LDAT: A LIDAR Data Analysis and Visualization Tool

Andrew Muñoz, Chase Carthen, Vinh Le,
Scotty D. Strachan, Sergiu M. Dascalu, Frederick C Harris, Jr.

Computer Science and Engineering
University of Nevada, Reno
Reno, Nevada 89557, USA
amunoz24@nevada.unr.edu, ccarthen@unr.edu, vle@unr.edu
strachan@unr.edu, dascalus@cse.unr.edu, fred.harris@cse.unr.edu

***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. As LiDAR point cloud data can be highly complex, it can often be difficult to conduct analysis and visualization of the data sets. A web tool was developed to analyze and visualize this type of data, ensuing in an interactive and readable representation of the data. The data obtained for this tool is from LiDAR sensors located on street lights directly adjacent to the University of Nevada, Reno to analyze traffic information. In order to ensure the effectiveness of the tool, a user study was conducted to test the functionality and assess possible improvements.**

Keywords: LIDAR, Big Data, Point Cloud, Containerization, Data Analysis, Data Visualization, User Interface, Web Framework, MQTT, Traffic Detection

I. INTRODUCTION

Sensor technology has been around since the mid 20th century and continues to advance in 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, specifically lasers, to detect both the distance and position of various objects within a real-world environment [1]. 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. The most common types of visualizations are graphs and charts, which are extremely useful in finding and determining patterns and relationships within a data set [2].

LDAT, aka the LiDAR Data Analysis and Visualization Tool, is a web application that is used to access point cloud data from street based LiDAR sensors and extract insightful information from the point clouds gathered. The LDAT tool is built upon the Angular and Flask web frameworks to create a robust web application that is able to process, clean, compress, and output the near real-time point cloud data steam. The output is generated through a 3D render of the mesh scene that corresponds with a particular LiDAR sensor. Both the data visualization graphs and charts, as well as the 3D render, are presented within a single web page as a data dashboard to make the data easily accessible and readable. LDAT was created as a proof of concept and therefore was designed with scalability, portability, and efficiency in mind.

The rest of this paper is organized as follows: Section II details the project background and provides a brief overview on some related work. Section III describes the system design, data workflow, and implementation of both the hardware and software components. Section IV presents the final prototype implementation of the LDAT application. Section V details the user study conducted and the results obtained from the study. Section VI lists the concluding thoughts on the application and contains study results, and outlines possible directions of future work.

II. BACKGROUND AND RELATED WORK

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 type is point cloud data that is captured through LiDAR sensors. The LDAT application gathers complex point cloud data directly from LiDAR sensors to analyze and visualize street traffic information. At the time of development, there were two active LiDAR sensors being utilized at the intersection of North Virginia Street and 15th Street which is directly adjacent to the University of Nevada, Reno. The LiDAR sensors stream over raw point cloud data (.pcd) files [3], which are then cleaned and processed to be used for analysis and visualization. Before diving into the design and development of the application, next there is a discussion on some related works.

There are a few works in recent years that relate to the processing, cleaning, and visualization of point cloud data from LiDAR sensors. The first is a study [4] that examines 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 determine the count of vehicles.

Another related project is the 3DSYSTEK viewer [5], which 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. It was made to address mobility and portability problems that are typically associated 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.

In a separate study [6], 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.

III. Software Design

LDAT was built using both the Angular [7] and Flask [8] web frameworks. Each of these components were designed to work as independent components to process, clean, and visualize the point cloud data sets. In this section, an overview of the system architecture is presented along with a breakdown of the data workflow and application interface design.

A. System Architecture

The system architecture was designed primarily with performance in mind while maintaining a smooth transition to create an optimal user experience. In order to fulfill this condition, the system would need to be designed with multiple servers to properly handle, manage, and process the large amounts of data that would be coming from the LiDAR sensors. Figure 1 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.

