

University of Colorado  
Department of Aerospace Engineering Sciences  
ASEN 4028

Project Final Report (PFR)  
May 3, 2021

**Functional LiDAR Analysis of Structural Health  
(FLASH)**

**Project Advisors**

Dr. Jelliffe Jackson jelliffe.jackson@colorado.edu	Dr. Dennis Akos dma@colorado.edu
---	-------------------------------------

**Project Customers**

Chris Prince cprince@astraspace.net	Erik Stromberg estromberg@astraspace.net	Cotton Anderson canderson@astraspace.net
--	---	---

**Team Members**

<b>Project Manager: Kunal Sinha</b> Kunal.Sinha@colorado.edu (303) 619-5611	<b>Jake Fuhrman</b> Jacob.Fuhrman@colorado.edu (720) 334-4867
<b>Shray Chauhan</b> Shray.Chauhan@colorado.edu (720) 347-1083	<b>Ishaan Kochhar</b> Ishaan.Kochhar@colorado.edu (719) 373-7353
<b>Courtney Kelsey</b> courtney.kelsey@colorado.edu (720) 335-8034	<b>Fiona McGann</b> fiona.mcgann@colorado.edu (801) 599-5103
<b>Erik Stolz</b> Erik.Stolz@colorado.edu (720) 355-2537	<b>Andrew Fu</b> Andrew.Fu@colorado.edu (224) 325-1228
<b>Julian Lambert</b> Julian.Lambert@colorado.edu (970) 987-1232	<b>Ricky Carlson</b> ricky.carlson@colorado.edu (303) 472-3573

# Contents

<b>1</b>	<b>Project Purpose</b>	<b>8</b>
<b>2</b>	<b>Project Objectives and Functional Requirements</b>	<b>9</b>
2.1	Levels of Success	9
2.2	FLASH CONOPS	10
2.3	Project Deliverables	10
2.4	Functional Block Diagram	11
2.5	Functional Requirements	11
2.5.1	Functional Requirement Motivation	12
<b>3</b>	<b>Final Design</b>	<b>13</b>
3.1	Structures	13
3.1.1	Design Requirements	14
3.1.2	Finalized Design	14
3.2	Software	16
3.2.1	Overview	16
3.2.2	Design Requirements	17
3.2.3	Finalized Design	17
3.3	Electronics and Communication	18
3.3.1	Overview	18
3.3.2	Design Requirements	18
3.3.3	Finalized Design	19
3.4	Integrated Design	19
3.4.1	Physical Design	19
3.4.2	Software Design	20
<b>4</b>	<b>Manufacturing</b>	<b>20</b>
4.1	Structures	20
4.2	Electrical	23
4.3	Software	23
4.4	Manufacturing Outcomes and Challenges	25
4.4.1	Hardware	25
4.4.2	Software	25
4.5	Integrated Design	27
<b>5</b>	<b>Verification and Validation</b>	<b>28</b>
5.1	Mount Pull Test (MPT)	28
5.1.1	Test Purpose/Objective	29
5.1.2	Test Equipment and Setup	29
5.1.3	Test Operation and Results	29
5.1.4	Model Verification and Requirements Satisfaction	30
5.2	Small-Scale LiDAR Testing (SSLT)	31
5.2.1	Test Purpose/Objective	31
5.2.2	Test Equipment and Setup	32
5.2.3	Test Operation and Results	33
5.2.4	Model Verification and Requirements Satisfaction	36
5.3	Comprehensive System Testing (CST)	36
5.3.1	Test Purpose/Objective	36
5.3.2	Test Equipment and Setup	37
5.3.3	Test Operation and Results	37

5.3.4	Model Verification and Requirements Satisfaction . . . . .	44
5.4	Validation of Functional Requirements and Success Criteria . . . . .	45
<b>6</b>	<b>Risk Assessment and Mitigation</b>	<b>46</b>
<b>7</b>	<b>Project Planning</b>	<b>49</b>
7.1	Organizational Chart (OC) . . . . .	49
7.2	Work Breakdown Structure (WBS) . . . . .	50
7.3	Work Plan (WP) . . . . .	51
7.4	Cost Plan (CP) . . . . .	53
7.5	Test Plan (TP) . . . . .	53
<b>8</b>	<b>Lessons Learned</b>	<b>54</b>
<b>9</b>	<b>Individual Report Contributions</b>	<b>55</b>
<b>10</b>	<b>References</b>	<b>56</b>
<b>11</b>	<b>Appendix A: LiDAR</b>	<b>58</b>
11.1	Additional Comprehensive System Test Results . . . . .	58
11.2	Thermal Analysis . . . . .	61
11.3	Error Analysis . . . . .	62
11.3.1	Sunlight . . . . .	62
11.3.2	Reflectivity . . . . .	62
11.3.3	Probability of Detection . . . . .	62
11.4	Point Volume and Data Budget . . . . .	63
11.5	Sensor Outputs . . . . .	64
11.6	Drawings . . . . .	64
11.7	Definitions . . . . .	65
11.7.1	Metrics . . . . .	65
11.7.2	Coordinate Frame . . . . .	65
<b>12</b>	<b>Appendix B: Structural Assessment</b>	<b>66</b>
12.1	Scan Obstructions . . . . .	66
12.2	Bridge Inspections . . . . .	67
<b>13</b>	<b>Appendix C: Trade Studies</b>	<b>69</b>
13.1	LiDAR Sensor Trade Study . . . . .	69
13.2	Mounting Mechanism Trade Study . . . . .	69
13.3	SLAM Trade Study . . . . .	70
13.4	Laptop Trade Study . . . . .	71

## List of Figures

1	FLASH Inspiration and Motivation . . . . .	8
2	FLASH Design Orientation . . . . .	9
3	FLASH CONOPS . . . . .	10
4	Functional Block Diagram . . . . .	11
5	FLASH System Functional Requirements . . . . .	12
6	Finalized LiDAR Aluminum Mount Design (Vertical Configuration) . . . . .	14
7	Finalized LiDAR Aluminum Mount Dimensions . . . . .	15
8	LiDAR Mounted atop the Mission Vehicle . . . . .	15

9	Final Software Architecture . . . . .	16
10	Final Electrical Interfacing Architecture . . . . .	19
11	3D-Printed Nylon/Carbon Fiber Composite Fit Check Mount . . . . .	20
12	3D-Printed Nylon/Carbon Fiber Crack . . . . .	21
13	LiDAR Mount Manufacturing Flow Down . . . . .	21
14	Manufactured Aluminum Mount (Old Configuration) . . . . .	22
15	Manufactured Aluminum Mount (New Configuration) . . . . .	22
16	Electrical Interfacing from the Vehicle to the DC/AC Power Inverter . . . . .	23
17	Electrical Interfacing from the Interface Box to the Laptop and DC/AC Power Inverter . . . . .	23
18	Stages of mesh generation of US 36 Underpass. . . . .	24
19	Comparison of expected LIOM output to actual. . . . .	26
20	Aluminum LiDAR Mount for Magnetic Attachment to the Mission Vehicle (with Associated Interface Box Cable) . . . . .	27
21	Full Functioning System on Mission Vehicle (External View) . . . . .	28
22	Full Functioning System on Mission Vehicle (Internal View) . . . . .	28
23	Pull Test Free Body Diagram . . . . .	30
24	Pull Test Setup . . . . .	30
25	Pull Test Force Measurement . . . . .	31
26	Small Scale LiDAR Test Setup (Schematic) . . . . .	32
27	Small Scale LiDAR Test Setup (Actual) . . . . .	33
28	Sample Point Cloud Measurement (Board Height) . . . . .	34
29	SSLT Results: Horizontal Resolution Measurements Compared to Model Prediction . . . . .	35
30	Underpass at Foothills Pkwy. and Walnut St. in Boulder, CO . . . . .	37
31	Point Cloud of Walnut/Foothills Underpass Bridge BEFORE Cleaning and Cropping . . . . .	38
32	Point Cloud of Walnut/Foothills Bridge AFTER Cleaning and Cropping . . . . .	39
33	Point Cloud of Walnut/Foothills Bridge AFTER Cleaning and Cropping, Colorized . . . . .	39
34	Point Density Measurement of Walnut/Foothills Bridge Underside (15 MPH Scan) . . . . .	40
35	CST Results: Average Point Density Measured on Bridge Underside from 5 to 20 MPH . . . . .	40
36	Walnut/Foothills Bridge Dimensions Referenced for Accuracy Estimation . . . . .	41
37	Overlay Comparison of Walnut/Foothills Point Cloud to Google Maps Ground Truth . . . . .	42
38	Mesh Generated from Walnut/Foothills Point Cloud . . . . .	43
39	Mesh Internals Rendered in Blender Environment from Walnut/Foothills Point Cloud . . . . .	43
40	3D Mesh Rendering of Walnut/Foothills Bridge in Blender . . . . .	44
41	Satisfaction of Design Requirements from CST Results . . . . .	44
42	Functional Requirements Verification . . . . .	45
43	Functional Requirements Verification Continued . . . . .	45
44	Simplified Levels of Success . . . . .	46
45	Initial Risk Matrix . . . . .	47
46	Failure Modes and Effects Analysis (FMEA) . . . . .	47
47	Risk Mitigation Methods . . . . .	48
48	Post-Mitigation Risk Matrix (as of CDR) . . . . .	48
49	Organization Chart (OC) Team FLASH . . . . .	50
50	Work Breakdown Structure for FLASH . . . . .	51
51	Software Work Plan . . . . .	52
52	Structures and On-board Setup Work Plan . . . . .	52
53	Finalized Team Budget (Updated: 4/27/2021) . . . . .	53
54	Testing Schedule with dependencies, margins and progress bars shown . . . . .	54
55	Point Cloud of Scanned Parking Structure . . . . .	58
56	Point Cloud of Scanned Parking Structure . . . . .	58
57	Early Attempt at Mesh Generation for Walnut/Foothills Bridge . . . . .	58
58	Early Attempt at Mesh Generation for Walnut/Foothills Bridge . . . . .	59

59	Mesh Generated from Point Cloud of Boulder Canyon Tunnel . . . . .	59
60	Boulder Canyon Tunnel Mesh Overlaid with Google Maps API . . . . .	59
61	Failed Mesh Generation in MeshLab Software . . . . .	60
62	SSLT Target Board Feature Discernment at 3.5 meters . . . . .	60
63	Point Cloud Frame Stacking Observed with Previous Sensor Orientation . . . . .	60
64	Governing Equations, Constants, and Assumptions for Thermal Analysis . . . . .	61
65	Thermal Analysis Results . . . . .	61
66	Various Objects and their Reflectivities . . . . .	62
67	Point Volume Calculation . . . . .	63
68	Data Volume Calculation . . . . .	63
69	Data Outputs from the Ouster OS1-32 Gen 1 . . . . .	64
70	Physical Representation of the Ouster OS1-32 Gen 1 with its Interface Box and Connectors . . . . .	64
71	Ouster OS1-32 Gen 1 Coordinate Frame Information . . . . .	65
72	Ouster OS1-32 Gen 1 Coordinate Frame Transformations . . . . .	66
73	Longitudinal Beams along Underside of Highway Bridge (Source: Getty) . . . . .	66
74	LiDAR Beam Blockage due to Bridge Girder . . . . .	67
75	Various Structural Deformations and their Characteristics . . . . .	67
76	Example of a Bridge Inspection . . . . .	68
77	Examples of Infrastructure Inspections and their Required Accuracies . . . . .	68
78	LiDAR Sensor Trade Study Decision Matrix. . . . .	69
79	Mount Mechanism Trade Study Decision Matrix . . . . .	70
80	Software Trade Study Decision Matrix. . . . .	70
81	On-board Computer Trade Study Decision Matrix . . . . .	71

## List of Tables

1	Levels of Success . . . . .	9
2	Structure Design Requirements . . . . .	14
3	Software Design Requirements . . . . .	17
4	Electronic and Communication Design Requirements . . . . .	18
5	Key LiDAR Sensor Specifications (Ouster OS1-32) . . . . .	19
6	Mount Pull Test Results . . . . .	29
7	SSLT Results: Test Board Measurements for Determination of Accuracy and Precision . . . . .	34
8	Satisfaction of Design Requirements from SSLT Results . . . . .	36
9	CST Results: Point Cloud Accuracy . . . . .	41
10	Test Plan for Spring 2021 Team FLASH . . . . .	54
11	Individual Report Contributions Table . . . . .	55

## List of Acronyms and Nomenclature

AC	Alternating Current
AIAA	American Institute of Aeronautics and Astronautics
API	Application Programming Interface
CAD	Computer-Aided Design
CDR	Critical Design Review
CNC	Computer Numerical Control

CONOPs Concept of Operations  
COTS Commercial off the Shelf  
CPE Critical Project Element  
CST Comprehensive System Test  
DC Direct Current  
DR Design Requirement  
EDPM Ethylene Diene Propylene Monomer  
FLASH Functional LiDAR Assessment of Structural Health  
FMEA Failure Mode and Effects Analysis  
FMVSS Federal Motor Vehicle Safety Standards  
FOV Field of View  
FR Functional Requirement  
GNSS Global Navigation Satellite System  
GPS Global Positioning System  
GUI Graphic User Interface  
IMU Inertial Measurement Unit  
IP Internet Protocol  
JSON Java Script Object Notation  
LiDAR Light Detection and Ranging  
LIO-SAM LiDAR Inertial Odometry via Smoothing and Mapping  
LIOM LiDAR Inertial Odometry and Mapping  
LOAM LiDAR Odometry and Mapping  
MPT Mount Pull Test  
NCHRP National Cooperative Highway Research Program  
OTIS Online Transportation Information Systems  
PDD Project Definition Document  
RFC Request for Comments  
ROS Robot Operating System  
RPN Risk Priority Number  
SFR Spring Final Review  
SLAM Simultaneous Localization and Mapping  
SLAM-LMAO SLAM LiDAR Mapping and Odometry

SNR Signal to Noise Ratio  
SSLT Small-Scale LiDAR Testing  
TCP Transmission Control Protocol  
UDP User Datagram Protocol  
USD United States Dollar  
VINS-Mono Monocular Visual-Inertial System

# 1 Project Purpose

*Author: Courtney Kelsey*

There is a growing and persistent need for monitoring and inspection of critical infrastructure around the country and the world. In the United States alone, there are more than 600,000 bridges, one third of which are over fifty years old. These critical infrastructures require surveillance and maintenance in order to ensure that they are structurally intact and not prone to failure. Traditional infrastructure inspection processes have an estimated industry value of USD 1.78 billion in 2019 and are projected to reach USD 5.38 billion by 2027 . Traditional inspection methods are slow, costly and infrequent. In remote areas, where there is a lack of GPS/GNSS location systems, or rough terrains in which the strenuous accessibility has led to infrequent examinations, there is a need for an autonomous mapping system for remote access and efficient methods of evaluation.



(a) Current Infrastructure Assessment



(b) Consequence of Poor Infrastructure Analysis

Figure 1: FLASH Inspiration and Motivation

The system being proposed for this project will reduce the amount of time and effort required to accurately scan bridges. The plan to accomplish this goal will include a low cost LiDAR as well as an accompanying computer and mounting system. The team has a goal for this system to be accurate enough to detect a wide variety of faults and possible failure points in bridges while operating at speed. This system is unique in that it combines the accuracy of a LiDAR scanning system with the time efficiency of a drive by inspection. Other projects exist for both these objectives separately but to the teams knowledge there has not been a project with the objective to optimize bridge scanning in particular at the speed that this project intends to reach.

Another important requirement that this project adheres to is the ability to produce a low budget functioning LiDAR system. Currently, for about \$0.5 million a similar system can be bought off the shelf that can accomplish similar goals [7]. What sets this project apart is that it will prove the feasibility of an accurate, low cost, fully functional LiDAR system at speed. When the system becomes fully operational it will decrease the time spent on bridge inspection and increase the number of bridges that can be scanned which will lead to quicker response times and fewer accidents caused by defective bridges.

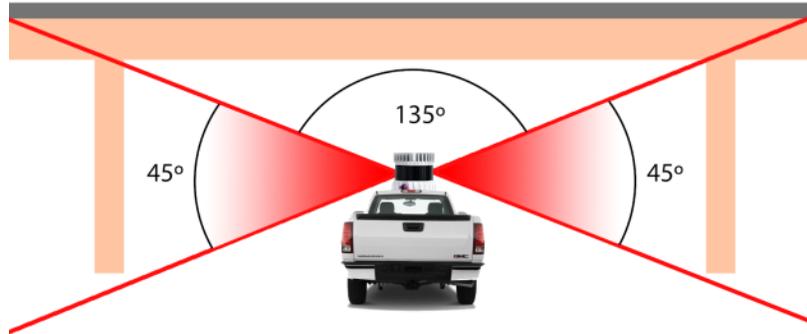


Figure 2: FLASH Design Orientation

## 2 Project Objectives and Functional Requirements

*Authors: Courtney Kelsey, Jake Fuhrman, Shray Chauhan, Kunal Sinha*

### 2.1 Levels of Success

Table 1 defines the teams levels of success which range from the absolute minimum that must be accomplished for the project to be considered a success (Level 1) up to the most that the project will plan to accomplish (Level 3). This table uses the most updated requirements and definitions as well as the most significant changes from when it was first created for PDD.

Table 1: Levels of Success

	<b>Mechanical &amp; Power</b>	<b>Data</b>	<b>Software</b>
<b>Level 1</b>	A structure capable of securing the system to one specific vehicle shall be manufactured.	A 10 cm feature shall be detected/resolved from the point cloud, with some noticeable noise in the data from a scan distance of 3.5m.	Generate a 3D point cloud-map and mesh in a stationary environment.
<b>Level 2</b>	The structure shall attach to and detach from multiple different vehicles.	A 5 cm feature shall be detected/resolved from the point cloud, with some noticeable noise in the data from a scan distance of 3.5m.	Generate a 3D point cloud map and mesh in a moving environment via self localization.
<b>Level 3</b>	The structure shall attach to and detach from any vehicle and shall be capable of highway speeds.	A 3 cm feature shall be detected/resolved from the point cloud, with minimal noticeable noise in the data from a scan distance of 3.5m.	Generate a 3D point cloud-map and mesh in a moving environment with enough accuracy and detail to enable structural analysis.

## 2.2 FLASH CONOPS

Figure 3 represents the concept of operation for the FLASH LiDAR system. This has been optimized multiple times in order to properly visualize the most important parts of the project while still highlighting the necessary sections as well.

Once the LiDAR unit is secured on a car, the vehicle will have to make a pass under a vehicle. 50 m before the entrance to the infrastructure, the LiDAR unit will start to collect data. This is the activation and deployment stage and is controlled by a passenger in the vehicle. The infrastructure is then scanned with raw singular point clouds and IMU data being saved. This on average would be less than 0.5 GB of data, with about 1000 pts/m<sup>2</sup> recorded. The data collection is stopped 50 m after the exit of the infrastructure. The point density will vary with speed of the vehicle, and hence the team plans to be under 60 mph at all times. Assuming a 5.1m bridge height, the system will record around 655 thousand points per second. This data will be transmitted onto an online drive or network server for post-processing at about 15 Mbps speed.

This data is then used for 3D map/model generation. This is done by transmitting the data to the processing computer with better hardware capabilities and running it through a SLAM (Simultaneous Localization and Mapping) algorithm using ROS (Robotic Operating System) to extract and move data. The outputs from SLAM is then fed into cloud compare for refinement of point clouds and creation of the final deliverable, the 3-D, smooth Mesh of the infrastructure.

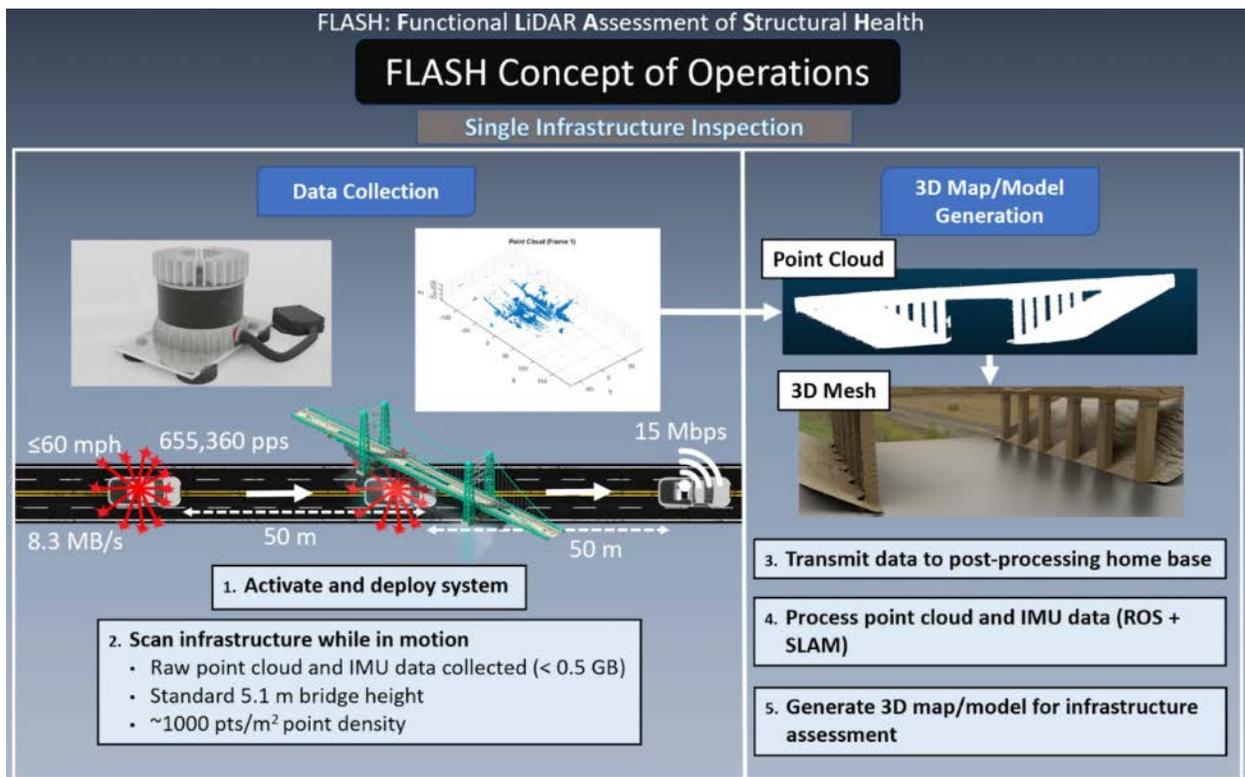


Figure 3: FLASH CONOPS

## 2.3 Project Deliverables

The project introduced to the group was to design and build a system that could scan and map the undersides of bridges and other structures in order to search for faults, erosion, or other structural points of failure. The team shall design, build, and deploy a vehicle-based infrastructure analysis system using LiDAR, that successfully maps its surroundings while in motion. The most important requirement that was

put forth by the customer was that the team needed to utilize a LiDAR sensor for this project. In addition to this deliverable, the system must also acquire data from the LiDAR system and be able to transmit the collected data via wireless transmission. A deliverable put forth by the course is that by the end of the project a full design must be defined and implemented into a functioning final project that can be tested for the required functionality. Finally, a deliverable put forth by the group is to acquire accurate data from this LiDAR system while at highway speeds passing under the bridge in question.

