Design and Integration of Semantic Mapping System for Forest Fire Mitigation

Winnie Kuang

CMU-RI-TR-23-48

July 26, 2023

The Robotics Institute
School of Computer Science
Carnegie Mellon University
Pittsburgh, PA

Thesis Committee:
Professor George Kantor, *chair*
Professor Sebastian Scherer
Daniel McGann

Submitted in partial fulfillment of the requirements for the degree of Master of Science in Robotics.

To my family.

Abstract

Remote sensing technologies can provide an automated approach to monitor and analyze conditions in the forest environment over a period of time for forest maintenance and wildfire mitigation efforts. In particular, unmanned aerial vehicles (UAVs) are a promising remote sensing modality since they can traverse uneven terrain and provide on-demand high-resolution surveys of the environment with various sensors.

In this work, we present the mechanical design and system integration of a multi-sensing payload to harness such capabilities of UAVs and collect meaningful data in forest environments with the ultimate aim of localizing clusters of flammable vegetation. Our payload primarily consists of LiDAR, visual, and IMU sensors. We use pose information from implementing a keyframe-based SLAM algorithm on our data to globally register semantically labeled point clouds. Since the pose updates from the keyframe-based SLAM system are sparse compared to the semantically labeled point clouds, we implement a relative frame pose correction interpolation method that uses keyframe poses as constraints to derive corrected relative frame poses for registration. We demonstrate our approach using real-world data collected with our sensing payload on a UAV operating in forest environments.

Acknowledgments

Firstly, I would like to thank my advisor Professor George Kantor for his support and guidance over the past two years. I would also like to thank my committee members, Professor Sebastian Scherer and Daniel McGann for their valuable insights during our discussions.

My team members in the SafeForest project, Francisco Yandun, David Russell, and Duda Andrada have been wonderful to work with as they are supportive, patient, helpful, and always kept a positive attitude during the trials of field testing and research. I am especially thankful for Francisco who is the backbone of the project as he orchestrated the project on CMU's side from the get-go. I will cherish all the valuable skills and lessons he passed on to me in preparation for field testing and hope to emulate the dedication and hard work that he puts forth in the project. A special thanks go to Babak Chehreh who piloted the drone and contributed to the Porto datasets used in this study.

I am grateful to the members of the Kantor Lab and the FRC community for their companionship, insights, and support during lab meetings, design, and presentation reviews. I would like to thank Jim Picard and Tim Angert who were very helpful during the design and fabrication process of the sensing payload.

I am very fortunate and lucky to be surrounded by inspiring, humble, and thoughtful people at CMU who have supported me throughout. My friends in RI have all made my whole graduate experience memorable. I am especially thankful for my close friends at CMU, Aarrushi Shandilya, and Prasanna Sriganesh who have wholeheartedly supported me while things were in disarray and gave me valuable guidance. They have all inspired me with their passion for robotics and thoughtful character.

Finally, my deepest gratitude goes to my family and close friends outside of CMU who have cheered me on right from the start and believed in me.

Funding

This project was supported by SafeForest, under the reference CENTRO-01-0247-FEDER-045931 and Agencia Nacional de Investigación y Desarrollo (ANID) /PFCHA/DOCTORADO NACIONAL CHILE/2019-21190471. We would also like to thank and acknowledge support from the AI Research Institutes program's AI Institute for Resilient Agriculture (AIIRA) which is funded by NSF and USDA-NIFA under AI Institute: for Resilient Agriculture, Award No. 2021-67021- 35329.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

In recent years, climate change has been a key driver in increasing the number of wildfires due to the rise in global temperature and drastic weather changes [28, 31]. Prolonged periods of higher temperatures and droughts create highly combustible material in the forest which allows wildfires to thrive. These conditions can cause wildfires to become more intense, frequent, and long-lasting when left untreated, [19, 45]. Wildfires and the resulting irreversible damage can be difficult to prevent and control without well-maintained and active measures to monitor and manage the forest. However, orchestrating systematic measures to control wildfires is especially challenging since forest environments are typically large in size and the labor involved in preventative maintenance procedures is particularly labor intensive. Furthermore, the danger posed by addressing wildfires on-the-fly is highly laborious and dangerous.

Monitoring and managing forest fires by using remote sensing technologies is an effective approach to studying the conditions and changes of large geographic areas over a period or at a given time [11, 25]. Satellite imagery has been a popular remote sensing method due to its ability to cover large regions. Past efforts at utilizing remote sensing to address the prevalence of

wildfires have commonly used satellite imagery to map fuel types [36, 38, 47], burned areas [14, 32], vegetation recovery [46], assessments of fire risk [13, 44], fire/burn severity [34, 40], and detection of wildfires [9, 62]. While satellite imagery has previously been used for forest fire detection [10], and risk zone mapping [41], the high operational costs and low spatial resolution make this method insufficient in providing timely, precise mapping.

Unmanned aerial vehicles (UAVs) have shown promise as a remote sensing approach since they can cover large areas and provide high-resolution surveys of the environment on-demand with various sensing modalities (i.e., visual, thermal, spectral, or laser range). They are especially useful for tasks that require collecting information on difficult or dangerous to-access areas. Past researchers have examined using UAVs to help in fire-fighting [53, 61], scout and detect fires [12, 66], or capture data of the fire for fire modeling [59].

An alternative yet effective approach is to mitigate wildfires by reducing the accumulation of highly combustible biomass in the forest environment. This combustible biomass ranges from smaller biomass such as dried pine needles, grasses, shrubs, and small twigs, to larger wood pieces, downed trees, and logs [22]. Robotic forestry maintenance can play a key role in prevention tasks, especially by using a combination of aerial and ground vehicles to cover the vast size of the forest. Recently, researchers have proposed and investigated a collaborative system in which a UAV will be used to survey the environment to generate a multi-layered map comprising information including traversability and areas with an accumulation of combustible material, and an unmanned ground vehicle will use this information to remove highly combustible material from the forest [6, 16, 54]. Works following this have developed a perception system for an unmanned ground vehicle (UGV) [6] and for a UAV [54] to generate a semantically labeled mapping of the forest environment, indicating classes including "fuel" or highly combustible material, canopies, and trunks.

This thesis aims to expand on the research of the works [6, 54] by developing a perception system for robotic systems, particularly UAVs, with a focus on

applying this to the forestry domain. The perception system includes the hardware consisting of a suite of sensors, and an integrated Simultaneous Localization and Mapping (SLAM) and semantic mapping system which is used to generate a multi-layered mapping of the forest environment with semantic labels localizing fuel clusters.

1.1 Summary of Contributions

In summary, our contributions are: (1) a custom-designed modular sensing payload that includes visual, inertial, and LiDAR modalities; (2) and the integration of SLAM and semantic mapping system with a pose correction approach so that a dense globally registered map with semantic labels of fuel clusters can be generated.

This thesis is organized as follows:

- Chapter 2 provides a background on using unmanned aerial vehicles independently or in conjunction with unmanned ground vehicles for forest fire management. Furthermore, this chapter also discusses the existing works in SLAM and mapping frameworks.
- Chapter 3 details the sensing payload that was designed for our work, including the mechanical design, system hardware, and assembly process.
- Chapter 4 provides an overview of the state-of-the-art Lidar Inertial Odometry via Smoothing and Mapping (LIO-SAM), which is the SLAM algorithm used in this study.
- Chapter 5 discusses the semantic mapping system and the implemented pose correction approach.
- Chapter 6 follows with experiments and testing of the SLAM and mapping system discussed in the previous two chapters.
- Chapter 7 reviews the results of implementing our system on our datasets.

- Finally, Chapter 8 ends with concluding notes on the future direction of this research.

Chapter 2

Background

2.1 Autonomous Robotic Systems for Forestry Applications

Autonomous robotic systems present an attractive alternative to manual workforce in the field of forestry for applications such as environmental preservation, monitoring, inventory operations, and planting, pruning, and harvesting [4, 42]. In addition, these systems could also be used for search and rescue and disaster management in forest environments. Many mobile robot platforms, including wheeled, tracked and legged robots that comprise unmanned ground vehicles (UGVs) can navigate along trails in forest environments. While steep, uneven terrain and impassible ground from excessive vegetation and wetlands can present difficulties to UGVs [23], these terrain challenges can be easily navigated with unmanned aerial vehicles (UAVs). When used together, UGVs and UAVs are appealing options to use in forest environments: both robots can be equipped with sensors (i.e., RGB-D, multispectral, and thermal cameras, LiDAR, IMU, etc.), cover large regions in the forest, and create high-resolution maps from under-the-canopy and over-the-canopy surveys.

There has been a growing effort to enable autonomous robotic platforms to address the prevalence of large-scale wildfires in the United States and Europe in recent years [35, 51]. Robots can play a key role in replacing humans in tasks that are otherwise too physically demanding or dangerous to mitigate or control wildfires. Many researchers have considered using robots for a more active wildfire relief, including fire-fighting [5, 30, 52, 58]. Forestry maintenance can have a significant impact on controlling the prevalence of wildfires since multiple studies have attributed the increase in wildfire frequency and duration to climate change, with one of the key factors being longer dry seasons which render vegetation to be highly combustible [57]. Researchers have mapped potential fire-prone regions with combustible vegetation using manual or remote sensing techniques (i.e., satellite imagery). More recently, Couceiro et al. [17] have proposed using a collaborative system of robotics in which a UAV will survey the environment and generate a map that contains information on fire-prone regions with dry vegetation and relay this information over to a UGV to clear the region of the combustible vegetation. Additional works followed to enable this area of research [6, 50, 54].

2.2 Simultaneous Localization and Mapping for Robotic Systems

A fundamental requirement to successfully deploy autonomous robotic systems for forestry applications is for robots to perceive and navigate the forest environment, which is often accomplished using mapping techniques. The robot estimates its state using its suite of sensors and simultaneously generates a mapping of the environment to use as a reference for localizing and planning. The map, at its basis, captures low-level information such as spatial features of the environment, but it can also contain high-level information that the robot can use to achieve high-level tasks. For the robot to be able to use the map for tasks in real-time, all the data structures and operations used to

store and access the map must be lightweight and computationally efficient.

