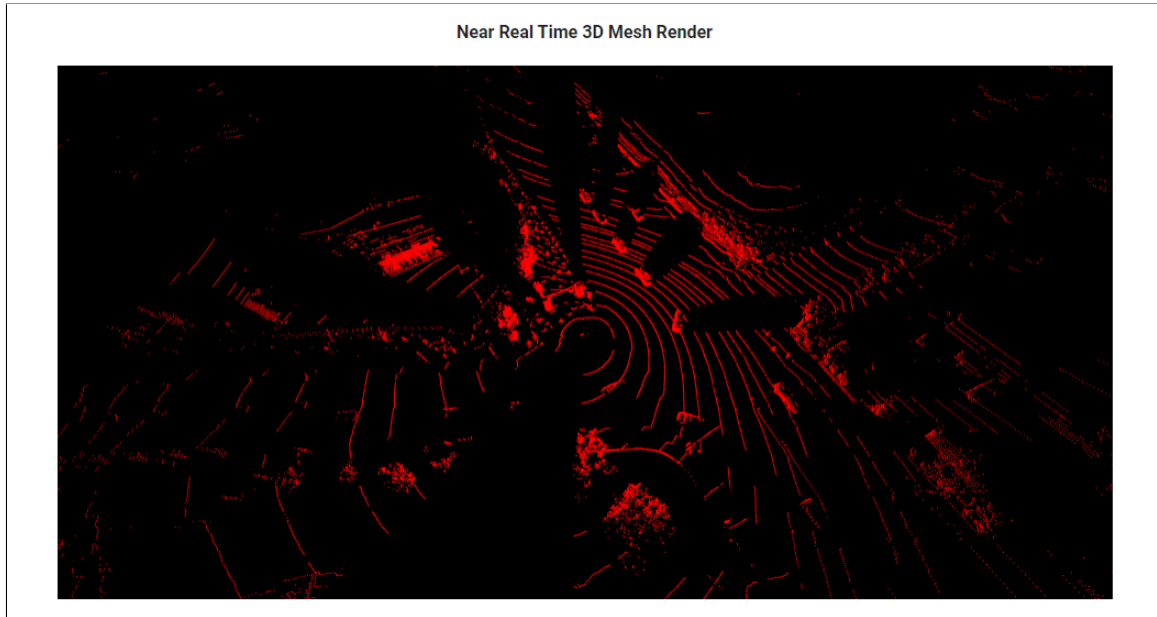


Figure 5.8: Screenshot of near real time 3D point cloud mesh render which displays the view from the sensor located on the Southeast corner.

### 5.3.2 Configurable Components

The settings page allows users to make changes to the point cloud render making it configurable based on the users preference. There are four changes that can be made to the appearance of 3D point cloud including changing the color of the point cloud environment, changing the color of each of the types of detected objects, and adjusting the size of the individual points in the point cloud. Figure 5.9 is a screenshot of the UI for these configurable elements. There are a total of three color pickers with one for the environment and two for the types of detectable objects. Each of these can be changed to a distinct color with the purpose to make the objects in the scene easily distinguishable from the environment. In order to change the color of each, the bar with the current color needs to be selected to open up the color picker window. Figure 5.10 shows the color picker window and the various options that can be selected. The color picker can be broken down into sections with the top of the window being the selected color and its variations from light to dark. The first bar is the color spectrum which the user can slide until the desired color is selected and



## Appearance

---

**Change Point Cloud Color:**    Red (Default)    Green    Blue

HEX Value: #0000ff

*\*Note: If the Red, Green, or Blue button is selected the color picker will be updated to match the selection.*

**Change Color of Detected Objects:**

*Car:*

HEX Value: #3efe1f

*Pedestrian:*

HEX Value: #ff09ce

**Change Point Cloud Size:**   0      1

Figure 5.9: UI layout for the Appearance section found on the Settings page.

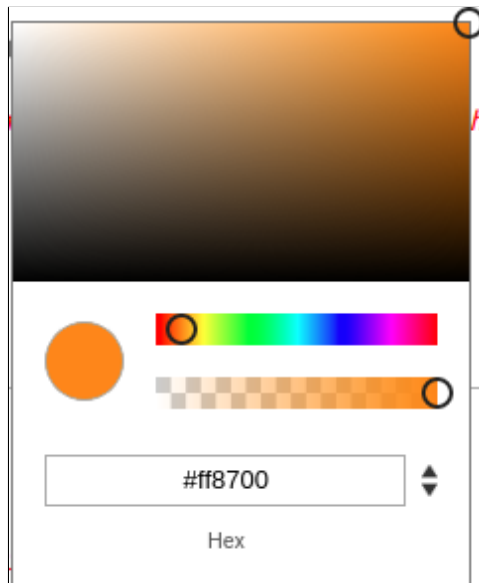


Figure 5.10: Color picker UI window.

lastly the second bar is the opacity of the selected color.

Once the color is selected for each of the three objects, the user can then make changes to the point cloud size. This means that the size of each individual point in

the point cloud can be changed to be more dense or thin through the slider. After all the selections are complete, the user can then set the settings to go and view the changes in the 3D point cloud mesh scene. Figure 5.11 displays the 3D scene with changes made to the colors as well as having the point cloud size changed from the default 0.5 to a much thinner 0.1 using the slider. The detected objects can also be clearly distinguished due to the stark color contrast between the environment and objects.

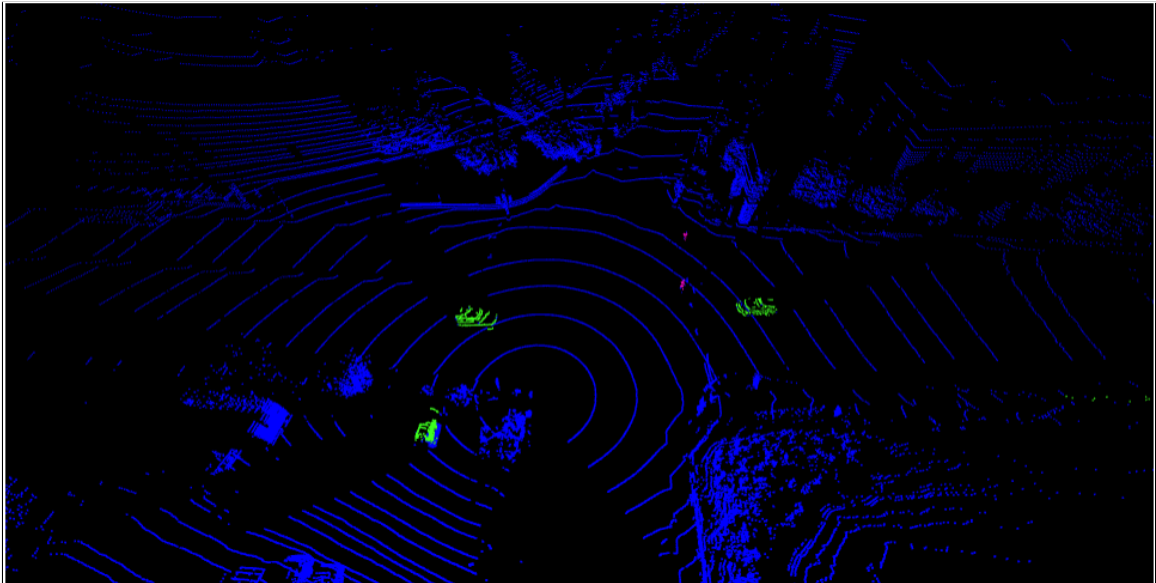


Figure 5.11: Screenshot of 3D point cloud mesh render scene with object colors changed and a point cloud size of 0.1. The sensor view shown is the Northwest corner of the intersection.

# Chapter 6

## User Study

### 6.1 Introduction

In Human Computer Interaction (HCI) related research, experiments are typically conducted to test interfaces to determine their usefulness and practicality. Software should be designed with the end user in mind as any and all software interfaces are built to be used by people, therefore, testing is essential to ensure for optimal interface design. In his book entitled *Human Computer Interaction*, author I.Scott Mackenzie details how to design HCI experiments and why they are so critical [29]. As MacKenzie states, HCI experiments take in a signal and noise variables with the signal representing the point of interest or investigation and noise representing external factors such as temperature, lighting, and fatigue each of which can slightly influence the results of the experiment. In the context of this thesis, a user study was conducted to act as an experiment to test the usability of the UI for the LDAT web application.

Before a user study can be conducted with human participants there are some steps that need to be covered to get approval from the Institutional Review Board (IRB) so that testing can begin. First, a certification is needed to show that the training on how to conduct experiments with humans as test subjects has been completed. This certification training is called the Collaborative Institutional Training Initiative (CITI) which is designed to teach how a study with human test subjects should be conducted as well inform of the rights of any person taking part in the study. The

course taken for this user study was the “CITI Group 1: Social Behavioral Research Investigators and Key Personnel Group” [53]. The next step in the process is to create a package on IRBNet [45] to submit for approval. The package created for this project includes the documents found in the Appendices. These documents cover the IRB Application Cover Sheet found in Appendix A, IRB Exempt Application found in Appendix B, Consent Form found in Appendix C, and the questionnaires and recruitment email found in Appendices D-F.

The rest of this Chapter will go over the general procedure and design of the user study conducted for this thesis. Section 6.2 details the participants that took part in the user study and explains the recruitment process. Section 6.3 covers the hardware and software tools that were used to conduct the user study. The procedure is outlined on Section 6.4 which encompasses the location where the user study was conducted and each step taken in the process. Section 6.5 presents the tasks that the users were asked to complete to test the LDAT interface. Finally, Section 6.6 covers the measurements taken from the tasks and some of the independent and dependent variables.

## 6.2 Participants

The participants that took part in this study were mostly current or former University of Nevada, Reno students as well as some friends and family. Participants were asked directly if they’d like to participate in the study through the recruitment email as exemplified in Appendix F. There were a total of 20 participants in the study with an age range from early 20s to mid 50s. The background of each participant varied as some had extensive experience with data visualization while others did not have much experience at all with data visualization. This was purposeful to get more varied results from a wide range of users.

## 6.3 Apparatus

The study was conducted on a Windows 10 Enterprise desktop with an Intel i7 64-bit computer processor and 12GB of RAM. There were software tools that helped to conduct the study as well including Docker Desktop for Windows, Visual Studio Code, Google Chrome web browser, and Mousotron [5]. Docker Desktop was used to initialize and run the containers in which the front and back end code was stored while Visual Studio Code was used to start up the containers and run the application in a Google Chrome browser window. Mousotron, pictured in Figure 6.1, was the software used to keep track of participants input statistics for each task including left mouse clicks, right mouse clicks, and mousewheel scrolls. Additionally, due to the COVID-19 pandemic Zoom was used to perform the study virtually in cases where the participant was unable or uncomfortable meeting in person.



Figure 6.1: Screenshot of Mousotron Interface.

## 6.4 Procedure

The procedure for the user study began with the recruitment email asking the participant if they would like to take part in the study. After the participant agreed to be in the study, a meeting time was set for either in person or over zoom. For those conducted in person, the study took place in the William Pennington Engineer Building (WPEB) room 436 located on the UNR campus. Upon starting the study, the participants had the entire study explained to them and were free to leave and no longer be part of the study if they felt uncomfortable with any part of the procedure. If the user agreed to still be part of the study they were asked to sign the consent form in Appendix C before proceeding.