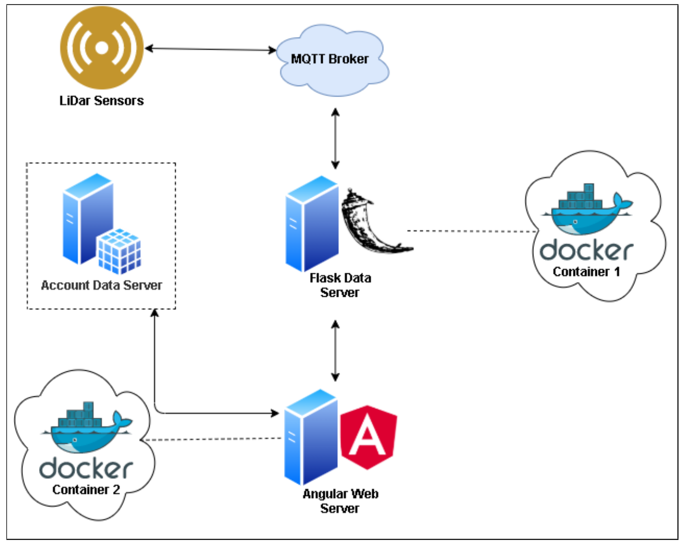

Fig. 1. System level diagram showcasing the main components of LDAT.

LDAT is comprised of four main components that make up the system architecture. The first of these components is the LiDAR sensors. The hardware used was the Velodyne Ultra Puck sensor [9] due to its 200 meter range and 360° view of the surrounding area. At the sensor level, the data is gathered in its raw format and pushed to the network through the second main component, the MQTT Broker [10]. 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 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 Interface. The Angular Web Interface 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 Interface are placed in separate Docker containers [11] to increase development efficiency and portability. Additionally, there is an Account Data Server which serves as a database server for user accounts such as log in information. This server is covered in more detail in Section VI.

B. Data Workflow

The point cloud data files are gathered by the LiDAR sensors which are located on an edge network. The high level

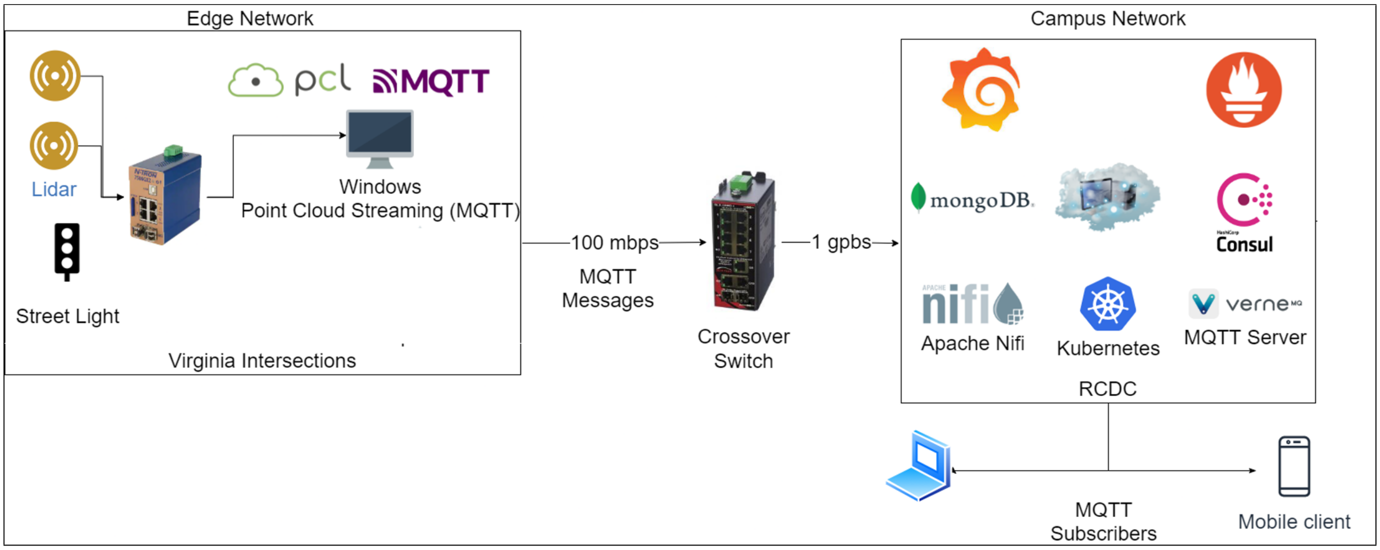

Fig. 2. High level diagram of the data workflow.

diagram [12] of the data workflow is shown in Figure 2. The two sensors illustrated in this figure represent the two different topics and data streams that were collected during the implementation of LDAT. Once a point cloud file is obtained by either sensor, the data is collected by the MQTT publisher [10] at the edge which then flows into the crossover switch. It is from the crossover switch that the data is published to the campus network and into Kubernetes and the MQTT Broker server. After the data stream has been published onto the MQTT Broker, the LDAT application is able to subscribe to the broker and pull in data from the two LiDAR sensor data streams.