Implementing an autonomous framework in a forest environment is challenging since: (1) GPS signals can be weaker in certain areas of the forest, especially under dense forest canopies. Using differential or Real Time Kinematics (RTK) GPS would require reliable communication with a base station; (2) the environment dynamic (e.g., changing lighting conditions throughout the day, moving objects in the scene such as leaves, etc); (3) the environment can be cluttered.

Automating survey methods that involve deploying autonomous robotic systems to navigate in the field requires the robots to have a robust Simultaneous Localization and Mapping (SLAM) system to estimate their state in 3D space. Over the years, SLAM research has matured with the development of different systems based on various combinations of sensor modalities [18]. SLAM systems relying on LiDAR sensor measurements can be used to capture very fine details and generate a 3D map with a large field of view. LiDAR-based systems are invariant to lighting conditions, however, it is susceptible to unstructured environments, especially with repeating features [49]. Visual-based SLAM systems can provide rich information at a lower price point and weight but fail in environments where rich texture information is lacking and is sensitive to changes in lighting conditions, viewpoints, and sudden rapid movements [8]. LiDAR and visual modalities are often fused together with IMU measurements to improve the SLAM system's performance since IMU is unaffected by the environment's visual and structural features and can provide high-frequency precise pose estimation [18]. However, since there are noise and bias in IMU sensors, errors and drift can accumulate over time. The fusion of different sensing modalities can account for each of the individual sensor's weaknesses.

Generally, various fusion frameworks of SLAM algorithms are classified as loosely coupled and tightly coupled schemes. In loosely-coupled sensor fusion frameworks, the sensor data are processed independently by individual

sensor-specific modules which provide their own estimation of the robot's pose. Loosely coupled frameworks provide flexibility in choosing specific sensor modalities which make it easier to integrate new sensors into the system. The sensor fusion step is accomplished at a higher level when sensor outputs are combined with sensor registration or probabilistic fusion algorithms. LiDAR Odometry and Mapping (LOAM) [67] is a state-of-the-art loosely-coupled LiDAR and inertial-based SLAM algorithm. In tightly-coupled sensor fusion frameworks, sensor data from different modalities are jointly processed and combined at a lower level, usually through a filtering or optimization framework, to estimate the robot's pose and update the map. However, tightly-coupled frameworks require more complex algorithms and computational resources to achieve joint estimation and optimization. The LiDAR odometry via smoothing and mapping (LIO-SAM) [55] is a tightly-coupled LiDAR and inertial-based SLAM algorithm which expands on the LOAM framework by including loop closure methods and feedback from absolute measurements such as the GPS and compass heading. Works have also looked into fusing LiDAR, visual, and inertial modalities, such as [56] which featured visual-inertial odometry (VIO) system and a LiDAR odometry (LO) system. The VIO subsystem is tightly-coupled and serves as a motion model for the LiDAR mapping which dewarps LiDAR points and registers the scans to the global map.

2.3 Mapping Representations

Fusing semantic information is especially important for forest fire mitigation efforts as it provides a globally registered 3D representation of the environment containing classification information. While semantic SLAM has gained traction over the recent years and has been applied in areas such as autonomous driving or assistive robotics, implementing SLAM with semantic labels directly into the forest domain is still in its early stages.

Several approaches have been studied to map 3D environments. Earlier works represent the environment using point clouds, elevation maps [26, 48], multi-level surface maps [60], and in a volumetric manner [37, 39]. Among these, the latter approach has gained traction since it can be used to represent obstacle-free and previously unexplored places, are more memory-efficient, and can fuse multiple measurements in a probabilistic manner to account for sensor noise. In this approach, 3D space is divided into equally-sized cubic volumes such as voxels, with each voxel storing the probability of occupancy [39]. The voxel at the highest resolution is known as the leaf node. [37] developed this avenue of work with octrees for 3D mapping, which is a hierarchical data structure that recursively divides the 3D space into eight octants and can allow queries at different resolutions. Following this, [27] developed OctoMap, in which the voxels (or nodes) in the octree, store the log-odds occupancy value to indicate whether a node is occupied, free, or unknown. The use of log-odds occupancy can better handle sensor noise and dynamic environments. The log-odds occupancy value is clamped to a value to allow for pruning which happens when all eight of the node's children are the same. Pruning the tree leads to a smaller tree so that traversing the tree is easier and memory usage is less. Using octree-based mapping allows for explicit representation of free-space and unmapped areas, which is important for exploration and collision avoidance tasks. However, creating an OctoMap requires more computational resources than point-based representations due to every depth measurement being ray-casted into the map.

Chapter 3

Sensing Payload Design

3.1 System Overview

We developed a multi-sensor payload to implement our SLAM and semantic mapping framework in the forest environment, as shown in Figure 3.1. Our implementation draws inspiration from previous works [56], [55] and features a combination of a LiDAR, an inertial measurement unit (IMU), and a stereo camera pair tightly coupled in a factor graph scheme to derive 6DOF pose information and 3D mapping of the environment using point clouds. We chose to include all three sensors in our sensing payload to have a broad range of sensing modalities with different fields of view and spectral characteristics, and to account for situations where any of the sensors is susceptible to poor performance. The aerial platforms we used in this study are the commercial drones, DJI M600 Pro, and Freefly Alta X, due to their abilities to carry larger payloads while sustaining longer flight times, as shown in Figure 3.2.

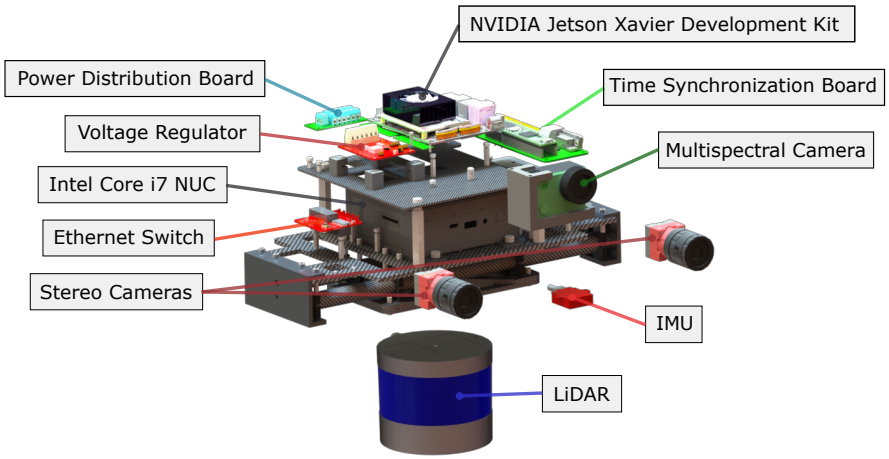
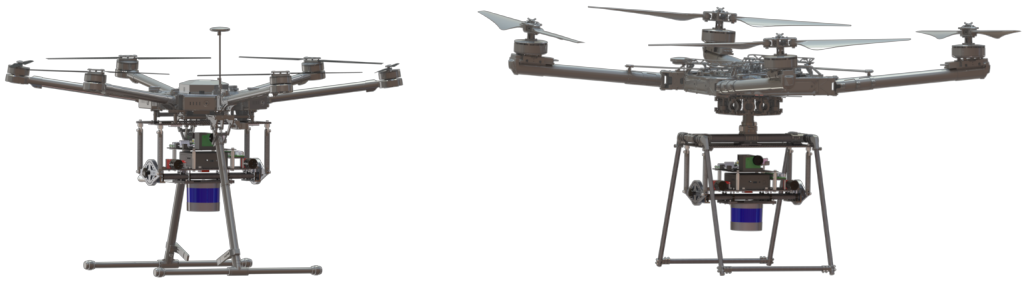


Figure 3.1: Multi-sensor payload



(a) (b)

Figure 3.2: Aerial platforms with sensing payload mounted: (a) DJI M600 Pro (b) Freefly Alta X

3.2 Design Requirements

The sensing payload will primarily be mounted onto an aerial platform. Therefore, the design of the payload factors in the key requirement of being low-weight and compact. A lighter payload will result in longer flight times. Furthermore, a compact design will minimize the impact of adverse motion of the UAV during flight. Additional design requirements hold to enable the payload's components to be accessible and conducive to experimental tests,

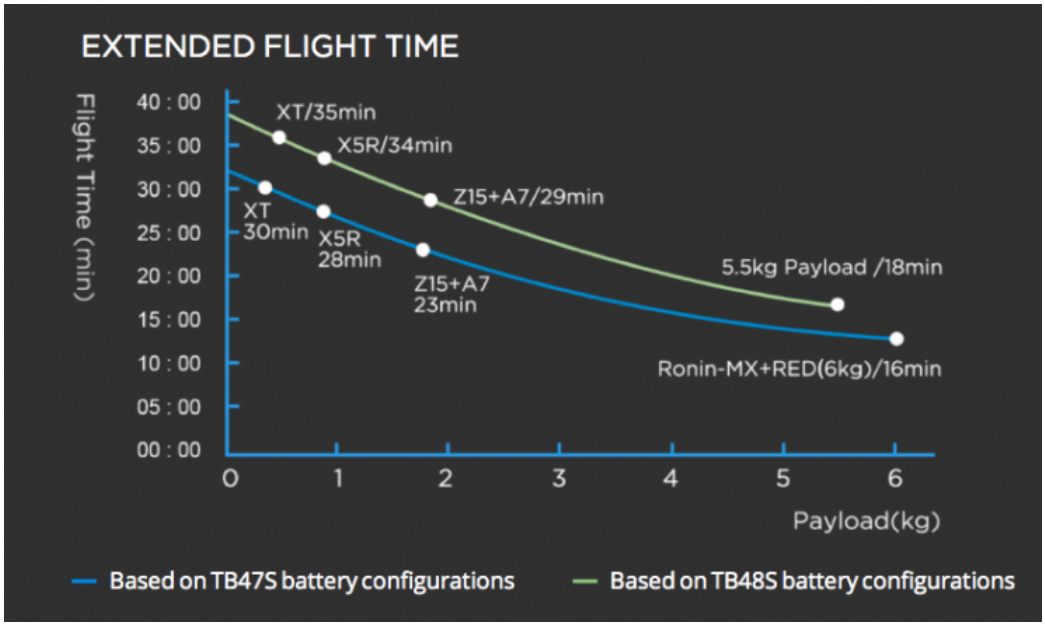


Figure 3.3: Estimated flight times for the DJI M600 Pro[2]