Following the signing of the consent form, the participant was asked to complete a pre-questionnaire as outlined in Appendix D. The pre-questionnaire asked participants some demographic information such as their age, gender, and educational status. In addition, they were also asked their familiarity with some of the topics covered in the study including LiDAR, data visualization, and websites. Upon completion of the questionnaire the participants were then guided to the LDAT web page where they were asked to complete four separate tasks to assess the usability and navigability of the application interface. These tasks included filling out the data explorer form, changing the point cloud size, and changing the point cloud color. The participant was asked to pause in between each task so that their input statistics from Mousotron could be logged. The Mousotron application allows for logging of the statistics which are outputted to a text file making the process quick and efficient.

After all the tasks were completed and the data logged the participants were then asked to fill out a post-questionnaire which can be seen in Appendix D. The post-questionnaire asked the participants general satisfaction with the application and whether or not they felt the application was useful in its current state. The first four questions were rated based on a scale of 1-5 and the last three questions were free response asking participants for thoughts on improvements and/or observations that

could make the application a better experience. It should also be noted that both questionnaires were created using Google Forms to make it easy to complete either in person or virtually over Zoom. The completion of the post-questionnaire signaled the end of the user study. The procedure was very similar for Zoom with the exception that the consent form needed to be signed, scanned, and sent over email as well as the participant needing to take control of the screen through the screen share and remote control feature in Zoom. This was due to the application needing to be run on a local host server that was connected directly to the UNR network. The entire procedure as a whole took about 20-30 minutes to complete for each participant.

## 6.5 Tasks

As stated previously, there were a total of four tasks that each participant was asked to complete during the user study. Table 6.1 outlines the tasks that participants performed within the study. The first task started on the LDAT home page and participants were then asked to navigate to the Data Explorer form and fill out the form and submit. Once the form was submitted, the participant was then guided to the Data Visualization Dashboard for the first time to view near real time chart

Table 6.1: Overview of tasks participants asked to perform in user study

User Study Tasks		
#	Name	Description
1	Fill Out Form	Participants asked to navigate to Data Explorer Form from Homepage to fill out form and submit to view visualizations.
2	Change Sensor Location	Participants asked to change location of the sensor and navigate around the 3D point cloud mesh render.
3	Change Point Size	Participants asked to navigate to Settings page from Dashboard and change point cloud size to a value of 1 and then again to a value of 0.1 to compare visual difference between the two point sizes.
4	Change Color	Participant asked to navigate to Settings page from Dashboard and change the color values for the point cloud environment and detected objects, vehicles and pedestrians, to three distinct colors of their choosing.

and graphs. The participants were allowed additional time if they desired to view and interact with the charts. The second task required the participant to change the location of the sensor. The participant was asked to make a note of the current location being viewed through a label on the top left corner of the dashboard as there were only two active sensors available during the time of testing. The participant would then go to the Data Explorer form for a second time to resubmit the form with the changed sensor location. After submitting and viewing the updated dashboard, participants were then asked to try controlling the 3D point cloud mesh through mouse control input.

This was followed by the third task which asked participants to go to the Settings page from the dashboard and change the point cloud size. Participants were able to change the size through a slider which had max value of 1 and min value of 0.1. Initially, participants were asked to change the points size to a value of 1 and view the dashboard. The participants were then asked to change the points size to a value of 0.1 to assess the visual difference between the two. Finally, the fourth and final task asked participants to once again go to settings but this time to change to point cloud color. The participant was asked to change the three color variables to three distinct colors. Once satisfied with their selection, participants would set the color settings and view the updated 3D mesh. The purpose was to highlight object detection and therefore make it visually easy to see what objects in the scene were being detected as well as distinguish vehicles and pedestrians from the environment.

## 6.6 Design

Measurements were taken during the study as each participant went through each task. There were a total of four measurements taken each of which are listed in Table 6.2. First measurement was the task completion time with each task being timed and logged. Second and third measurements were the total number of left and right mouse clicks respectively. Lastly, the fourth measurement tracked the number of mouse wheel scrolls per task. These measurements were meant to test the ease of



Table 6.2: Measurements taken from tasks

<b>Task Measurements</b>	
#	Measurement
1	Time taken to complete each task.
2	Number of left mouse clicks.
3	Number of right mouse clicks.
4	Number of mouse wheel scrolls.

navigation and general usability of the application.

There are two types of test conditions in user studies as explained by MacKenzie [29] in his book, those being within-subjects and between-subjects. This utilized the within-subjects method meaning that the conditions and measurements were repeated for each participant. In terms of variables, the single independent variable of the study was the LDAT web application interface while the dependent variables of the study covered user input which includes the previously discussed measurements. In addition, the control variables included the hardware and software used as these stayed the same throughout the study.

# Chapter 7

## Results and Analysis

This Chapter details the results obtained during the conduction of the user study. The organization of the chapter is as follows. Section 7.1 briefly presents the data gathered from the pre-questionnaire including backgrounds of participants and their familiarity with the technology involved. Section 7.2 covers the measurements obtained from each task along with an explanation of the results. Section 7.3 provides an overview of the data obtained from the post-questionnaire detailing participant opinion of the application and feedback. Lastly, Section 7.4 goes over the statistical significance of the results obtained.

### 7.1 Data From Pre Questionnaire

The participants in this study ranged mostly between the ages of 21-31 with three participants in their mid 50s. Figure 7.1 shows the range of ages of the participants within this study. A majority of the participants were current or former students at UNR and had either their Bachelor's or Master's degrees while some were in pursuit of their Ph.D. The participants came from a wide range of backgrounds and majors including from computer science and engineering, accounting, biology, political science, mathematics, and statistics. Those who were no longer in school came from different career paths including data engineering, software engineering, hydro-geology, accounting, human resources, and optometry. By including participants from various backgrounds this helped to ensure that there were different perspectives to obtain

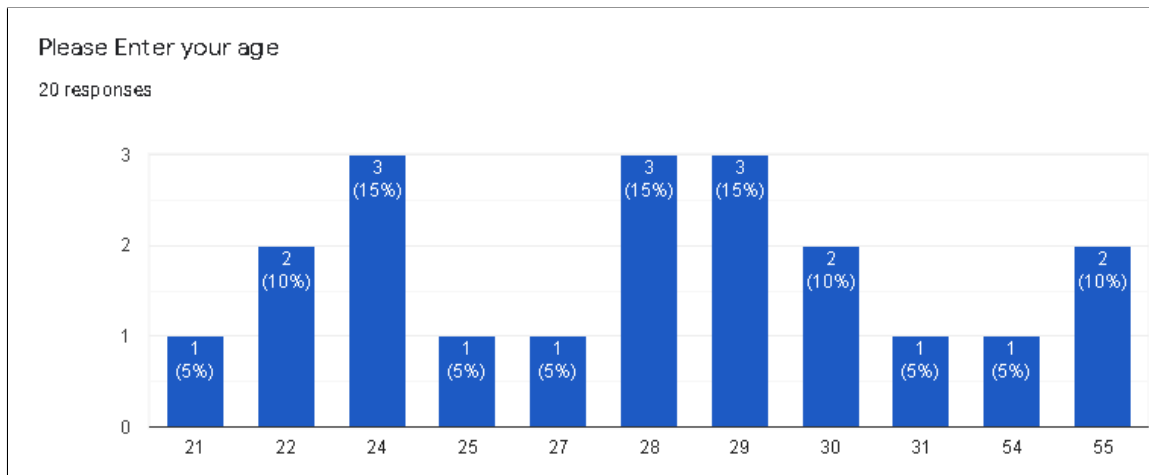


Figure 7.1: Bar chart displaying the range of ages of the participants in the study.

varied and valuable results.

Following the background questions, participants were then asked their familiarity with the specific technical areas used to implement the application being tested. The participants were first asked their familiarity with LiDAR sensor technology with the results being pictured in Figure 7.2. Most participants were unfamiliar with LiDAR sensors as half the participants had no familiarity or understanding while only three expressed some basic familiarity with the technology. This was followed by a question regarding the participants familiarity with point cloud data to which 13

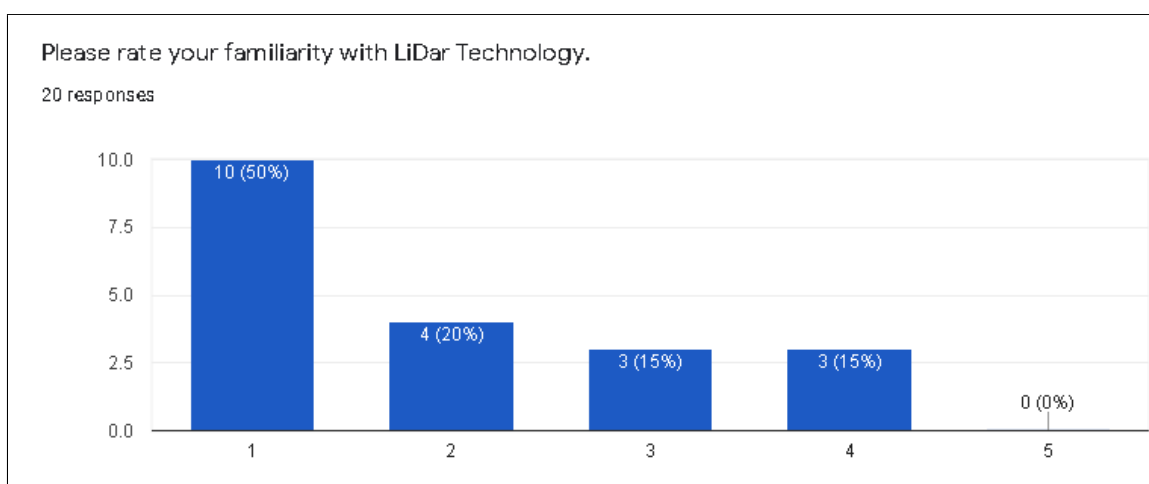


Figure 7.2: Bar chart showing participant response to their familiarity with LiDAR sensor technology.

participants responded not familiar at all. The participants were then asked their familiarity with data visualization and how useful they think it is. Figure 7.3 shows participant response for the frequency in which they use data visualization. About half the participants responded stating that they use data visualization very frequently to frequently while the other half use it occasionally or not at all. Lastly, a majority of participants found data visualization to be a useful tool with approximately 14 participants providing a response of very useful.

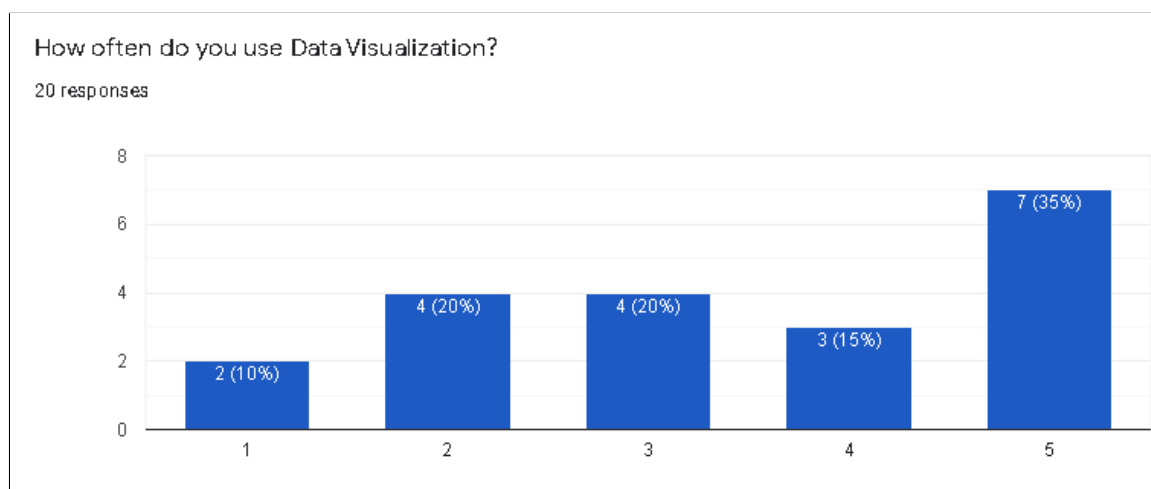


Figure 7.3: Bar chart that displays how often participants use data visualization.