## 2.4 Functional Block Diagram

The functional block diagram (Figure 4) is a high level description of how the system works together. Power is provided by the car’s auxilliary power outlet which provides unregulated power to a power inverter, which produces wall power output. This is then used to power the sensor package outside the car through an interface box which serves as an intermediary connection to the sensor for both power and data. The LiDAR sensor rests on top of the vehicle and is mounted onto a vehicle platform that was designed by the team. The inverter also provides power to an on-board passenger laptop which stores the data from the sensors. This data can be uploaded to a network server or a cloud drive via generic WiFi, or it can be processed on-board depending on the hardware capabilities of the laptop. This data will be downloaded onto a Post Processing Unit for data verification and mesh generation. The software pipeline for post processing will be discussed further in the Final Design section.

Aside from certain customized portions of the software pipeline and vehicle sensor housing platform, all FLASH components in the functional block diagram will be supplied/acquired from commercial sources, and the team is responsible for integrating all components.

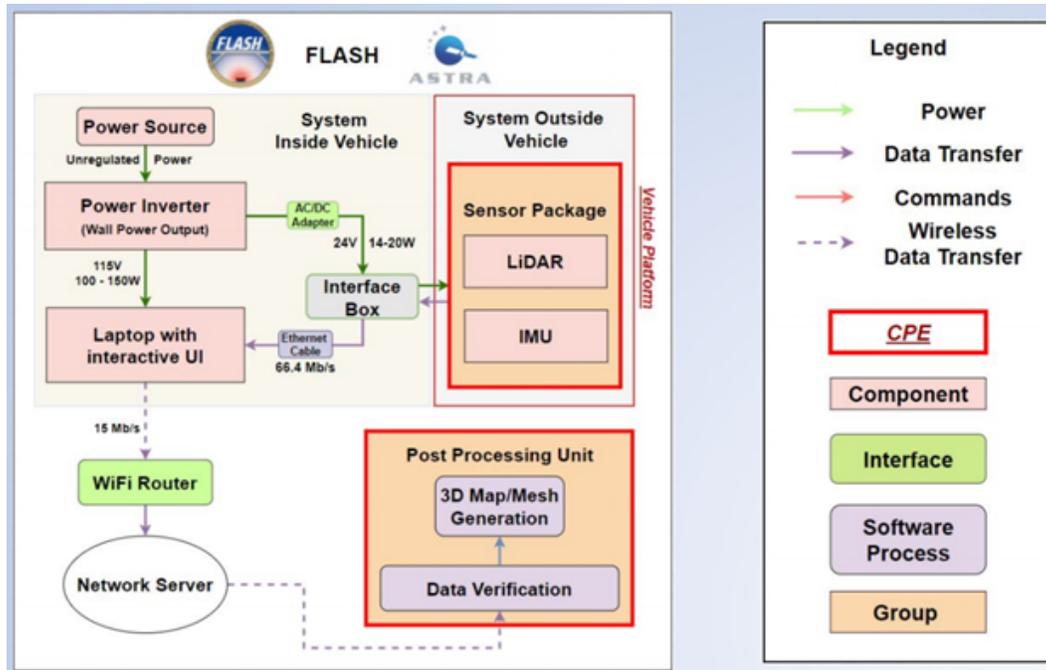


Figure 4: Functional Block Diagram

## 2.5 Functional Requirements

Figure 5 lists the Functional Requirements the team needs to verify. They include all of the necessary high level requirements that define a successful system that can achieve the project’s main objective. The rationale and mode of verification for these functional requirements are discussed in detail in Section 2.5.1.

<b>FR 1</b>	The system shall utilize a 3D LiDAR sensor to survey infrastructure of interest.
<b>FR 2</b>	The LiDAR sensor shall collect and output usable 3D point cloud data (x, y, z coordinates).
<b>FR 3</b>	The system shall be capable of localizing itself during normal driving conditions even when GNSS services are not readily available.
<b>FR 4</b>	The on-board processing unit shall be capable of data storage, handling, and interfacing between components.
<b>FR 5</b>	The system shall be capable of mounting onto a vehicle and operating while the vehicle is in motion.
<b>FR 6</b>	The system shall incorporate a power source that is capable of continuously supplying power to all applicable components.
<b>FR 7</b>	The point cloud and localization data shall be consolidated and post-processed into an interactive digital 3D map/model.
<b>FR 8</b>	The on-board communications unit shall be capable of wirelessly transferring point cloud and localization data directly to a network server.
<b>FR 9</b>	The system shall be capable of initiating and terminating data collection with minimal passenger interaction.
<b>FR 10</b>	The system shall conform to all relevant safety regulations and guidelines.

Figure 5: FLASH System Functional Requirements

### 2.5.1 Functional Requirement Motivation

**FR 1 The system shall utilize a 3D LiDAR sensor to survey infrastructure of interest.**

Current technologies for infrastructure fault detection are expensive and inefficient in terms of time and resources. Creating a 3D point cloud gives structural engineers an effective alternative to detecting faults within infrastructure of interest. The proposed non-stationary terrestrial LiDAR system by this project would serve as a more effective and less costly solution to traditional methods used today for infrastructure analysis.

**FR 2 The LiDAR sensor shall collect and output usable 3D point cloud data (x,y,z coordinates).**

This requirement is very similar in motivation to FR 1. The point cloud data from this system will be useful in determining the structural health of a structure and the most efficient way for this to occur is to have the system output usable 3D point cloud data for immediate post-processing by the customer.

**FR 3 The system shall be capable of localizing itself during normal driving conditions even when GNSS services are not readily available.**

In order to collect data for certain infrastructure such as wide bridges or long tunnels, the system must be able to operate as expected without having connection to GNSS services. This aspect sets the system apart when compared to traditional non-stationary LiDAR collection systems.

**FR 4 The on-board processing unit shall be capable of data storage, handling, and interfacing between components.**

The incorporation of these components into an onboard computer are common and necessary for the correct functionality of a computer system. There needs to be a large capacity for the data storage in order to handle the sizable files that will be produced from each scan. There will need to be a specialized handling system for converting the data received into usable 3D point cloud data. Finally, the interfacing between components is necessary for completing an accurate scan of a structure.

**FR 5 The system shall be capable of mounting onto a vehicle and operating while the vehicle is in motion.**

The system is meant to improve the process of infrastructure fault detection. Traditional methods used today seldom involve non-stationary terrestrial LiDAR systems. Even when these methods are employed, the associated costs are far beyond the budget of this project. The system being mounted to a moving vehicle will drastically reduce the time it takes to collect the data when compared to a stationary system and through the innovation of the team it will be much less costly than any current off the shelf model.

**FR 6 The system shall incorporate a power source that is capable of continuously supplying power to all applicable components.**

Power supply is an incredibly important part of an electrical and mechanical system. Without a power source on board there would be no sensor measurements. Therefore, the system needs to have a power source onboard to supply enough power to run all applicable components within the system.

**FR 7 The point cloud and localization data shall be consolidated and post-processed into an interactive digital 3D map/model used by the customer to conduct structural analysis.**

The purpose of this mission is to provide data to the client in order to search for structural faults. This needs to be done in a time efficient fashion therefore the customer needs to have the data in the correct format to immediately start analyzing for potential faults.

**FR 8 The on-board communications unit shall be capable of wirelessly transferring point cloud and localization data directly to a designated headquarters.**

This requirement is based on the need to quickly and efficiently transfer the data received from scanning to the customer who will post-process it for fault detection. The transmission of data wirelessly will allow for a hands free way of transferring the data as none of the hardware needs to be removed once the vehicle has returned to the homing site.

**FR 9 The system shall be capable of initiating and terminating data collection with minimal passenger interaction.**

The customer required this project to have minimal driver interaction to ensure safe operation of the vehicle and as much automated functionality as possible.

**FR 10 The system shall conform to all relevant safety regulations and guidelines.**

This project involves a moving vehicle and a laser device, so safety must be addressed.

## 3 Final Design

*Authors: Kunal Sinha, Andrew Fu, Ricky Carlson, and Fiona McGann*

### 3.1 Structures

The final mount design was motivated by the systems requirement to have the LiDAR mounted up right on a moving vehicle. The mount is constructed from a single plate of  $\frac{1}{8}$  inch thick scrap Aluminum found in the machine shop of CUs Aerospace Engineering Department and neodymium magnets (see Appendix D for the trade study of attachment methods). The LiDAR has mounting points machined into its construction and is bolted directly to this Aluminum plate. Originally, the system called for the LiDAR to be mounted 90 ° in a horizontal orientation. This first structure was based on the same base plate which now serves as the final mount. This original mount was constructed from five separate plates of scrap Aluminum which were cut in a CNC machine so that it could be bolted together. Because this orientation did not allow the existing attachment points of the LiDAR to be mounted to the base plate, the original manufactured structure had side walls to house the sensor. The protective cage included an opening on top to allow for the LiDAR to still scan above the vehicle without obstructing any of the lasers. Additionally the mount included an opening where the heat sink fins of LiDAR could be open to airflow. The protective cage and the openings were removed from the final design as they were unnecessary.

### 3.1.1 Design Requirements

Table 2: Structure Design Requirements

DR	Requirement	Justification
DR-5.1	The mounting structure shall withstand drag forces associated with a vehicle speed of no more than 65 mph.	While data collection is not necessarily required to be taken at high speeds, the vehicle should be able to safely travel in between speeds at highway speeds.
DR-10.1	The system shall adhere to all applicable Federal Motor Vehicle Safety Standards (FMVSS)	As this system is meant to be operated around other civilian vehicles, it is imperative FLASH adheres to the FMVSS regulations.

### 3.1.2 Finalized Design

For comprehensive mission success, the structural mount must be securely attached to the exterior roof of a moving vehicle. This mount must be easily detachable from the vehicle without compromising the security of the attached LiDAR sensor, even at highway speeds. To accomplish this, four,  $\frac{1}{2}$  inch N42 neodymium magnets with EDPM rubber coatings and integrated bolts were screwed into the corners of an  $\frac{1}{8}$  inch thick rectangular, Aluminum base plate. The dimensions of the base plate are 13 cm by 8.3 cm. The aluminum base plate also acted as a heat sink so the LiDAR sensor does not exceed its functional thermal limits. The LiDAR is attached to the plate by aligning screw holes on the base plate to the vendor manufactured mounting holes at the bottom of the OS1. Figure 6 a shows the finalized structural design with the OS1 attached, this final mount would then be attached to the testing vehicle as seen in Figure 8.

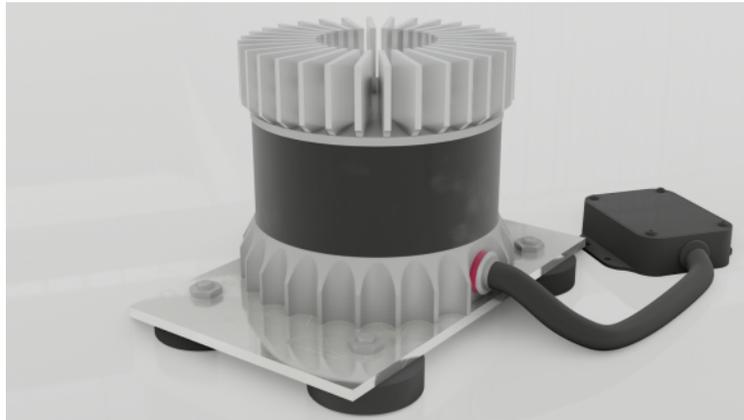


Figure 6: Finalized LiDAR Aluminum Mount Design (Vertical Configuration)

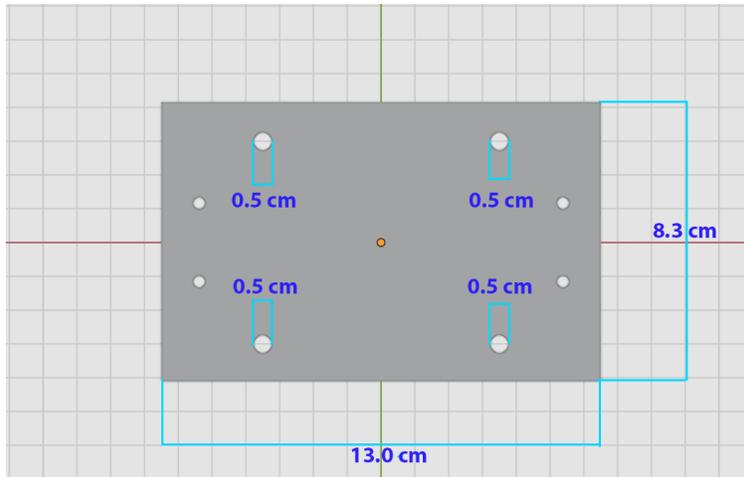


Figure 7: Finalized LiDAR Aluminum Mount Dimensions



Figure 8: LiDAR Mounted atop the Mission Vehicle

## 3.2 Software

### 3.2.1 Overview

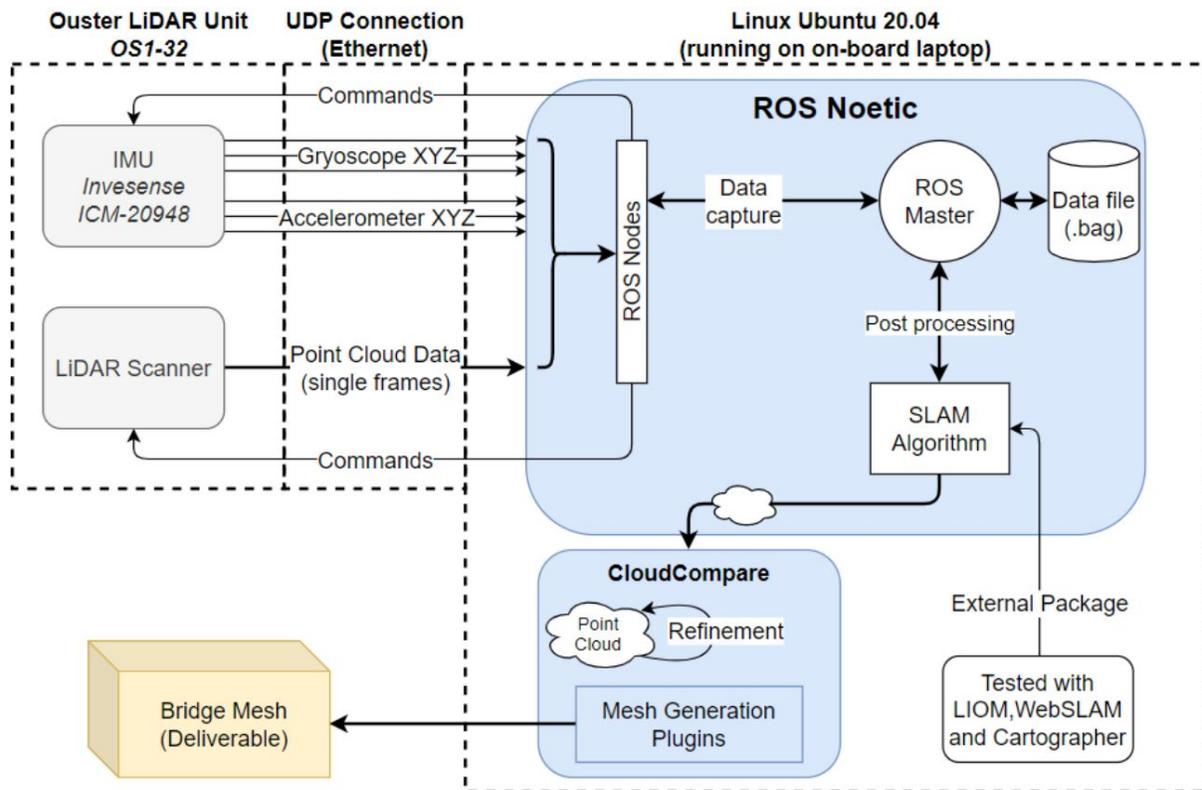


Figure 9: Final Software Architecture

The software pipeline for this project must be capable of transforming outputs of the OS1 into editable 3D Meshes. This covers all stages of data processing performed by the system, beginning from collecting the raw sensor outputs all the way through to generating the final 3D mesh representation of the infrastructure, the deliverable output of our system. The two main stages of the pipeline are data capturing and post-processing. The finalized software pipeline is shown in Figure 9.

The diagram starts from the top left going through the processes in the clockwise manner. First, raw singular point cloud frames, and accelerometer data is collected from the Ouster OS1-32 Gen 2 sensor. The data collection is controlled by an on-board passenger laptop with ROS (the Robot Operating System) installed. ROS forms nodes with the sensor and is used to start and stop data collection. This raw data collected is saved on board as a '.bag' extension file. This file is read during post processing for applying a SLAM Algorithm to it. SLAM, or Simultaneous Localization and Mapping Algorithm combines the accelerometer and singular point clouds to form a single stitched point cloud. These algorithms are available online through external packages such as GitHub projects and open source softwares. The team tried several such packages, including LIO-SAM, LIOM, MATLAB SLAM and Kudan SLAM but found that WebSLAM (also called SLAM-LMAO) provides the best results to Ouster sensor collected data. The trade study and selection process of the SLAM algorithm can be found in Appendix D.

After obtaining a single all-inclusive stitched point cloud as the output from the SLAM Algorithms, the point cloud can be uploaded to a network server to be opened in computers with better hardware capabilities. Once downloaded, the data is sent to Cloud Compare for refinement of the SLAM output.

Here, outlier points are removed, and any misaligned artifacts observed are removed through refinement algorithms in Cloud Compare. This point cloud is then converted into a smooth mesh using mesh generation algorithms available in cloud compare itself, to produce our final output, a mesh of the infrastructure scanned.

### 3.2.2 Design Requirements

Table 3: Software Design Requirements

DR	Requirement	Justification
DR-2.1	The point cloud shall have an instantaneous point density (resolution) of at least 400 points per square meter directly above the sensor.	400 points per square meter is the estimated resolution needed to be able to visually capture the targeted structural faults.
DR-3.1	The system shall implement a GNSS-independent post-processing technique to produce a point-cloud map from the raw data.	This aspect of the design was specifically requested by the customer.
DR-4.2	The on-board computer shall be compatible with Linux.	Linux is required because it is needed to support the ROS framework needed to process the LiDAR data.
DR-7.1	The point cloud shall be used to create a 3D mesh which can be visualized, interacted with, and modified as necessary.	The interactive 3D mesh is meant to make structural analysis easier for the end product user.
DR-9.1	The system shall begin data collection no less than 50 m away from the infrastructure and shall terminate 50 m after infrastructure of interest.	This requirement is meant to minimize the size of the .bag files produced by the LiDAR in order to limit processing time.
DR-9.2	The system shall provide a means of manual data collection initiation and termination via a passenger operated interface.	Manual data collection will allow the passenger to have more control of the scanning process and thus the finalized scan.

### 3.2.3 Finalized Design

The first step in the pipeline is to record and save data collected from the OS1. The FLASH team benefited from the sensor manufacturer, Ouster, releasing a generically written sample ROS-based communication architecture. The FLASH team chose to model the communications architecture based on this, allowing a reliable data acquisition framework to be developed before even receiving the hardware. When the LiDAR is connected via Ethernet, it is assigned an IP address according to the laptop's local network subnet mask, as defined by IEE 802.3 [14]. This allows ROS to communicate with the sensor using the TCP/IP protocol, as defined by RFC 793 [15]. Since the sensor is operating based upon the subnet mask of the laptop, this does not require an external network connection (such as the internet) to successfully assign the sensor package a uniquely addressable IP, meaning the system can be used with the on-board laptop which will be completely "offline". Once the IP address was established, the data captures could be started and stopped manually in the terminal through the "record\_all\_topics\_to\_bag.bash" command.

Post-processing begins with the ingestion of raw sensor outputs from the LiDAR unit and IMU to form a single cohesive point cloud. This is done by manually loading the saved .bag files and JSON configuration files into WebSLAM. WebSLAM is then responsible for fusing individual frames together in order to have a continuous representation of the scanned geometry and the trajectory of the sensor package during data collection for the LiDAR and IMU outputs respectively. It is also responsible for correlating the two together and potentially making corrections to one based on the other's results (this is known as data fusion). The fused data results in a point-cloud representation of the infrastructure and anything else picked up by the LiDAR sensor during the scanning session. Once the point-cloud has been generated via WebSLAM, it is saved as a .ply and cleaned to isolate the desired infrastructure and eliminate any extraneous points.

After the scanned infrastructure is separated from the surroundings, the .ply is passed into a point cloud refinement and visualization tool. Though it may have been possible to keep this within the ROS pipeline,

the FLASH team chose to use the open-source program *CloudCompare*, which offers a comprehensive GUI and feature suite for both refining point clouds and generating mesh representations of the geometry. The refinement process is not intended to downgrade any potential fidelity of the point cloud, so it does not down-sample the point cloud, plane smooth, de-ghost, edge sharpen, etc.. It merely removes any points not a part of the infrastructure geometry, such as background scenery, and remove any obvious outliers caused by poor sensor data. Constructing the 3D mesh involves two main steps of processing: estimating normal planes within the point cloud and constructing surfaces based on those normal vectors. Similarly to the many features within the software pipeline, there are many strategies and algorithms available to achieve these steps, but for the generally flat, well-defined geometric features of bridges and other infrastructure, FLASH determined the best solution for normal vector calculation and surface reconstruction were quarticly iterated normals with an initial guess aligned in the  $+Z$  direction, and a 3-dimensional integration of those normals according to Poisson’s equation respectively [17]. The mesh generations were an involved process that took approximately 30-60 mins for the Comprehensive System Test files. Once the mesh was finalized, it was then exported as an .stl file. The software pipeline from uploading the .bag file to WebSLAM to exporting the .stl took (on average) 1-2 hours.

### 3.3 Electronics and Communication

#### 3.3.1 Overview

The electronics component of this program ensures both the LiDAR sensor and laptop receive the correct amount of voltage from a standard car 12VDC output. The communication aspect focuses on wirelessly uploading .bag files to a drive that can be accessed by the technicians or engineers in order to continue through the software pipeline.

#### 3.3.2 Design Requirements

Table 4: Electronic and Communication Design Requirements

DR	Requirement	Justification
DR-4.1	The system shall accommodate a cumulative data size of at least 64 GB.	A minimum storage of 64 GB is needed to store the LiDAR (especially if multiple scans are taken per trip).
DR-6.1	The power system shall require no more than a 12VDC input.	This requirement makes the design more modular because it allows the system to be operated from a standard car 12VDC outlet.
DR-6.2	The power system shall be capable of supplying at least 175W of continuous steady-state power.	25W of power is needed to operate the LiDAR, and the on-board passenger laptop uses 150W to charge.
DR-8.1	The system shall be capable of transmitting data at a range of 10 meters.	This requirement was specified by the customer.
DR-8.2	The system shall be capable of transmitting data at a minimum rate of 15 Mbps.	The goal of wireless transmission is to save time uploading the data as opposed to handing off a computer or flash drive. Based on the size of the .bag files, it was determined 15 Mbps was the minimum acceptable rate for this.