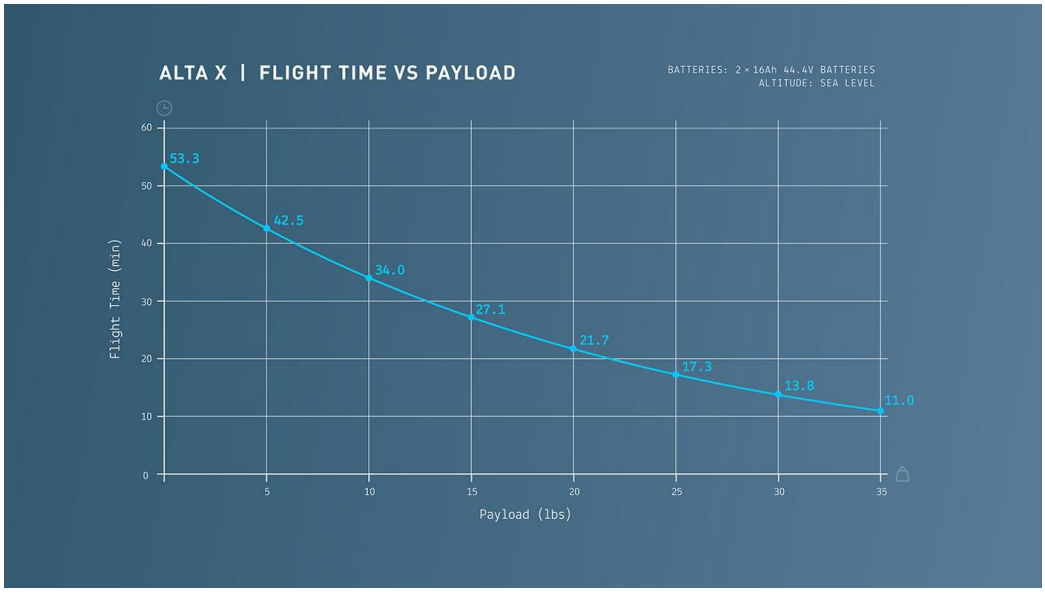


Figure 3.4: Estimated flight times for the Freefly Alta-X [3]

and allow the payload to be integrated with the chosen aerial platforms. The design requirements are summarized as follows:

1. **Component placement:** The stereo cameras need to be distanced with a baseline of 25.0 cm, according to the desired depth of field calculations. The IMU should ideally be close to the center of mass of the payload. Furthermore, the components of the payload can be easily accessed and removed. Finally, the ports of the components, which are required for operation, can be easily accessed.
2. **Structure and form factor:** The design should be compact and the center of mass of the payload is as close to the center of the payload as the height where the axis of mounting is. The structure of the payload is lightweight for mounting onto the aerial platform and robust to support the components and sustain environmental and operating conditions.
3. **Mounting:** The payload can be easily mounted and unmounted onto an aerial platform without interference, especially if it is mounted below so that the landing gear can still be used. The design of the payload should also have a mechanism to maintain and keep track of pitch angles allowing the payload to collect data at different orientations, ranging from horizontal level to nadir.

3.3 Constraints

There were a number of constraints considered during the design process, including the following:

1. **Aerial platform:** The two aerial platforms considered are the DJI M600 Pro and the Freefly Alta X, as shown in Figure 3.2. These two aerial platforms are capable of supporting larger payloads. The flight times are documented in Tables 3.2 and 3.1, and Figures 3.4 and 3.3.
2. **Safety of operation:** Mounting additional components on the UAV should not significantly change the center of gravity of the UAV since doing so would result in unfavorable effects on the UAV's stability. Furthermore, components added on should not interfere with the signals and accessibility of the other components.
3. **Vibration:** Vibration is present during flight and can influence sensor measurements. To mitigate the effects of vibration (i.e., the loosening of components), components were constrained very tightly with split lock washers or with internal tooth washers and additional Loctite. Furthermore, the vibration isolator unit was used when flying with the Freefly Alta X drone.

Table 3.1: DJI M600 Pro specifications

Specification	**Value**
Weight*	9.5 kg
Max takeoff weight	15.5 kg
Diameter**	1668 mm
Battery capacity	4.5 Ah
Battery voltage	22.2 V
Hovering time	32 min (no payload), 16 min (6 kg payload)

* with six TB47S batteries
** including propellers

Table 3.2: Freefly Alta X specifications

Specification	Value
Weight*	19.8 kg
Max takeoff weight	34.86 kg
Diameter**	2273 mm
Battery capacity	16 Ah
Battery voltage	44.4 V
Hovering time	50 min (no payload), 26.6 min (6.8 kg payload)

* with two 16 Ah batteries
** including propellers

3.4 System Hardware

Our sensing payload primarily comprises an inertial measurement unit (VN-200 Rugged GNSS/INS), a three-dimensional laser range sensor with 32 channels (Velodyne VLP-32), a 20.2 Megapixel stereo camera with 8.5 mm focal length lenses (Allied Vision Alvium 1800 U-2040c), a multispectral camera (Mapir Survey3W, R, G, NIR), two onboard computers (NVIDIA Jetson Xavier Development Kit and the Intel NUC 11 Core i7), an ethernet switch (BotBlox GigaBlox), voltage regulator board (Vishay SIC438AEVB-B), and a custom time synchronization board.

Table 3.3: Sensors and operation frequency

Sensor Name	Sensor Type	Frequency (Hz)
VN-200 Rugged GNSS/INS	IMU	400
Velodyne VLP-32C	LiDAR	10
Allied Vision Alvium 1800 U-2040c	RGB camera	10
Mapir Survey 3	multispectral camera	15

3.4.1 Sensor Selection

Components were selected based on their performance, form factor, and weight. An ethernet switch (BotBlox, GigaBlox) was used so that sensor information can be exchanged between the two onboard computers. Table 3.3 shows the frequencies of all the sensor components included in the payload.

LiDAR

We have compared multiple LiDARs, such as the Velodyne Puck-32MR, Velodyne Puck LITE, and Velodyne Ultra Puck, due to their superior performance. Ultimately, the LiDAR selected is the Velodyne Ultra Puck (VLP-32C), as shown in Figure 3.5. This LiDAR model is commonly used for state-of-the-art robotic systems with the capability to perform SLAM since it has high performance and can capture a denser point cloud representation of the map. The Velodyne Ultra Puck was chosen among the other options since it has a compact form factor and can produce 32 channels to provide a full horizontal view of the environment and generate accurate real-time 3D point cloud data with a range of 200 m. In the forestry domain, the range and accuracy of this particular LiDAR are excellent as a range of experiments, including under and over-canopy surveys, will be conducted.

Figure 3.5: Velodyne VLP-32c

Table 3.4: Specifications for the LiDAR

Specifications	**Values**
Number of channels	32
Range	200 m
Accuracy	± 3 cm
Horizontal field of view	360°
Vertical field of view	40°
Angular resolution (horizontal/azimuth)	$0.1° - 0.4°$
Mass	925 g

Inertial Measurement Unit

Since we are implementing SLAM algorithms with our payload and fusing IMU measurements with other sensor measurements, we are opting for a robust, high-performance IMU. Furthermore, since we are collecting data on our payload attached to a UAV flying in the forest environment, we are also aiming to choose an IMU that can handle such conditions present. Two IMU models stand out which are the Xsens MTi 30 AHRS and the Vectornav (VN) 200 Rugged GNSS/INSS. Ultimately, the VN-200 Rugged GNSS/INS was selected since it offered a high heading and pitch/roll accuracy, a small form factor, and is lightweight. The VN-200 Rugged GNSS/INS is a combination of MEMS inertial sensors and a high-sensitivity GNSS receiver. Having the option of the GNSS receiver would be beneficial to provide ground truth data.

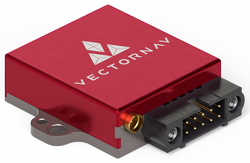


Figure 3.6: VN-200 Rugged GNSS/INS

Table 3.5: Specifications for the IMU

Specifications	**Values**
Horizontal position accuracy	1 m RMS
Vertical position accuracy	1.5 m RMS
Angular resolution	0.001°
Velocity accuracy	<0.05 m/s RMS
Heading accuracy	0.2°
Pitch/roll accuracy	0.03°
In-run bias stability gyroscope	<10°/h
Mass	16 g
GPS	Yes

Stereo Cameras

The stereo cameras that were chosen in this study are the the Allied Vision Alvium 1800 U-2040 with the TECHSPEC 8.5 mm fixed focal length lens. This camera has a Sony IMX541 CMOS sensor, which is a high resolution sensor. The maximum frame rate is at 21 fps (however, as shown in Table 3.3, during operation, the frequency is 10 fps). Figure 3.9 shows the field of view achieved when the object distance is $0.5 - 1.0$ m away, with various focal lengths. This focal length was chosen because it was appropriate for the proximity the UAV will be flying to objects (i.e., trees, canopy, ground, etc) in the forest environment while providing a wide-angle of view and lower magnification. A wider view allows the UAV to capture representative imagery of the environment. The baseline of the stereo pair was chosen to be 25.0 cm.