## 7.2 Data Analysis

As discussed, the participant was asked to complete four tasks using the LDAT application interface following the pre-questionnaire. Once these tasks were completed all of the data was aggregated and then placed into box plots which showed the analysis of variance (ANOVA) for each task. Table 7.1 shows each participants task completion time for the four tasks conducted in the study as well as the mean average completion time and standard deviation for each task. The ID number in the table refers to each participant who took part in the study. From here, graphs are created to analyze the collected data. Box plots can be broken down to highlight the minimum, mean, and maximum values found in a data set.

Table 7.1: Task Completion Time for each task

<b>Task Completion Time (in seconds)</b>				
ID	Task 1	Task 2	Task 3	Task 4
1	56	93	101	92
2	51	70	45	47
3	80	42	55	58
4	99	48	74	84
5	114	113	141	144
6	96	89	133	164
7	116	142	138	95
8	90	80	108	160
9	89	129	135	198
10	134	126	170	152
11	126	178	123	221
12	59	68	99	86
13	87	95	96	124
14	137	131	149	141
15	121	158	150	119
16	137	89	100	184
17	151	155	106	112
18	111	132	121	147
19	113	62	100	105
20	68	98	87	92
<b>Mean</b>	101.75	104.9	111.55	126.25
<b>SD</b>	28.41	37.14	31.27	44.74

Figure 7.4 presents the box plot for the task completion time of each task. Box plots are also referred to as Box and Whisker plots with the whiskers represented by the upper and lower lines from the box. The lower whisker shows the minimum result value while the upper whisker shows the maximum result value. The line in the center of each box indicates the average time for the tasks and also separates the upper and lower bound of the data set with the cross representing the mean. At first glance, it is clear that Task 1 had the least amount of variance and therefore was fairly consistent among all the participants had no extreme outliers. Task 2 had slightly more variance with a larger upper bound while Task 3 had outliers mostly on the lower bound. Task 4 had the largest range between its minimum and maximum values when compared to the other tasks.

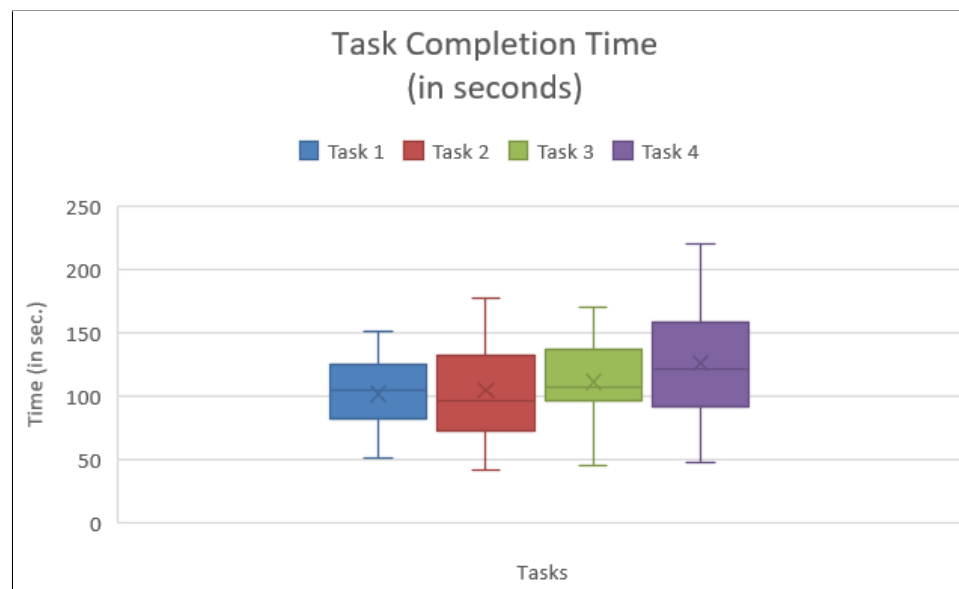


Figure 7.4: ANOVA chart for Task Completion Time. Shows the mean average completion time in seconds and indicates the min and max values for each task.

The following measurement of left mouse clicks per task can be seen in Table 7.2. The mean and standard deviation can be seen in the last two rows of the table with the highest mean count of left mouse clicks coming in Task 2. The box plot pictured in Figure 7.5 shows the variance in the number of clicks for each task. Task 1 had only a few outliers in the upper bound while the lower bound was very small and

Table 7.2: Number of Left Clicks for each task  
**Left Mouse Clicks**

ID	Task 1	Task 2	Task 3	Task 4
1	10	28	9	11
2	18	19	8	12
3	14	13	9	15
4	10	8	8	13
5	9	19	8	16
6	12	14	15	16
7	9	20	11	18
8	11	12	18	44
9	9	11	11	11
10	9	18	10	14
11	9	16	11	30
12	8	19	13	11
13	21	20	7	9
14	16	8	7	10
15	9	16	9	8
16	17	17	7	25
17	22	22	18	12
18	15	21	14	15
19	14	11	6	13
20	10	15	5	11
<b>Mean</b>	12.6	16.35	10.2	15.7
<b>SD</b>	4.19	4.87	3.63	8.27

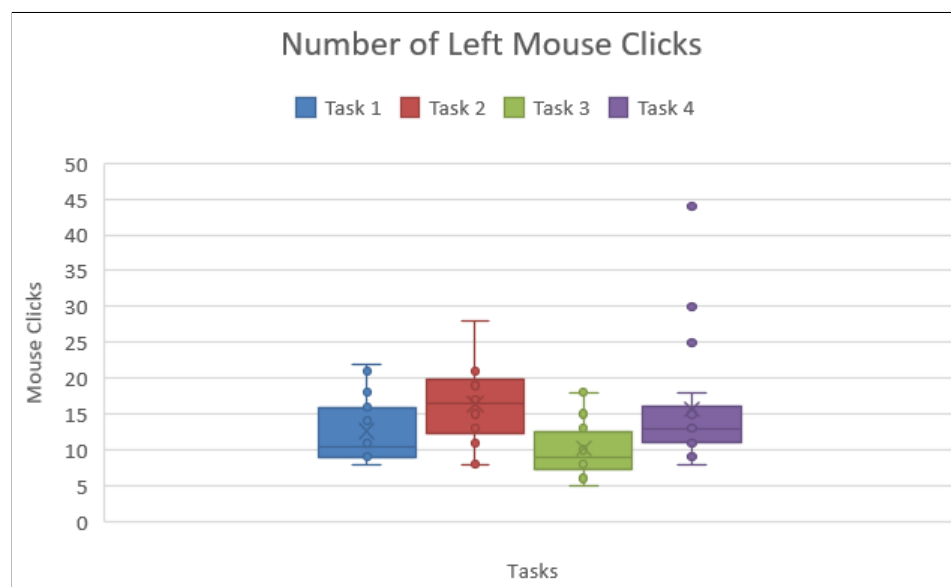


Figure 7.5: Analysis chart for number of left mouse clicks per task.

consistent throughout the study. Task 2 had the most variance out of the four tasks with the largest range between the min and max result. Lastly, Task 3 and 4 both had a similar distribution of results. Task 3 having a slightly higher upper bound, however, Task 4 had the most outliers with three results far surpassing the the normal upper bound.

The resulting number of right clicks per tasks is listed in Table 7.3. The right click button was only used when interacting with the 3D point cloud mesh which therefore resulted in a very low use for most of the tasks. As can be seen from the data, Task 2 had the highest mean average of 2.6 right clicks with Task 4 having the second highest mean average of 1.45 right clicks. Task 1 and 3 both had little to no results for a majority of the study.

Table 7.3: Number of Right Clicks for each task

<b>Right Mouse Clicks</b>				
ID	Task 1	Task 2	Task 3	Task 4
1	3	0	0	0
2	0	5	0	2
3	0	1	3	1
4	0	6	3	4
5	0	6	2	3
6	0	0	0	0
7	0	0	2	0
8	0	0	0	0
9	0	0	0	2
10	0	6	0	0
11	0	5	2	3
12	0	6	3	3
13	0	0	0	0
14	0	0	3	3
15	0	1	0	0
16	0	0	0	0
17	0	7	0	6
18	0	2	0	0
19	0	1	3	2
20	0	6	0	0
<b>Mean</b>	0.15	2.6	1.05	1.45
<b>SD</b>	0.65	2.75	1.32	1.72



The anova results for the number of right clicks is shown in Figure 7.6. It is clear by looking at the chart that participants hardly used the right mouse button during the first task of the study with an average below 1 and only a single outlier of 3 clicks. As stated Task 2 had the most number of right clicks and therefore had the most variance out of all of the tasks. Task 3 and 4 had similar results, however, Task 4 had a higher max result.

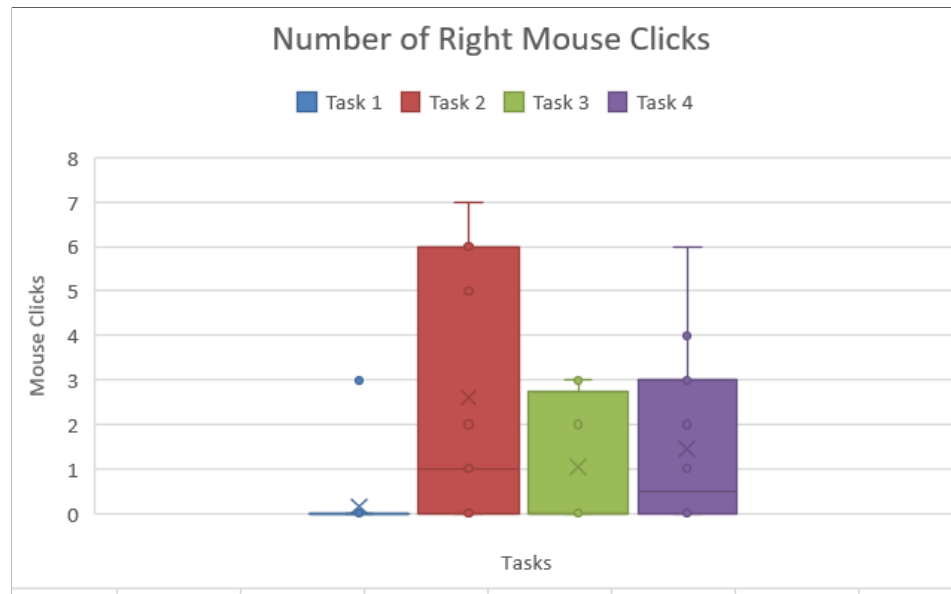


Figure 7.6: Analysis chart for number of right mouse clicks per task.