C. 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 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 is welcomed to the application on the Home Page and from there is able to go to the Data Explorer Form to create a data visualization dashboard. Once the form is submitted, the user can navigate 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 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 the user on how to use LDAT. A more detailed breakdown along with additional screenshots of the UI are provided in Section IV.

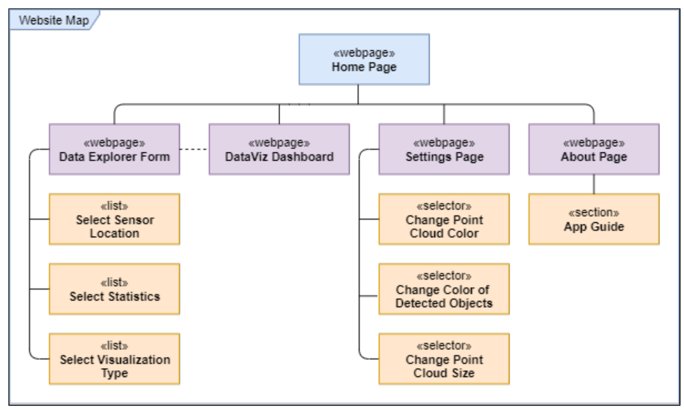

Fig. 3. Website map for the LDAT application.

IV. Prototype

LDAT was developed with simplicity in mind: a web application 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 guiding the user in the process of accessing the data.

A. User Interface

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 4. To proceed the user must click the ”Get Started”

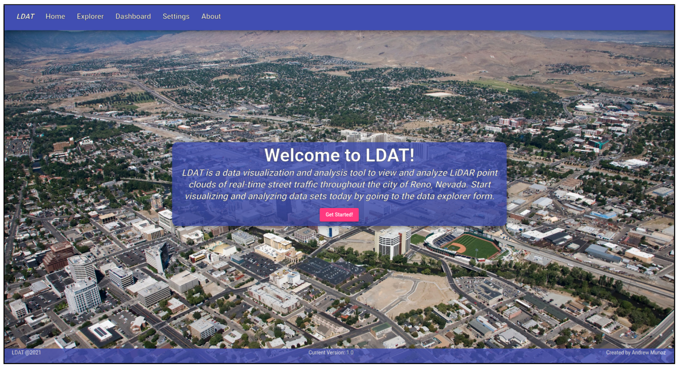

Fig. 4. LDAT Home page.

button at the center which will take users to the Data Explorer form. This 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 available visualization charts 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 feed.

After submitting the form, the user is routed to the Data Visualization Dashboard. The current version of the dashboard 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 laid 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. In addition to the dashboard, there is a Settings Page, which allows for configurations to be made to the 3D visualization such as changing the color of the environment, changing the color of detected objects, and changing the size of the points.

B. Data Visualizations

The three main data visualizations displayed 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, exemplified in Figure 5, that 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. The line graph, presented in Figure 6, displays the count per second of total objects detected in the scene. 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.

The third and final visualization is that of the near real time 3D point cloud mesh shown in Figure 7. It should be noted that the near real time 3D point cloud mesh can be configured in the Settings page through options to change the color of objects

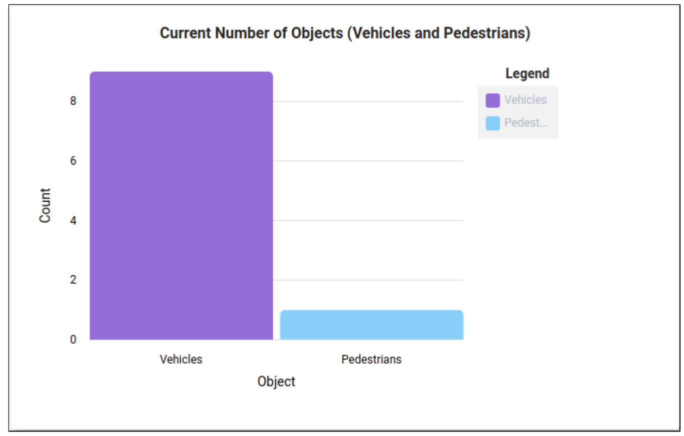

Fig. 5. Bar chart that displays the current count of each object being detected in the point cloud. The objects that can be detected are vehicles and pedestrians.