Figure 3.7: Allied Vision Alvium 1800 U-2040c

Figure 3.8: TECHSPEC 8.5 mm C Series fixed focal length lens

Table 3.6: Specifications for the stereo cameras

Specifications	**Values**
Sensor type	CMOS
Sensor model	Sony IMX541
Sensor size	14.10 mm × 10.30 mm
Pixel size	2.74 ν× 2.74 ν
Resolution	4512 pixels (H) × 4512 pixels (V)
Max frame rate at full resolution	21 fps at 450 MByte/s, Mono8
Mass	65 g
Mass of lens	80 g

Focal length [mm]	Field of view (H × V in [mm])	
	Object distance = 500 mm	Object distance = 1000 mm
6	1018 × 1018	2048 × 2048
8	760 × 760	1533 × 1533
12	503 × 503	1018 × 1018
16	374 × 374	760 × 760
25	235 × 235	482 × 482
35	164 × 164	341 × 341
50	111 × 111	235 × 235
75	70 × 70	152 × 152
85	60 × 60	133 × 133

Figure 3.9: Alvium 1800 U-2040m/c with Type 1.1 (17.5 mm diagonal) sensors: focal length vs. field of view [1]

Multispectral Camera

Multispectral cameras are used to measure light emissions and can be used to aid in fuel mapping and vegetation classification in forests. We chose the Mapir Survey 3W camera which has wide lenses and red, green, and Near Infrared band (NIR) bands. In the forest environment, this spectrum is especially useful for detecting dry and flammable vegetation. This camera features a rolling shutter and a resolution of 12 Megapixels.

Figure 3.10: Mapir Survey 3W multispectral camera (RGN, NDVI)

3.4.2 Computers

We decided to include the two onboard computers, namely the NVIDIA Jetson Xavier NX Development Kit and the Intel NUC 11 Core i7 (NUC11PAHi7 Panther Canyon Mini PC i7-1165G7), to divide the computing load for recording and processing the data. These two computers were chosen because they have excellent computational capabilities and the connection interfaces allow the integration of the other components of the payload into the system. Furthermore, their small form factor and low power consumption make these two computers desirable for a UAV payload for forestry applications.

The NVIDIA Jetson Xavier NX Development Kit was chosen to run machine learning models. The Xavier has the NVIDIA Volta architecture with 384 NVIDIA CUDA cores and 48 Tensor cores, and the 6-core NVIDIA Carmel ARM v8.2 64-bit CPU. The Intel NUC 11 computer was intended to handle all the heavier CPU calculations (i.e., SLAM processes). This computer's processor is the 11th Generation Intel Core i7-1165G7 processor 2.8 GHz 4.7 GHz Turbo, which has 4 cores, 8 threads, and 12 MB Cache.

(a) (b)

Figure 3.11: Onboard computers: (a) NVIDIA Jetson Xavier NX Development Kit, (b) Intel NUC 11

3.4.3 Ethernet Switch

The ethernet switch was used as a router to connect the two onboard computers together with the time synchronization board and the SSH port. This allowed information and data acquisition between the two computers to be synchronized. The ethernet switch that was used is the GigaBlox Small GigaBit Switch, which is the most lightweight and has the smallest form factor among many of the ethernet switchboards available.

Figure 3.12: BotBlox GigaBlox

3.4.4 Synchronization Board

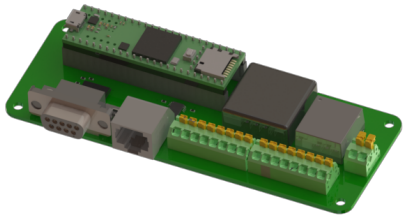


Figure 3.13: Custom synchronization board

Time synchronization among the sensors and computers is a critical issue and has a significant impact on the quality of the SLAM system. Since we had two separate onboard computers, a time synchronization board was necessary to synchronize the time between the two. Furthermore, we needed a method to trigger the stereo cameras. We designed a custom time synchronization board featuring a microcontroller, Teensy 4.1 Development Board, that was used to synchronize the two onboard computers and therefore the sensors connected. These sensors such as the LiDAR, IMU, and cameras were synchronized by having the synchronization board simulate Global Positioning System (GPS) signals, namely, pulse per second (PPS) signals and National Marine Electronics Association (NMEA) messages. NMEA messages are a type of formatted data that is supported by all GPS manufacturers and includes information such as position, velocity, and time. Having the microcontroller mimic GPS signals is desirable since forest environments can contain certain areas which would cause the GPS' PPS signal to be unreliable. Furthermore, the time synchronization board also contains the DC-DC converters: PDQ30-Q24-S12-D (12V, 30W) and Mornsun URB2405MT-3WR3 (5V, 3W).

The NVIDIA Jetson Xavier computer receives NMEA messages via serial communication and the pulse-per-second (PPS) signal which is the main clock source. The GPSD service is a service used to process GPS data from a receiver and is used to read the NMEA strings and the PPS signals so that they can be usable. The system time is synchronized with the NMEA strings

and PPS signal by using the implementation of the Network Time Protocol (NTP) called chrony. Chrony is typically used to synchronize the system clock across different NTP servers.

3.4.5 Voltage Regulator

A voltage regulator board was used so that the appropriate voltage was distributed to the respective components. Figure 3.15 and Table 3.7 detail the wiring and the voltage specifications for all components, respectively. The wiring was designed so that the sensing payload could be powered by the UAV's batteries. This would decrease the total weight of the payload since a separate battery is not required.

Figure 3.14: Vishay SIC438AEVB-B

Table 3.7: Voltage specifications for components

Component	Voltage (V)
IMU	3.2 - 5.5
LiDAR	10.5 - 18..0
Stereo cameras	5.0
Multispectral camera	5.0
Xavier	9.0 - 20.0
Intel NUC	12.0 - 19.0
Ethernet board	5.0 - 60.0
Synchronization board	18.0 - 36.0

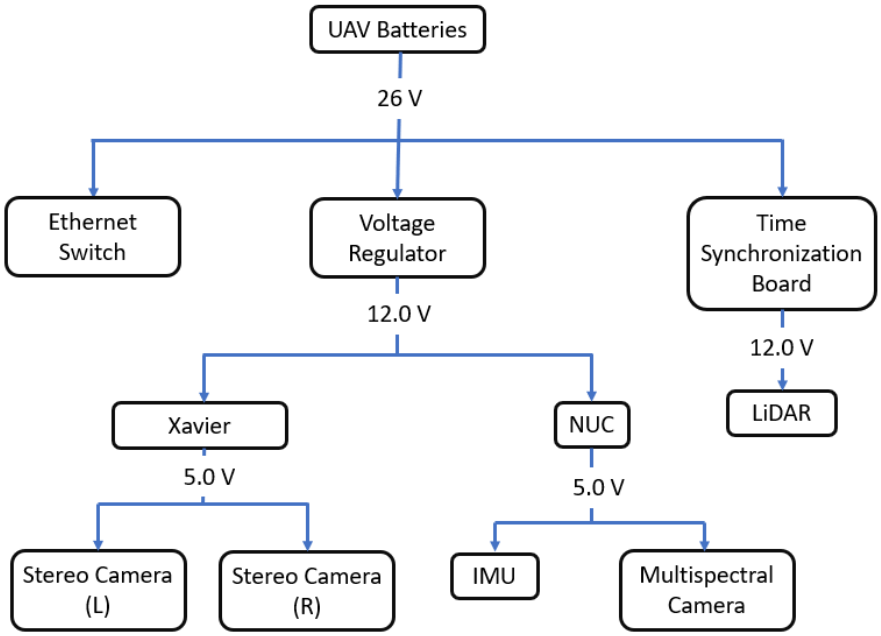


Figure 3.15: Overview of voltage breakdown among components

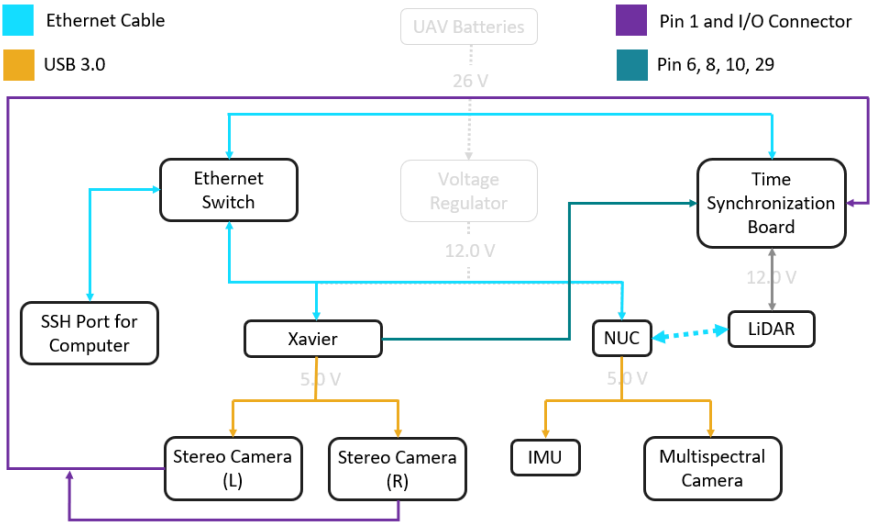


Figure 3.16: Overview of connection type among components

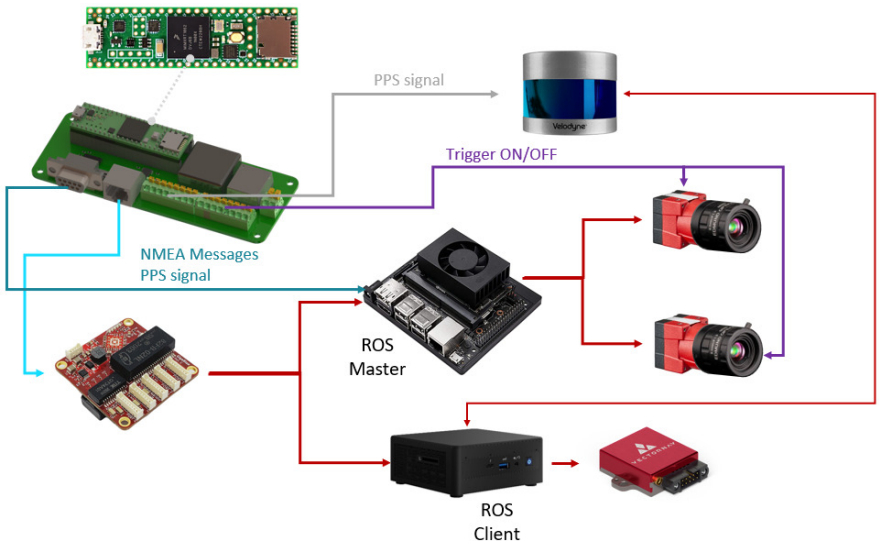


Figure 3.17: Overview of ROS connection among components

3.4.6 Wiring

The multi-sensor payload acquires its power from the drone's battery. As shown in Figure 3.15, the aerial platform, namely the DJI M600 Pro, provides a voltage of 22.2 V, which is regulated using the power distribution board to 12 V for components such as the stereo cameras, Xavier board, the ethernet board, and the IMU. The remaining components, such as the intel NUC, and the LiDAR can operate on the unregulated 22.2 V. We note that the Freefly Alta X does not need a voltage regulator since it is able to regulate the voltage of the batteries to the appropriate voltages. In either case, the NUC and LiDAR can directly be connected to the drone's batteries. If the DJI M600 Pro is used, the remaining components need to be connected to a voltage distribution board that regularizes the voltages to suitable values.