The last measurement of data analyzed was the number of mouse wheel scrolls participants used for each task. The data gathered for this measurement is shown in Table 7.4. The results here varied greatly with some participants utilizing the mouse wheel and others hardly using at all during the study. This measurement was taken to assess navigation of the web pages as well as navigation of the 3D mesh render. Figure 7.7 displays the mouse wheel scrolls analysis results from each task. As can be seen from the plot, the mean value for each task is fairly consistent throughout. However, there are many outliers most notably in Task 1 as well as a few in Task 4. It should be noted that one participant had zero scrolls for each task in the study. This was due to some technical issues with the participants input device which could not be fixed or changed as the study was being conducted over Zoom. Therefore, this

Table 7.4: Number of Mouse Wheel Scrolls for each task

Mouse Wheel Scrolls				
ID	Task 1	Task 2	Task 3	Task 4
1	0	40	46	71
2	6	25	28	38
3	107	76	117	80
4	60	12	96	124
5	7	13	82	112
6	880	2066	2257	3085
7	0	338	0	0
8	1222	813	206	125
9	21	119	46	152
10	1652	176	367	650
11	1432	744	150	711
12	11	93	110	34
13	64	84	146	23
14	13	53	63	111
15	43	104	66	91
16	45	55	55	108
17	106	161	177	398
18	0	0	0	0
19	86	28	181	78
20	13	23	21	29
<b>Mean</b>	303.58	264.37	221.79	316.84
<b>SD</b>	530.09	480.25	486.83	681.20

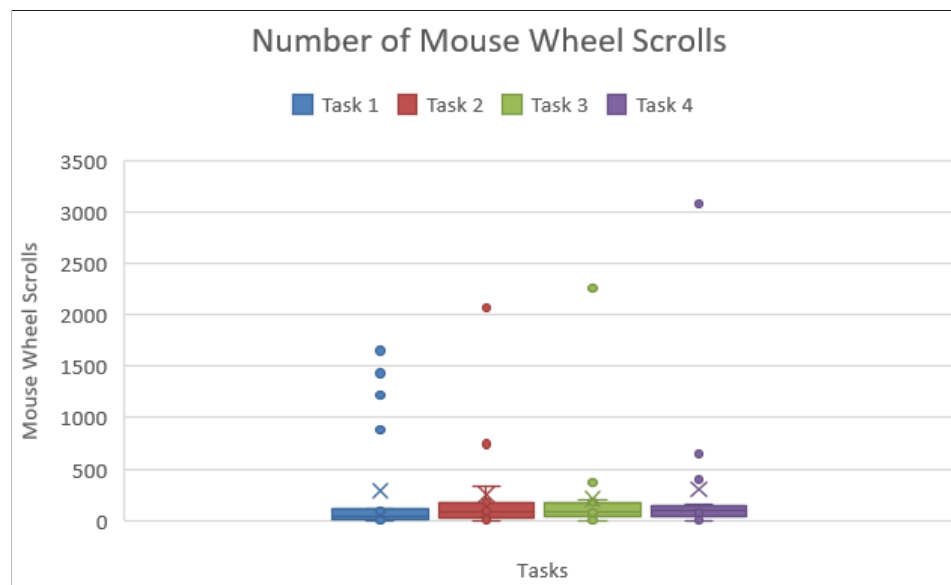


Figure 7.7: Analysis chart for number of mouse wheel scrolls per task.

measurement only includes data from 19 participants.

### 7.3 Data From Post Questionnaire

The post-questionnaire followed the completion of the tasks and was meant to act as a qualitative means to assess the participant satisfaction with the website design and whether the website met its overall design goals. The questionnaire began by asking participants their thoughts on the usefulness of the current website, general satisfaction, and difficulty completing the tasks assigned during the study. Each of these questions was set on a scale from 1-5 with 1 being the lowest possible rating and 5 being the highest.

The first question of the post-questionnaire asked participants how useful they felt the website was in its current state. The results of the responses are shown in Figure 7.8. A majority of the participants found the website fairly useful in its current state with approximately 16 participants responding with a high rating of either a 4 or 5. There were only 4 participants who rated usefulness at either a 3 or 2 meaning they found the website somewhat to not very useful. The weighted average of the responses equaled a 4.15 rating for question 1.

The second question of the post-questionnaire asked participants their overall

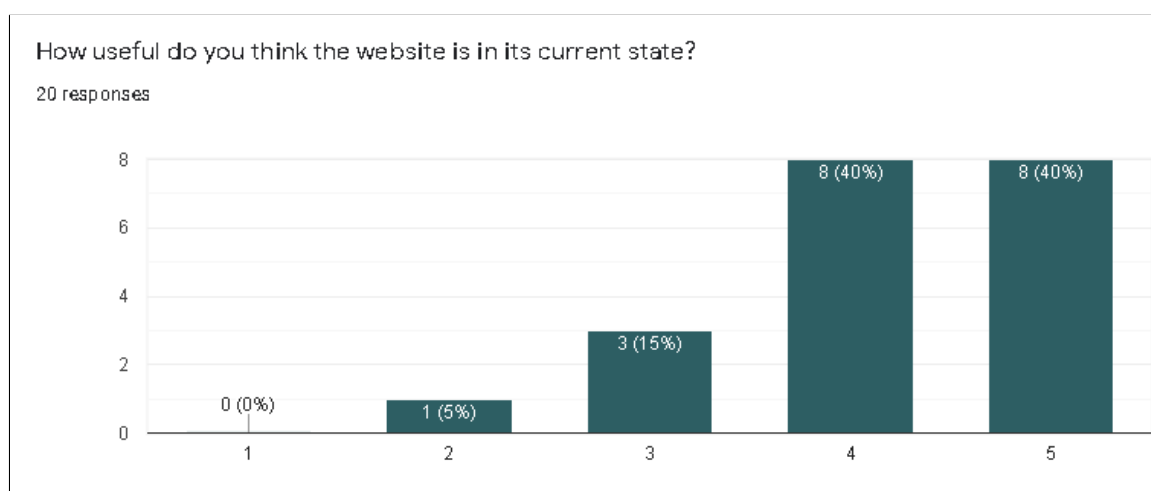


Figure 7.8: Results from Question 1 of the post-questionnaire.

satisfaction with the design of the website. Figure 7.9 displays the graph with the results obtained from the second question responses. The results from this question were mostly positive with 19 participants providing a rating of either 4 or 5 for their overall satisfaction of the website while only one participant rated a 3 for website design. The overall weighted average for question 2 resulted in an average rating of about 4.65.

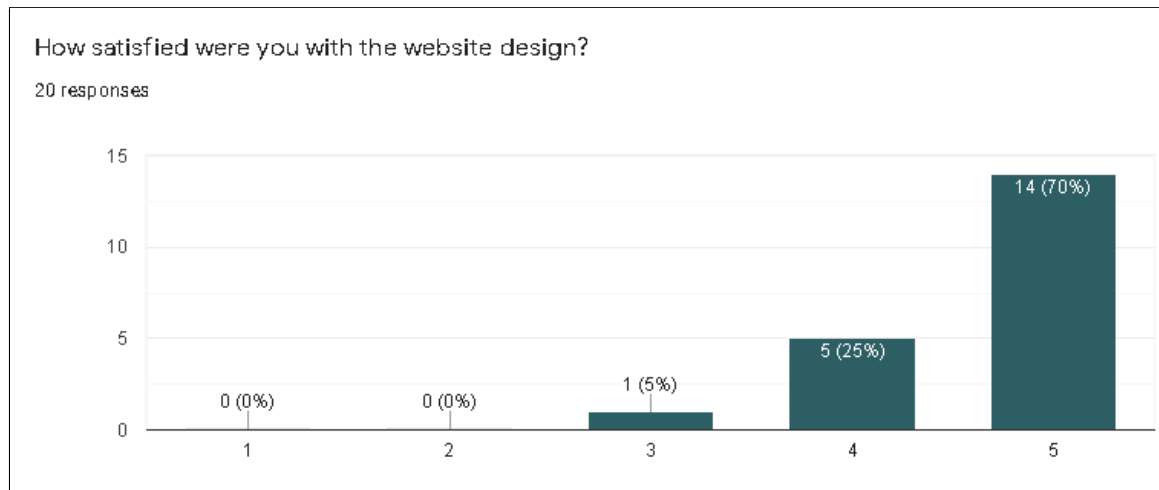


Figure 7.9: Results from Question 2 of the post-questionnaire.

The third question of the post-questionnaire asked participants the general difficulty to complete the four tasks assigned. The results chart from the responses can be seen pictured in Figure 7.10. Most of the participants found the tasks to be either simple or easy with about 15 rating very easy (5) and one rating somewhat easy (4). There were some participants who found the tasks to be difficult with four participants giving a rating on 1 for very difficult. For this question, the weighted average of response was roughly 4.15.

Finally, the fourth question of the post-questionnaire asked the participants to rate the website based on their overall satisfaction with their experience using the website. Figure 7.11 highlights the results acquired from the responses to question 4. These results were predominately positive with all response being either a 4 or 5 thereby suggesting that all the participants had a positive experience using the web

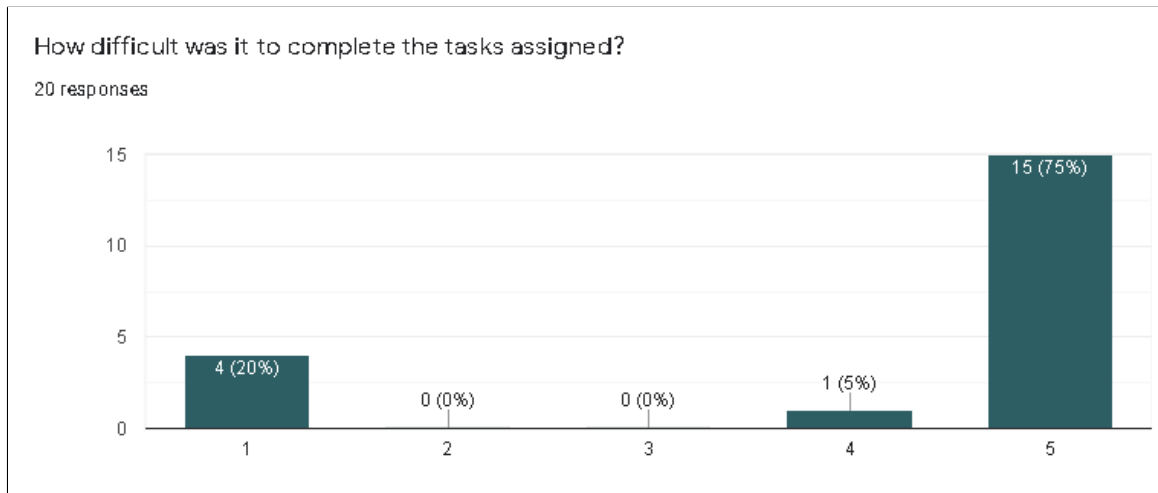


Figure 7.10: Results from Question 3 of the post-questionnaire.

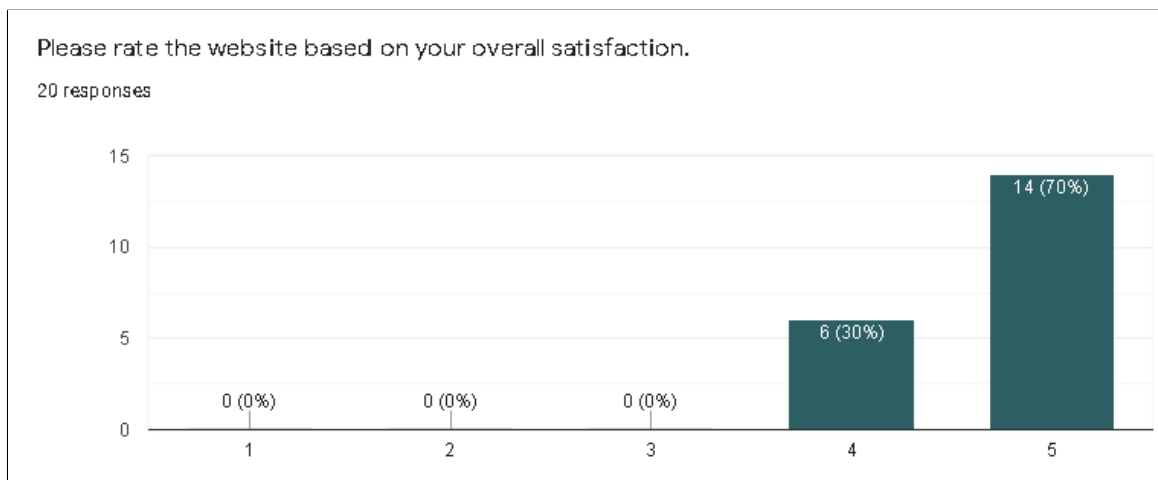


Figure 7.11: Results from Question 4 of the post-questionnaire.

interface. The weighted average for this question ended up resulting in a 4.7 average rating. Table 7.5 lists out the calculated weighted average for each of the ranged questions in the post-questionnaire.