### 3.3.3 Finalized Design

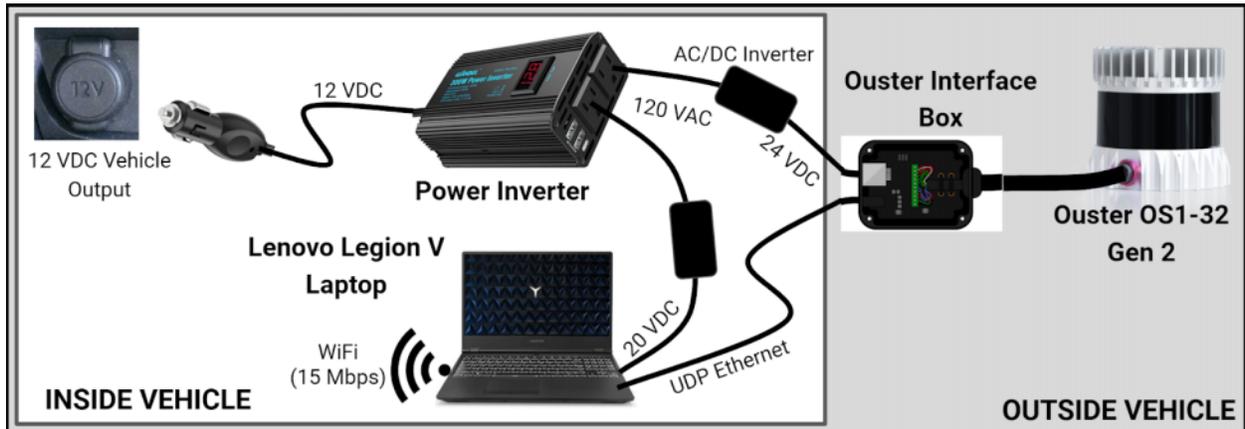


Figure 10: Final Electrical Interfacing Architecture

The final electrical interface design is as follows. The selected power inverter will be plugged into a general 12 VDC vehicle with a auxiliary power lighter adapter. An AC/DC inverter will be plugged into one of the standard 120 VAC outlets on the inverter in order to power the OS1. The Lenovo Laptop will be connected to the other standard 120 VAC outlet on the inverter via the accompanying charger. The Ouster Interface Box will then be connected to the laptop via a UDP Ethernet connection. The Laptop will then be able to transmit data over WiFi at 15 Mbps.

## 3.4 Integrated Design

### 3.4.1 Physical Design

The finalized integrated design will have a passenger sitting with the laptop on their lap. As discussed in Section 3.3.3, the laptop will be plugged into the power inverter which is plugged into the vehicle's 12 VDC auxiliary power outlet. This will ensure the laptop does not run out of battery during data collection. The UDP Ethernet cord that connects the laptop to the Ouster Interface box will be within the car and will also be connected to the power inverter. The cord connecting the OS1 sensor to the interface box will need to pass from the inside of the car to the exterior either through the passenger window or a sun roof. It will be ensured this cord does not obstruct the driver's view of the road or any mirrors. The LiDAR sensor package will be sitting on the roof of the near the front of the vehicle as depicted in Figure 8. The sensor package consists of all sensing equipment needed to collect 3D point cloud data. This includes the Ouster OS1-32 LiDAR system, which has both a LiDAR scanner and a built-in inertial measurement unit (IMU). Table 5 below shows the primary datasheet specifications of the sensor.

Table 5: Key LiDAR Sensor Specifications (Ouster OS1-32)

<b>Max Range</b>	120 m
<b>Precision</b>	+/-1.5 to 10 cm
<b>Field of View</b>	45° (V), 360° (H)
<b>Data Output</b>	8.3 MB/s (66 Mbps)
<b>Power Consumption</b>	14-20 W (Steady State)

### 3.4.2 Software Design

The passenger will start the ROS Master Node from the laptop and then manually start and stop the data collection script at the appropriate distances before and after driving under a bridge. The way the script is set up, stopping the script will automatically save the data as a .bag file in a designated folder. After the trip is complete, the passenger will upload the '.bag' file and the JSON configuration file to WebSLAM in order to output a point cloud ('.ply' file) of the scanned infrastructure. If needed, the passenger could also wirelessly upload the '.bag' files to a drive where they could be accessed by someone else and turned into '.ply' files. This point cloud will then be cleaned in order to distinguish the infrastructure of interest from its surroundings and save the file to the laptop. The cleaned '.ply' file will then be uploaded to CloudCompare and made into a 3D mesh. If needed, this mesh can also be cleaned before being saved as a '.stl' file and sent off for evaluation.

## 4 Manufacturing

*Authors: Jake Fuhrman, Ishaan Kochhar*

### 4.1 Structures

Before the finalized aluminum LiDAR mount was manufactured through CNC, the team wanted to complete a "fit check" using a low-cost, yet durable, 3D-printed prototype. This prototype (shown in Figure 11 as the old mounting configuration) was constructed by the CU Boulder Smead Aerospace Machine Shop staff from a CAD file supplied by the team. This 3D print was made from a Nylon/Carbon Fiber composite material for minimal mass and maximum durability. This fit check proved that the LiDAR would fit into an aluminum mount of the same dimensions. However, the hole allocated for the interface box cable was proven to be too small and improperly oriented. This prompted the team to make a design change, lowering the flat wall containing the insertion hole (as will be seen in Figure 14). It was decided at this point to continue with the aluminum mount concept, since a baseline thermal analysis of the 3D printed mount showed that it would not conduct heat away from the LiDAR unit as well as a mount made from aluminum. Furthermore, the 3D-printed mount did not contain the same level of durability as possessed by aluminum. During a preliminary drop test of the aluminum mount (without the LiDAR sensor attached) proved that it was structurally compromised around the thin circular area near the top LiDAR face, as seen in Figure 12.



Figure 11: 3D-Printed Nylon/Carbon Fiber Composite Fit Check Mount



Figure 12: 3D-Printed Nylon/Carbon Fiber Crack

Figure 13 shows the nominal structural manufacturing flow diagram for the project. Once the 3D-print fabrication, fit check, and design adjustments had all been made, a CNC request was submitted to the Aerospace Machine Shop for the aluminum mount. The aluminum material was sourced from machine shop scrap metal, and therefore did not require any purchase or pre-processing. Once all five aluminum components were machined (as seen in the middle right side of Figure 13), they were screwed together, using the machined screw holes and purchased threaded screws. The magnets (which were traded to contain built-in threaded screws) were inserted into the bottom aluminum plate and fastened. The completed aluminum mount was placed onto the mission vehicle to assure secure attachment before the LiDAR sensor was screwed into place.

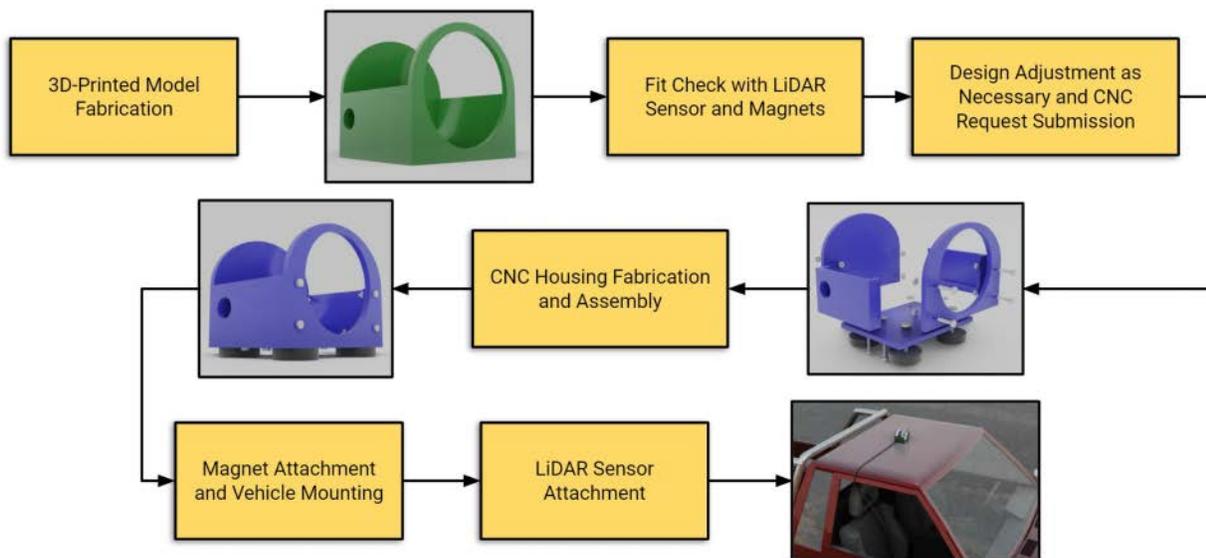


Figure 13: LiDAR Mount Manufacturing Flow Down

Figure 14 shows the LiDAR sensor attached to the aluminum mount (old configuration). There was a slight discrepancy in the locations of the machined screw holes for the LiDAR. Therefore, the team drilled

larger holes which would accommodate the LiDAR built-in screw locations. The sensor was then fastened to the aluminum mount. The aluminum wall on the left side of Figure 14 was lowered in the machining process to accommodate the interface box cable, leaving it sufficient space for connection to the LiDAR sensor.



Figure 14: Manufactured Aluminum Mount (Old Configuration)

Figure 15 shows the modified aluminum mount after the LiDAR orientation change was made. This new mount utilized the bottom aluminum plate with new screw holes drilled in for the LiDAR sensor. A benefit of this design is its versatility, namely that the materials and screw holes exist for either a vertically or horizontally configured LiDAR.



Figure 15: Manufactured Aluminum Mount (New Configuration)

## 4.2 Electrical

The electronics subsystem (as outlined in Figure 10) consisted of only purchased, COTS components. The interfacing architecture itself was "manufactured" by the team, since all the required connections were unique to this project.

Figure 16a shows the DC/AC Power Inverter interfacing with the vehicle's 12 VDC power output. This power source activates the power inverter, which then is connected to the LiDAR interface box cord, as seen in Figure 16b.

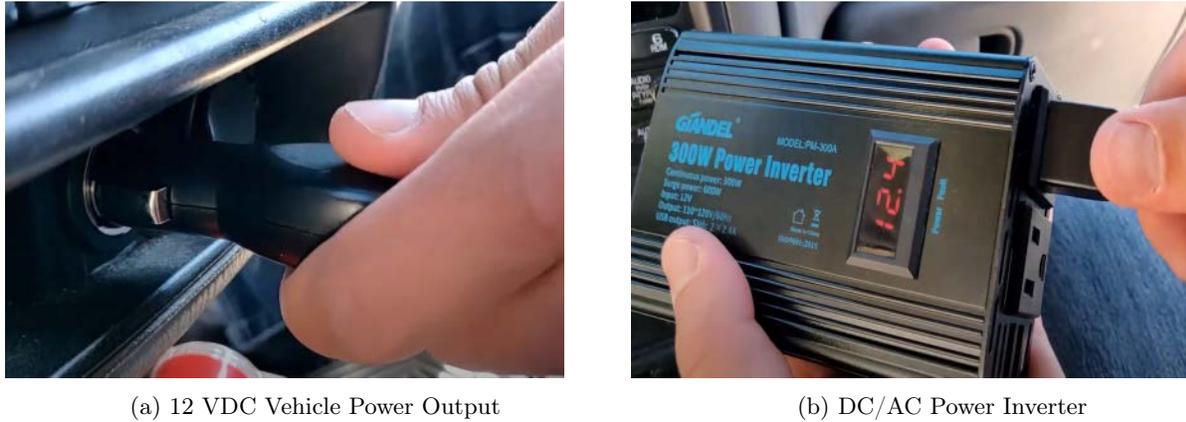


Figure 16: Electrical Interfacing from the Vehicle to the DC/AC Power Inverter

The electricity through the LiDAR interface box power cord then travels through an AC/DC converter and onto the interface box, as seen by the black cable in Figure 17a. The blue Ethernet cord in this Figure travels to the mission laptop, as seen in Figure 17b. Not pictured here is the interface box cord connection with the LiDAR sensor (shown in Figure 15) and the power connection for the laptop charger (plugged into the laptop and the second AC power outlet on the power inverter).

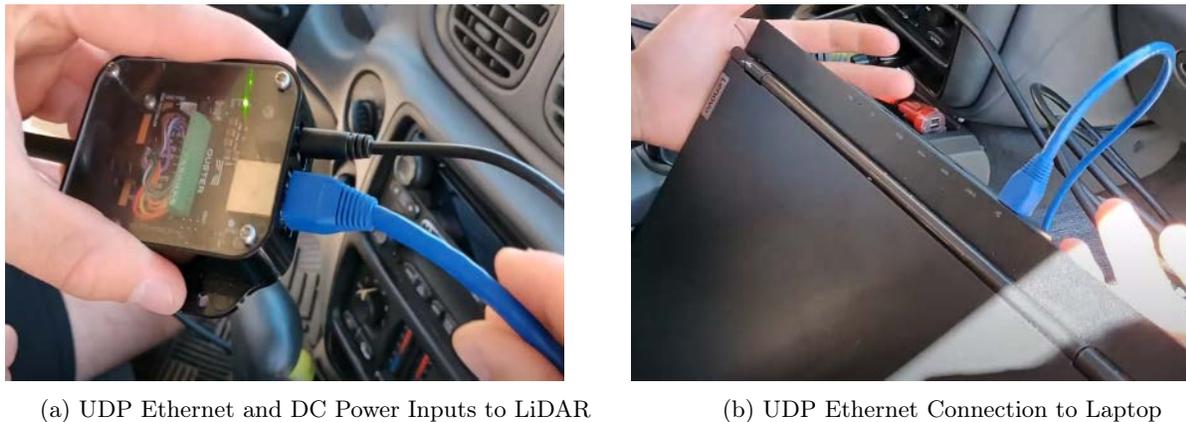


Figure 17: Electrical Interfacing from the Interface Box to the Laptop and DC/AC Power Inverter

## 4.3 Software

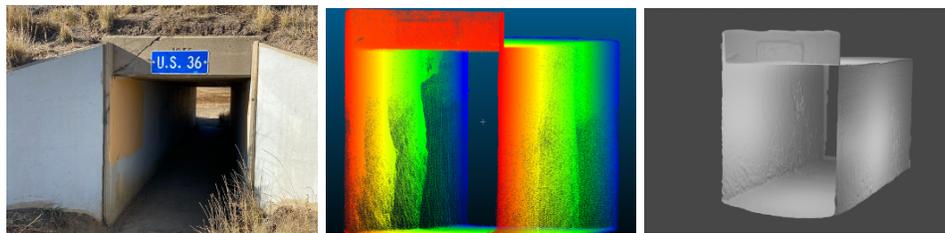
The data that was collected by our system was sent through a specific software pipeline to generate the final deliverable. This pipeline involved interfacing between the data collection and data storage devices,

processing the data into a 3D point cloud, isolating the target infrastructure within the point cloud, and constructing a mesh from the refined point cloud output.

For interfacing between components, FLASH used ROS (Robot Operating System). ROS is a generic networking interface tool, designed specifically to consolidate and streamline communications to and from hardware that is used commonly in robotics, including sensors, processing routines, and data acquisition tools. It has support for all mainstream networking protocols and contains a host of tools for representing data in a compatible format to the protocol of choice, all without the user having to engage with the complexities of interfacing between hardware and software. As per the functional and design requirements, the FLASH system was not limited in the choice of auxiliary sensors to aid the LiDAR unit's localization capabilities when processed by a SLAM algorithm, with the exception of a GNSS receiver. The team determined that an IMU sensor would benefit the raw LiDAR data, and as such necessitated the software be capable of handling data from multiple isolated sensors. ROS had the ability to coordinate this requirement.

The data that was collected by the LiDAR and IMU was consolidated and used to produce a single cohesive point cloud representation of the target infrastructure. The team performed a trade study on multiple SLAM algorithms to determine the version that would be best suited for the application. The SLAM algorithm is responsible for combining individual frames together in order to create a continuous visualization of the scanned geometry and the trajectory of the sensor package during data collection. The SLAM algorithm correlates the outputs from both the LiDAR and IMU sensors and attempts to make corrections to one based on the other's results. The output of SLAM algorithms is a single point cloud which contains aligned frames based on the trajectory calculated from the IMU data, computer-vision based alignments of geometric features within the point cloud data, as well as corrections made between each during the SLAM process.

The final step in the software pipeline is refining the SLAM-generated point cloud map. As mentioned in the *Finalized Design, Software* section, the process of "refinement" refers to removing unwanted points and outliers, *not* reducing the fidelity of the scanned infrastructure data in any way. Generating a mesh involves identifying candidate points that lie on the outermost surfaces of geometries and fitting plane normals which define the surface's orientation and size. This normal estimation and the actual generation of the surfaces is generally performed in two separate steps, each with parameters available to tune depending on the nature of the data being produced. Figure 18 depicts this visualization process of the software pipeline. While the point cloud in this image was not generated from the OS1, the data follows the rest of the software pipeline flow.



(a) Photograph of underpass. (b) Point cloud of underpass. (c) 3D mesh of underpass.

Figure 18: Stages of mesh generation of US 36 Underpass.

During development and testing, the FLASH team experimented with several combinations of both SLAM and meshing packages. As described in the *Testing* section, the team ultimately was able to get usable results which met our accuracy and precision requirements only after iterating upon the software pipeline's design several times during implementation, namely the specific SLAM and meshing algorithms used as well as how their respective parameters were tuned.

## 4.4 Manufacturing Outcomes and Challenges

### 4.4.1 Hardware

Through manufacturing the housing structure, several challenges were met with and faced. The first such challenge was utilizing the 3D-printed model as a baseline design. Such a design was considered, but ultimately became too faulty to utilize for the final product. The 3D-printed model had issues in crack propagation while being drilled into. It was a brittle material that ended up being insufficient for the project’s needs thermally and structurally. The next manufacturing challenge included the LiDAR orientation change. The original aluminum mount was a five plate CNC design that had to be screwed together and only functioned for a vertical (i.e. FOV looking up) LiDAR orientation. This challenge was overcome by removing extraneous plates and leaving only the bottom plate. Screw holes were then drilled into this plate for the LiDAR to attach vertically (i.e. outwards-looking field of view). This simple change saved materials and also added project versatility. Depending on the nature of the LiDAR scan, the housing structure can now accommodate both a vertical and horizontal field of view, while having no wasted parts or a requirement for new parts.

The end product on the structures side included a versatile, machined aluminum mount to support the LiDAR sensor. Through baseline testing and analysis, a 3D-printed module was used preliminarily for sensor fit checks, but proved to be ineffective in the unlikely event that the structure was knocked off the car.

### 4.4.2 Software

Manufacturing the FLASH software pipeline was a multi-step process with each facing its own unique set of difficulties. This was broken up into three rough stages of development, beginning with implementing the communications software, followed by configuring SLAM packages, and generating meshes once we were successfully generating point clouds.

As mentioned previously, the choice of implementing a ROS-based architecture streamlined the development of the communications code so the hardware could be interfaced with immediately upon arrival. This software was primarily designed to function according to the sample ROS communication framework using TCP, provided by the sensor manufacturer, Ouster. This proved to be a great success, as the team was able to successfully design and implement the majority of this code before even receiving the sensor package. Only minor modifications were necessary to enable data collection within less than 24 hours of receiving the sensor package. Once data was able to be obtained from the live sensor, the software team swiftly transitioned to focusing performing the first rounds of tests. These preliminary tests focused on verifying the sensor specifications, as well as confirming the data acquisition software was stable while deployed in the field, and overall was considered a success.

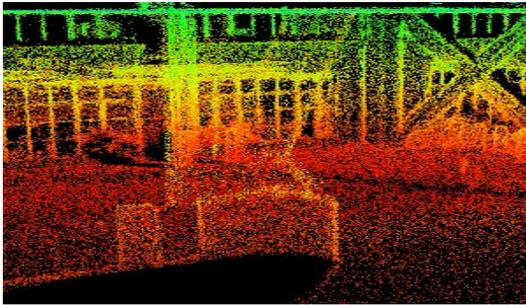
The next phase of development was implementing and configuring the SLAM algorithm which would be responsible for taking raw sensor data and generating a single cohesive point-cloud representation of the scanned infrastructure, as described in previous sections. This introduced the first round of challenges which the software team faced, involving gaining an in-depth understanding of inner mathematical workings of the various SLAM packages we ended up testing. It was found that the originally selected algorithm, LIO-SAM[9] was fundamentally incompatible with the sensor package’s 6-axis IMU due to requiring an extra 3 magnetometer axes to calculate absolute roll, pitch, and yaw<sup>1</sup> which was only necessary if LIO-SAM was configured to perform loop-closure using GNSS data, neither of which we were going to be incorporated into FLASH.

From here the software team focused on LIO-SAM’s predecessor package, LIO-Mapping[13], also known as LIOM, which was compatible with a 6-axis IMU by incorporating trajectory pre-integration routines borrowed from VINS-Mono [14]. This was able to adequately calculate trajectories using the IMU data, however completely failed to produce viable maps. This was due to an inadequate feature map within

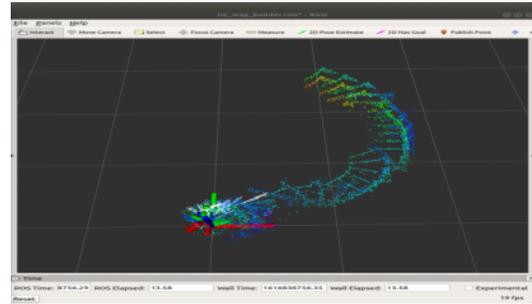
---

<sup>1</sup>During the design phase of this project the FLASH team did not anticipate this being an issue due to the included IMU being listed as outputting 9-axes according to its datasheet[12]. However, it was later discovered that the OS1 is unable to output the IMU’s magnetometer data channels due to the large electromagnetic disturbances caused by the system’s brushless motor.

the LiDAR data itself. In LIOM's case, a view of the ground plane was required at all times in order to adequately orient itself within the 3-dimensional environment due to the vertical sensor orientation only ever viewing bridge undersides.



(a) LIOM's expected output from research.



(b) Our LIOM output consequence of no 9 axis IMU or ground reference.

Figure 19: Comparison of expected LIOM output to actual.

Upon trying more SLAM alternatives, it was discovered that due to the nature of aligning computer-vision based geometrical features in a pose graph rather than doing any direct point cloud alignment, modern LiDAR-based SLAM implementations struggle to perform any sort of localization if the feature map is sparse in any of the three spatial dimensions. This required the software and hardware teams to collaborate on how to revert the mount to be compatible in a horizontal orientation as well, as discussed in *Manufacturing Difficulties, Hardware*.

Ultimately, the SLAM team found that the SLAM algorithm developed by the sensor manufacturer Ouster, named SLAM-LMAO, provided the best results for our hardware configuration without requiring excessive tuning of interdependent parameters. Although this is one of the only proprietary pieces of software to make its way into the final FLASH software pipeline, the software team had an opportunity to work with Ouster engineers directly to understand the inner workings of the package and optimize our data collection to suit its functionality. In brief, SLAM-LMAO behaved somewhat similarly to LIOM in that it performed data fusion using an EKF (Extended Kalman Filter), IMU pre-integration, and rotation constraints using planar orientation. This combination of changes finally produced viable point cloud results from our sensory data.

Once we had complete point cloud representations of the scanned infrastructure, the team began implementing and tuning the mesh generation algorithms. Similarly to the SLAM algorithms, this process ended up having several packages experimented with throughout the development phase, each with their own unique set of interdependent parameters which needed to be tuned. This introduced yet another hurdle towards the software pipeline's completion, but upon gaining more experience with refinement techniques and the different meshing algorithms, the team was able to get reasonably accurate results using Poisson surface reconstruction, as described in the *Finalized Design, Software* section.