The ethernet board allows information to be communicated between the Xavier and the NUC computers. The two stereo cameras are connected to the Xavier. The IMU, multispectral camera and the LiDAR are connected to the NUC. The LiDAR, NUC, the Xavier, and the time synchronization

board are connected to the ethernet board. Figure 3.16 shows the connection type, and Figure 3.17 displays an overview of the ROS connection. Since the Xavier is the ROS master and thus, will allow ROS nodes to locate one another, the synchronization board is directly connected to the Xavier.

3.5 Mechanical Design

The mechanical design of the sensing payload features a stacked, layered design so that the weight of the sensing components can be distributed rather than concentrated on a main base plate and the payload can feature a pitching design. Carbon fiber was the main material used for the two levels of the payload, which contain stand-offs and mounting for the sensors and computers, to reduce weight without compromising the structural integrity of the payload. Two carbon fiber rods are used as the main support and to provide spacing between the top and bottom platforms so that the IMU can be positioned close to the center of mass of the payload. Positioning the IMU as close to the center of mass of the payload will make processing transformations of sensor information from the other components with respect to the IMU easier.

To satisfy the form factor requirement, the width and height were determined according to sensor requirements and how well the payload can be integrated with the aerial platforms. In this manner, the width of the payload was determined so that the stereo cameras can be placed with a baseline of 25.0 cm and the payload can fit underneath the payload with the landing gear deployed for both aerial platforms. The height of the payload was kept at a minimum to fit the components so that the center of mass is low and the payload can be mounted at the bottom of the aerial platform. Mounting the payload at the bottom of the aerial platform would allow the LiDAR to have minimal occlusion when capturing scans during over-the-canopy flights.

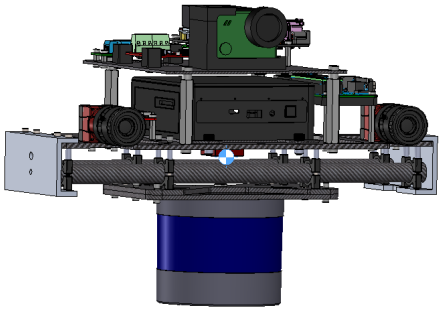


Figure 3.18: Calculated center of mass of the sensing payload (isometric)

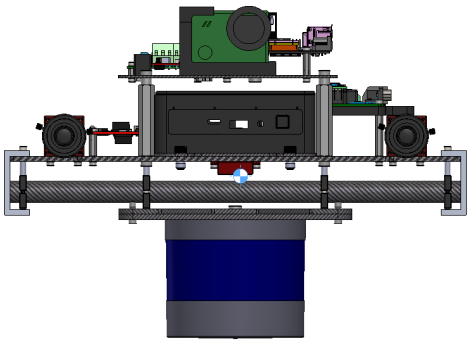


Figure 3.19: Calculated center of mass of the sensing payload (front)

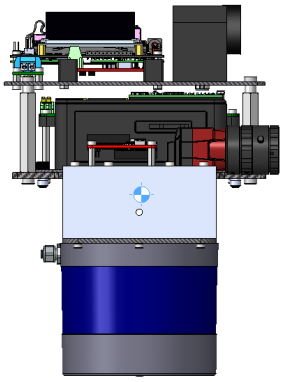


Figure 3.20: Calculated center of mass of the sensing payload (side)

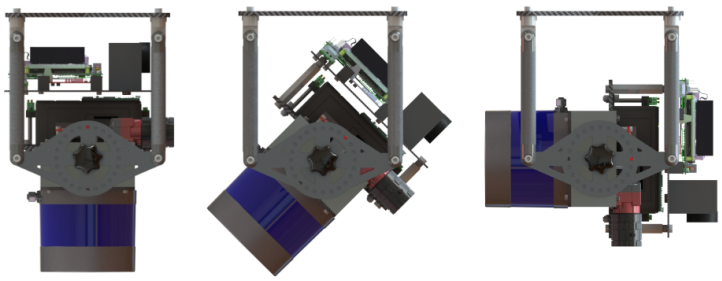


Figure 3.21: Pitching the sensing payload: horizontal (left) to nadir (right)

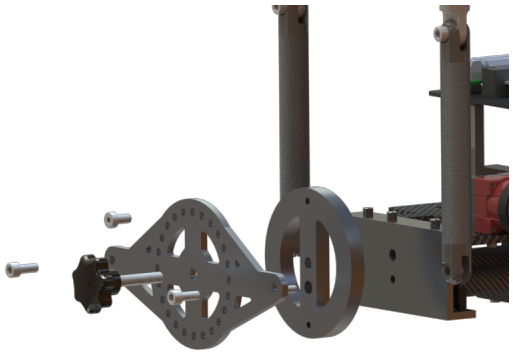


Figure 3.22: Aluminum mounting side brackets for maintaining the angling of the payload (exploded view)

Aluminum mounting brackets on the side of the payload feature holes in increments of 15° so that the payload can be oriented at different pitch angles with respect to the horizontal plane, as shown in Figure 3.21. As shown in Figure 3.22, the aluminum mounting brackets on the side of the payload consist of three main parts: the hand, the bracket with slots to keep track of the pitch angles, and the connector support that connects to the sides of the payload and stabilizes the bracket.

The payload design is made so that sensors and components can be removed without disassembling the entire payload. In order to prevent the

components of the payload from having an adverse effect on the movement of the aerial platform during flight, the components were placed so that the center of mass of the payload was as close to its volumetric center as possible. Table 3.8 provides a mass breakdown of the primary components. Figures 3.18, 3.19, and 3.20 show the location of the center of mass of the payload.

Table 3.8: Mass of payload components

Component	**Mass (g)**
IMU	16
LiDAR	925
Stereo cameras (L)	145
Stereo cameras (R)	145
Multispectral camera	49.8
Xavier	180
Intel NUC	600
Ethernet board	18
Synchronization board	63
Voltage regulator board	25
Voltage distribution board	20
External SSD	10
Wiring and structure	1,635.2
Total	3,832

3.6 Integration with Aerial Platform

The aerial platforms considered in this study are the DJI M600 Pro and the Freefly Alta X. A mounting plate was machined out of carbon fiber using the CNC router machine. There are two distinct mounting plates used to connect the payload system to the two different aerial platforms. Figure 3.23 is the mounting plate designed for the DJI M600 Pro, and Figure 3.24 is the mounting plate designed for the Freefly Alta X which is attached to the landing gear.

(a) (b)

Figure 3.23: Mounting plate for DJI M600: (a) top view, (b) isometric view

(a) (b)

Figure 3.24: Mounting plate for Freefly Alta X: (a) top view, (b) isometric view

3.6.1 Material Selection

The payload's main structure is composed of carbon fiber due to its high strength-to-weight ratio. Aluminum was also used as supports on the sides and parts were used as connections because it is lightweight, has good strength, and is corrosion-resistant. The remaining integral parts used for the assembly process, including screws, nuts, and bolts were chosen to be high-strength

anti-corrosive stainless steel.

Figure 3.25: Front view of the main structure of the sensing payload. Color code: aluminum (yellow), nylon (cyan), carbon fiber (red/black), steel (orange)

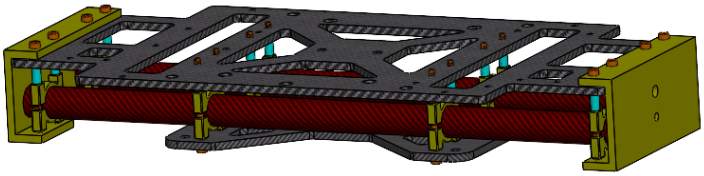


Figure 3.26: Isometric view of the main structure of the sensing payload. Color code: aluminum (yellow), nylon (cyan), carbon fiber (red/black), steel (orange)

3.6.2 FEA Analysis

Finite element analysis (FEA) was performed using SolidWorks to determine how well the sensing payload's structure behaves under loading conditions from the weight of all the components. Tables A.4, A.3, A.2 and A.1 show the material properties of all the relevant materials. As a preliminary step, the structural rigidity of the main carbon fiber plate as a standalone was tested. Following this, FEA was conducted on the main carbon fiber plate with two carbon fiber tubes added. Adding the carbon fiber tubes will help improve the payload's structural rigidity. The carbon fiber tubes also provide clearance to fit the IMU close to the center of mass of the payload and allow the payload to be rotated on an axis to vary the pitch.

The material properties selected to conduct FEA were chosen based on the discussion in the Material Selection section. The loading conditions across the FEA studies on the base plate and the base plate with the carbon fiber rods were determined such that the loads represented the components that were placed in specific areas on the base plate while keeping the sides of the payload constrained. In this sense, the total mass of all the components, shown in Table 3.8, excluding the weight of the structure and the wiring, was used to determine the load. This load value was then multiplied by a factor of two for a conservative study.

Figure 3.27 shows the results, and Table 3.9 summarizes the results from this study. From the results, it is evident that adding the two carbon fiber rods as supports reduces the maximum displacement values, however, the stress appears to be higher. This may be a result of the stress being concentrated along certain corners of the payload since the assembly with the supports is more complex. Figure 3.27 shows that much of the main structure of the base plate is at the lower end of the stress range.