Table 7.5: Weighted average of ratings for post-questionnaire ranged-based questions

<b>Weighted Average of Questions</b>						
Q#	Bad (1)	Somewhat Bad (2)	Neutral (3)	Somewhat Good (4)	Very Good (5)	<b>Weighted Average</b>
1	0	1	3	8	8	4.15
2	0	0	1	5	14	4.65
3	4	0	0	1	15	4.15
4	0	0	0	6	14	4.7

There were three additional questions in the post-questionnaire that followed the range based questions. These questions were short answer questions which asked the participants what improvements they thought could be made, how it could be best tailored to their work, and any other observations or comments. For the first short answer question, many of the users suggested adding in tool tips or some short captions to inform how to control the 3D LiDAR point cloud mesh as some felt the controls were not very intuitive. Another suggested point of improvement was to add on screen arrows to the 3D mesh to control as opposed to using the right click thereby making it clear how to pan the scene. The inclusion of a reference image to see the actual location in addition to the LiDAR sensor stream was also mentioned as a possible improvement. Furthermore, it was suggested to add in edit buttons for the visualizations due to the mouse commands overwriting when hovering over the visualizations along with a message informing users that the graphs are editable.

As for the second question of how it could be tailored to your work, there were a variety of responses. These included being able to export a summary of the data and figures, being able to backtrack through past point cloud captures, and to research traffic patterns and city planning. Some participants did not work in jobs that required data visualization so they responded with 'N/A'. Lastly, the final question asked the participants if they had any additional observations or comments. Most

of the responses made references to the UI and simple and straightforward design. One participant made a comment about input latency or lag from their perspective when interacting with the 3D point cloud mesh over Zoom. While this is a single occurrence, it is a potential affecting factor in the study especially over a virtual setting.

## 7.4 Discussion

From the results, it can be gleaned that most participants found the website to be simple and straightforward in terms of usability. The data analysis of the four tasks performed indicated that participants were fairly consistent in terms of performance and in the way in which they interacted with the graph. Most participants were able to complete the tasks in a certain range of time as can be seen from Table 7.1 and Figure 7.4. There were a few notable outliers, especially on the higher end of the spectrum. This appeared to be directly correlated with the participants interest in the field of data visualization as some asked more questions as they were completing tasks for increased understanding which resulted in a longer amount of time taken to complete the tasks. Additionally, some participants enjoyed viewing and interacting with the visualizations which also caused a lengthier completion time for some of the tasks.

The number of left mouse clicks, results shown in Table 7.2 and Figure 7.5, varied widely over the four tasks with Task 2 and 4 being the highest. This suggests that most participants spent additional time navigating the 3D point cloud mesh as both of these tasks were geared towards participants interacting with the visualizations to some extent. Task 1 and 3 were both designed to test general navigation which was fairly consistent throughout the study. The measurement of right mouse clicks, as seen in Table 7.3 and Figure 7.6, was the least used out of the four measurements taken. A majority of its use was in Task 2 when participants would first interact with 3D point cloud mesh and the researcher asked the participants to try using right click within the scene. This suggests that the right click was not an intuitive means of control,

and therefore, another means may need to be considered in the future. The number of mouse wheel scrolls, pictured in Table 7.4 and Figure 7.7, had a lot of variance over the four tasks. These results were difficult to measure as it was potentially affected based on the participants' input device. The number of scrolls was greatly increased for some participants who were both over Zoom and using the touch pad on their laptop as opposed to a traditional mouse and keyboard. Therefore, it is difficult to truly assess each participant's use of the mouse wheel for each task.

In terms of performance, the LDAT application was pretty consistent throughout user study testing and only had one noticeable occurrence of input latency or lag over Zoom as stated by a participant in the post-questionnaire. A past study [30] tested the effects of lag and how it increases difficulty and in turn usability of an application. Therefore, this is a potential issue that may need to be improved upon to ensure for the best performance. The application was also tested on a few different machines to assess whether or not the performance varied due to any network issues. Table 7.6 shows a list of the devices and connection types used to test the number of frames per second (fps) coming through the data stream. The maximum fps value obtained by the LiDAR sensor is currently set to 10 fps. The devices used to test the performance were a RCDC machine, WPEB lab machine, and laptop. The best performance of about 8-10 fps was obtained at the RCDC machine as this machine was located on the same network as the data stream, therefore, had a direct connection. The WPEB machine had decent performance with a consistent range of 6-8 fps, however, it tied on the lower end with the laptop which connected to the network over a secure VPN which was able to bypass firewalls to get a direct connection to the data stream.

Table 7.6: FPS performance of point cloud data stream measured using three different methods.

<b>Point Cloud FPS Performance</b>		
<b>#</b>	<b>Device and Connection Type</b>	<b>FPS Value</b>
1	RCDC Machine (Direct Connection)	9-10
2	WPEB-436 Lab Machine (UNR Network)	6-8
3	Laptop (WiFi)	6-9



# Chapter 8

## Conclusion and Future Work

### 8.1 Conclusions

The purpose of this thesis was to develop a web application tool called LDAT that could take in data from LiDAR sensors and output data visualizations based on the data stream and user input. The overall goal of the application was to make the LiDAR data easily accessible for researchers and scientists while also providing the data in a format that was both readable and usable to assess traffic patterns at any given time of day. Traffic patterns were assessed through the use of object detection, which currently is able to detect moving object including both vehicles and pedestrians within a scene. The final application included a dashboard which contained three different types of visualizations: a bar chart, line graph, and 3D LiDAR point cloud mesh scene render. Each of these was continuously being updated every 2 seconds to keep up with the near real time data stream. Configuration functionality was also added to the application to ensure each user was able to adapt the visualizations based on their own preferences.

In order to test the application, a user study was developed and conducted that was a comprehensive means to evaluate the usability and general UI design of the application. The study took into account four measurements each of which focused on input and participant performance. The measurements covered completion time, number of left mouse clicks, number of right mouse clicks, and number of mouse wheel scrolls. This was to test both ease of navigation within the current web site

design as well as the core functionality in addition to performance. The user study was designed to be short and simple containing only four principle tasks which were conducted over a 30 minute span.

The results from the user study were analyzed based on each measurement per task. It was found that the participants tended to take longer to complete the tasks depending on if they had an interest and past experience with data visualizations. In turn, other participants were able to complete the tasks fairly quickly which demonstrated that the application was easy to navigate. The number of left mouse clicks varied over the study but indicated participant engagement with the data visualizations as the left mouse click was the primary input to control the 3D point cloud mesh render. The number of right mouse clicks resulted in very little data as many participants rarely used it during the study. It was found that the right click was not an intuitive secondary control input for the 3D point cloud mesh, therefore, was often not used. Lastly, the number of mouse wheel scrolls did not contain any significant findings as it resulted in inaccurate data due to the variance in input devices with the study partly being conducted over Zoom. The main takeaway from the study is that the performance throughout was mostly consistent and the participants were able to easily navigate through the website to test out the different functionality.

## 8.2 Future Work

As this application was a proof of concept, there are a few different additions that have been discussed for any future iterations of this application. First is the addition of statistical analysis to be able to detect and display the speed and trajectory for any and all moving objects within a scene. Next would be the ability to stitch streams from two sensors together to create one large 3D mesh. Currently, the sensors are located on two separate corners of the same intersection and can only be viewed one at a time. As the sensor network expands and grows, it would be nice to be able to select based on a location and view streams from multiple sensors as one single stream, thereby providing more detail and increasing ease of use.

Another addition that has been discussed is the inclusion of a database to store past point cloud captures. This would afford users with the ability to go back and look at past data streams and conduct comparative analysis of the current data. As it stands, the application right now is limited to only the near real time data stream so it would be ideal to add that as an option for the future. Also, in terms of accessibility and security the application would need to include a user account database which would allow users to create an account and sign into the website to gain access to the data. Other potential additions include being able to record and export the data set, tool tips to assist users when using the application, and editing capability for the visualizations to allow users to make adjustments adding in another layer of customization.

# Bibliography

- [1] Andrew Munoz. LiDAR-DataAnalysis-and-Visualization. URL: <https://github.com/amunoz247/LiDAR-DataAnalysis-and-Visualization> (visited on 07/30/2021).
- [2] Angular. What is Angular? URL: <https://angular.io/guide/what-is-angular> (visited on 05/27/2021).
- [3] Damien Arrachequesne. Socket.io:Introduction. URL: <https://socket.io/docs/v4> (visited on 06/03/2021).
- [4] avatseav. three-orbitcontrols-ts. URL: <https://www.npmjs.com/package/@avatsaev/three-orbitcontrols-ts> (visited on 07/31/2021).
- [5] BlackSun Software. Mousotron: Mouse and keyboard usage counter. URL: <http://www.blacksunsoftware.com/mousotron.html> (visited on 08/04/2021).
- [6] Car and Driver. ADAS: Everything you need to know. URL: <https://www.caranddriver.com/research/a31880412/adas/> (visited on 05/13/2021).
- [7] Adam Cassell, Andrew Muñoz, Brianna Blain-Castelli, Nikkolas Irwin, Feng Yan, Sergiu M. Dascalu, and Frederick C. Harris. Cars: a containerized amazon recommender system. In Shahram Latifi, editor, *ITNG 2021 18th International Conference on Information Technology-New Generations*, pages 489–496, Cham. Springer International Publishing, 2021. ISBN: 978-3-030-70416-2. DOI: 10.1007/978-3-030-70416-2\_63.
- [8] D. Deibe, M. Amor, and R. Doallo. Big data storage technologies: a case study for web-based lidar visualization. In *2018 IEEE International Conference on Big Data (Big Data)*, pages 3831–3840, 2018. DOI: 10.1109/BigData.2018.8622589.
- [9] Docker. What is a Container? URL: <https://www.docker.com/resources/what-container> (visited on 06/04/2021).
- [10] Stephanie D.H. Evergreen. *Effective Data Visualization: The Right Chart for the Right Data, 2nd Ed.* SAGE Publications, Inc., Thousand Oaks, California, 2019.
- [11] Flask. Flask: User’s Guide. URL: <https://flask.palletsprojects.com/en/2.0.x/> (visited on 05/27/2021).