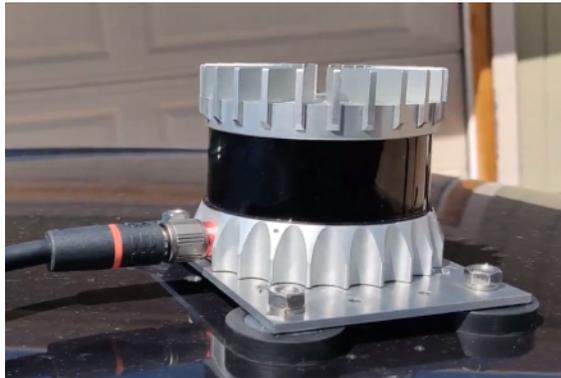
As shown in the *Testing* section, within the limits of reasonable computation times and hardware required, the FLASH team was able to successfully generate meshes using the many CST datasets. This generated a mesh representation of the infrastructure with both impressive dimensional and visual accuracy, but the FLASH team ultimately needed to conclude that the process of generating meshes from point clouds *for the purposes of granular structural analysis* is unviable for our system due to the many false-positives and false-negatives that are inherently produced by closing surfaces over unrefined point clouds.

The team was still confident that the data, in both point cloud and mesh form, has several practical alternative uses that are closely related to the original goals. For example, possessing a comprehensive 3-dimensional visualization tool to assist in the planning and execution of surveys (that are conducted with "traditional" surveying equipment), which in context of FLASH's functional requirements this would be an extremely useful tool for inspectors that would undoubtedly save significant net time in the overall inspection

and assessment of infrastructure. Through surmounting all of the aforementioned obstacles the software team learned many things about DAQ, SLAM, meshing, and software development practices. The software pipeline’s design benefited from the many iterations in packages used demanding increasing generality in its design, the final product being highly modular, robust, and compatible with several mainstream hardware and software packages. Overall, the software team believes the project’s final implementation a great success.

## 4.5 Integrated Design

The fully integrated system is split into two locations, the exterior and interior of the mission vehicle. The exterior of the mission vehicle, as seen in Figures 20 and 21, includes the LiDAR sensor, which is connected to the aluminum, magnetic mounting structure. There is a black cord that connects to the LiDAR sensor and runs through the mission vehicle’s passenger window. **NOTE:** This means the passenger window must be unrolled by approximately one inch during scanning operations.



(a) Aluminum Mount (Zoomed-In)



(b) Aluminum Mount (Zoomed-Out)

Figure 20: Aluminum LiDAR Mount for Magnetic Attachment to the Mission Vehicle (with Associated Interface Box Cable)

The black cord that runs through the passenger-side window then connects to the interface box, which is placed in a location such that it will not interfere with the passenger and such that the passenger will not inadvertently interfere with it. The interface box then has two cable outputs, one for power and one for Ethernet data transfer. These cords run to the mission vehicle’s auxiliary power supply (via the power inverter) and the mission laptop, respectively (these connections are shown in Figures 16 and 17). The mission laptop, highlighted in Figure 22, is responsible for sending commands to the LiDAR sensor, data storage, and post-processing of data.



Figure 21: Full Functioning System on Mission Vehicle (External View)

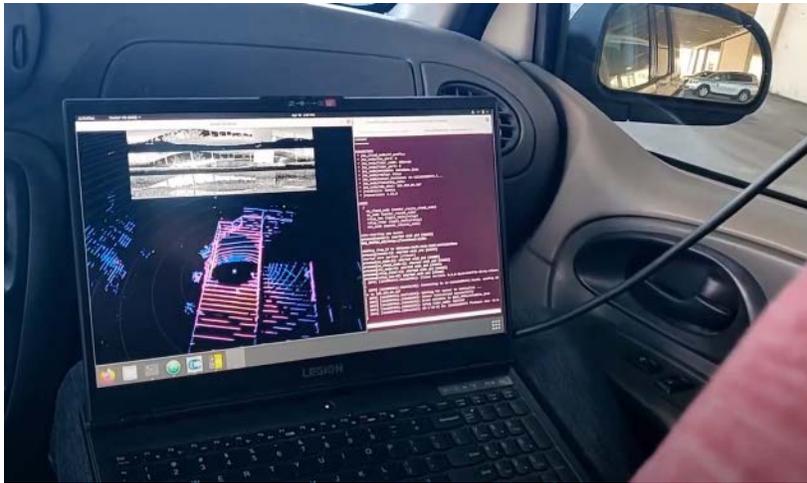


Figure 22: Full Functioning System on Mission Vehicle (Internal View)

The functioning system has two significant external interfaces, the mission vehicle and the occupant of the passenger seat. The mission vehicle is responsible for providing 12 VDC power and a magnetic mounting surface (via the roof). The passenger is responsible for operation of the mission laptop, with essential tasks of initiating/terminating data collection and saving raw point cloud data.

## 5 Verification and Validation

*Authors: Shray Chauhan, Jake Fuhrman, Ishaan Kochhar, Fiona McGann*

### 5.1 Mount Pull Test (MPT)

The requirement for the LiDAR sensor housing structure include:

- DR 5.1: The mounting structure shall withstand drag forces associated with a vehicle speed of no more than 65 mph.

### 5.1.1 Test Purpose/Objective

In order to satisfy this requirement, a Mount Pull Test (MPT) was conducted to ensure that the structure could withstand drag forces far exceeding 65 mph. To model this theoretical drag force associated with wind speeds of driving 65 mph, a pull test was designed. This drag force was adapted to the cross-sectional area of the housing structure, and the associated drag force (with 1.5x built-in factor of safety) was calculated to be 1.6 lbf. In summary, a team member will subject the housing structure and 1 lb dummy weight (to simulate the LiDAR sensor weight), to an axial force in the direction with which wind would affect the exterior system while driving. The objective of the pull test is to effectively demonstrate that the housing structure can survive an axial force of 1.6 lbf, showing that it is above a factor of safety of 1.5 for wind drag associated driving speeds of 65 mph.

### 5.1.2 Test Equipment and Setup

The equipment required for this test included:

- Dummy LiDAR Weight - plastic bag with coins weight 1 lb
- Hook scale - borrowed from CU Smead Aerospace Machine Shop
- Food scale - to ensure Dummy LiDAR Weight reaches 1 lb
- Plastic 3D-printed housing structure
- Purchased COTS magnets screwed into bottom of housing structure
- Attachment mechanism from hook scale to bottom of housing structure - belt
- Vehicle for structure to attach to

The setup for the MPT included first taking the 3D-printed plastic mount and attaching the COTS magnets to it. Then, the dummy weight was developed by adding coins to a bag until the food scale reading indicated it weight 1 lb. The bag was then added to the plastic structure and mounted on top of the vehicle used for all subsequent testing. Lastly, the attachment was hooked around the magnets at the bottom of the structure and connected to the hook scale. The setup can be visualized in figure 24.

### 5.1.3 Test Operation and Results

Once setup was complete, a team member would pull axially on the hook scale, which would measure the amount of force the mount was subjected to. This would simulate the drag force associated with driving at high speeds. The pull test was conducted such that a team member would pull in 5-10 lb increments (based on hook scale readings, see Figure 25), and another team member would observe whether or not the housing structure displaced. Though this observation is qualitative, the team felt that it characterized the test well, since the required force to withstand was already so low. These values are included in Table 6:

Table 6: Mount Pull Test Results

Hook Scale Reading	Observations
5 lb	Sturdy (no slippage)
10 lb	Sturdy (no slippage)
20 lb	Sturdy (no slippage)
30 lb	<b>Earliest observed slipping</b>
35+ lb	Steady, consistent slipping as load increases

The end result of the pull test showed earliest signs of the housing structure displacement at a hook scale reading of 30 lb. This means that the housing structure was validated for drag forces up to a factor of safety of approximately 30, far exceeding the original factor of safety of 1.5.

#### 5.1.4 Model Verification and Requirements Satisfaction

The pull test is based off of the modeled drag force caused by relative wind on the top of the moving vehicle. The four magnets on the base must be able to resist the force of this relative wind ( $F_w$ ) that is given by Equation 1. The horizontal model required that the magnets be able to withstand 1.0809 lbf (4.8079 N) to travel at highway speeds of 65 mph.

$$F_w = \left(\frac{1}{2}\rho v^2\right) \cdot A \quad (1)$$

The pull test was originally conducted with the LiDAR still in its horizontal orientation (see Figures 23 and 24).

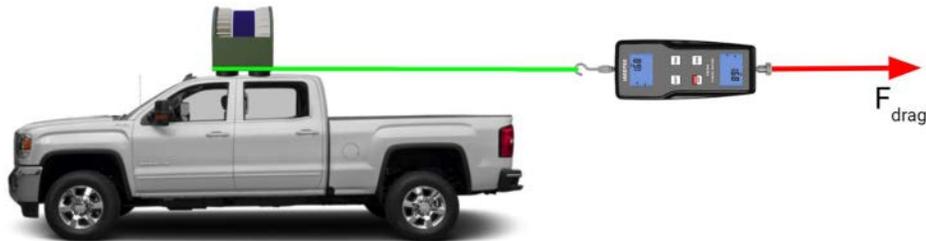


Figure 23: Pull Test Free Body Diagram



Figure 24: Pull Test Setup

In each trial of the pull test, the mount remained motionless up to 30 lbf as measured by a manual hook scale provided by CU's Aerospace Engineering Department. In the three conducted tests, slippage began to occur between 30-35 lbf and consistently slipped after 35 lbf was applied. Figure 25 shows the recorded hook scale measurement after slippage first occurred on the first pull test.



Figure 25: Pull Test Force Measurement

Since the pull test was conducted, the housing mount was modified to accommodate a vertical orientation (see Section 3.1). However, the only change this would have in the model is the exposed area of the structure to the wind force. The area change was so small that it did not significantly impact the required resistance force of the structure. Keeping in mind the original determined factor of safety was around 30, FLASH did not find it necessary to reconduct this test with the new orientation. After multiple Comprehensive System Tests recorded no slippage of the mount, this decision was validated.

## 5.2 Small-Scale LiDAR Testing (SSLT)

### 5.2.1 Test Purpose/Objective

Small-Scale LiDAR Testing (SSLT) aimed to determine sensor performance in both static and dynamic states in regard to design requirements. The design requirements to be verified by this test included:

- DR 1.1: Maximum measurement range of greater than or equal to 30 m.
- DR 2.1: Point spacing (resolution) of less than or equal to 5 cm (point density of greater than or equal to 400 pts/m<sup>2</sup>).
- DR 2.2: Point cloud accuracy less than or equal to 10 cm as compared to ground truth data (the test board).
- DR 2.3: Point cloud precision less than or equal to 10 cm (variation in individual point locations between trials).

In addition to static and dynamic operational modes, the testing setup allowed for both sunlit and shaded conditions to determine the correlation between noise in the data and amount of ambient sunlight on the test board. Further discussion of the test equipment and setup is included next.

The SSLT served as a predecessor to the Comprehensive System Test (CST) - which aimed at verifying the same requirements but without the controlled testing space as given in the SSLT. The objective here was to determine the capabilities of the LiDAR sensor and as a result, set expectations and limitations for scanning real bridges and underpasses.

### 5.2.2 Test Equipment and Setup

Listed below is the equipment that was required for conducting this series of tests:

- Ouster OS1-32 Gen 2 LiDAR Sensor
- Target Test Board
- Lenovo Legion V Laptop
- LiDAR Data Cable
- LiDAR Power Cable
- Tape Measure

Prior to testing, the LiDAR sensor was plugged into a wall outlet for power and connected to the laptop for data capture. The manufactured target test board was mounted off the ground and the sensor was oriented normal to the board surface in preparation for scanning. Once the Ouster/ROS interface on the laptop was initialized for data viewing and storage, the sensor was held at a specified distance from the board (measured via tape measure) so that scanning could commence. Details of the test operation and procedure are outlined in the following section. Testing was conducted inside a garage with ambient sunlight present to simulate conditions expected under a bridge in daytime. Figures 26 and 27 depict the test setup.

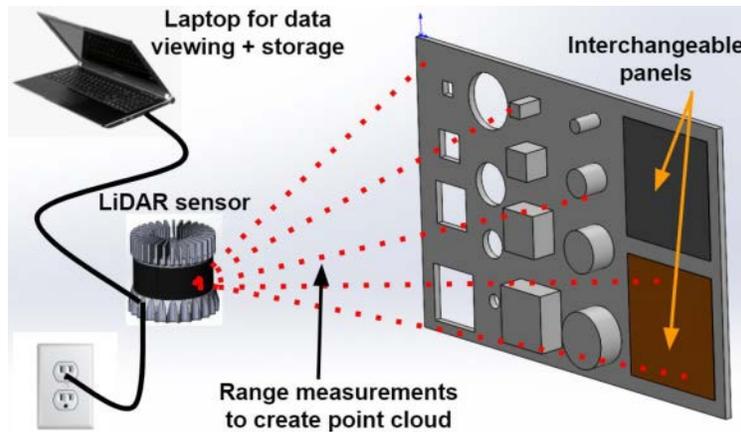


Figure 26: Small Scale LiDAR Test Setup (Schematic)



Figure 27: Small Scale LiDAR Test Setup (Actual)

### 5.2.3 Test Operation and Results

Trials were conducted with the LiDAR sensor at four different distances (1.5 m, 3 m, 3.5 m, 4.5 m) from the target test board. For each distance, the board was scanned once with the sensor stationary, twice with the sensor translating horizontally parallel to the board, and twice with the sensor translating vertically parallel to board. Activation and termination of data collection was controlled via ROS Noetic code (Ubuntu 20) on the laptop. After each scan, raw data was saved in a .bag file for later point cloud processing. Once all trials were completed, the .bag files were processed through a SLAM package to convert the raw data into usable 3D point clouds. Next, the point clouds were cleaned and cropped in CloudCompare to isolate points on/near the target test board. In order to evaluate the designated LiDAR performance metrics (accuracy, precision, resolution), an in-built tool was used to measure the dimensions of the test board (width and height). Additionally, the distance between adjacent scan points was measured to estimate resolution (point spacing). An example measurement for a single point cloud is shown in Figure 28.

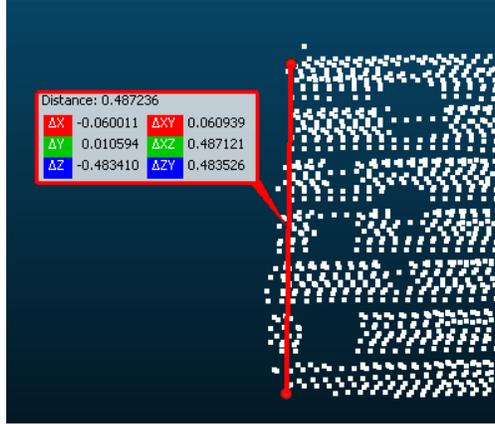


Figure 28: Sample Point Cloud Measurement (Board Height)

The smallest board feature that could be perceived/identified for each trial was also noted. Accuracy/error was determined by comparing the known text board dimensions to their corresponding point cloud measurements. For example, with a known test board width of 76.2 cm, a point cloud width measurement of 76.4 cm yields an accuracy (or error) of 0.2 cm. An accuracy value was computed for each distance and scanning condition (i.e., stationary or dynamic). On the other hand, precision (repeatability) was determined by comparing identical point cloud measurements across multiple equivalent trials. For example, if the board was measured to be 51.6 cm high in the point cloud generated from a horizontal sensor translation trial, and 51.7 cm high in the next horizontal sensor translation trial, the resulting precision is 0.1 cm. As mentioned earlier, resolution was determined by measuring the horizontal spacing between adjacent points. Results for accuracy and precision are tabulated in Table 7 below, while results for resolution are displayed in Figure 29. Note that the leftmost column in Table 7 indicates scan distance.

Table 7: SSLT Results: Test Board Measurements for Determination of Accuracy and Precision

		Horizontal Sensor Translation			Vertical Sensor Translation		Mean Error	Precision (Variation)
		Static Trial	Dynamic Trial 1	Dynamic Trial 2	Dynamic Trial 3	Dynamic Trial 4		
1.5 m	Board Width	76.4 cm	77.4 cm	75.7 cm	76.6 cm	73.6 cm	0.98 cm	1.29 cm
	Board Height	55.1 cm	55.4 cm	54.9 cm	59.2 cm	57.8 cm	5.68 cm	1.72 cm
3 m	Board Width	75.0 cm	75.5 cm	75.3 cm	77.1 cm	75.9 cm	0.80 cm	0.73 cm
	Board Height	52.9 cm	50.3 cm	51.8 cm	53.4 cm	54.6 cm	2 cm	1.49 cm
3.5 m	Board Width	77.3 cm	75.7 cm	75.7 cm	76.1 cm	74.9 cm	0.70 cm	0.78 cm
	Board Height	53.5 cm	60.1 cm	53.5 cm	51.1 cm	58.6 cm	4.56 cm	3.41 cm
4.5 m	Board Width	75.4 cm	78.8 cm	75.3 cm	79.6 cm	74.1 cm	1.92 cm	2.15 cm
	Board Height	46.9 cm	51.5 cm	59.7 cm	57.9 cm	56.1 cm	5.18 cm	4.65 cm

The mean error values in Table 7 were computed by averaging the errors of each individual measurement (rather than a singular error of the average measurement). The precision values were computed by taking the standard deviation of the board measurements for each distance. The results demonstrate that the measurement error stayed within the 10 cm accuracy requirement (DR 2.2) for all trials. Similarly, the variation in measurements across trials at the same distance did not exceed the 10 cm precision requirement (DR 2.3). Best estimates for accuracy and precision were determined by taking the mean of the values the last two columns of Table 7. The best estimate for accuracy is 2.73 cm ( $\sigma = 1.94$  cm) and the best estimate for precision is 2.02 cm ( $\sigma = 1.27$  cm). Note that these values are slightly different than those reported at SFR because these are averaged over multiple distances. The apparent deviation can be attributed to the difference in error between the board height and width measurements. The board height measurements were consistently observed to have worse accuracy than the board width measurements. This can likely be attributed to the fact that the LiDAR sensor scans more points horizontally (parallel to board width) than vertically (parallel to board height).

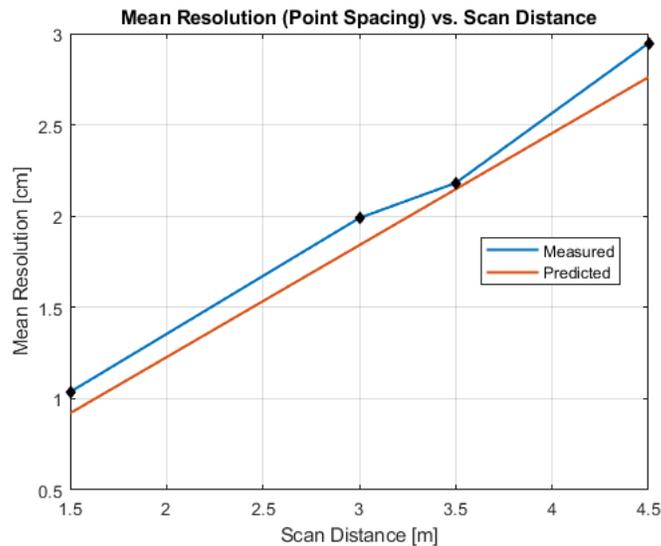


Figure 29: SSLT Results: Horizontal Resolution Measurements Compared to Model Prediction

The results plotted in Figure 29 show that the measured horizontal point spacing (resolution) remained below the 5 cm requirement and was in agreement with the prediction. The predictive model in this case was a simple trigonometric calculation which uses knowledge of the sensor’s angular resolution to determine spacing between points at any given distance. The formula for point spacing is given by  $2d \tan(a/2)$  where  $d$  is the scan distance and  $a$  is the sensor’s horizontal angular resolution.

In terms of feature extraction from test data, it was found that the smallest test board elements, which were 2.5 and 5 cm in size, were roughly visible in the point clouds up to 4.5 meters away, but their geometry was best captured at distances inside about 3.5 meters. This corresponds to Level 2 project success. In terms of more qualitative results, it was found that the material panels with the higher reflective properties yielded the least noise and "point wiggle", which is in agreement with expectations. Also, prior to the main set of trials, preliminary data was collected in sunlight and shade for comparison of noise. While the sunlit condition demonstrated slightly more noise, the overall data quality was not affected significantly – this is why the primary testing (as previously mentioned) was conducted in shade inside a garage. From an operational standpoint, the stationary test verified the on-board computer’s ability to store data, handle data, and interface between components.

### 5.2.4 Model Verification and Requirements Satisfaction

The primary "models" against which the design was verified through this testing include expected sensor performance specifications from the manufacturer-provided datasheet and the trigonometric model developed for horizontal point spacing. Table 8 summarizes how the SSLT results validated functional requirements and overall project success criteria.

Table 8: Satisfaction of Design Requirements from SSLT Results

DR	Satisfied?	Associated CPE	Associated Level of Success	Explanation
1.1 Max Range	YES	CPE-1	N/A	Data collection successful at distance beyond 30 m requirement
2.1 Resolution	YES	CPE-1 CPE-2	Level 2	Point spacing less than 5 cm requirement for scans up to 4.5 m
3.5 m Accuracy	YES	CPE-1 CPE-2	Level 2	Average measurement error fell within 10 cm requirement
4.5 m Precision	YES	CPE-1 CPE-2	Level 2	Measurement variation across trials did not exceed 10 cm requirement

## 5.3 Comprehensive System Testing (CST)

### 5.3.1 Test Purpose/Objective

Comprehensive System Testing (CST) aimed to determine the performance and capabilities of the fully integrated system. This test verified that the system could collect raw data and process it into a deliverable 3D mesh. Project elements that were validated by this test include:

- Magnetic attachment of mount
- All electrical interfacing
- LiDAR 3D point cloud data collection
- Saving/registering point cloud data
- Generating a 3D mesh model

These project elements map to specific design requirements as referenced below:

- DR 1.2: Scanning bridges at least 5.1 m (16.7 ft) in vertical clearance above the road.
- DR 1.3: Scanning coverage width of 7.2 m (24 ft).
- DR 2.1: Point spacing (resolution) of less than or equal to 5 cm (point density of greater than or equal to 400 pts/m<sup>2</sup>).
- DR 2.2: Point cloud accuracy less than or equal to 10 cm as compared to ground truth data (from Google Earth).
- DR 7.1: Interactive 3D mesh generation.

The requirements outlined above are only a key selection of the DR's verified by this test. This CST directly tests verification of each design requirement in the project, since each DR involves specifications related to an individual component of the system and/or the system as a whole.

While SSLT aimed to verify DR's 2.1 and 2.2 in a controlled environment, the CST was meant to verify these same requirements in a real, day-in-the-life of the system. This is because there are several factors in the field (road conditions, lighting conditions, vehicle speed, etc.) that can not be simulated by SSLT.

To verify the above requirements, several predictive models were used for comparison. The Google Maps API provided ground truth data to which our accuracy requirement (DR 2.2) could be verified (or not

verified). The Ouster OS1-32 Gen 2 Datasheet provided LiDAR-specific resolution, precision, and accuracy values which could be compared against. The National Cooperative Highway Research Program (NCHRP) LiDAR Infrastructure Scanning guidelines (included in the Appendix) provided expected point cloud density values (DR 2.1) to be associated with a "high accuracy and fine density" bridge inspection.