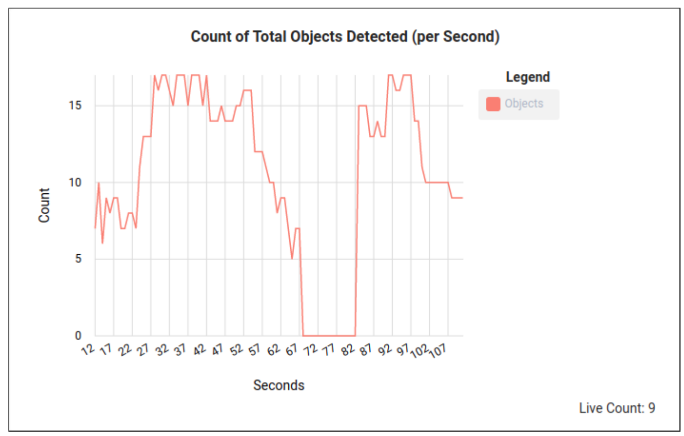

Fig. 6. Line graph that presents the total number of objects detected within the point cloud.

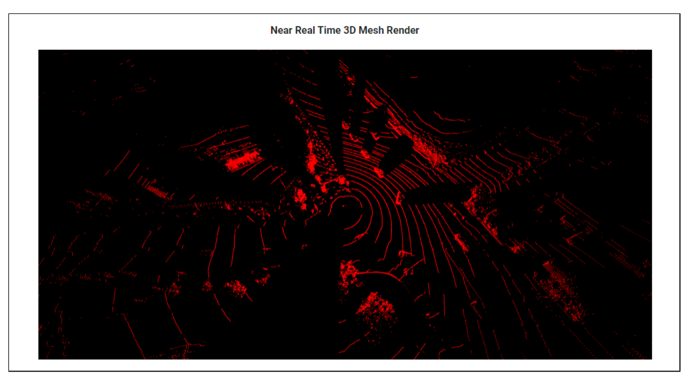

Fig. 7. Screenshot of near real time 3D point cloud mesh render which displays the view from the sensor located on the Southeast corner.

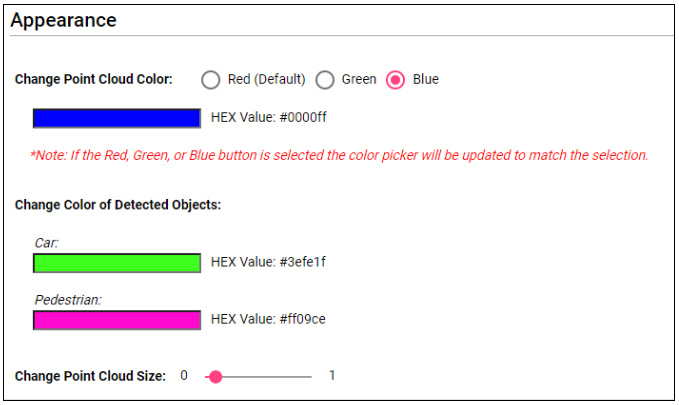

Fig. 8. UI layout of the configuration options on the Settings page.

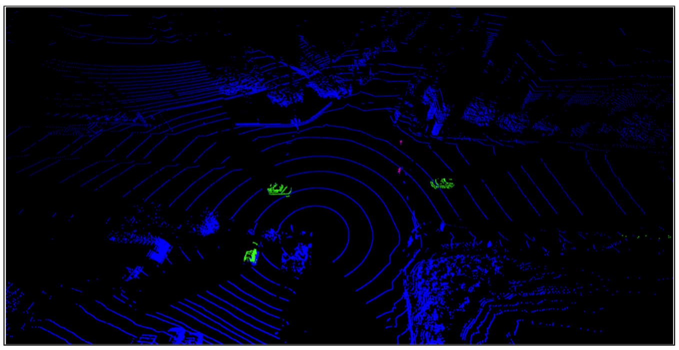

Fig. 9. Screenshot of 3D point cloud mesh render scene with object colors changed and a point cloud size of 0.1.

and size of the point cloud as displayed in Figure 8. These changes are visualized in Figure 9 with the colors changed and point cloud size changed to 0.1. 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.