- [12] C. Galle, J. Amelung, T. Dallmann, and S. Brueggenwirth. Vehicle environment recognition for safe autonomous driving: research focus on solid-state lidar and radar. In *AmE 2020 - Automotive meets Electronics; 11th GMM-Symposium*, pages 1–3, 2020. URL: <https://ieeexplore.ieee.org/document/9094573>.
- [13] Zhenzhen Gao, Luciano Nocera, and Ulrich Neumann. Visually-complete aerial lidar point cloud rendering. In (SIGSPATIAL '12), 289–298, Redondo Beach, California. Association for Computing Machinery, New York, USA, 2012. ISBN: 9781450316910. DOI: 10.1145/2424321.2424359.
- [14] Google. Angular - Singleton services. URL: <https://angular.io/guide/singleton-services> (visited on 07/30/2021).
- [15] Google. N Virginia St I& East 15th Street. URL: <https://www.google.com/maps/place/N+Virginia+St+%26+East+15th+Street,+Reno,+NV+89503/@39.5438691,-119.8207192,17z/data=!3m1!4b1!4m5!3m4!1s0x8099473df9c6a305:0x7d573f3a1dc1205d!8m2!3d39.5438691!4d-119.8185305> (visited on 07/30/2021).
- [16] Rob Harrap and Matt Lato. An overview of lidar: collection to application. *NGI publication*, 2:1–9, 2010. URL: [https://www.academia.edu/1360215/An\\_Overview\\_of\\_LIDAR\\_collection\\_to\\_applications](https://www.academia.edu/1360215/An_Overview_of_LIDAR_collection_to_applications).
- [17] Humans of Data. 4 Data Collection Techniques: Which One’s Right for You? URL: <https://humansofdata.atlan.com/2017/08/4-data-collection-techniques-ones-right/> (visited on 05/22/2021).
- [18] import.io. What is Data Visualization and Why Is It Important? URL: <https://www.import.io/post/what-is-data-visualization/> (visited on 04/21/2021).
- [19] John Pike. AN/GVS-5 Laser Rangefinder. URL: <https://fas.org/man/dod-101/sys/land/an-gvs-5.htm> (visited on 05/13/2021).
- [20] Khronos.org. Getting Started: What is WebGL? URL: [https://www.khronos.org/webgl/wiki/Getting\\_Started](https://www.khronos.org/webgl/wiki/Getting_Started) (visited on 06/05/2021).
- [21] Yuichiro Kinoshita, Satoshi Tsukanaka, and Kentaro Go. Strolling with street atmosphere visualization: development of a tourist support system. In *CHI '13 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '13, 553–558, Paris, France. Association for Computing Machinery, New York, USA, 2013. ISBN: 9781450319522. DOI: 10.1145/2468356.2468454.
- [22] Cole Nussbaumer Knaflic. *Storytelling With Data: A Data Visualization Guide for Business Professionals, 1st Ed.* John Wiley & Sons, Inc., Hoboken, New Jersey, 2015.
- [23] Shrinidhi Kulkarni, Manjunath Chandrashekaraiyah, and Raghunandan S. 3d annotation tool using lidar. In *2019 Global Conference for Advancement in Technology (GCAT)*, pages 1–4, 2019. DOI: 10.1109/GCAT47503.2019.8978301.

- [24] Matthew J. Lato. *Geotechnical Applications of LiDAR Pertaining to Geomechanical Evaluation and Hazard Identification*. PhD thesis, 2010. URL: <https://qspace.library.queensu.ca/handle/1974/5451>.
- [25] Paul Lewis, Conor P. Mc Elhinney, and Timothy McCarthy. Lidar data management pipeline; from spatial database population to web-application visualization. In *Proceedings of the 3rd International Conference on Computing for Geospatial Research and Applications*, COM.Geo '12, Washington, D.C., USA. Association for Computing Machinery, New York, USA, 2012. ISBN: 978-1-45-0311137. DOI: 10.1145/2345316.2345336.
- [26] Ning Li, Tao Luo, and Bo Su. Fast extrinsic calibration for 3d lidar. In *Proceedings of the 2019 4th International Conference on Automation, Control and Robotics Engineering*, CACRE2019, Shenzhen, China. Association for Computing Machinery, New York, USA, 2019. ISBN: 9781450371865. DOI: 10.1145/3351917.3351986.
- [27] Weiping Liu, Jia Sun, Wanyi Li, Ting Hu, and Peng Wang. Deep learning on point clouds and its application: a survey. *Sensors*, 19(19), 2019. ISSN: 1424-8220. DOI: 10.3390/s19194188.
- [28] Bharat Lohani and Suddhasheel Ghosh. Airborne lidar technology: a review of data collection and processing systems. In *Proceedings of the National Academy of Sciences, India Section A: Physical Sciences*, volume 87, pages 567–579, 2017. DOI: 10.1007/s40010-017-0435-9.
- [29] I. Scott MacKenzie. *Human Computer Interaction: An Emphirical Research Perspective*. Elsevier, Inc., 2013.
- [30] I. Scott MacKenzie and Colin Ware. Lag as a determinant of human performance in interactive systems. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, CHI '93, 488–493, Amsterdam, The Netherlands. Association for Computing Machinery, New York, USA, 1993. ISBN: 0897915755. DOI: 10.1145/169059.169431.
- [31] E. Maravelakis, A. Konstantaras, K. Kabassi, I. Chrysakis, C. Georgis, and A. Axaridou. 3dsystek web-based point cloud viewer. In *IISA 2014, The 5th International Conference on Information, Intelligence, Systems and Applications*, pages 262–266, 2014. DOI: 10.1109/IISA.2014.6878726.
- [32] Paul McManamon, Gary Kamerman, and Milton Huffaker. A history of laser radar in the united states. *Proceedings of SPIE - The International Society for Optical Engineering*, April 2010. DOI: 10.1117/12.862562.
- [33] Yiming Miao, Yuan Tang, Bander A. Alzahrani, Ahmed Barnawi, Tarik Alafif, and Long Hu. Airborne lidar assisted obstacle recognition and intrusion detection towards unmanned aerial vehicle: architecture, modeling and evaluation. *IEEE Transactions on Intelligent Transportation Systems*:1–10, 2020. DOI: 10.1109/TITS.2020.3023189.

- [34] Karl Mohring, Trina Myers, and Ian Atkinson. A controlled trial of a commodity sensors for a streetlight-mounted traffic detection system. In *Proceedings of the Australasian Computer Science Week Multiconference, ACSW '18*, Brisbane, Queensland, Australia. Association for Computing Machinery, New York, USA, 2018. ISBN: 9781450354363. DOI: 10.1145/3167918.3167922.
- [35] Mozilla. JavaScript. URL: <https://developer.mozilla.org/en-US/docs/Web/JavaScript> (visited on 06/05/2021).
- [36] MQTT. MQTT: The Standard for IoT Messaging. URL: <https://mqtt.org/> (visited on 06/03/2021).
- [37] Shubham Nagla. 2d hector slam of indoor mobile robot using 2d lidar. In *2020 International Conference on Power, Energy, Control and Transmission Systems (ICPECTS)*, pages 1–4, 2020. DOI: 10.1109/ICPECTS49113.2020.9336995.
- [38] npmjs.com. ngx-charts. URL: <https://www.npmjs.com/package/@swimlane/ngx-charts> (visited on 07/30/2021).
- [39] Pandas. pandas - Python Data Analysis Library. URL: <https://pandas.pydata.org/> (visited on 07/20/2021).
- [40] ParaView. LidarView: The ParaView Lidar app. URL: <https://www.paraview.org/lidarview/> (visited on 05/26/2021).
- [41] Point Cloud Library. Point Cloud Library - About. URL: <https://pointclouds.org/about/> (visited on 04/05/2021).
- [42] Point Cloud Library. The PCD (Point Cloud Data) file format. URL: [https://pointclouds.org/documentation/tutorials/pcd\\_file\\_format.html](https://pointclouds.org/documentation/tutorials/pcd_file_format.html) (visited on 04/06/2021).
- [43] Python Software Foundation. threading - Thread based Parallelism. URL: <https://docs.python.org/3/library/threading.html> (visited on 07/20/2021).
- [44] Python Software Foundation. Welcome to Python.org. URL: <https://www.python.org/> (visited on 06/05/2021).
- [45] Research Dataware, LLC. IRBNet: Innovative Solutions for Compliance and Research Management. URL: <https://irbnet.org/release/index.html> (visited on 07/26/2021).
- [46] Cristina Romero-González, Álvaro Villena, Daniel González-Medina, J. Martínez-Gómez, Luis Rodríguez, and I. García-Varea. Inlida: a 3d lidar dataset for people detection and tracking in indoor environments. In *VISIGRAPP*, 2017. DOI: 10.5220/0006148704840491.
- [47] Radu Bogdan Rusu and Steve Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011. DOI: 10.1109/ICRA.2011.5980567.
- [48] Sergey Dryabzhinsky. python-zstd. URL: <https://github.com/sergey-dryabzhinsky/python-zstd> (visited on 07/20/2021).

- [49] J. Sochor, R. Juránek, J. Špaňhel, L. Maršík, A. Široký, A. Herout, and P. Zemčík. Comprehensive data set for automatic single camera visual speed measurement. *IEEE Transactions on Intelligent Transportation Systems*, 20(5):1633–1643, 2019. DOI: 10.1109/TITS.2018.2825609.
- [50] Ian Sommerville. *Software Engineering, 10th Edition*. Pearson Education, 2015.
- [51] Tesla. Autopilot - Future of Driving. URL: <https://www.tesla.com/autopilot> (visited on 2021).
- [52] Three.js. Three.js Fundamentals. URL: <https://threejsfundamentals.org/threejs/lessons/threejs-fundamentals.html> (visited on 06/05/2021).
- [53] University of Nevada, Reno. Research Integrity - Training. URL: <https://www.unr.edu/research-integrity/training> (visited on 07/26/2021).
- [54] Velodyne Lidar. Ultra Puck. URL: <https://velodynelidar.com/products/ultra-puck/> (visited on 07/16/2021).
- [55] Visual Paradigm. What is Use Case Diagram? URL: <https://www.visual-paradigm.com/guide/uml-unified-modeling-language/what-is-use-case-diagram/> (visited on 05/16/2021).
- [56] Xin Wang, HuaZhi Pan, Kai Guo, Xinli Yang, and Sheng Luo. The evolution of LiDAR and its application in high precision measurement. *IOP Conference Series: Earth and Environmental Science*, 502:012008, June 2020. DOI: 10.1088/1755-1315/502/1/012008.
- [57] Aloysius Wehr and Uwe Lohr. Airborne laser scanning—an introduction and overview. *ISPRS Journal of Photogrammetry and Remote Sensing*, 54(2):68–82, 1999. ISSN: 0924-2716. DOI: 10.1016/S0924-2716(99)00011-8. URL: <https://www.sciencedirect.com/science/article/pii/S0924271699000118>.



## Appendix A

# UNR IRB Application Cover Sheet

University of Nevada, Reno  
Institutional Review Board  
**Part I, Cover Sheet**

**Last edited by:** Andrew Munoz

**Last edited on:** April 28, 2021

[\[click for checklist\]](#)

[1756566-1] Near Real-time Visualization of LiDar Data

Answer all questions on this form completely, include attachments and obtain signatures of Principal Investigator and the Responsible official prior to final submission on IRBNet.