### 5.3.2 Test Equipment and Setup

Listed below is the equipment that was required for conducting this series of tests:

- Ouster OS1-32 Gen 2 LiDAR Sensor
- LiDAR Interface Box Cable
- Lenovo Legion V Laptop
- Lenovo Legion V Laptop Charger
- LiDAR Data/Ethernet Cable
- LiDAR Power Cable
- Magnetic Aluminum Sensor Mount
- DC/AC Power Inverter (for the mission vehicle 12 VDC output)
- AC/DC Power Inverter (for the LiDAR)
- Mission Vehicle

To begin, the LiDAR sensor was secured atop the designated testing vehicle via the magnetic aluminum mount. The LiDAR Interface Box cable was routed through the passenger-side window into the vehicle for connection with the interface box. Other connections (power and data) were established between the laptop, the interface box, and the power inverter. Next, the ROS code and live data stream visualizer were launched on the laptop (by the passenger - two team members are required to operate this test, one driver and one laptop operator/passenger) to verify active communication with the sensor and a quick scan was taken to ensure that .bag files were saved correctly and to ensure the software pipeline was performing nominally. The steps outlined here were conducted in the safety of a team member's driveway prior to deployment. The finalized setup outside and inside the mission vehicle can be seen in Figures 21 and 22, respectively.

### 5.3.3 Test Operation and Results

Because the Comprehensive System Tests (CSTs) were designed to emulate a "day-in-the-life" of the FLASH system, the general testing procedure closely followed the mission CONOPS (Figure 3). Following system setup/activation (outlined in the previous section), the testing vehicle was deployed to the chosen infrastructure of interest: Foothills Parkway underpass on Walnut Street in Boulder (40°01'09.9"N, 105°14'38.9"W). A Google Street View image of this bridge is shown in Figure 30. Note that the bridge has a maximum clearance of 6.7 meters and an approximate width of 46.63 meters.



Figure 30: Underpass at Foothills Pkwy. and Walnut St. in Boulder, CO

Beyond the convenience of its proximity, this bridge was specifically chosen because it does not receive much traffic, so low-speed testing could occur without posing a safety risk. Upon approach to the bridge

(approximately 50 meters away), LiDAR data collection was initiated on the laptop via a ROS command. Scanning continued as the vehicle passed under the bridge and was subsequently terminated via another ROS command once the bridge was behind the vehicle. Capture of the .bag file (containing raw point cloud data) was verified by checking the predesignated storage folder and the file was then renamed for later access. Multiple CSTs were conducted over the course of about 4 weeks so that the results from different scanning conditions (vehicle speed, day/night, etc.) could be characterized. Upon completion of field testing, the vehicle was returned to the setup location for system deactivation and wireless data off-loading.

To begin post-processing the test data, the raw .bag files (one per bridge pass) were fed into the SLAM algorithm portion of the FLASH software pipeline. For each .bag file, the algorithm outputted a single 3D point cloud in the form of a .ply file, which is an industry standard format often used in LiDAR applications. The .ply files were then opened in CloudCompare and in-built segmentation tools were used to manually clean and crop the point clouds – this primarily involved eliminating extraneous points (e.g., trees, other buildings, etc.) so that the bridge itself could be isolated in each point cloud. Displayed in Figure 31 is a point cloud of the Walnut/Foothills underpass bridge and its surroundings prior to cleaning and cropping. Figure 32 shows the point cloud after manual processing, and Figure 33 shows the same point cloud colored by coordinates.

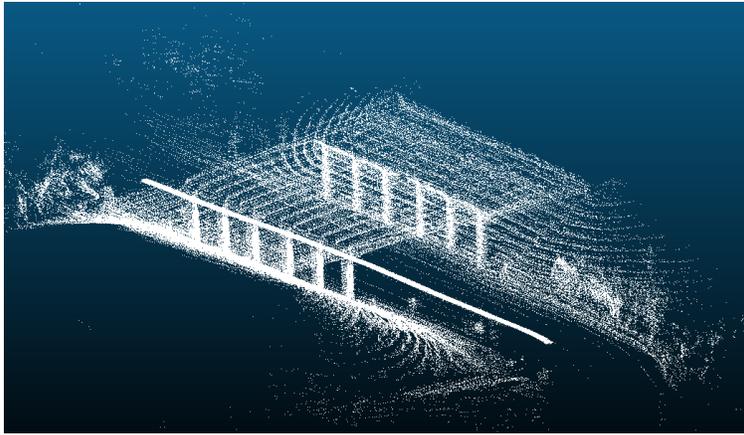


Figure 31: Point Cloud of Walnut/Foothills Underpass Bridge BEFORE Cleaning and Cropping

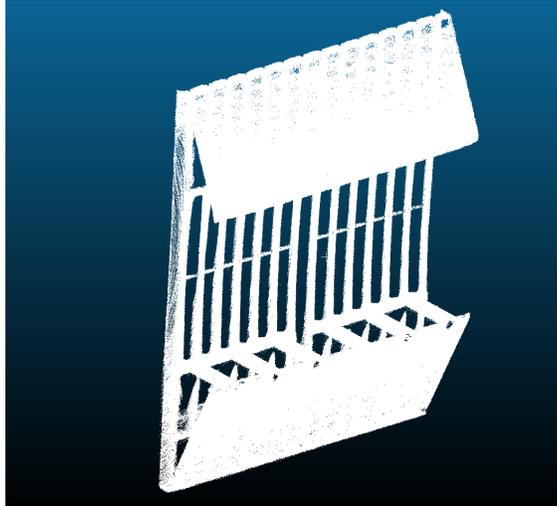


Figure 32: Point Cloud of Walnut/Foothills Bridge AFTER Cleaning and Cropping

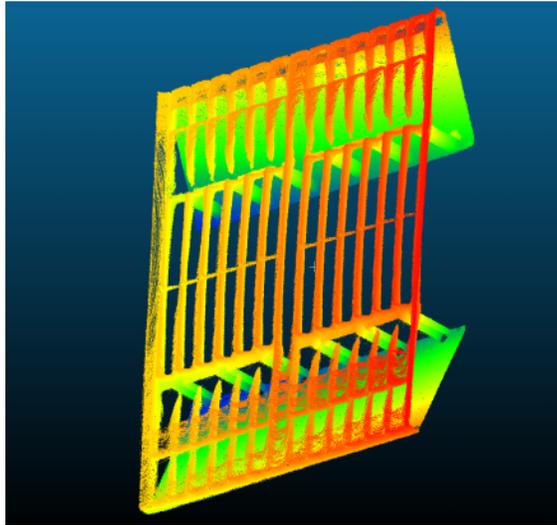


Figure 33: Point Cloud of Walnut/Foothills Bridge AFTER Cleaning and Cropping, Colorized

Before converting the point clouds into 3D meshes, data quality (average point density and accuracy) was assessed in CloudCompare for comparison to predictive models and requirements. As outlined earlier, Design Requirement 2.1 states that each point cloud shall have an average point density (resolution) of at least 400 points per square meter on the bridge underside. Prior modeling/analysis with the previous sensor configuration (scanning upwards) yielded an expected point density of approximately 1440 pts/m<sup>2</sup>. Point density was expected to be less in the new sensor configuration, but still above the requirement of 400 pts/m<sup>2</sup>. In order to verify this, point density of the isolated bridge underside was measured within CloudCompare for four CST scans taken at 5, 10, 15, and 20 mph. Figure 34 is a screenshot of the CloudCompare point density measurement result for a vehicle speed of 15 mph. Figure 35 shows the results for all four testing speeds.

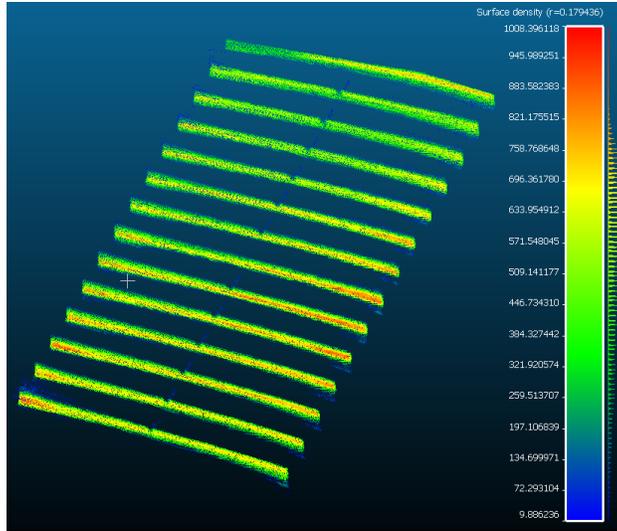


Figure 34: Point Density Measurement of Walnut/Foothills Bridge Underside (15 MPH Scan)

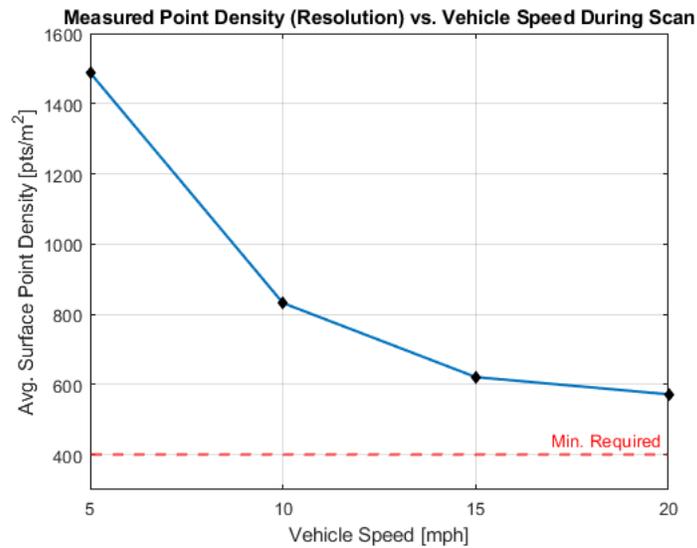


Figure 35: CST Results: Average Point Density Measured on Bridge Underside from 5 to 20 MPH

As seen in Figure 35, the average measured point density remains above the resolution requirement for scans up to 20 mph. Although speeds above 20 mph were not tested, quadratic extrapolation of the data suggests that the point density can be maintained above the minimum required for speeds up to roughly 30 mph. As expected, point density on segments of the bridge other than the underside (e.g., support pillars) was found to be much higher than the values shown in Figure 34 and 35 because those areas more directly fall within the sensor’s vertical field of view. Accuracy was estimated by comparing point cloud measurements of the overall bridge dimensions (Figure 36) to ground truth measurements provided by Google Earth. Measurements of bridge span and width were averaged over nine CST trials at the same speed. Results are tabulated in Table 9.

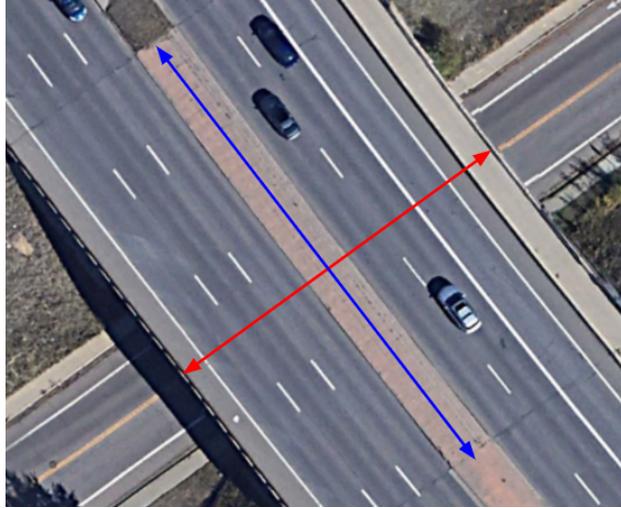


Figure 36: Walnut/Foothills Bridge Dimensions Referenced for Accuracy Estimation

Table 9: CST Results: Point Cloud Accuracy

	Measurement (Point Cloud)	Ground Truth (Google Earth)	Accuracy (Error)	10 cm Req. Satisfied?
<b>Bridge Span (Avg.)</b>	34.77 m ( $\sigma = 0.58$ m)	34.74 m	3 cm (0.08%)	YES
<b>Bridge Width (Avg.)</b>	45.01 m ( $\sigma = 0.45$ m)	46.63 m	162 cm (3.47 %)	NO

Table 9 shows the 10 cm accuracy requirement was satisfied for the bridge width measurement but not for the bridge span measurement. The reason for this is twofold – first, there was error involved in selecting bridge measurement endpoints in Google Earth for acquiring ground truth values (i.e., no explicitly defined "start" and "end"). Second, because the spanwise bridge endpoints are further away from the sensor than the bridge width endpoints, the span measurements inherently have more error (higher scanning range required). Even though accuracy fell short of the requirement, the point clouds matched up with the Google Maps API reasonably well (Figure 37). Note that this alignment was done without scaling the point cloud.

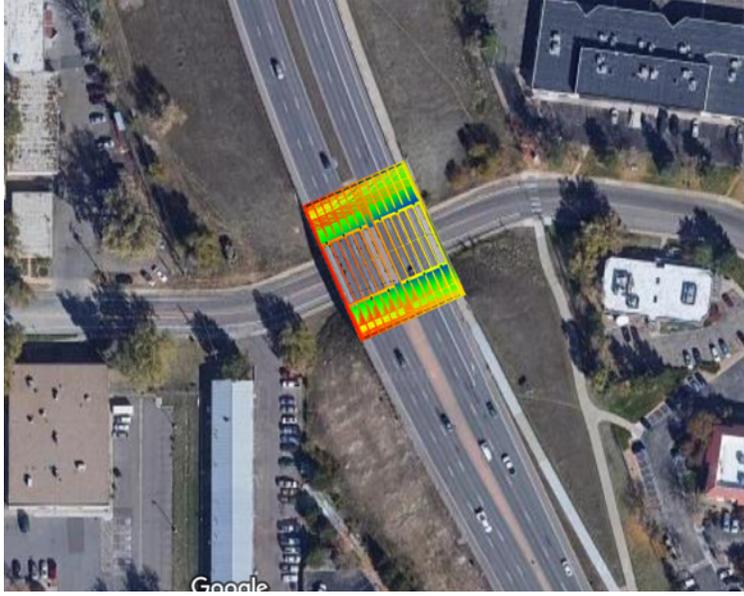


Figure 37: Overlay Comparison of Walnut/Foothills Point Cloud to Google Maps Ground Truth

Because of the team’s inability to directly access and measure key structural features on the bridge underside to provide truth values, the target board from SSLT served as an analog for feature identification purposes. In regard to results from system testing on the road (CSTs), it quickly became clear that large, sharp, and well-defined geometric structural features, such as the concrete girders that span the length of the bridge, showed up best in the point clouds. Hence, the definition and clarity of these features’ geometry in the point clouds was used as a qualitative verification of accuracy, but because so much “lumpiness” and non-uniformity (false positives) were observed in the data, the team realized that tracking individual defects would not yield desirable results regardless. Accordingly, the iterative scanning efforts during CSTs were more broadly focused on improving overall point cloud and mesh fidelity on a large scale, so this included things like minimizing mapping drift and frame skewing. This, of course, directly led into the measurement and verification of overall bridge dimensions for characterizing accuracy on a higher level, as discussed previously.

Following assessment of point density and measurement accuracy, the point clouds were converted into 3D mesh models via a Poisson surface reconstruction algorithm within CloudCompare. Mesh generation was found to be quite computationally expensive, and results varied based on input parameters to the algorithm. Figure 38 below shows a mesh output created directly from the Walnut/Foothills bridge point cloud (refer back to Figure 32). The mesh was then rendered in a virtual 3D environment (via Blender), with the result presented in Figure 39.

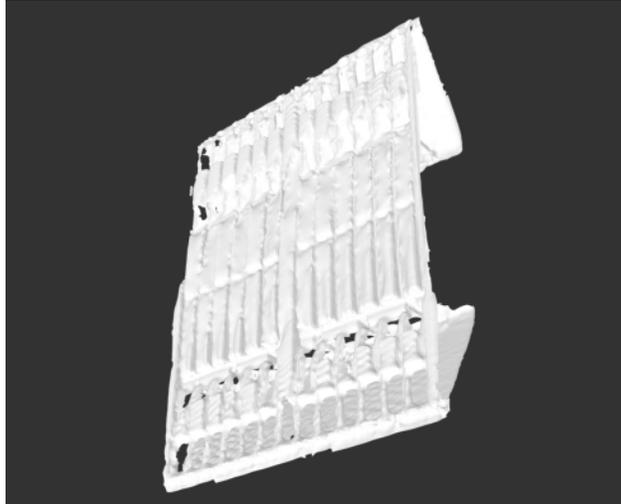


Figure 38: Mesh Generated from Walnut/Foothills Point Cloud

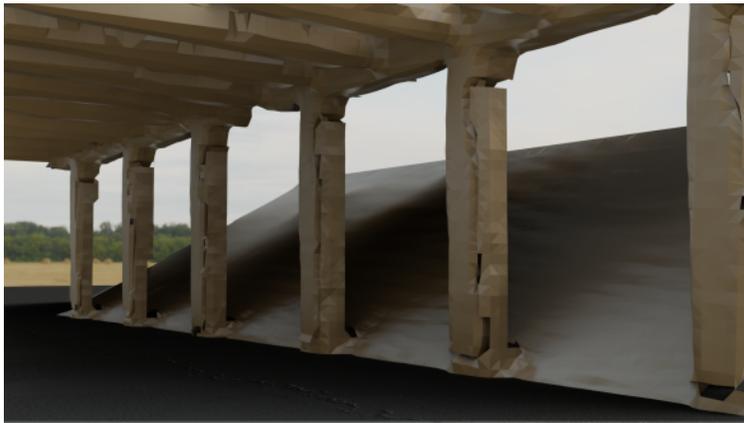


Figure 39: Mesh Internals Rendered in Blender Environment from Walnut/Foothills Point Cloud

Even though the collected data is dense and rich in point cloud form, problems arise when the point cloud is converted into a 3D mesh model (evident in Figures 38 and 39 above), which is what the team envisioned would be used by inspectors and engineers to conduct structural assessment. Through exhaustive experimentation with different mesh generation techniques and trial & error with various input parameters, it became clear that surface reconstruction from the dense point clouds leads to many false representations of geometry in the final generated mesh. Even when the point clouds were thoroughly cleaned and cropped to eliminate outliers and noise, it was concluded that mesh algorithms may inherently filter out the features of interest -- for example, cracks get closed over and small deformities get smoothed out simply due to how surface reconstruction works. Even the best meshes struggled with replicating precise geometry. That's why the team believes the 3D models produced by the FLASH system, at least in its current state, may not be appropriate for full-fledged structural analysis. However, the team is confident that the generated models and their associated point clouds will still prove useful for large scale infrastructure visualization and measurement as a planning tool for more efficiently conducting on-site surveying. Figure 40 is another mesh rendering of the Walnut/Foothills bridge created from a CST point cloud.



Figure 40: 3D Mesh Rendering of Walnut/Foothills Bridge in Blender

#### 5.3.4 Model Verification and Requirements Satisfaction

Two primary "models" were verified through the Comprehensive System Tests (CSTs). Google Maps/Earth served as the predictive model (or ground truth) against which point cloud accuracy was verified. LiDAR scanning range, as applicable to bridge height and width, was verified against expected performance specifications from the manufacturer-provided sensor datasheet. A model was developed for prediction of point cloud density (DR 2.1); however, because of the unforeseen sensor orientation change, its result is no longer applicable. Figure 41 below summarizes how the CST results validated functional requirements and overall project success criteria.

DR	Satisfied?	Associated CPE	Associated Level of Success	Explanation
1.2 Height	YES	CPE-1	Level 2	The scanned Walnut Underpass height was 6.7 m. This is greater than the requirement of 5.1 m.
1.3 Coverage	YES	CPE-1	Level 2	Scanning coverage width of the Walnut Underpass exceeded the requirement of 7.2 m
2.1 Point Density	YES	CPE-1 CPE-2	Level 2	572 pts/m <sup>2</sup> for the Walnut Underpass (underside only) is greater than the required 400 pts/m <sup>2</sup> .
2.2 Accuracy	NO	CPE-1 CPE-2	Level 2	The calculated width error (between the point cloud and ground truth data) for the Walnut Underpass was 162 cm. This is more than the maximum 10 cm error.
7.1 3D Mesh	YES	CPE-2	Level 2	An interactive 3D mesh was created for the Walnut Underpass (while in motion and self-localizing).

Figure 41: Satisfaction of Design Requirements from CST Results

## 5.4 Validation of Functional Requirements and Success Criteria

Throughout this project the team stuck as closely to their requirements and test plans as possible, in order to definitively claim success of the project. Despite the teams amazing work over the past two semesters and the numerous requirements they did meet there were a couple requirements and levels of success that they did not meet in the end which will be fully explained. A listing of all the teams functional requirements, their satisfaction, and which test they were satisfied by can be seen in Figure 42 and 43. A more condensed version of the levels of success can be found in Figure 44 where the specific level item that was not fully satisfied can be seen.

Req. ID	Functional Requirement	Satisfied?	Verification Test(s)
FR 1	The system shall utilize a 3D LiDAR sensor to survey infrastructure of interest.	YES	SSL, CST
FR 2	The LiDAR sensor shall collect and output usable 3D point cloud data (x, y, and z coordinates).	YES	SSL, CST
FR 3	The system shall be capable of localizing itself during normal driving conditions even when GNSS services are not readily available.	YES	CST
FR 4	The on-board processing unit shall be capable of data storage, handling, and interfacing between components.	YES	SSL, CST
FR 5	The system shall be capable of mounting onto a vehicle and operating while the vehicle is in motion.	YES	Pull Test, CST

Figure 42: Functional Requirements Verification

Req. ID	Functional Requirement	Satisfied?	Verification Test(s)
FR 6	The system shall incorporate a power source that is capable of continuously supplying power to all applicable components.	YES	CST
FR 7	The point cloud and localization data shall be consolidated and post-processed into an interactive 3D map/model.	PARTIAL	SSL, CST
FR 8	The on-board communications unit shall be capable of wirelessly transferring point cloud and localization data directly to a network server.	YES	CST
FR 9	The system shall be capable of initiating and terminating data collection with minimal passenger interaction.	YES	CST
FR 10	The system shall conform to all relevant safety regulations and guidelines.	YES	Pull Test, CST

Figure 43: Functional Requirements Verification Continued

	Structure	Data	Software/Mapping
Level 1	Capable of securely mounting system to one particular vehicle	10 cm size feature identified within point cloud from a scan distance of 3.5 m	Generate a 3D point cloud map and mesh in a stationary environment
Level 2	Capable of securely mounting system to multiple vehicles	5 cm size feature identified within point cloud from a scan distance of 3.5 m	Generate a 3D point cloud map and mesh in a moving environment via self-localization
Level 3	Capable of securely mounting system to multiple vehicles up to highway speeds	3 cm size feature identified within point cloud from a scan distance of 3.5 m	Generate a 3D point cloud map and mesh in a moving environment with enough accuracy and detail to enable structural analysis

Figure 44: Simplified Levels of Success

The team learned valuable lessons about continuously returning to the functional and design requirements throughout the project and more importantly, even if this tactic is deployed accurately, sometimes there will be roadblocks that you can not get over in the time allotted. Although the team successfully generated interactive 3D models directly from point cloud and localization data, Functional Requirement 7 is considered partial completion because, as discussed earlier, the produced models are not quite at the level appropriate for structural analysis. Further refinement and tuning of the mesh generation process may yield results that are more suitable for detailed structural assessment, but the team was unable to achieve this degree of fidelity given time constraints. Hence, Level 3 success for "Software/Mapping" (see Figure 44) was not reached. Additionally, Level 3 success for "Data" was not achieved because features smaller than 5cm could not be reliably discerned from SSLT results at distances above 3.5 meters. In conclusion, while the FLASH final product did not satisfy each and every requirement, the project can still be considered a success overall, and the system in its current state offers plenty of utility to the customer in terms of large scale infrastructure visualization.