V. User Study and Results

Software in general should be designed with the end user in mind, and this is especially important for data visualization interfaces such as LDAT. Software interfaces are built to be used by people, therefore, extensive testing is essential to ensure for optimal user experience. In order to properly test the LDAT application, a user study was conducted to evaluate the usability of the UI. Before the user study could be conducted with human participants, a certification was needed to get approval from the Institutional Review Board (IRB) [13]. Once the study was approved then the testing of the application could proceed.

A. User Study

There were approximately 20 participants who took part in the study, each of which came from STEM fields and were not necessarily affiliated with software development in the hopes of obtaining varied results. The study itself was conducted in the William Pennington Engineering Building (WPEB) room 436 located on the University of Nevada, Reno campus. A telecommunication option via Zoom was also provided for virtual participation due to consideration made in light of the ongoing COVID-19 pandemic.

There were three main components to the study including: a pre-questionnaire, a provided set of tasks, and a post-questionnaire. The pre-questionnaire asked participants some demographic information such as their age, gender, and educational status as well as their familiarity with some of the topics covered in the study including LiDAR, data visualization, and websites. This was followed by a set of tasks that participants had to complete to test different aspects of the interface which were as follows:

- Fill Out Form (Task 1) - Participants were asked to navigate to Data Explorer Form from Homepage to fill out the form and submit it to view visualizations.
- Change Sensor Location (Task 2) - Participants were asked to change the location of the sensor and navigate around the 3D point cloud mesh render.
- Change the Point Size (Task 3) - Participants were asked to navigate to the Settings page from the Dashboard and change the point cloud size.
- Change Color (Task 4) - Participants were asked to navigate from the Dashboard to the Settings page and change the color values for the point cloud environment and detected objects.

After the completion of the set of tasks outlined above, the participants were then asked to answer a post-questionnaire. The post-questionnaire asked the participants aspects of their general satisfaction with the application and whether or not they felt the application was useful in its current state.

B. Results

User performance was measured during the completion of the four main tasks which included tasks completion time, number of left mouse clicks, number of right mouse clicks, and number of mouse wheel scrolls. The two measurements that had the most significant results were the task completion time and number of left mouse clicks. Figure 10 presents the box plot for the task completion time of each task. At first glance, it is clear that Task 1 had the least amount of variance and therefore was fairly consistent among all the participants (it 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.

The box plot for the number of left mouse clicks is depicted in Figure 11. Task 1 had only a few outliers in the upper bound while the lower bound was very small and consistent

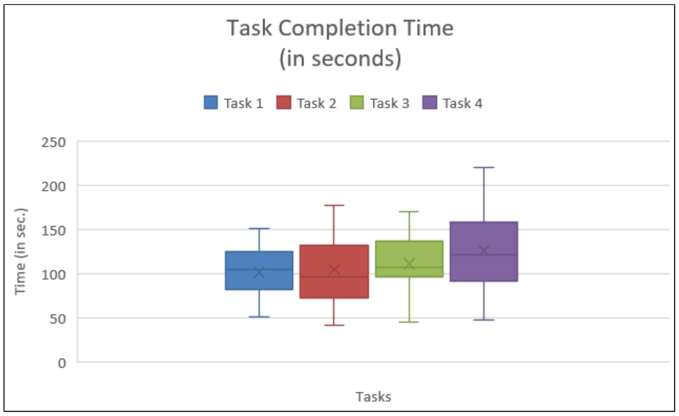

Fig. 10. Box plot for task completion time showing the mean average completion time in seconds and the min and max values for each task.