### I. Principal Investigator

**Name:** Sergiu Dascalu, PhD  
**Institution:** University of Nevada, Reno  
**Department:** College of Engineering  
**Telephone:** (775) 636-5060 **Email:** dascalus@nevada.unr.edu  
**Address:** 1664 N Virginia St MS-171 Reno, NV 89557-0171, USA

### II. Contact Person

N/A

**Name:** Andrew Munoz  
**Telephone:** (209) 402-1583 **Email:** amunoz24@nevada.unr.edu

### III. Co-Investigator(s) Including Student Investigator(s)

N/A

**Name:** Andrew Munoz, BS  
**Institution:**  
**Telephone:** (209) 402-1583 **Email:** amunoz24@nevada.unr.edu

### IV. Research Assistant(s)/Coordinator(s)

N/A

**Name:**  
**Telephone:** **Email:**

### V. Responsible Official

**Name:** Eelke Folmer  
**Title:** Professor and Chair  
**Institution:** University of Nevada, Reno  
**Department:** College of Engineering  
**Telephone:** (775) 784-7592 **Email:** efolmer@unr.edu

## VI. Project Information

### Research Type:

- Biomedical
- Social Behavioral/Educational

### Research Location:

- VA Sierra Nevada Healthcare System
- Saint Mary's Regional Medical Center
- Renown Health
- UNR Campus
- Other -

### Renown Health Research Locations:

- |   |   |
|---|---|
| <input type="checkbox"/> Renown Regional          | <input type="checkbox"/> Renown Pharmacy        |
| <input type="checkbox"/> Renown South Meadows     | <input type="checkbox"/> Renown Emergency Room  |
| <input type="checkbox"/> Renown Pregnancy Center  | <input type="checkbox"/> Renown Skilled Nursing |
| <input type="checkbox"/> Renown Outpatient Clinic | <input type="checkbox"/> Renown Hospice Care    |
| <input type="checkbox"/> Renown Urgent Care       | <input type="checkbox"/> Renown Home Health     |
| <input type="checkbox"/> Renown Imaging           | <input type="checkbox"/> Renown Rehabilitation  |
| <input type="checkbox"/> Renown Lab               |   |

### Requested Review Path:

- Requesting a determination about whether a project is human subject research  
*Complete Request for Human Research Determination*
- Requesting authorization to use an external IRB  
*Complete Request to Use an External IRB*
- Reporting emergency use of an FDA-regulated drug or device  
*Complete Emergency Use of a Test Article*
- Minimal risk exempt research involving contact with participants  
*Complete Exempt Core Application Research with Participants*
- Minimal risk exempt research involving existing records or specimens  
*Complete Exempt Core Application Review of Existing Records or Specimens*
- Exempt, IRB Flex: Minimal Risk, Non Federally Funded Research  
*Complete Exempt, IRB Flex: Minimal Risk, Non Federally Funded Research*
- Exempt, IRB Flex: Ethnographic and Naturalistic Field Research  
*Complete Exempt, IRB Flex: Ethnographic and Naturalistic Field Research*
- Expedited IRB Review  
*Complete Part II for Social/Educational Research or Part II for Biomedical Research*
- Full Board Review  
*Complete Part II for Social/Educational Research or Part II for Biomedical Research*

### Risk Level:

- Minimal risk

- Greater than minimal risk (requires full board review)
- No known risk

**Involvement of Vulnerable Populations:**

- N/A, research will not involve vulnerable populations
- Pregnant women and fetuses  
*Complete Population: Pregnant Women and Fetuses*
- Prisoners  
*Complete Population: Prisoners*
- Children (persons under 18 years of age)  
*Complete Population: Children*
- Adults with impaired decision-making capacity  
*Complete Population: Impaired Decision-Making Capacity*
- Persons Living in a Foreign Country  
*Complete International Research*

**VII. Funding Information**

N/A

**Sponsor Type:**

- |   |   |
|---|---|
| <input type="checkbox"/> Federal Government | <input type="checkbox"/> Other Government (State/Local) |
| <input type="checkbox"/> Industry Sponsor   | <input type="checkbox"/> Other Private Funds            |
| <input type="checkbox"/> Departmental       | <input type="checkbox"/> Subcontract                    |
| <input type="checkbox"/> Other:             |   |

**Sponsor Name:**

**Grant/Contract Title and Number:**

**VIII. Researcher Conflict of Interest**

**Significant Financial Interests Related to this Research:**

- Yes
- No

**Significant Financial Interest or Performance Commitment:**

- Yes
- No

**IX. Federal Agencies with Additional Requirements to Protect Human Participants**

- DoD  
*Complete Research Involving Department of Defense*
- DoE  
*Complete Research Involving Department of Energy*
- DoEd  
*Complete Educational Records and Classroom Survey Research (FERPA and PPRA)*
- DoJ or NIJ

*Complete Research Involving Department of Justice or National Institute of Justice*

- EPA  
*Complete Research Involving the Environmental Protection Agency*
- NSF  
*Additional training requirement only; no supplemental form. See our website for training details.*
- VA  
*Complete forms with "VA" in the title.*
- N/A

#### X. FDA-Regulated Research (Clinical Trials Involving Investigational Drugs or Devices)

- N/A, research does not involve drugs or devices
- Drug research

Trade Name

Generic Name

- Device research

Name of Device

Device Manufacturer

#### XI. External Committee Approvals

- Thesis Committee
- Radiation Safety Committee
- Biosafety Committee
- Other:
- N/A

#### INSTRUCTIONS TO RESEARCHERS

[\[top\]](#)

You have completed Part I of the application process. **Preview** Part I and correct if needed. Print the last page so you have the list of the researcher forms required for this research. Click **Save and Exit**. Log-in to IRBNet, go to My Projects, select the new project, and **Add** the remaining required documents (listed below or referenced in the researcher forms/applications), and electronically **Sign** and **Submit** the project.

If you have any questions, refer to the [IRBNet pages of the RI website](#).

#### Additional required researcher forms:

- Complete *Exempt, IRB Flex: Minimal Risk, Non Federally Funded Research*

## Appendix B

### UNR IRB Exempt Application



## Exempt, IRB-Flex: Minimal Risk Research with No Federal Funding/Support

<p><b>A <a href="#">University of Nevada, Reno Part I, Cover Sheet</a> is required for each project submitted in IRBNet.</b></p>	
<p><b>Study Purpose and Narrative</b></p>	
<p>1. Describe the study background, significance, and purpose/hypothesis/research questions.</p>	<p>LiDar is a term that stands for Light Detection and Ranging and is used primarily to quickly measure the distance between objects or in other words detect the position of objects in a real space. This is accomplished through the use of sensors that contain many lasers embedded within them. The laser beams are emitted from a mounted sensor which receives a return signal from the laser once an object is detected providing accurate position data over large areas and long distances.</p> <p>Data visualization is an important tool in data analysis as it allows for quick insight to the reader through the form of a graph or chart. The project that this study is based upon is a web based data visualization tool that is streaming near-real time street traffic data through LiDar sensors. Through a user study, the participant is given tasks to perform on the web site which will help assess the usability and design flow of the user interface. Quantitative analysis will include data relating to left mouse clicks and task completion time. Qualitative analysis involves the usability of the website and user satisfaction which will be assessed through pre and post questionnaires.</p> <p>The main goal for this study is to determine the overall functionality of the website, improvements that could be made, and overall user satisfaction.</p> <p>Related Work can be found in the following: M. Kuder and B. Zalik. Web-based lidar visualization with point-based rendering. In 2011 Seventh International Conference on Signal Image Technology Internet-Based Systems, pages 38–45, 2011. doi:10.1109/SITIS.2011.36.</p>
<p>2. Provide a detailed narrative, non-technical description of the project and of the planned activity for the participants and/or data collection.</p>	<p>Data visualization tools are essential to be able to output data in a format that is readable which can provides readers a way to easily discover patterns and trends. This study focuses on the design and usability of a web based data visualization tool which utilizes LiDar point cloud data. The study</p>

	<p>begins when the participant arrives at the campus lab where the study will be held. The whole process of the study will be described to the participant at which time they can decide to leave at any point if they feel uncomfortable with the testing. Once the process of the study is explained the participant will take a pre-test questionnaire that will ask their age, gender, education level, current enrollment status, and their familiarity with data visualization tools, websites, as well as LiDar technology.</p> <p>Following the pre-test questionnaire, the participant will be given a quick demo overview of the website. The participant will then be given control of the computer to access the website. The participant will be asked to complete between 3 and 5 tasks pertaining to data visualization (e.g. selection, zoom in/out, filtering, customization).</p> <p>The study will conclude with a post-test questionnaire that will gather qualitative data by asking users their general thoughts on the website, improvements that could be made, and their personal rating on a Likert scale, likely between 1-5 (could also be 1-10). Quantitative data related to user performance will be collected which the participant will be informed of at the beginning of the study. These data points will include number of mouse clicks, number of key presses, and task completion time.</p>
<b>Study Sites</b>	
3. List or describe the study sites.	<input type="checkbox"/> N/A, Internet or telephone survey Study sites: University of Nevada, Reno in the William Pennington Engineering Building Lab Room 436; Also remotely by Zoom
4. Is approval from an external entity required for this research (e.g., TMCC, school district, oversight committee or board)? Add documentation of approval, or submit via subsequent amendment package if site requires IRB approval. IRB program staff will coordinate WCSD Accountability Office approval.	<input checked="" type="checkbox"/> No <input type="checkbox"/> Yes, specify:
<b>Study Populations</b>	
5. How many participants or records will the project involve?	12-25 participants
6. Describe the study populations or the sources of	The participants for this project will mostly be



the records for this project.	University students who are between 18-35 years of age from the Reno area.
7. Will any participants be younger than 18 years old?	<input checked="" type="checkbox"/> No <input type="checkbox"/> Yes, explain why it's necessary to include children:
8. Will this project involve pregnant women, fetuses, or neonates?	<input checked="" type="checkbox"/> No <input type="checkbox"/> Yes, explain why it's necessary to include pregnant women/fetuses:
<b>Recruitment and Informed Consent</b>	
9. Will prospective participants be invited to consider participation in the research? Add recruitment scripts/flyers.	<input type="checkbox"/> N/A, recruitment combined with consent <input type="checkbox"/> N/A, records review or covert observation only <input type="checkbox"/> No, explain why not: <input checked="" type="checkbox"/> Yes, describe the invitation process: Participants will be gathered through word of mouth and email contact. During this process a brief description of the experiment will be given. Once the participant agrees to be in the study, a full overview of the study will be provided.
10. Will potential participants be told about the research and be given the opportunity to agree to participate? Add information scripts, sheets, or letters.	<input type="checkbox"/> N/A, records review or covert observation only <input type="checkbox"/> No, explain why not: <input checked="" type="checkbox"/> Yes, describe when, where, and how: At the time when the researcher approaches the participant, the researcher will verbally explain the study to the participant. The participant will be invited to enter the lab if the participant is still interested in taking part in the experiment. Once in the lab, the participant will be given a consent form to fill which includes all the necessary details regarding the study. Next, the participants will be allowed a quick overview of the website. After this, the participant will be asked again if they want to continue with the study. At any point of time, even during the experiment the participant is free to opt out of the study. At the end of the study the participant will be given a copy of the consent form with our contact information. If this study will be administered remotely, the same steps as above will be applied.
11. Will incentives be provided to enhance enrollment?	<input checked="" type="checkbox"/> No <input type="checkbox"/> Yes, specify the incentives (include dollar value and odds of winning):
12. When, how, and by whom will incentives be distributed?	<input checked="" type="checkbox"/> N/A, no incentives <input type="checkbox"/> N/A, using SONA; standard procedures apply Explanation:

Research Activities/Study Procedures	
13. Indicate the types of research activities that will be used.	Select all that apply: <input type="checkbox"/> Tests (cognitive, diagnostic, aptitude) <input type="checkbox"/> Survey, paper/pencil (including via mail) <input type="checkbox"/> Survey, telephone <input checked="" type="checkbox"/> Survey, Internet <input type="checkbox"/> Interviews, including focus group interviews <input checked="" type="checkbox"/> Assessment of participant performance of simple tasks <input checked="" type="checkbox"/> Computer simulations <input type="checkbox"/> Training simulations <input type="checkbox"/> Non-invasive physical assessments (e.g., eye tracking, body fat assessment using calipers, obtaining blood via heel or finger stick, cheek swab, assessment of physical activity) <input type="checkbox"/> Review of existing records NOTE: another exempt category may apply <input type="checkbox"/> Review of records to be obtained prospectively <input type="checkbox"/> Other, describe:
14. What is the date range for selecting records for this project (e.g., month/year of first and last records obtained)?	<input checked="" type="checkbox"/> N/A, project does not involve existing records Explanation:
15. How will the researchers obtain information from existing records?	<input checked="" type="checkbox"/> N/A, information will not be obtained from existing records Description:
16. Describe in detail and in chronological order what participants will be asked to do:	<input type="checkbox"/> N/A, records review only Description: Once the participant agrees to take part in the study, they will be told in detail what the study entails. The participant will then be given a pre questionnaire which asks participants about their demographics, familiarity with LiDar, data visualizations, and websites. This is then followed by a brief demo where the participant is given an overview of the website. Once this is complete, participants are then verbally instructed the tasks that need to be completed one by one. The tasks are related to data visualizations and will consist of typical actions such as selection, zoom in/out, filtering, and customization.  Once each of the tasks are completed, the participant will then be given a post questionnaire which will serve as qualitative data.
17. Will participants be video-recorded or photographed? Add the video/photo release form participants	<input checked="" type="checkbox"/> No <input type="checkbox"/> Yes, explain why videos or photos are necessary for the research:

will sign.	
18. How much time will a participant spend in the study and how long will her/his participation last?	<input type="checkbox"/> N/A, records review only <input type="checkbox"/> N/A, single contact <input type="checkbox"/> Number of contacts <input type="checkbox"/> 30 minutes <input type="checkbox"/> Time spent participating in minutes/hours <input type="checkbox"/> Length of time/duration in weeks/months
19. List all study materials/research instruments (e.g., surveys, questionnaires, interview guides, data collection logs) and equipment. Add study materials and equipment manuals.	Questionnaires, data collection logs, desktop lab computers, laptops, tablets
20. List the data that will be collected for this study:	Data that will act as quantitative data for this study include:  Number of left mouse clicks  Number of key presses  Task completion time
21. How will the data be analyzed?	Using standard ANOVA and Friedman tests for analyzing data
<b>Participant Privacy and Data Confidentiality</b>	
22. Will participants' Protected Personally Identifiable Information (PII) (see <a href="#">Policy Manual Definitions</a> ) be recorded?	<input checked="" type="checkbox"/> No <input type="checkbox"/> Yes, list the PII that will be recorded and explain why it's necessary to record PII:
23. How will participants' privacy rights be protected?	We are recruiting participants' individually and not keeping any record of the contact. The only record kept of a person's participation in the experiment is the questionnaire responses. No other records will be kept.
24. How will research records be handled and stored to ensure confidentiality?	The participants will be given a randomized ID that will have no identifiable information. All collected data will be stored under the ID numbers and saved on a computer that requires specific login information. In addition, the computer is located in a lab that requires key card access.
25. If a master list will be used to link participant research IDs with PII, when will the code list be destroyed?	<input checked="" type="checkbox"/> N/A, PII not recorded or not linked Specify:
<b>HIPAA Requirements</b>	
26. Will Protected Health Information (PHI) (see <a href="#">Policy Manual Definitions</a> ) be obtained or recorded for this research?	<input checked="" type="checkbox"/> No, <b>SKIP</b> remaining questions <input type="checkbox"/> Yes, <b>Requires</b> signed HIPAA authorization from participant or IRB approval of waiver of HIPAA authorization (HIPAA templates are in researcher library in IRBNet)
27. Will the PHI be provided as a Limited Data Set	<input type="checkbox"/> No

## Appendix C

# UNR IRB Consent Information Sheet

**University of Nevada, Reno**  
**Social Behavioral or Educational Research Consent Form**

**Title of Study:** Near Real-time Visualization of LiDar Data  
**Principal Investigator:** Sergiu Dascalu, PhD  
**Co-Investigators /** Andrew E Munoz  
**Study Contact:** Andrew E Munoz  
**Study ID Number:** 1756566-1  
**Sponsor:** N/A

---

**SUMMARY OF KEY ELEMENTS:**

**Introduction**

You are being invited to participate in a research study. Before you agree to be in the study, read this form carefully. It explains why we are doing the study; and the procedures, risks, discomforts, benefits and precautions involved.

At any time, you may ask one of the researchers to explain anything about the study that you do not understand.

You do not have to be in this study. Your participation is voluntary. If you do not agree to participate, you will receive the **care/education** you would have received if the study was not taking place.

Take as much time as you need to decide. If you agree now but change your mind, you may quit the study at any time. Just let one of the researchers know you do not want to continue.

**Why are we doing this study?**

We are doing this study to examine the usability, efficiency, and design of a web based data visualization user interface.

**Why are we asking you to be in this study?**

We are asking you to be in this study because you are an adult who can speak English.

**How many people will be in this study?**

We expect to enroll 12-25 participants at University of Nevada, Reno.

**What will you be asked to do if you agree to be in the study?**

If you agree to be in this study you will be asked to complete a pre and post questionnaire at the beginning and end of the experiment along with tasks to be performed to assess the usability and efficiency of the website.

The pre-test questionnaire will ask you questions in regards to your age, enrollment status, and other demographics data.

The tasks for the study are related to data visualization as follows: selection of data subsets for visualization, selection of visualization type (e.g. line graph, bar graph), switching between visualization types, zooming in and out of visualization, filtering related information, interface customization.

The post questionnaire will ask questions in relation to the participant's thoughts on the usability and design of the website as well as any improvements that think could be made.

**How long will you be in the study?**

The study will take about 30 minutes of your time.

**What are your choices if you do not volunteer to be in this research study?**

If you decide not to be in the study, tell the investigator and you will be allowed to leave.

**What if you agree to be in the study now, but change your mind later?**

You do not have to stay in the study. You may withdraw from the study at any time by leaving the room after informing the investigator.

**What if the study changes while you are in it?**

If anything about the study changes or if we want to use your information in a different way, we will tell you and ask if you if you want to stay in the study. We will also tell you about any important new information that may affect your willingness to stay in the study.

**Is there any way being in this study could be bad for you?**

If you participate in this study, your mental and physical health is not at risk.

**What happens if you become injured because of your participation in the study?**

In the event that this research activity results in an injury, treatment will be available. This includes first aid, emergency treatment, and follow-up care as needed. Care for such injuries will be billed in the ordinary manner to you or your health insurance carrier.

**Will being in this study help you in any way?**

Probably not, except it will add to your experience with using websites.

**Who will pay for the costs of your participation in this research study?**

No costs are associated with participation in this study.

**Will you be paid for being in this study?**

You will not receive any payment for being this study.

**Who will know that you are in in this study and who will have access to the information we collect about you?**

The researchers and the University of Nevada, Reno Institutional Review Board will have access to your study records.

**How will we protect your private information and the information we collect about you?**

We will treat your identity with professional standards of confidentiality and protect your private information to the extent allowed by law.

We will not use your name or other information that could identify you in any reports or publications that result from this study.

**Do the researchers have monetary interests tied to this study?**

The researchers and/or their families have no monetary interests tied to this study.

**Whom can you contact if you have questions about the study or want to report an injury?**

At any time, if you have questions about this study or wish to report an injury that may be related to your participation in this study, contact Sergiu Dascalu at (775) 636-5060 and/or Andrew Munoz at (209) 402-1583.

**Whom can you contact if you want to discuss a problem or complaint about the research or ask about your rights as a research participant?**

You may discuss a problem or complaint or ask about your rights as a research participant by calling the University of Nevada, Reno Research Integrity Office at (775) 327-2368. You may also use the online *Contact the Research Integrity Office* form available from the [Contact Us page](#) of the University's Research Integrity Office website.

**Agreement to be in study**

If you agree to participate in this study, you must sign this consent form. We will give you a copy of the form to keep.

---

 Participant's Name Printed

---

 Signature of Participant

---

 Date

---

 Signature of Person Obtaining Consent

---

 Date

## Appendix D

### UNR IRB Pre-Test Survey



**University of Nevada, Reno  
Social Behavioral Research**

**Title of Study:** Near Real-time Visualization of LiDar Data  
**Principle Investigator:** Sergiu Dascalu, Ph.D.  
**Co-Investigators:** Andrew Munoz  
**Study ID Number:** 1756566-1  
**Sponsor:** N/A

---

**Pre Test Questionnaire**

Participant ID #: \_\_\_\_\_

Please Enter Your Age: \_\_\_\_\_

What is your gender? \_\_\_\_\_

What is your highest level of completed education? \_\_\_\_\_

Please rate your familiarity with LiDar technology.

(1 – Not Familiar, 5 - Very Familiar)

Not Familiar	1	2	3	4	5	Very Familiar
-----------------	---	---	---	---	---	------------------

Please rate your familiarity with point cloud data.

(1 – Not Familiar, 5 - Very Familiar)

Not Familiar	1	2	3	4	5	Very Familiar
-----------------	---	---	---	---	---	------------------

How often do you use data visualization? (1 – Never, 5 - Very Often)

Never	1	2	3	4	5	Very Often
-------	---	---	---	---	---	---------------

How useful do you think data visualization is? How do you like it?

(1 – Not Useful, 5 - Very Useful)

Not Useful	1	2	3	4	5	Very Useful
---------------	---	---	---	---	---	----------------

Please rate your familiarity with web-applications and websites.

(1 – Not Familiar, 5 - Very Familiar)

Not	1	2	3	4	5	Very
Familiar						Familiar

## Appendix E

### UNR IRB Post-Test Survey

**University of Nevada, Reno  
Social Behavioral Research**

**Title of Study:** Near Real-time Visualization of LiDar Data  
**Principle Investigator:** Sergiu Dascalu, Ph.D.  
**Co-Investigators:** Andrew Munoz  
**Study ID Number:** 1756566-1  
**Sponsor:** N/A

---

**Post Test Questionnaire**

Participant ID #: \_\_\_\_\_

How useful do you think the website is in its current state?

(1 – Not at all, 5 - Very Useful)

Not at all            1            2            3            4            5            Very Useful

How satisfied were you with the website design?

(1 – Not at all, 5 - Very Satisfied)

Not at all            1            2            3            4            5            Very Satisfied

How difficult was it to complete the tasks assigned?

(1 – Very Difficult, 5 - Very Easy)

Very Difficult        1            2            3            4            5            Very Easy

Please rate the website based on your overall satisfaction.

(1 – Not at all, 5 - Very Satisfied)

Not at all            1            2            3            4            5            Very Satisfied

What improvements, if any, do you think will help make the website more useful or easy to use?

---



---



---



---

---

---

---

---

If applicable, how could this software be best tailored to your work/needs?

---

---

---

---

---

---

---

---

Please write any other comments or observations that you might have.

---

---

---

---

---

---

---

---

## Appendix F

### UNR IRB Participant Recruitment Email

### Recruitment Email Script

#### General Recruitment Email:

Good [morning, afternoon, etc.],

I am conducting a research study to assess the usability and overall design flow of a data visualization web application. I am currently looking for 12-25 participants to conduct a user study to advance this research. Participants in the study will be asked to browse the website and complete between 3 and 5 tasks pertaining to data visualization (e.g. selection, zoom in/out, filtering, customization) which will take no longer than 30 minutes to complete. This user study will be conducted in WPEB Lab Room 436 but can also be completed virtually via Zoom. If you are interested in participating in this user study, please email Andrew Munoz at amunoz24@nevada.unr.edu to set up a time that works best for you to complete the study.

Sincerely,

Andrew Muñoz

#### Personal Recruitment Email:

Dear [participant],

I am conducting a research study to assess the usability and overall design flow of a data visualization web application. I am currently looking for 12-25 participants to conduct a user study to advance this research. Participants in the study will be asked to browse the website and complete between 3 and 5 tasks pertaining to data visualization (e.g. selection, zoom in/out, filtering, customization) which will take no longer than 30 minutes to complete. This user study will be conducted in WPEB Lab Room 436 but can also be completed virtually via Zoom. Let me know if you are interested and if so please let me know your availability to complete the user study.

Sincerely,

Andrew Muñoz