Table 3.9: FEA results for base plate studies

Static Study	**Total load (N)**	**Max stress (MPa)**	**Max displacement (mm)**
Base plate	43.454	16.92	0.07627
Base plate with supports	43.454	39.79	0.06020

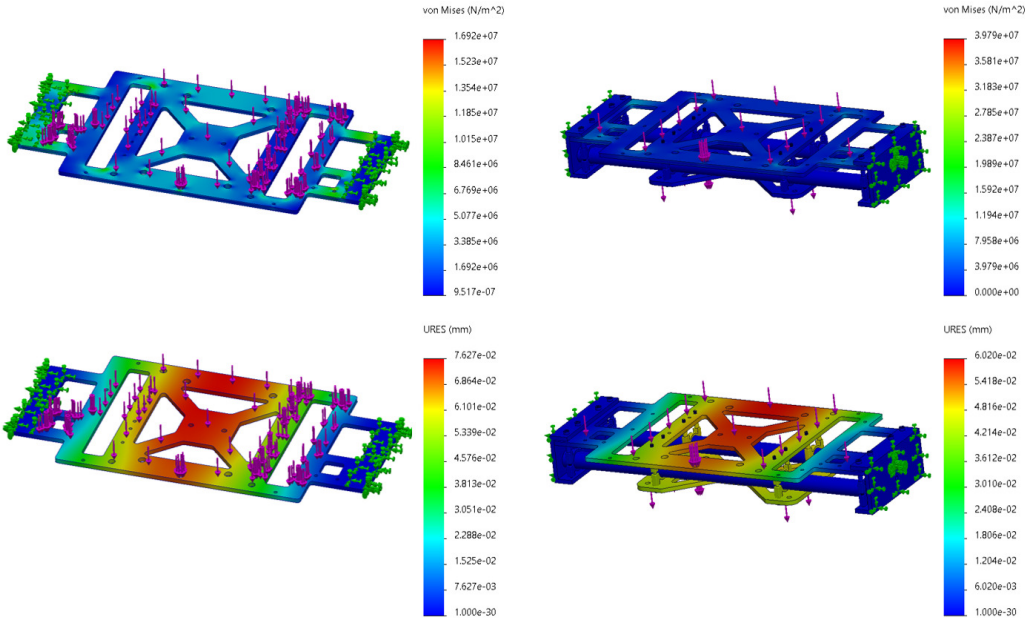


Figure 3.27: Base plate (left) and reinforced base plate (right) of the sensing payload - stress (top) and displacement (bottom) results: constraints (green), loads (purple)

FEA was also performed to determine the behavior of the mounting plates for the DJI M600 Pro and Freefly Alta X under loading conditions that simulated the entire weight of the payload, including the supports to connect the sensing payload to the payload plate. Figures 3.28 are the results corresponding to the mounting plate for both aerial platforms. Table 3.28 summarized the results for both mounting plates. These results show that the mounting plates designed for both aerial platforms can support the weight of the sensing payload.

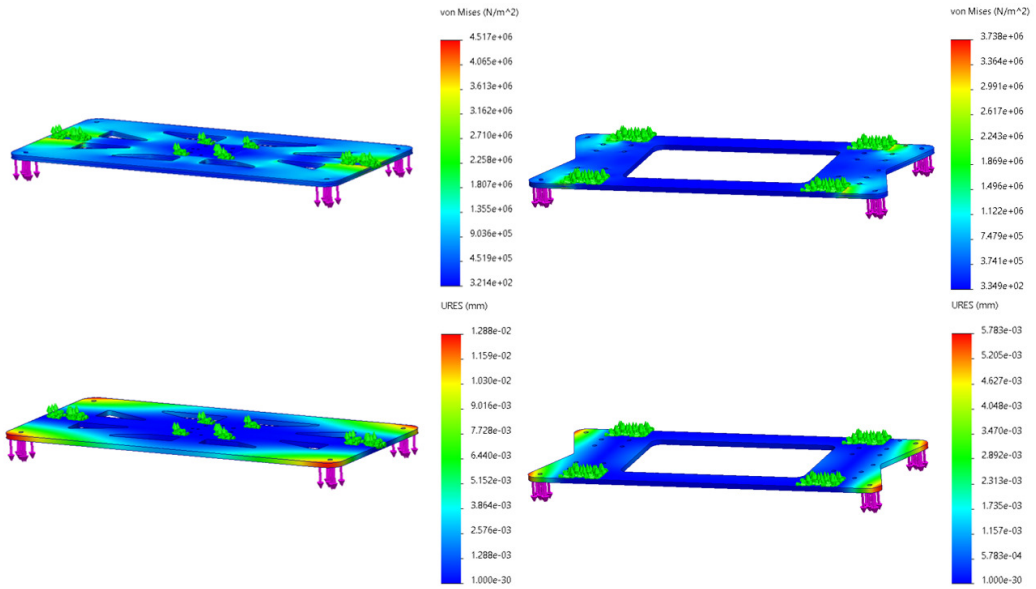


Figure 3.28: Freefly Alta X mounting plate (left) and DJI M600 Pro mounting plate (right) stress (top) and displacement (bottom) results

Table 3.10: FEA results for mounting plate studies

Static Study	**Total load (N)**	**Max stress (MPa)**	**Max displacement (mm)**
DJI M600 Pro Mounting	60	3.738	0.005783
Freefly Alta X Mounting	60	4.517	0.01288

3.7 Assembly

Figure 3.29 shows the placement of all the components in the payload. The IMU is oriented underneath the NUC onboard computer so that it is close to the center of mass of the payload. To access the IMU, the sides of the payload need to be removed. Ideally, the IMU should include the wiring before mounting onto the payload.

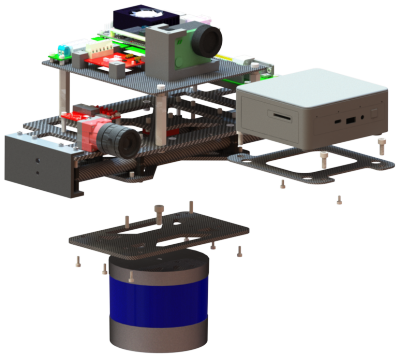


Figure 3.29: Exploded view of payload showing the removable plates for the NUC computer and the LiDAR

3.8 Fabrication

Carbon fiber was used as the main structure of the payload. The payload consists of two layers, fabricated using carbon fiber sheets and supported with two carbon fiber rods. The CNC router was used to cut the carbon fiber plates from the carbon fiber sheet. Rivet nuts were used to connect the carbon fiber plates together without the use of hex nuts. Standoffs were chosen so that there was enough clearance below the boards so that there could be circulation underneath the board and the components could be stacked. Epoxy was used to hold the carbon fiber tube supports which connect the main structure of the payload to the mounting plate.

The sides of the payload are manufactured out of aluminum since it is lightweight and rigid. An aluminum column was used as the side brackets and then modified by cutting holes to insert the knob and screw so that the payload can be angled to have different pitching degrees and be fixed.

Many of the mounts for components such as the power distribution board, and the multispectral camera were 3D printed. This allowed the mounts to be custom-designed and lightweight.

Chapter 4

Simultaneous Localization and Mapping Framework

4.1 System Overview

In our study, we chose to focus on LiDAR-inertial-based SLAM systems. While cameras are excellent at providing color and texture information about the environment, they are susceptible to variable illumination conditions and motion blur. LiDARs are robust and invariant to illumination conditions and can obtain accurate depth information of the scene. Results from [54] have found that the visual odometry had higher RMSE compared with LiDAR odometry. Hence, we use LIO-SAM, which is a state-of-the-art SLAM algorithm. Furthermore, as will be discussed in our results section, we have compared LIO-SAM with another state-of-the-art SLAM algorithm, Fast LiDAR Inertial Odometry with Scan Context (FASTLIO-SC) [33, 64], on our datasets with ground truth and have determined that LIO-SAM has consistently had lower RMSE errors [7].

LIO-SAM performs scan matching and state smoothing and mapping using a factor graph approach where relative and absolute measurements

are used as factors [55]. The IMU measurements provide an initial guess of the robot's pose and are used to deskew point clouds. The initial guess is then integrated over the LiDAR scan rate and further optimized using LiDAR measurements. Hence, since the IMU measurements are used in the optimization process, this algorithm is tightly coupled. The resulting LiDAR odometry is subsequently used to estimate the bias of the IMU which is jointly optimized with the LiDAR odometry factors in the factor graph. GPS measurements are optionally included to further refine the estimated pose.

FASTLIO-SC is also a tightly coupled SLAM algorithm that derives a kinematic model, similar to LIO-SAM, with the IMU measurements, and then uses this estimation in a backpropagation which estimates the LiDAR pose of each point in the LiDAR scan with respect to the pose at the scan at the end time based on IMU measurements The iterated Kalman Filter approach is then used to propagate on each IMU measurement and perform an iterated update on each LiDAR scan to derive the LiDAR odometry. The estimated odometry can then be improved by GPS measurements and the movement restrictions imposed by a scan context approach [33]. FASTLIO-SC, unlike LIO-SAM, does not use the LiDAR odometry as feedback for estimating the IMU bias.

Due to LIO-SAM having better performance, as will be discussed in the results section of this thesis, we have primarily used LIO-SAM for deriving the poses of the UAV and semantic map generation in the forest environment. This chapter will primarily focus on discussing the fundamental theory of LIO-SAM.