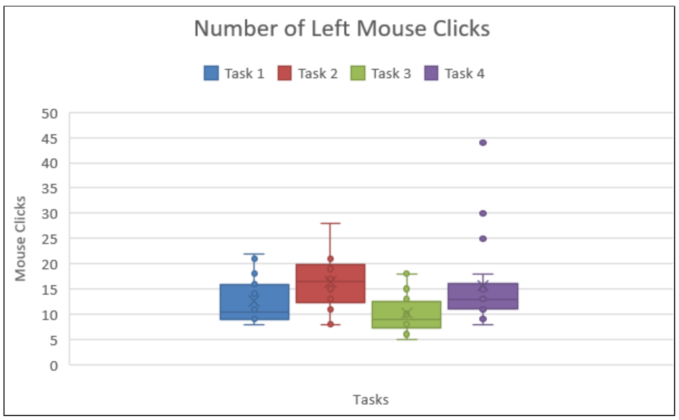

Fig. 11. Box plot for number of left mouse clicks per task.

throughout the study. Task 2 had the most variance out of the four tasks with the largest range between the min and max. Lastly, Tasks 3 and 4 had a similar distribution of results. The number of right mouse clicks was the lowest of the four measurements suggesting that it was not an intuitive control mechanism for navigating the 3D mesh. Lastly, the number of mouse wheel scrolls had the most variance but was difficult to measure as it was potentially affected based on various input devices used from virtual and in-person participants.

VI. Conclusion and Future Work

The overall goal of the LDAT 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: bar chart, line graph, and 3D LiDAR point cloud mesh scene render. Each of these were continuously updated in order to keep up with the near real time data stream. Configuration functionality was also added to the application to provide an element of control, customization, and ensure the user was able to adapt the visualizations based on their own preferences or need.

As for future work, there are a several different additions that have been discussed. A useful addition to this visualization would be a statistical analysis or classifier able to detect and display the speed and trajectory for any and all moving objects within a scene. A major development and very useful addition to work would be the fusing of these two streams from two sensors in order to create one large 3D mesh. Another addition could be the inclusion of a user account database which would convert this visualization application into a fully function user portal, capable of logins, controlled access, and security. Finally, a scalable database to store past point cloud captures could be implemented. This would afford users with the ability to go back and look at past data streams and conduct comparative analysis of the current data.

Acknowledgements

This material is based in part upon work supported by the NSF OAC CC* award #1827186, NSF OAC CC* award #2019164, and the Nevada Center for Applied Research. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation or the Nevada Center for Applied Research.

References

[1] R. Harrap and M. Lato, "An overview of LIDAR: collection to application," *NGI publication*, vol. 2, pp. 1–9, 2010.
[2] C. N. Knaflic, *Storytelling with Data: A Data Visualization Guide for Business Professionals, 1st Ed.* Hoboken, New Jersey: John Wiley & Sons, Inc., 2015.
[3] Point Cloud Library, "The PCD (Point Cloud Data) file format." https://pointclouds.org/documentation/tutorials/pcd_file_format.html. [Accessed 2021-04-06].
[4] K. Mohring, T. Myers, and I. 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, Association for Computing Machinery, New York, USA, 2018.
[5] 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*, pp. 262–266, 2014.
[6] S, Kulkarni, and M, Chandrashekaraiah and S, Raghunandan, "3d annotation tool using LiDAR," in *2019 Global Conference for Advancement in Technology (GCAT)*, pp. 1–4, 2019.
[7] Angular, "What is Angular?." https://angular.io/guide/what-is-angular. [Accessed 2021-05-27].
[8] Flask, "Flask: User's Guide." https://flask.palletsprojects.com/en/2.0.x/. [Accessed 2021-05-27].
[9] Velodyne Lidar, "Ultra Puck." https://velodynelidar.com/products/ultra-puck/. [Accessed 2021-07-16].
[10] MQTT, "MQTT: The Standard for IoT Messaging." https://mqtt.org/. [Accessed 2021-06-03].
[11] Docker, "What is a Container?." https://www.docker.com/resources/what-container. [Accessed 2021-06-04].
[12] "Data infrastructure for advanced traffic sensor edge networks," in *PEARC '21: Practice and Experience in Advanced Research Computing*, Association for Computing Machinery, 2021.
[13] Research Dataware, LLC., "IRBNet: Innovative Solutions for Compliance and Research Management." https://irbnet.org/release/index.html. [Accessed 2021-07-26].