## 6 Risk Assessment and Mitigation

*Authors: Courtney Kelsey, Jake Fuhrman*

Once the project had been well defined and the team had a good idea of all the project goals and deliverables they started identifying possible risks. These were first identified back in the first semester and were tracked and mitigated throughout the project. The team started by identifying the most important risks that they thought could have the biggest impact on the project. The result of this research can be found in the Initial Risk Matrix Figure 45.

The team identified that the two intolerable risks to the project were a failure in point cloud resolution and registration failure. The team also defined two more high level risks in the tolerable range which were mesh generation difficulties and excessive vibrations. All four of these high level risks were identified separately and further broken down in order to begin a mitigation plan. These two steps can be found in Figures 46 and 47.

		Consequence:			
		Acceptable	Tolerable	Intolerable	
Probability	Very Likely (5)				
	Likely (4)			Excessive Vibrations	
	Possible (3)			Scanning Obstructions	Mesh Generation Difficulties
	Unlikely (2)			IMU Incompatibility	Insufficient IMU
	Very Unlikely (1)				Power Supply Insufficient
			Negligible (1)	Minor (2)	Moderate (3)
					Severe (5)
		Severity			

Figure 45: Initial Risk Matrix

Risk	Subsystem	Description	Effect	SEV	PROB	Risk Priority Number (RPN)
Point Cloud Resolution	LiDAR	Insufficient point cloud resolution for defining structural flaws.	Catastrophic structural flaws could exist but not detected by the LiDAR if they are smaller than the maximum LiDAR point cloud resolution.	5	3	15
Registration Failure	Software	Registration is the process of merging the time-sequenced measurements to generate a final 3D point cloud.	The outputted dataset will be unusable for structural analysis whatsoever.	5	3	15
Mesh Generation Difficulties	Software	From the 3D point cloud a 3D mesh will be created to represent the geometry of the bridge.	The outputted 3D mesh will be unusable for structural analysis.	4	3	12
Excessive Vibrations	Structures	Excessive vibrations causing data collection inaccuracies.	Accuracy and precision of the LiDAR-generated point cloud could be compromised.	3	4	12

Figure 46: Failure Modes and Effects Analysis (FMEA)

Risk	Mitigation Method
Point Cloud Resolution	Apply maximum LIDAR data collection setting (maximum horizontal channels and rotation rate); reduce vehicle speed during data collection if needed.
Registration Failure	Design ROS pipeline with maximal compatibility for interchanging SLAM routines if LIO-SAM fails to produce high-quality output. (i.e. Google Cartographer)
Mesh Generation Difficulties	Survey and prepare for experimenting with alternative competing mesh generation algorithms that are compatible with CloudCompare.
Excessive Vibrations	Apply thermal paste and/or shock-absorbing material to structural housing; research effects of vibrations on LiDAR performance.

Figure 47: Risk Mitigation Methods

These four identified risks ended up varying in their importance as the project progressed through the semester. The excessive vibration risk ended up being irrelevant for this project as the infrastructure in question would always be on paved roads and if there was excessive pot holes and other obstructions the driver could work to avoid them as they are not distracted with the software and the self localizing capabilities could account for the movement.

Point cloud resolution, registration failure and mesh generation difficulties became bigger issues as has been discussed. Despite these drawbacks the team can confidently say that they saw these issues from the beginning and carefully tracked the progress throughout the project and addresses these various issues each time they came up. Therefore, the mitigation strategies laid out in the fall allowed the team to easily track and address all the risks that came up and engineered the best solutions possible.

		Consequence:			
		Acceptable	Tolerable	Intolerable	
Probability	Very Likely (5)				
	Likely (4)				
	Possible (3)	Scanning Obstructions	Excessive Vibrations		
	Unlikely (2)		Insufficient IMU	Mesh Generation Difficulties	Registration Failure
	Very Unlikely (1)	IMU Incompatibility	Power Supply Insufficient	Mounting Mechanism Detachment	Point Cloud Resolution
		Negligible (1)	Minor (2)	Moderate (3)	Significant (4) Severe (5)
		<b>Severity</b>			

Figure 48: Post-Mitigation Risk Matrix (as of CDR)

Figure 48 shows the anticipated post-mitigation risk probabilities and severities as of CDR (end of Fall Semester). Of the four most impactful risks identified previously, excessive vibrations were mitigated to its approximate position on the post-mitigation risk matrix. While poor road conditions (such as uneven paving and potholes) were still probable (and frequently encountered during comprehensive system testing), it was concluded that they did not have a significant impact on the resulting point cloud. This is mainly because data was captured at slow speeds and over short distances where pothole occurrence was low.

Achieving a desirable point cloud resolution was also mitigated to its approximate location on the post-mitigation risk matrix. While realization of this risk would have been severe, it was verified that our point cloud density and point spacing (via the SSL Test) was within our design requirements, and with a healthy margin (see the Verification and Validation Section for more details). Therefore, the realization probability of this risk was reduced to very unlikely.

Registration failure and mesh generation difficulties were the two most impactful risks that were not able to be mitigated as planned in CDR. As of this current point in the project (end of Spring Semester), the team would identify both of these risks as severe if realized. The probabilities of both these risks occurring would also be moved to likely.

The team's chosen SLAM algorithm varied throughout the Spring semester, mainly because of point cloud registration failures. It was found that the best mitigation to this risk was to use Ouster's proprietary WebSLAM tool. This SLAM algorithm only then worked if the LiDAR was placed in a horizontal configuration as opposed to the previously planned vertical configuration. Once the LiDAR was placed in the horizontal configuration, it was found that both ground reference data and easily distinguishable bridge geometries were needed in order for the algorithm to compile properly. While some of the teams compiled point clouds were accurate to ground truth data, many proved to not be.

Generating a 3D mesh, usable for structural analysis, first required an accurate, registered point cloud. Therefore, these two risks were somewhat linked throughout the semester. It was found that given an accurate point cloud (.ply file), 3D mesh generation was feasible. However, none of the 3D meshes prepared by the team would have been significantly usable (on their own) for structural analysis. Since structural analysis was a goal of this project, the realization of this risk was severe and probable given the lack of detail in the 3D meshes. Mitigation of this risk was attempted by using different mesh construction tools, but even the best, most detailed meshes proved to be ineffective for structural analysis.

The mitigation and realization of the teams identified risks had a significant impact on the success of the project. As seen in the achieved Levels of Success Table (Figure 44), Level 3 success was achieved for structures, but data and software/mapping only achieved Level 2 success. The mitigation of the excessive vibrations risk and other related structural risks proved to be effective and allowed the team to achieve complete success in regard to structures. As previously mentioned, the failure of generating accurate 3D meshes lead the data and software/mapping areas of the project to be one level less successful than the structural area. Although the team did not achieve all Level 3 successes, the associated risks and their realizations were mitigated to the point where this project still produced useful results to supplement structural inspections of bridges.

## 7 Project Planning

*Authors: Jake Fuhrman, Kunal Sinha*

### 7.1 Organizational Chart (OC)

Figure 49 defines the roles of each team member on the project as well as the teams customer, ASTRA, and the team's advisor, Dr. Dennis Akos. The team is broken down into four main technical sub-teams: software, sensors, electronics, and structures.

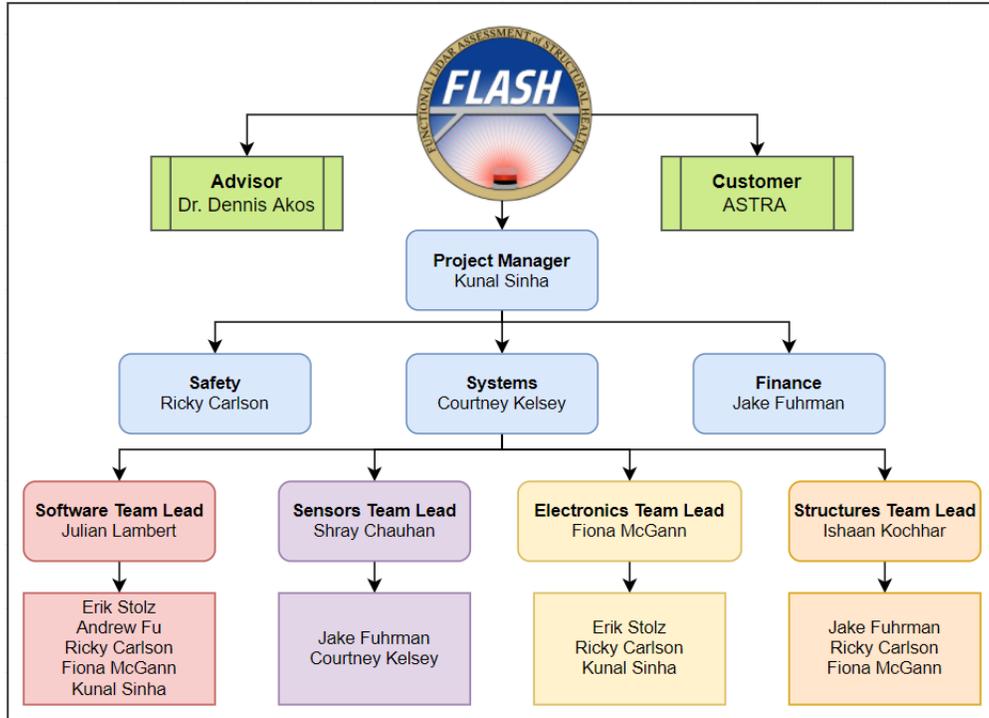


Figure 49: Organization Chart (OC) Team FLASH

## 7.2 Work Breakdown Structure (WBS)

The team's work breakdown structure has been broken down into five main sections; Processing Software, On-board setup, Structures, fall deliverables, and spring deliverables. They are then divided by the different phases in the project. These were the major tasks to be carried out by the team and the schedules were made primarily by inferring the diagram in Figure 50.

During the project definition section, the primary tasks for the software team was to develop a software pipeline for us to process our raw collected LiDAR and IMU data. This was followed with planning how the data will flow through the several software packages. This is where the team learnt about the different mapping, mesh generation and data collection softwares and planned to integrate them into a single pipeline for smooth data processing. The On-board setup teams (including the Sensors and Electronics sub-teams) parallely worked to choose a sensor that would meet the functional requirements the project. They conducted trade studies for different COTS components and then planned the power system on-board. The structures team meanwhile worked on modelling a mount for the component selected by the sensors team. This concluded majority of the work done by the team during the fall semester.

In the spring semester, the teams first worked on manufacturing the designed items. For Software, this meant scripting the pipeline and installing the different packages that the team wanted to test. For On-board, it meant acquiring the hardware required and assembling them in the car, along with installing the preliminary software required to collect data from the LiDAR unit. Structures made a mount first by 3-D printing it, and then ordering the aluminum mount from the Aerospace Machine Shop. Attaching the magnetic attachments into the mount then concluded the manufacturing section of the project which was then ready to be tested.

The team started its testing with a mount pull test to make sure the drag forces experienced by the mount does not lessen the safety of the sensor. Meanwhile the On-Board setup team conducted a Small Scale Lidar test to make sure the sensor meets the functional requirements as predicted during the planning phase. The full system along with on-board electronics, mounts and software pipelines were then integrated

together for the Comprehensive Systems Tests. The team conducted several data collection runs and ran it through the designed pipelines. This became an iterative process till the team found the best software package and LiDAR unit orientation for our purposes of the project. The meshes generated from these tests were further compared to truth data as given by a Google Maps API. This test served as the final verification and validation test that had to be conducted.

Once the tasks were completed, the team documented their progress in terms of deliverables for Fall and Spring semesters for external input on the project. With the submission of the Project Final Report, the team has finished all tasks that were planned.

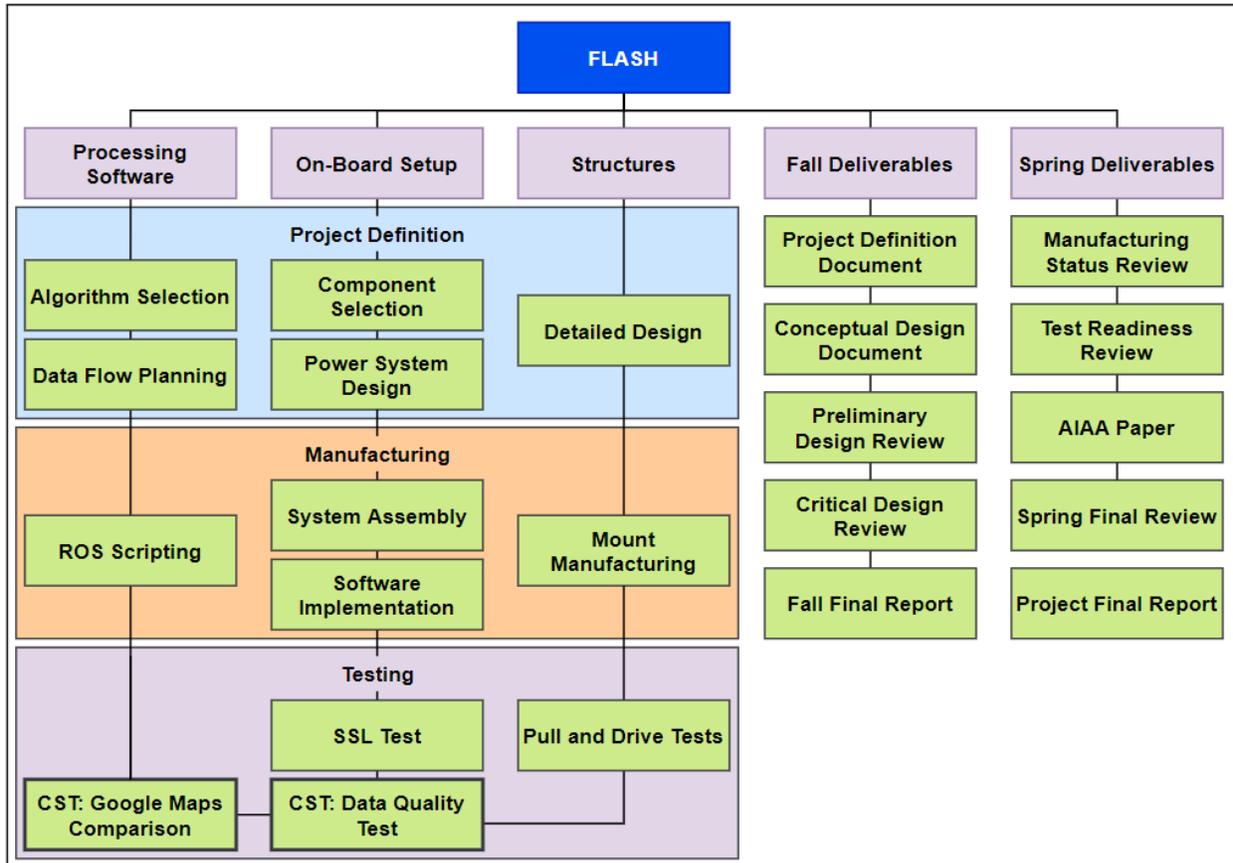


Figure 50: Work Breakdown Structure for FLASH

### 7.3 Work Plan (WP)

The team decided to break the work plan into two sections. The first is the software and its defined schedule for the spring semester. This project is very software heavy so a large amount of time and resources had been allocated in the semester. The second section is the structures and on-board set up work plan. This represents all the hardware building and sensor control and tasks that need to be accomplished. Even though these two sections exist separately their tasks become closely linked near the middle of the semester and each will rely on the other to be complete in time to perform a final system tests. The team planned to finish all testing by the Spring Final Review submission so there can be extra time allotted for any further testing and anomaly work that becomes necessary by the Final Project Report. As per the schedule, the team will be done with manufacturing and testing **by 20th April**.

The critical path for each sub team, the major milestones, the progress bars and margins to the schedule

can be seen in the Gantt charts (figures 51 and 52). The margins were allocated based on the complexity of a task. For example, the comprehensive system testing was allocated about two weeks of margin time as it was planned to be the most difficult task which may require repetition. Whereas tasks like software implementation in on-board setup (installing Ouster's data collection software) had lower margins due to their simplicity.

The team was successful in its goal of its end date of testing, but challenges like delay in purchasing the LiDAR unit, change of SLAM Algorithms and change of sensor orientations caused our schedule to change frequently during the semester.

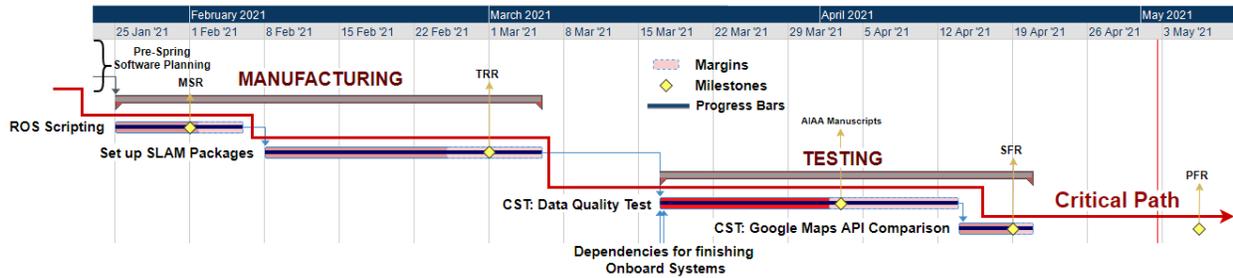


Figure 51: Software Work Plan

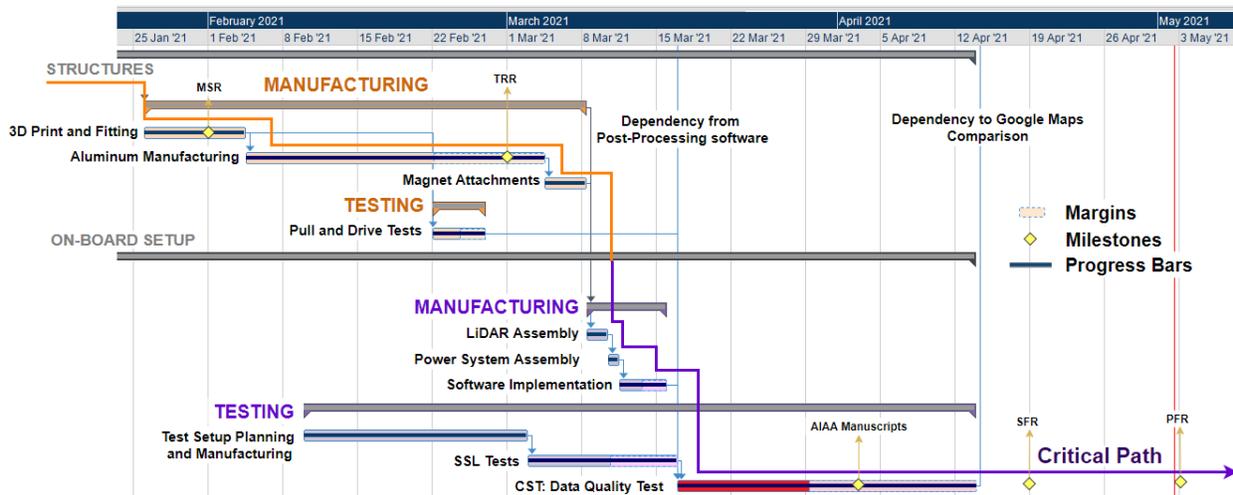


Figure 52: Structures and On-board Setup Work Plan

## 7.4 Cost Plan (CP)

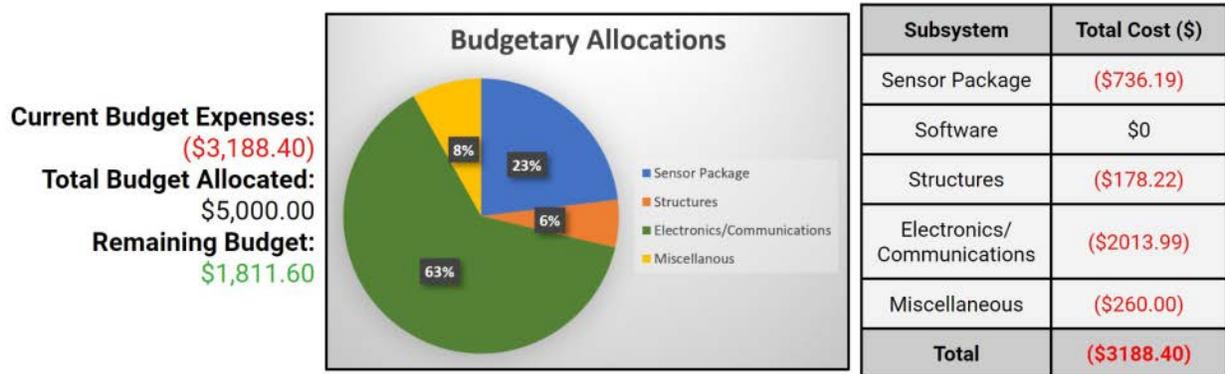


Figure 53: Finalized Team Budget (Updated: 4/27/2021)

Figure 53 shows the total budget for Team FLASH. As seen in the Figure, the team spent a total of \$3,188.40 throughout the academic year. Given a total budget of \$5,000.00, this left the team with \$1,811.60 remaining. This total budget is \$154.98 **under** the originally planned budget of \$3,343.38 (as presented at CDR).

The reasons for this slight discrepancy in the final budget are as follows. The budget presented at CDR accounted for a \$1,500.00 external IMU to be included as part of the sensor package. At the time, it was decided this would be necessary in order to obtain the needed precision for the chosen SLAM algorithm. As Spring semester began, this item was de-scoped, as WebSLAM was told to be satisfactory without the high precision of an external IMU. Therefore, this \$1,500.00 was saved. Toward the end of the Spring semester, it was decided to purchase items focused on the continuation of the project. These items included mass data storage devices (allowing for long periods of scanning and storage of the associated .ply (point cloud) files) and transportation cases for delivery to ASTRA. In total, these items cost approx. \$1,700.00. After removing the 20% cost margin given at CDR, the team was decidedly under budget.

The central pie chart in Figure 53 shows the approximate cost percentage breakdown by subsystem. The Electronics/Communications subsystem included relatively expensive items such as the laptop and mass data storage devices, which is why it accounts for 63% of the total budget. The Sensor Package subsystem includes all the transportation cases for delivery to ASTRA. The Structures subsystem includes all expenses associated with the mounts and magnets for attachment to the mission vehicle. The Miscellaneous category accounts for any non-related technical expenses, such as the PILOT deposit and AIAA Student Conference registration fees.