LIO-SAM estimates the state of the robot based on sensor data from a 3D LiDAR, an IMU, and GPS (optional). The state estimation is formulated as a maximum a posteriori (MAP) problem, with the assumption of a Gaussian noise model. The state of the robot is optimized in a factor graph scheme, where there are four factors: (1) IMU preintegration factors; (2) LiDAR odometry factors; (3) GPS Factors; and the (4) loop closure factors. The

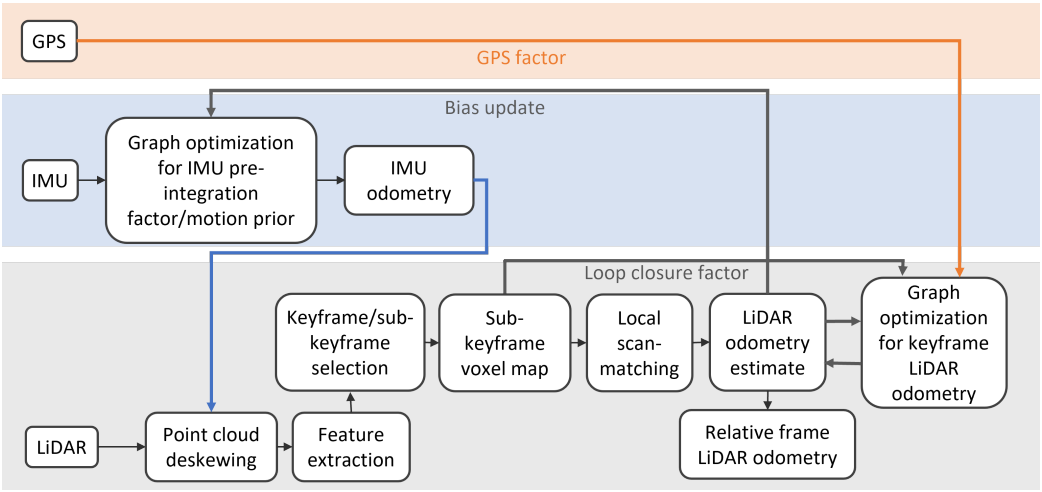


Figure 4.1: LIO-SAM system overview

state node x is added to the factor graph when there is a significant change in the robot's pose such that the change exceeds a user-specified threshold. Figure 4.1 shows an overview of the factors involved in LIO-SAM and details LIO-SAM's system.

The robot state is defined by a 6 DoF, consisting of the position and the rotation described as a quaternion. The world frame is given as W and the body frame of the robot is B. We assume that the IMU frame is the same as the body frame B of the robot. Therefore, the robot state is expressed as,

$$x = [\mathbf{R}^T, \mathbf{t}^T, \mathbf{v}^T, \mathbf{b}^T]^T. \tag{4.1}$$

In this expression, the rotation matrix is in the special orthogonal group, $\mathbf{R} \in SO(3)$, the position vector is $\mathbf{t}$, the velocity vector is $\mathbf{v}$, and the IMU bias is $\mathbf{b}$. The vectors, $\mathbf{t}$, $\mathbf{v}$, and $\mathbf{b}$ are $\in R^3$. Note that the odometry values outputted in SLAM contain $\mathbf{R}$ and $\mathbf{t}$, which can form the transformation matrix T. We also have a transformation T_{WB} which is used to transform the robot state from frame B to frame W. This transformation T_{WB} is in the special Euclidean group, $T_{WB} \in SE(3)$, and is expressed as,

$$T_{WB} = \begin{bmatrix} R_{WB} & t_{WB} \\ 0 & 1 \end{bmatrix} \tag{4.2}$$

where R_{WB} is the rotation matrix that transforms the robot state from frame B to frame W, and t_{WB} is the translation vector that translates the robot state from frame B to frame W. Note that this notation in the transformation matrix is used in the pose correction section of this thesis.

4.2 Factor Graphs

The factor graph is modelled as maximum a posteriori optimization (MAP) problem. The MAP estimators requires the densities of the measurement models to be defined as well as their log-likelihood.

When optimizing using the MAP problem formulation, we have the following where we are trying to find the $\hat{x}^*$ which maximizes the probability of $\hat{x}$ given the measurement z.

$$\begin{aligned} \hat{x^*} &= \underset{\hat{x}}{\operatorname{argmax}}\, p(\hat{x}|z) \\ &= \underset{\hat{x}}{\operatorname{argmax}}\, \frac{1}{p(z)} p(z|\hat{x})p(\hat{x}) \\ &= \underset{\hat{x}}{\operatorname{argmax}}\, p(z|\hat{x})p(\hat{x}) \\ &= \underset{\hat{x}}{\operatorname{argmax}}\, p(\hat{x})\Pi_i p(z_i|\hat{x}) \\ &= \underset{\hat{x}}{\operatorname{argmin}}(-\log(p(\hat{x})\Pi_i p(z_i|\hat{x}))) \\ &= \underset{\hat{x}}{\operatorname{argmin}}(-\log(p(\hat{x})) - \sum_i \log(p(z_i|\hat{x}))) \end{aligned} \tag{4.3}$$

We assume that the measurements are normally distributed, hence, we have,

$$p(z_i|\hat{x}) \sim N(\mu = z_i, \sigma) = \frac{1}{\sqrt{(2\pi)^n|\sum|}} e^{-\frac{1}{2}(h_i(\hat{x})-z_i)^T \sum^{-1}(h_i(\hat{x})-z_i))} \quad (4.4)$$

When we account for this, we have,

$$\begin{aligned} \hat{x^*} &= \underset{\hat{x}}{\operatorname{argmin}}(-\log(p(\hat{x})) - \sum_i \log(p(z_i|\hat{x}))) \\ &= \underset{\hat{x}}{\operatorname{argmin}}(-\log(p(\hat{x})) - \sum_i (h_i(\hat{x}) - z_i)^T \Sigma^{-1}(h_i(\hat{x}) - z_i)) + c \quad (4.5) \end{aligned}$$

where c is a constant. Note that the term $(h_i(\hat{x} - z_i)$ is the residual and the factor. $h_i(\hat{x})$ is the observation model, and Σ is the covariance matrix.

To leverage factor graphs to only update a typically small subset of variables impacted by a new measurement, incremental smoothing and mapping with the Bayes tree (iSAM2) is used. Incremental Smoothing and Mapping decreases the computational complexity of solving the MAP problem as a sparse nonlinear optimization and allows fast incremental updates of the square root information matrix. Using the Bayes tree instead of a Bay net structure allows the MAP problem to be relinearized with a reduced set of variables and maintain sparsity and accuracy.

4.2.1 IMU Preintegration Factor

The motion of the robot is first estimated using the measurements from the IMU, which typically includes a three-axis accelerometer and a three-axis gyroscope. This would give us measurements such as the acceleration and rotation rate of the sensor with respect to an inertial frame. Thus, we have the following measurements defined, where ${}_B\hat{a}(t)$ is the measurement of the acceleration, and ${}_B\hat{\omega}(t)$ is the measurement of the rotation rate of the IMU in the body frame B.

$$
\begin{aligned}
{}_B\hat{a}(t) &= R_{BW}(t)({}_Wa(t) - {}_W\, g) + b^a(t) + \eta^a(t) && (4.6)\\
{}_B\hat{\omega}(t) &= {}_B\omega(t) + b^\omega(t) + \eta^\omega(t) && (4.7)
\end{aligned}
$$

.

These two measurements, ${}_B\hat{a}(t)$ and ${}_B\hat{\omega}(t)$, are affected by additive white noise η and a slowly varying sensor bias b. $R_{BW}(t)$ is the rotation matrix from frame B to frame W. In our context, we define frame B to be the frame of the body and frame W to be the frame of the world, since we are neglecting the effects due to earth's rotation. The gravity vector in the frame W is defined by ${}_Wg$.

From this point, we introduce the following kinematics equations, so that we can further derive the motion of the robot using the IMU measurements,

$$
\begin{aligned}
\dot{R}_{WB} &= R_{WB}(t)[{}_B\omega(t)]_\times && (4.8)\\
{}_W\dot{v}(t) &= {}_W\, a(t) && (4.9)\\
{}_W\dot{p}(t) &= {}_W\, v(t) && (4.10)
\end{aligned}
$$

If we assume that ${}_Wa$ and ${}_B\omega_{WB}$ are constant in the time interval $[t, t+\Delta t]$, we have the following motion models derived from applying Euler integration on the IMU measurements,

$$
\begin{aligned}
{}_Wv(t+\Delta t) &= {}_Wv(t) + {}_W\, g\Delta t + R_{WB}(t)({}_B\hat{a}(t) - b^a(t) \\
&\quad -\eta^{a*}(t))\Delta t && (4.11)\\
{}_Wp(t+\Delta t) &= {}_Wp(t) + {}_W\, v(t)\Delta t && (4.12)\\
&\quad +\frac{1}{2}{}_W g\Delta t^2 + \frac{1}{2}R_{WB}(t)({}_B\hat{a}(t) - b^a(t) - \eta^{a*}(t))\Delta t^2 \\
R(t+\Delta t)_{WB} &= R_{WB}(t)\exp(({}_B\hat{\omega}_{WB}(t) - b^\omega(t) - n^{\omega*}(t))\Delta t) && (4.13)
\end{aligned}
$$

Note that the $\eta(t)^{a*}, \eta(t)^{\omega*}$ are the noise terms for discrete-time. Hence, the covariance of the noise is a function of the sampling rate and relates to the continuous time noise by $Cov(\eta^{a*}(t)) = \frac{1}{\Delta t} Cov(\eta^{a}(t))$ and $Cov(\eta^{\omega*}(t)) = \frac{1}{\Delta t} Cov(\eta^{\omega}(t))$. The full derivation of the previous equations can be found in [20]. We note that the left subscript to denote the world frame may not be consistent in the semantic mapping pose correction section of this thesis. For convenience and ease of representation, especially with additional subscripts and superscripts, we have opted to use the left subscript to denote the world frame in this section.