## 7.5 Test Plan (TP)

Table 10 shows the test plan for the team which was executed in the spring semester. This table shows the duration of each test described in Section 5 as well as any predecessors. This table also lists the resources and locations necessary for each test. **NOTE:** Homebase refers to any location with access to WiFi.

Figure 54 shows how the tests were scheduled. For this the team estimated the time and margins allotted based on the complexity of the task. The SSL and CST tests were given large blocks of time because of the iteration required while changing data collection parameters (like orientation, velocity etc.). The Google Maps Comparison used data collected during CST and hence was scheduled last. The specific locations in which testing was carried out is mentioned in the Verification and Validation Sections.

Test	Test Name	Duration	Pre.	Reources	Location
1	Structures: Pull and Drive Test	2 days	N/A	Hook Scale	Homebase (open parking space)
2	Small Scale LiDAR Test (SSL)	2 weeks	N/A	Ouster OS1-32 Processing Computer	Homebase (with Wifi small test setup)
3	Comprehensive System Test (CST): Data Quality	4 weeks	2	Processing Computer Ouster OS1-32 CDOT Highway Database (OTIS)	Low-traffic road with a highway underpass
4	Comprehensive System Test (CST): Google Maps API Comparison	1 week	3	Processing Computer	Homebase (with Wifi)

Table 10: Test Plan for Spring 2021 Team FLASH

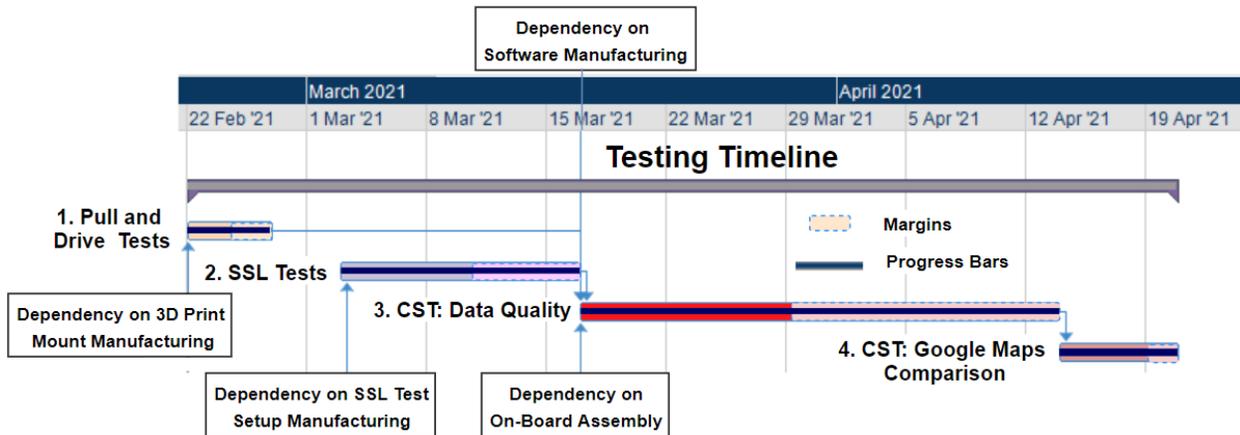


Figure 54: Testing Schedule with dependencies, margins and progress bars shown

## 8 Lessons Learned

*Authors: Courtney Kelsey, Kunal Sinha*

By the end of this project the team came away more experienced and better prepared to engineer a bigger project from start to finish. Taking a project from the ground up to final product was a very useful process to go through before going out into the industry and the team is grateful to have had the opportunity. Of course along the way there were many lessons learned.

From a systems engineering standpoint some of the most important lessons learned included the importance of well developed design requirements. Even just the specific and careful wording of each requirement could determine the difference between a fully functioning system and a potential failure. The team also learned that tracking the outcomes from every test iteration so that it could be compared to the functional requirements was also very important. Without sticking close to the original requirements that team would have had trouble verifying their success criteria and making sure that the project performed to the standards that it needed to. Another lesson was to prioritize risk assessment to prevent requirement failures as every good project should as well as returning to the original risks and making sure they are not forgotten as the project progresses. Finally, keeping backups of all COTS products and open-source software was an important lesson as when in a time sensitive environment a failure can be much more easily fixed with a

replacement. This allows the project to continue on schedule and a failure analysis can be done separately or at a later time.

The team also learned some lessons from a project management standpoint. These included the designation of time based on risk factors for scheduling through the integration and testing phases. More specifically, time management needed to be placed with a higher priority on the more sensitive and difficult aspects of a project sooner rather than later. Another lesson was to have an Agile approach for software based projects. The testing required small sprints to test software while changing small parameters in each iteration. The team would have been better prepare schedule-wise if we planned our sprints rather than the test as one block of time. This created major changes in our schedule. And hence the team found it important to find a management technique best fit for the project, rather than choosing a technique and applying our project to it. Another lesson learned was to keep real expectations of scheduling and work better with team as software portions can change quickly as they learned with the schedule slips resulting from the underestimation on the software difficulty in the fall semester. Finally, having tag-ups with full team as frequently as possible so that whole team is up to date was always important in order to keep the team on track and up to date with the progress of every section in the team. If this is carried out, then it takes lesser time to distribute man-power as team members won't have to work to catch up as much. In projects with small teams such as this, it saves a lot of time. Overall, the team learnt how to carry out long and expensive projects like FLASH, along with how to work with industry customers.

## 9 Individual Report Contributions

Table 11: Individual Report Contributions Table

	<b>Description of Contributions</b>
<b>Kunal Sinha</b>	Projectives Objectives, Final Design, Project Planning, Lessons Learned
<b>Courtney Kelsey</b>	Project Purpose, Objectives, Functional Requirements, Risk Assessment
<b>Jake Fuhrman</b>	Manufacturing, V&V - SSLT, V&V - CST, Risk Assessment, Project Planning - Cost Plan
<b>Shray Chauhan</b>	Project Objectives, Functional Requirements, Verification and Validation
<b>Ishaan Kochhar</b>	Final Design, Verification and Validation, Manufacturing
<b>Andrew Fu</b>	Manufacturing - Software, V&V - CST
<b>Ricky Carlson</b>	Final Design, Verification and Validation
<b>Fiona McGann</b>	Final Design, Verification and Validation
<b>Julian Lambert</b>	Software related subsections
<b>Erik Stolz</b>	Manufacturing, Final Design

## 10 References

- [1] Comeron, A., Constantino Muñoz-Porcar, Francesc Rocadenbosch and Michaël Sicard (2017). Current Research in Lidar Technology Used for the Remote Sensing of Atmospheric Aerosols. [online] ResearchGate. Available at: [https://www.researchgate.net/publication/317714590\\_Current\\_Research\\_in\\_Lidar\\_Technology\\_Used\\_for\\_the\\_Remote\\_Sensing\\_of\\_Atmospheric\\_Aerosols](https://www.researchgate.net/publication/317714590_Current_Research_in_Lidar_Technology_Used_for_the_Remote_Sensing_of_Atmospheric_Aerosols) [Accessed 14 Sep. 2020].
- [2] American Geosciences Institute. (2017). What is Lidar and what is it used for? [online] Available at: <https://www.americangeosciences.org/critical-issues/faq/what-lidar-and-what-it-used> [Accessed 14 Sep. 2020].
- [3] Bridges (2017). Bridges. [online] ASCE’s 2017 Infrastructure Report Card. Available at: <https://www.infrastructurereportcard.org/cat-item/bridges/: : text=The%20U.S.%20has%20614%2C387%20bridges,structurally%20deficient%20bridges%20each%20day.> [Accessed 14 Sep. 2020].
- [4] Mathworks.com. (2020). What Is SLAM (Simultaneous Localization and Mapping) – MATLAB Simulink. [online] Available at: <https://www.mathworks.com/discovery/slam.html> [Accessed 14 Sep. 2020].
- [5] Sun, J., Timurdogan, E., Yaacobi, A., Hosseini, E.S. and Watts, M.R. (2013). Large-scale nanophotonic phased array. *Nature*, [online] 493(7431), pp.195–199. Available at: <https://www.nature.com/articles/nature11727> [Accessed 14 Sep. 2020].
- [6] Matthew Berger, Andrea Tagliasacchi, Lee Seversky, Pierre Alliez, Gael Guennebaud, et al.. A Survey of Surface Reconstruction from Point Clouds. *Computer Graphics Forum*, Wiley, 2016, pp.27. [ff10.1111/cgf.12802](https://doi.org/10.1111/cgf.12802). [ffhal-01348404v2](https://doi.org/10.1111/cgf.12802)
- [7] “Leica ScanStation P50 – Long Range 3D Terrestrial Laser Scanner.” Leica ScanStation P50 | Leica Geosystems, 2020, [leica-geosystems.com/en-us/products/laser-scanners/scanners/leica-scanstation-p50](https://www.leica-geosystems.com/en-us/products/laser-scanners/scanners/leica-scanstation-p50).
- [8] “Mitigation Strategies For Design Exceptions - Safety: Federal Highway Administration.” Safety, [safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3/3verticalclearance.cfm](https://www.safety.fhwa.dot.gov/geometric/pubs/mitigationstrategies/chapter3/3verticalclearance.cfm).
- [9] Shan, Tixiao Englot, Brendan Meyers, Drew Wang, Wei Ratti, Carlo Rus, Daniela. (2020). LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping.", Link:<https://github.com/TixiaoShan/LIO-SAMprepare-imu-data> (LIO-SAM)
- [10] Qin, Tong, et al. “VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator.” *IEEE Transactions on Robotics*, vol. 34, no. 4, Aug. 2018, pp. 1004–20, doi:10.1109/tro.2018.2853729. Link: <https://github.com/HKUST-Aerial-Robotics/VINS-Mono>
- [11] Google Maps, Google, 2020, [www.google.com/maps/@40.6990124,-111.8023971,12.41z](https://www.google.com/maps/@40.6990124,-111.8023971,12.41z).
- [12] MPU-9250 Product Specification Revision 1.1, InvenSense Inc., 1745 Technology Drive, San Jose, CA 95110 U.S.A, retrieved March 23rd, 2021 <https://www.invensense.com/download-pdf/mpu-9250-datasheet/>
- [13] Ye, H., Chen, Y., and Liu, M., “Tightly Coupled 3D Lidar Inertial Odometry and Mapping,” *arXiv.org* Available: <https://arxiv.org/abs/1904.06993>.
- [14] T. Qin, P. Li and S. Shen, "VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator," in *IEEE Transactions on Robotics*, vol. 34, no. 4, pp. 1004-1020, Aug. 2018, doi: 10.1109/TRO.2018.2853729.

- **[15]** Information Sciences Institute University of Southern California, "Transmission Control Protocol," University of Southern California, Sept. 1981, Available: <https://tools.ietf.org/html/rfc793>.
- **[16]** Wikipedia, "Poisson's equation", Wikimedia, Mar. 2021, Available: [https://en.wikipedia.org/wiki/Poisson%27s\\_equation](https://en.wikipedia.org/wiki/Poisson%27s_equation).
- **[17]** Kautz, Jan, et al. Pacific Graphics 2011 SSD: Smooth Signed Distance Surface Reconstruction. no. 7, 2011, [mesh.brown.edu/ssd/pdf/Calakli-pg2011.pdf](http://mesh.brown.edu/ssd/pdf/Calakli-pg2011.pdf).

# 11 Appendix A: LiDAR

## 11.1 Additional Comprehensive System Test Results

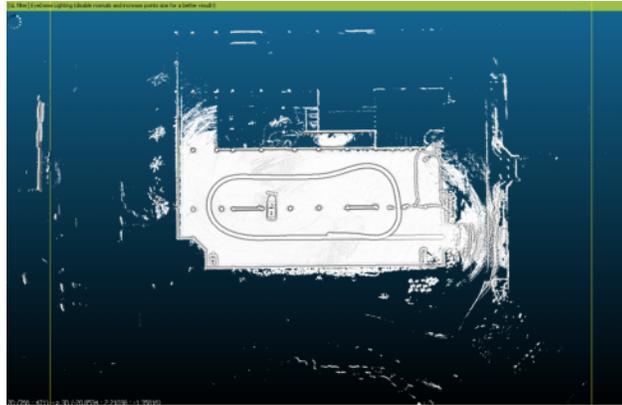


Figure 55: Point Cloud of Scanned Parking Structure

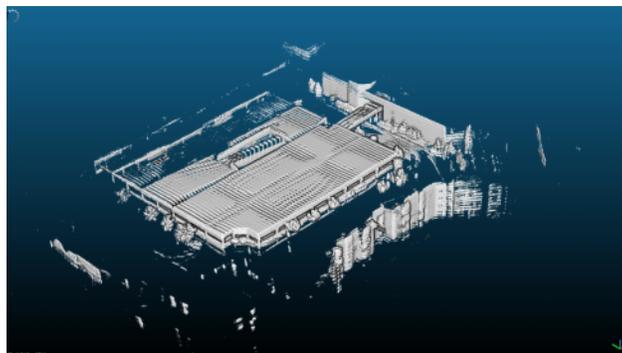


Figure 56: Point Cloud of Scanned Parking Structure

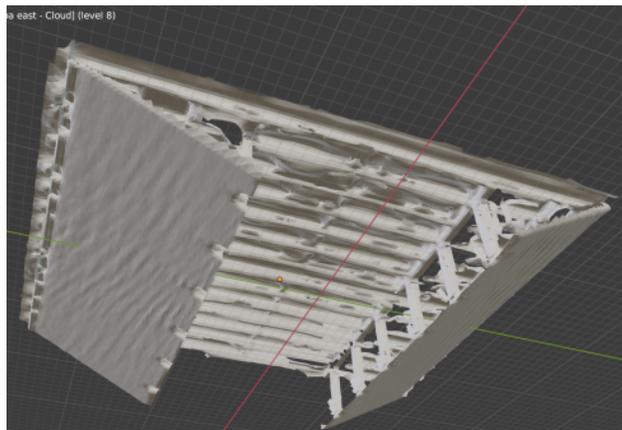


Figure 57: Early Attempt at Mesh Generation for Walnut/Foothills Bridge

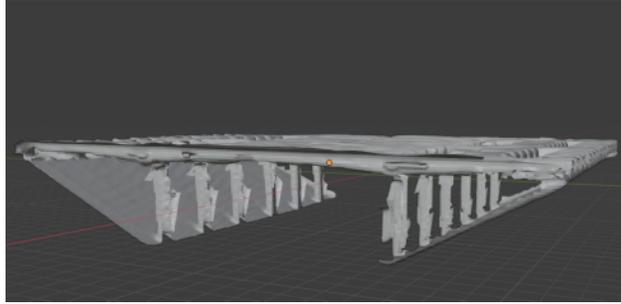


Figure 58: Early Attempt at Mesh Generation for Walnut/Foothills Bridge

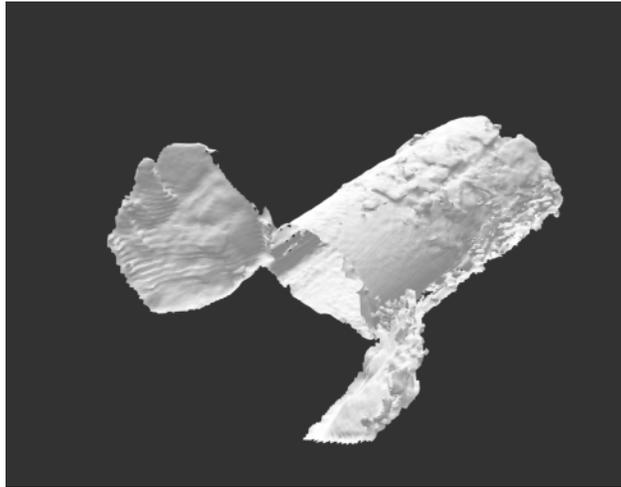


Figure 59: Mesh Generated from Point Cloud of Boulder Canyon Tunnel



Figure 60: Boulder Canyon Tunnel Mesh Overlaid with Google Maps API

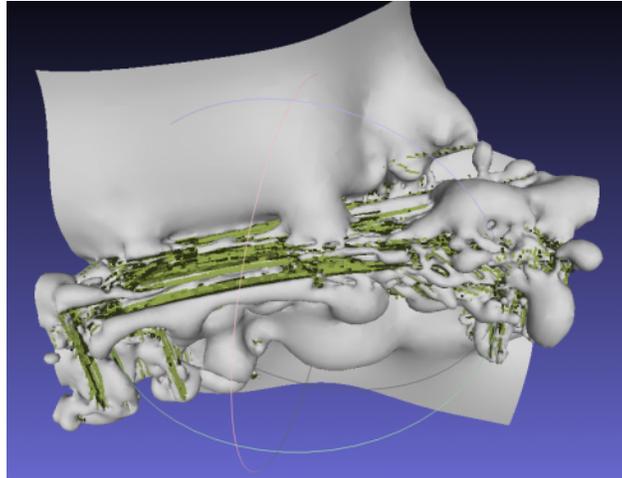


Figure 61: Failed Mesh Generation in MeshLab Software

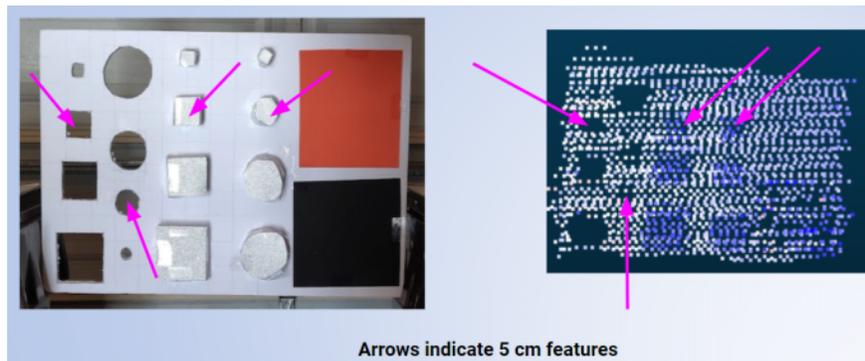


Figure 62: SSLT Target Board Feature Discernment at 3.5 meters

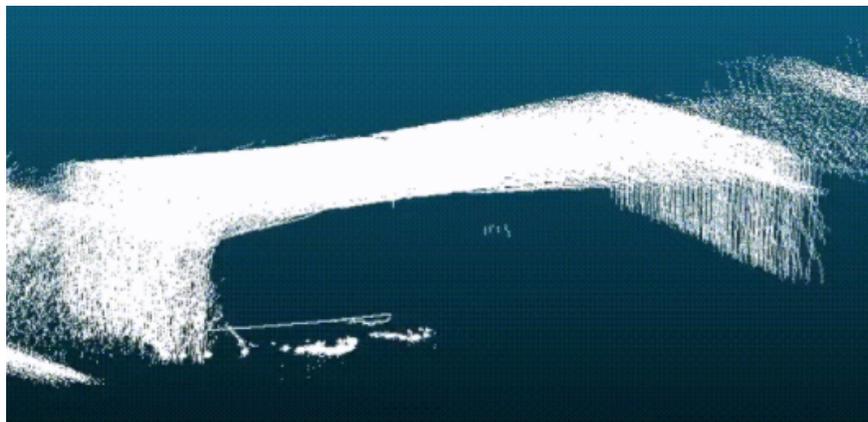


Figure 63: Point Cloud Frame Stacking Observed with Previous Sensor Orientation

## 11.2 Thermal Analysis

The thermal analysis included in figures 64 and 65 indicate that even at maximum operational temperatures (50°C), the LiDAR system will conduct and convect away more heat than it will radiate itself and obtain heat from direct sunlight (radiation). This is the case for both the initial 3D printed plastic design (ABS) and the final CNC Aluminum 6061 design.

$$Q_{in} \leq Q_{out}$$

$$Q_{in,rad} + Q_{gen} \leq Q_{out,rad} + Q_{out,conv} + Q_{out,cond}$$

$$q_{in,rad}A_L + Q_{gen} \leq q_{out,rad}A_L + q_{out,conv}A_L + q_{out,cond}A_B$$

$$q_{in,rad}A_L + Q_{gen} \leq \varepsilon\sigma T_L^4 A_L + h(T_L - T_\infty)A_L + \frac{k(T_L - T_\infty)}{L_A} A_B$$

LiDAR Surface Area =  $A_L = 0.0496 \text{ m}^2$   
 Emmissivity of Anodized Aluminum =  $\varepsilon = 0.77$   
 Stefan Boltzmann Constant =  $\sigma = 5.67 \times 10^{-8} \frac{\text{W}}{\text{m}^2\text{K}^4}$   
 LiDAR Operating Temperature =  $T_L = 315 \text{ K}$   
 Air Temperature =  $T_\infty = 298 \text{ K}$   
 Conductivity of Aluminum =  $k = 167 \frac{\text{W}}{\text{mK}}$   
 Conductivity of ABS Plastic =  $k = 0.1 \frac{\text{W}}{\text{mK}}$   
 Length of Aluminum Plate (at LiDAR Base) =  $L_A = 0.0039 \text{ cm}$   
 Diameter of LiDAR Base =  $A_B = 0.005 \text{ m}^2$   
 Forced Convection Coefficient =  $h = 167 \frac{\text{W}}{\text{m}^2\text{K}}$

Figure 64: Governing Equations, Constants, and Assumptions for Thermal Analysis

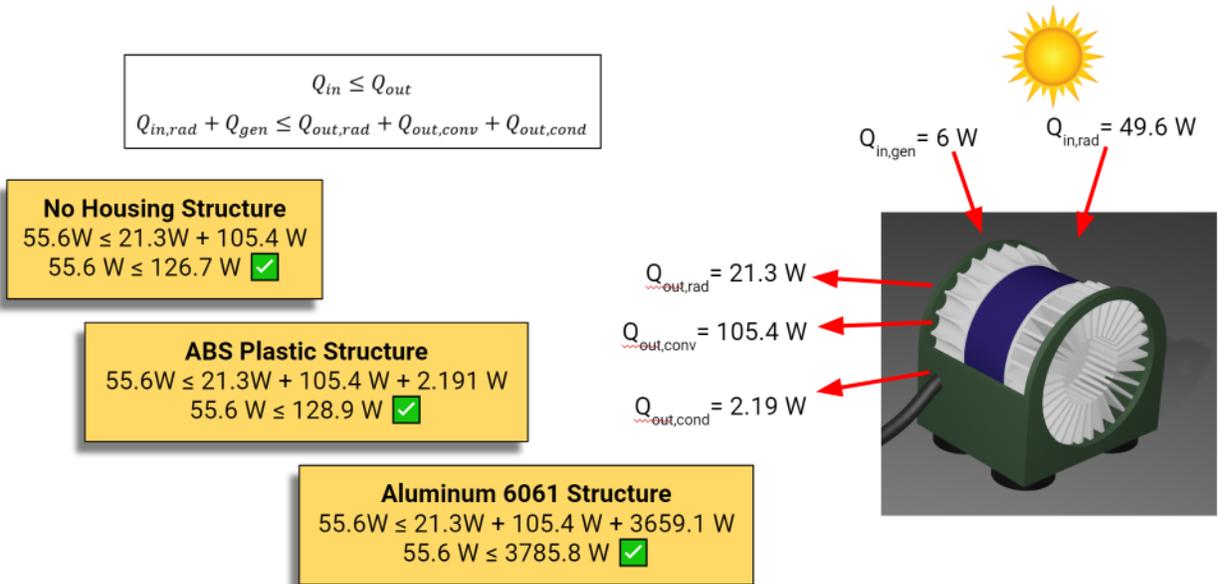


Figure 65: Thermal Analysis Results

## 11.3 Error Analysis

### 11.3.1 Sunlight

Operating the OS1-32 Gen 1 on a sunny day increases the amount of noise present in the data, therefore decreasing the signal to noise ratio (SNR), where SNR is the ratio of laser signal strength to sunlight noise strength. It is estimated that sunlight and its effects contribute towards 8% of the total error.

### 11.3.2 Reflectivity

Reflectivity depends on the physical properties of objects, and can range from lava at 9% reflective to retroreflectors (street signs) at 90%. The most common material used in bridge construction, concrete, has a reflectivity of 30%. The higher the reflectivity, the more likely the LiDAR will be able to detect the object.

### Range: Lambertian Reflectivity Examples

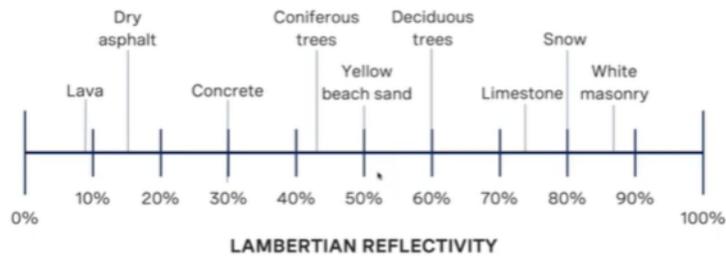


Figure 66: Various Objects and their Reflectivities

### 11.3.3 Probability of Detection

Probability of detection (PD) represents the likelihood that the LiDAR will be able to detect a certain object. PD is defined as the fraction of true positive measurements over total measurements. This equation assumes a single point of a known target at a fixed distance. The higher the probability of detection, the more likely the object is to be reflected in the point cloud.

## 11.4 Point Volume and Data Budget

Vertical Points	32
Horizontal Points	2048
Frame Rate	10 Hz
<b>Points per Second</b>	<b>655360</b>

Assuming vehicle speed of 60 MPH (26.82 m/s)

Every 26.82 meters traveled → 655360 points collected

50 meter travel distance under bridge → **1.22 million points total**

Figure 67: Point Volume Calculation

Assuming vehicle speed of 10 mph\* (4.47 m/s) + bridge width of 50 m  
= 11.2 seconds under bridge

Sensor data rate of 66.23 Mbps + 11.2 seconds under bridge  
= 740.8 Mb = **92.6 MB of data**

Upload speed of 15 Mbps + 92.6 MB of data  
= **49 seconds to upload**

Figure 68: Data Volume Calculation

## 11.5 Sensor Outputs

<b>Range</b>	Distance of point from beam origin in mm
<b>Signal Photons</b>	Intensity/strength of return signal
<b>Ambient Photons</b>	Estimated ambient light/noise
<b>Reflectivity</b>	Estimated reflectance of target
<b>Timestamp</b>	Timestamp of measurement in ns
<b>Measurement ID</b>	Sequentially incrementing azimuth measurement (0 to 2047)
<b>Frame ID</b>	Index of scan, increments every rotation
<b>Encoder Count</b>	Azimuth angle as a raw encoder tick
<b>Beam Altitude</b>	Angle of range measurement above sensor XY plane
<b>Beam Azimuth</b>	Angle of range measurement w.r.t. radial line from center

Figure 69: Data Outputs from the Ouster OS1-32 Gen 1

## 11.6 Drawings

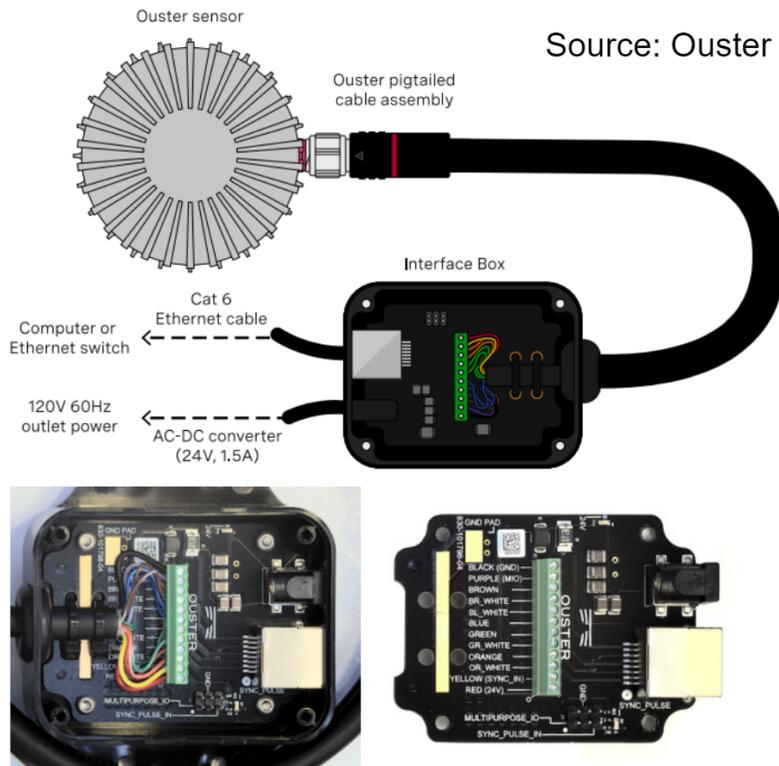


Figure 70: Physical Representation of the Ouster OS1-32 Gen 1 with its Interface Box and Connectors

## 11.7 Definitions

### 11.7.1 Metrics

**Range Resolution** - Indicates the smallest increment by which range measurements can be made → analogous to “ticks on a ruler”

**Accuracy** - How close are the measured points to the true/actual position of the structure being scanned?

**Resolution** - How far apart are the measured points? How dense is the point cloud?

**Precision** - How repeatable are the measurements? How much noise is observed in the point cloud?

### 11.7.2 Coordinate Frame

The Lidar Coordinate Frame follows the right-hand rule convention and is defined at the intersection of the lidar axis of rotation and the lidar optical midplane (a plane parallel to Sensor Coordinate Frame XY plane and coincident with the 0° elevation beam angle of the lidar).

**The Lidar Coordinate Frame axes are arranged with:**

- positive x-axis pointed at encoder angle 0° and the red external connector
- positive y-axis pointed towards encoder angle 90°
- positive z-axis pointed towards the top of the sensor

The Lidar Coordinate Frame is marked in both diagrams below with  $X_L$ ,  $Y_L$ , and  $Z_L$ .

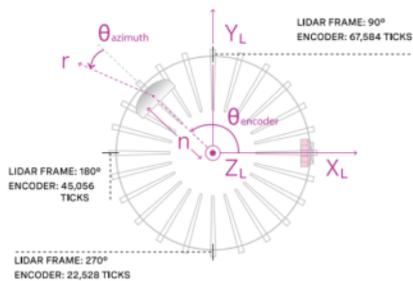


Figure8.1: Top-down view of Lidar Coordinate Frame

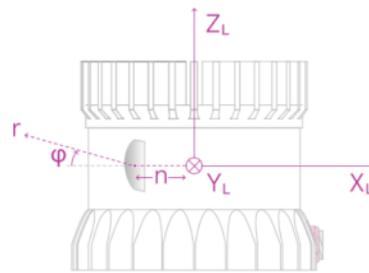


Figure8.2: Side view of Lidar Coordinate Frame

Figure 71: Ouster OS1-32 Gen 1 Coordinate Frame Information

**From an azimuth data block from the UDP packet:**

- `encoder_count` of the azimuth block
- `range_mm` value of the data block of the  $i$ -th channel

**From the `get_beam_intrinsics` TCP command:**

- `lidar_origin_to_beam_origin_mm` value
- `beam_altitude_angles` array
- `beam_azimuth_angles` array

The corresponding 3D point can be computed by

$$\begin{aligned}
 r &= \text{range\_mm} \\
 n &= \text{lidar\_origin\_to\_beam\_origin\_mm} \\
 \theta_{\text{encoder}} &= 2\pi \cdot \left(1 - \frac{\text{encoder\_count}}{90112}\right) \\
 \theta_{\text{azimuth}} &= -2\pi \frac{\text{beam\_azimuth\_angles}[i]}{360} \\
 \phi &= 2\pi \frac{\text{beam\_altitude\_angles}[i]}{360} \\
 x &= (r - n) \cos(\theta_{\text{encoder}} + \theta_{\text{azimuth}}) \cos(\phi) + n \cos(\theta_{\text{encoder}}) \\
 y &= (r - n) \sin(\theta_{\text{encoder}} + \theta_{\text{azimuth}}) \cos(\phi) + n \sin(\theta_{\text{encoder}}) \\
 z &= (r - n) \sin(\phi)
 \end{aligned}$$

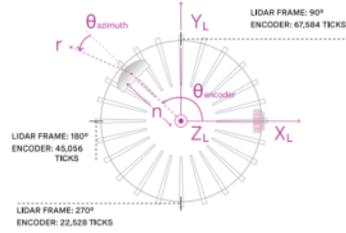


Figure 8.1: Top-down view of Lidar Coordinate Frame

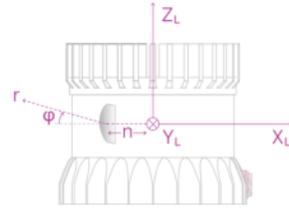


Figure 8.2: Side view of Lidar Coordinate Frame

Figure 72: Ouster OS1-32 Gen 1 Coordinate Frame Transformations

## 12 Appendix B: Structural Assessment

### 12.1 Scan Obstructions

Bridges come in all shapes and sizes, so it is important to acknowledge a limitation of the FLASH system in terms of what can't be scanned. The system may come across bridges that are supported with longitudinal beams/girders as shown in Figure 73 below.



Figure 73: Longitudinal Beams along Underside of Highway Bridge (Source: Getty)

Unfortunately, these girders will block out portions of the underside simply due to line-of-sight obstruction, so LiDAR data will not be captured for these areas (see Figure 74). However, given the fact the bottom flange width of these I-beams is typically small, shadowed areas are expected to be minimal as compared to areas of captured data. Hence, this does not pose a major threat to overall system feasibility.

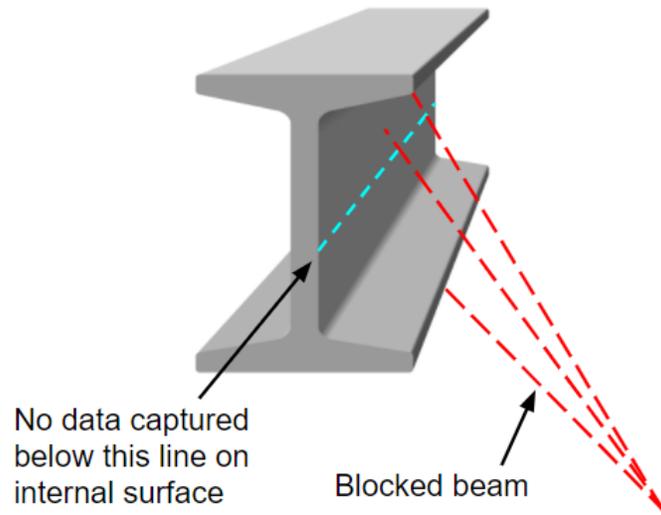


Figure 74: LiDAR Beam Blockage due to Bridge Girder

## 12.2 Bridge Inspections

<ul style="list-style-type: none"> <li>• <b><u>Types of damage/defects to be identified</u></b> <ul style="list-style-type: none"> <li>○ <b>Concrete spalling</b> <ul style="list-style-type: none"> <li>■ ~15 cm or more in diameter</li> <li>■ ~2.5 cm or more in depth</li> </ul> </li> <li>○ <b>Concrete delamination</b> <ul style="list-style-type: none"> <li>■ ~2.5 to 7.5 cm in size</li> </ul> </li> <li>○ <b>Destructive losses due to impact</b> <ul style="list-style-type: none"> <li>■ Size varies, but typically largest form of damage</li> </ul> </li> <li>○ <b>Corrosion in reinforcement</b> <ul style="list-style-type: none"> <li>■ ~5 to 20 cm in size</li> </ul> </li> </ul> </li> <li>• <b><u>Limitations</u></b> <ul style="list-style-type: none"> <li>○ Long-term deformation/displacement <ul style="list-style-type: none"> <li>■ On the mm scale</li> </ul> </li> <li>○ Cracking <ul style="list-style-type: none"> <li>■ On the mm scale</li> </ul> </li> </ul> </li> </ul>	<p>Source: Rollanet</p>
	<p>Source: GSG Distribution</p>

Figure 75: Various Structural Deformations and their Characteristics

Cause	Observations	Required resolution	Cause	Observations	Required resolution
<b>Bridge deck</b>					
<i>Sun shadow</i>	Shading	1m	Abutment shift	Relative displacement	0.025m
<i>Rain dampness</i>	Shading	0.5m	Pier displacement		0.025m
Car accident		1m	Bridge deck displacement		
<b>Section loss</b>		<b>0.5m</b>	Deck punch-through	Large openings	0.5m
Deterioration		0.1m	Deck corrosion		0.5m
Chemical spill	Discoloring	0.1m	Wear at joint	Gap at expansion joints	0.1m
Collision	Deformation	0.1m			
<b>Wearing surface</b>					
<i>New wear surface</i>	Discoloring	1.0m	Cracking	Shading	0.005m
Raveling	Local discoloring	0.5m	Potholing		0.1m
			Rutting		0.1m
<b>Railing</b>					
Missing railing		0.5m	Cracking	Shading	0.005m
<b>Cracking</b>	Shading	<b>0.005m</b>	Spalling		0.1m
Section loss		0.1m	Alignment	Curb edge detection	0.5m
Spalling		0.1m	Collision damage	Shading, edge detection	0.1m
<b>River bank (1 miles)</b>					
Pollution	De-vegetation	1m	Deterioration	Shading	0.1m
Smaller flow	River channel widening	0.5m			
<b>Traffic</b>					
Increase in ADT		1m	Scaling potion		0.1m
Increase in trucking			<i>Surrounding land use</i>	Changes in image	1m
Rush hour traffic					
Loading condition					
<b>Geometry of bridge</b>					
			Edge detection	Horizontal misalignment	0.5m
<b>Utilities</b>					
<i>Light shape, cables</i>		0.1m	<i>Traffic line</i>		1m

Source: UNC Charlotte

Figure 76: Example of a Bridge Inspection

Accuracy	HIGH < 0.05 m (< 0.16 ft)	MEDIUM 0.05 to 0.20 m (0.16 to 0.66 ft)	LOW > 0.20 m (> 0.66 ft)
Density	1A	2A	3A
FINE >100 pts/m <sup>2</sup> (>9 pts/ft <sup>2</sup> )	<ul style="list-style-type: none"> <li>Engineering surveys</li> <li>Digital Terrain Modeling</li> <li>Construction Automation/ Machine Control</li> <li>ADA compliance</li> <li>Clearances</li> <li>Pavement analysis</li> <li>Drainage/flooding analysis</li> <li>Virtual, 3D design</li> <li>CAD models/baseline data</li> <li>BIM\BRIM</li> <li>Post-construction quality control</li> <li>As-built/As-is/repair documentation</li> <li>Structural inspection</li> </ul>	<ul style="list-style-type: none"> <li>Forensics/Accident Investigation</li> <li>Historical Preservation</li> <li>Power line clearance</li> </ul>	<ul style="list-style-type: none"> <li>Roadway condition assessment (general)</li> </ul>
INTERMEDIATE 30 to 100 pts/m <sup>2</sup> (3 to 9 pts/ft <sup>2</sup> )	1B	2B	3B
	<ul style="list-style-type: none"> <li>Unstable slopes</li> <li>Landslide assessment</li> </ul>	<ul style="list-style-type: none"> <li>General Mapping</li> <li>General measurements</li> <li>Driver Assistance</li> <li>Autonomous Navigation</li> <li>Automated/semi-automatic extraction of signs and other features</li> <li>Coastal change</li> <li>Safety</li> <li>Environmental studies</li> </ul>	<ul style="list-style-type: none"> <li>Asset Management</li> <li>Inventory mapping (e.g. GIS)</li> <li>Virtual Tour</li> </ul>
COARSE <30 pts/m <sup>2</sup> (<3 pts/ft <sup>2</sup> )	1C	2C	3C
	<ul style="list-style-type: none"> <li>Quantities (e.g., Earthwork)</li> <li>Natural Terrain Mapping</li> </ul>	<ul style="list-style-type: none"> <li>Vegetation Management</li> </ul>	<ul style="list-style-type: none"> <li>Emergency Response</li> <li>Planning</li> <li>Land Use/Zoning</li> <li>Urban modeling</li> <li>Traffic Congestion\ Parking Utilization</li> <li>Billboard Management</li> </ul>

Source: National Cooperative Highway Research Program (NCHRP)

Figure 77: Examples of Infrastructure Inspections and their Required Accuracies

## 13 Appendix C: Trade Studies

### 13.1 LiDAR Sensor Trade Study

As there were many COTS options for LiDAR sensors, the team conducted a trade study during the 2020 Fall semester to select the LiDAR unit (see Figure 78). The eight criteria were accuracy, range, field of view, cost, data output, platform integration, mass, and power draw. These criteria were used to down select from the Velodyne Puck (High Res), Velodyne Puck, Ouster OS0, Ouster OS1, SICK MRS 1000, and Livox Mid-100.

		Velodyne Puck Hi-Res	Ouster OS0-32	Ouster OS1-16 (Gen 1)	SICK MRS1000	Livox Mid-100	Velodyne Puck
Criteria	Weight	Score	Score	Score	Score	Score	Score
Accuracy	7.5%	4	3	2	1	5	4
Range	7.5%	4	1	2	3	5	4
Field of View	30%	3	5	4	2	1	3
Cost	20%	1	2	4	3	5	3
Data Output	20%	3	5	4	2	3	3
Platform Integration	5%	4	3	3	5	3	4
Mass	5%	3	4	5	2	1	3
Power	5%	5	3	3	4	1	5
<b>Total</b>	<b>100%</b>	<b>2.9</b>	<b>3.7</b>	<b>3.65</b>	<b>2.45</b>	<b>2.9</b>	<b>3.3</b>

Figure 78: LiDAR Sensor Trade Study Decision Matrix.

Due to the team's limited budget, cost was a very important consideration in this selection. Based on the results of the trade study and conversations with both the manufacturer (Ouster) and the customer (ASTRA), the team decided to go with the Ouster OS1. Upon ordering this product, Ouster upgraded the team to the OS1-32 Gen 2 without increasing the price.

### 13.2 Mounting Mechanism Trade Study

Figure 79 shows the trade study done during the 2020 Fall Semester to determine the method of attachment. Keeping in mind the FLASH's objectives and design requirements, methods of attachment were traded based on their ease of attachment/detachment, stability risks, mass, cost, manufactured, and size. The three configurations considered were a fixed attachment, a magnetic attachment, and a clamping attachment.

		Fixed Attachment	Magnetic Attachment	Clamping Attachment
Criteria	Weight	Score	Score	Score
Ease of Attach/Detach	30%	2	5	4
Stability Risk	25%	5	3	4
Mass	20%	1	5	3
Cost	10%	1	5	4
Manufacturability	10%	2	5	4
Size	5%	2	5	2
<b>Total</b>	<b>100%</b>	<b>2.45</b>	<b>4.5</b>	<b>3.7</b>

Figure 79: Mount Mechanism Trade Study Decision Matrix

The magnetic attachment method won the trade study and also aligned with the FLASH objective. As will be discussed in Section 5, this attachment method was proven to meet all of the functional and design requirements related to the mount, so there was no need to run another trade study on the mount attachment.

### 13.3 SLAM Trade Study

Figure 80 shows the trade study conducted during the 2020 Fall Semester to determine the optimal SLAM algorithm for the FLASH software pipeline. Based on what the algorithm needed to accomplish and the inputs given by the OS1, the algorithms were traded on whether or not they would require an external IMU, whether or not they were independent of GPS input, their overall compatibility with the FLASH design, and their ease of use. At the time, the four best options were determined to be Google Cartographer, LIO-SAM, LIOM, and LOAM.

		Google Cartographer	LIO-SAM	LIOM	LOAM
CRITERIA	WEIGHT	Score	Score	Score	Score
External IMU	15%	5	5	4	1
Indep. GPS	30%	4	5	4	3
Compatibility	25%	3	5	3	3
User Engagement, Ease of Use	30%	3	3	2	2
<b>Total</b>	<b>100%</b>	<b>3.6</b>	<b>4.4</b>	<b>3.15</b>	<b>2.4</b>

Figure 80: Software Trade Study Decision Matrix.

When this trade study was conducted, the clear winner was LIO-SAM; however, this decision was made in part because the literature of the time made it seem that LIO-SAM could be run with a 6-axis IMU input when paired with VINS MONO initialization to null out yaw measurements. After many months of trying to combine LIO-SAM with VINS MONO, it was discovered this was not nearly as simple as the literature suggested and that LIO-SAM would require a 9 axis IMU input. At the time of this realization, FLASH's schedule would not allow for an external IMU to be incorporated. A decision was made to pivot to Ouster's proprietary SLAM algorithm WebSLAM.

WebSLAM was initially not considered because FLASH was unable to determine what was going on behind the scenes. However, after talking to Ouster engineers, it seemed that many comparisons could be made between WebSLAM and LIOM (the SLAM algorithm LIO-SAM was tuned from). With the finalized vertical orientation of the OS1 sensor, WebSLAM was able to produce point clouds that met the design requirements.

### 13.4 Laptop Trade Study

The biggest decision that needed to be made for this subteam was what to store and transmit data with. In the initial design phase in the fall, the team focused on which on-board computer had the most data storage capabilities, best Linux compatibility, lowest cost, and was easiest to use. FLASH was not weight or size limited in this capacity. Figure 81 shows the trade study between a Single Board Computer (such as a RaspberryPi), a Mini Computer (such as an Intel Nuc), and a Laptop Computer that was conducted in the Fall Semester of 2020.

		Single Board Computer	Mini Computer (Intel Nuc)	Laptop Computer
Criteria	Weight	Score	Score	Score
Data Storage	30%	2	4	5
Linux Compatibility	30%	1	5	5
Cost	10%	5	3	2
Ease of Use	30%	1	2	5
Total	100%	1.7	3.6	4.7

Figure 81: On-board Computer Trade Study Decision Matrix

As FLASH was not attempting to minimize the size, weight, or computing ability of the on-board computer, the Laptop Computer was selected. This design choice also made it easier to download and run all the necessary operating systems and software packages and save multiple data files because the laptop was specifically purchased for this project and did not require partitioning a team member's personal drive.

A trade study was not done for the selection of the specific laptop. Instead, the team chose a laptop capable of performing the necessary data collection and post-processing functions (based on design requirements) at a reasonable cost. The selected laptop was the Lenovo Legion 5 Gaming Laptop. This laptop has an AMD Ryzen 7 4800H Processor with 8 cores, 16 threads and capable of 4.2GHz boost clock to run CPU intensive programs. It satisfies DR 4.1 by having a 16GB DDR4 RAM and a 512GB SSD. Additionally the work station comes with an NVIDIA GTX 1660Ti capable of computing GPU heavy tasks such as point cloud rendering. It is also equipped with 802.11ax WiFi capable of delivering the teams data at the required rate (DR 8.2).