To obtain a constraint in the factor graph using this motion model, we show that the motion constraint between two consecutive measurements with distinct change, which we will term keyframes k, at times $k = i$ and $k = i + 1$, can be expressed as a single measurement which is the preintegrated IMU measurement. Assuming that the IMU is synchronized with the camera and that it provides measurements at discrete times k, we can derive the following, as shown in [20],

$$\begin{aligned}
\Delta v_{i,i+1} &= R_i^T (v_{i+1} - v_i - g\Delta t_{i,i+1}) \\
&= \sum_{k=i}^{(i+1)-1} \Delta R_{ik} (\hat{a_k} - b_k^a - \eta_k^{a*}) \Delta t \qquad (4.14) \\
\Delta p_{i,i+1} &= R_i^T (p_{i+1} - p_i - v_i \Delta t_{i,i+1} - \frac{1}{2} g \Delta t_{i,i+1}^2) \\
&= \sum_{k=i}^{(i+1)-1} [\Delta v_{i,i+1} \Delta t + \frac{1}{2} \Delta R_{ik} (\hat{a}_k - b_k^a - \eta_k^{a*}) \Delta t^2] \qquad (4.15) \\
\Delta R_{i,i+1} &= R_i^T R_{i+1} \\
&= \Pi_{k=i}^{(i+1)-1} exp((\hat{\omega}_k - b_k^{\omega} - \eta_k^{\omega*}) \Delta t) \qquad (4.16)
\end{aligned}$$

We assume that the bias remains constant between two keyframes. Thus we have, for a set of m keyframes,

$$b_i^a = b_{i+1}^a = ... = b_{i+m}^a, b_i^\omega = b_{i+1}^\omega = ... = b_{i+m}^\omega$$

A more thorough discussion on accounting for the bias by assuming that b_i is known, and avoiding the repeating integration when the bias estimate changes can be found in [20].

The residuals correlated to the IMU preintegration factors are given by,

$$r_{\Delta R_{i,i+1}} = \log\left(\Delta R_{i,i+1}(b_k^\omega) exp(\frac{\partial \Delta R_{i,i+1}}{\partial b_k^\omega}\delta b_k^\omega))^T R_i^T R_{i+1}\right) \quad (4.17)$$

$$r_{\Delta v_{i,i+1}} = R_i^T(v_{i+1} - v_i - g\Delta t_{i,i+1}) - (\Delta v_{i,i+1}(b_k^\omega, b_k^a) + \frac{\partial \Delta v_{i,i+1}}{\partial b_k^a}\delta b_k^a + \frac{\partial \Delta v_{i,i+1}}{\partial b_i^\omega}\delta b_i^\omega) \quad (4.18)$$

$$r_{\Delta p_{i,i+1}} = R_k^T(p_{i+1} - p_i - v_i\Delta t_{i,i+1} - \frac{1}{2}g\Delta t_{i,i+1}^2) - (\Delta p_{i,i+1}(b_k^\omega, b_k^a) + \frac{\partial \Delta p_{i,i+1}}{\partial b_k^a}\delta b_k^a + \frac{\partial \Delta p_{i,i+1}}{\partial b_k^\omega}\delta b_k^\omega) \quad (4.19)$$

$$||r_{b_{i,i+1}}|| = ||b_{i+1}^\omega - b_i^\omega||_{\sum^{b\omega *}}^2 + ||b_{i+1}^a - b_i^a||_{\Sigma^{ba*}}^2 \quad (4.20)$$

4.2.2 LiDAR Odometry

To compute the LiDAR odometry, during each sweep of the laser scan, the combined point cloud during sweep k forms P_k. Following this, P_k is processed by first computing the motion between two consecutive sweeps, and then the estimated motion is used to correct the distortion in P_k. Afterward, the P_k with the distortions corrected are further processed by matching and registering the local undistorted cloud onto a global cloud. The final odometry is given by the LiDAR pose from the undistorted LiDAR scan with respect to the global map.

Since we would like to compute the motion between two consecutive sweeps, we first perform feature extraction on the point cloud, P_k, by extracting a set of feature points that are either edge or planar features. We denote $X_{(k,i)}$

to be a 3D point, $i \in P_k$. We determine the smoothness of the points over a local region within the range of S number of adjacent points to $X_{(k,i)}$ by calculating the smoothness value c with the following equation,

$$c = \frac{1}{|S| \cdot ||X_{k,i}||} ||\Sigma_{j \in S, j \neq i}(X_{k,i} - X_{k,j})|| \tag{4.21}$$

Half of the points of S are on either side of $X_{(k,i)}$. We calculate the smoothness value for all the points $X_{(k,i)}$ in the point cloud P_k. Note that $X_{(k,i)}$ contains the 3D position of the point in the point cloud. From this point, each of these sub-scans is assigned the value c that denotes the smoothness value. A threshold c_{th} is used to distinguish between edge features and planar features. Thus, if c is larger than c_{th}, then it is classified as an edge feature; otherwise, it will be classified as a planar feature. The threshold c_{th} was chosen so that the feature points within the scan are evenly distributed and the scans are separated into subregions where each subregion can have a maximum number of edge points and planar points. Points that are unreliable and not classified as feature points are: (1) points that lie on a surface that is roughly parallel to the LiDAR's laser beam; (2) points that lie on the edge/boundary of an occluded region.

Following feature extraction, we need to find correspondences between two LiDAR clouds. Given a point cloud P_k, we combine all the features extracted from the point cloud P_k at time k into a frame in the frame L, ${}_L F_k = \{{}_L F_k^e, {}_L F_k^p\}$, with ${}_L F_k^e$ denoting edge features and ${}_L F_k^p$ denoting planar features from the scans at time k.

To calculate the odometry, we need a relationship relating the previous frame to the current frame. If we use every LiDAR frame to compute factors to the factor graph for optimization, this would be computationally expensive. We use keyframes, which are frames where the change in the robot's pose from the previous state exceeds a defined threshold. Let F_{i+1} be the keyframe associated with the robot state x_{i+1}, and the preceding keyframe and robot state are F_i and x_i, respectively. The frames in between in which the robot

does not have a significant change in position and rotation are discarded. We transform the keyframes $\{{}_L F_i^e, {}_L F_i^p\}$ which is in the frame of the LiDAR to world frame with the extrinsic relating the LiDAR to the IMU, and the calculated robot motion T_i from the IMU, which is Equation 4.12.

A sliding-window approach and sub-keyframes are also used for computational efficiency. In this manner, a fixed number n of recent LiDAR scans, called sub-keyframes or relative frames (which we note in the semantic mapping and pose correction sections of this thesis) are extracted. The set of relative frames, $\{F_j, ..., F_{j+n}\}$ are registered into the world frame W with the transformations $\{T_j, ..., T_{j+n}\}$ into a voxel map, M_j. The transformation $\{T_j, ..., T_{j+n}\}$ between the two relative frames is found by finding the corresponding features for each point, using scan-matching methods such as the iterative closest point (ICP). The key concept of ICP is to compute the transformation that minimizes the distance between corresponding points in the two frames F_j and F_{j+1}.

Therefore, M_j contains sub voxel maps with edge features M_j^e and planar features M_j^p. We have,

$$M_j = \{M_j^e, M_j^p\}, \tag{4.22}$$

$$M_j^e = {}_W F_j^e \cup_W F_{j-1}^e \cup ... \cup_W F_{j-n}^e \tag{4.23}$$

$$M_j^p = {}_W F_j^p \cup_W F_{j-1}^p \cup ... \cup_W F_{j-n}^p \tag{4.24}$$

Let $\mathcal{E}_j$ and $\mathcal{E}_{j+1}$ denote the set of edge points and its corresponding edge points at time j and $j+1$, respectively. Similarly, we have $\mathcal{H}_j$ and $\mathcal{H}_{j+1}$ denoting the set of planar points and their correspondences. We can derive a geometric correlation between the set of edge points $\mathcal{E}_{j+1}$ and $\mathcal{E}_j$. Given that the corresponding edge points at the indices p at time $j+1$ are indices q and r at time j, compute the distance of the edge points as the following,

$$d_{E_j} = \frac{|(X^e_{j+1,p} - X^e_{j,q}) \times (X^e_{j+1,p} - X^e_{j,r})|}{|X^e_{j,q} - X^e_{j,r}|} \tag{4.25}$$

In this equation $X^e_{j+1,p}$ denotes the 3D point p projected at time $j+1$, and $X^e_{j,q}$ and $X^e_{j,r}$ are the corresponding edge points, q and r, of p in time j.

We can also derive a geometric correlation between the set of planar points, $\mathcal{H}_{j+1}$ and $\mathcal{H}_j$. When the corresponding planar points of the points in indices p at time $j+1$ are points of the indices q, r, and s at time j.

$$d_{H_j} = \frac{\begin{vmatrix} X^e_{j+1,p} - X^e_{j,q} \\ (X_{j,q} - X_{j,r}) \times (X_{j,q} - X_{j,s}) \end{vmatrix}}{|(X_{j,q} - X_{j,r}) \times (X_{j,q} - X_{j,s})|} \tag{4.26}$$

The scan-to-motion estimation is found with this geometric correlation by minimizing the overall distances of these feature points. Therefore, we use the following Gauss-Newton method to solve for the transformation that minimizes the following expression,

$$\min_{T_{j+1}} \{ \sum_{X^e_{j+1,p} \in W F^e_{j+1}} d_{E_j} + \sum_{X^p_{j+1,p} \in W F^p_{j+1}} d_{H_j} \} \tag{4.27}$$

Hence, we compute the relative transformation, which is the LiDAR odometry factor, between the two states, x_j and x_{j+1} as the following,

$$\Delta T_{j,j+1} = T_j^T T_{j+1} \tag{4.28}$$

4.2.3 GPS Factor

As an added measure in case the state estimation and mapping system suffers drift during longer-duration tasks, GPS measurements are used to derive a factor. Although GPS is typically weaker in forest environments with dense canopies, we were able to have reliable GPS measurements in the environments we tested. Furthermore, typically, the drift of the odometry

from the LiDAR and IMU grows very slowly and a GPS factor is added only when the estimated position covariance is greater than the GPS position covariance.

To utilize the GPS measurements in a factor graph scheme, the transformation between ground truth (GPS measurements) and the state estimated from the IMU and LiDAR is introduced as a new node in the factor graph. If there is no direct synchronization between the GPS signal and the LiDAR frames, the GPS measurements are interpolated linearly based on the timestamp of the LiDAR frames.

4.2.4 Loop Closure Factor

The loop closure factor is especially useful to correct drift in the robot's altitude and when the GPS is the only ground truth available. The loop closure factor is calculated with a Euclidean distance-based approach in which prior states that are close to the new state x_{i+1} in Euclidean space are searched. Suppose the prior state, x_2 is the state that is the closest to x_{i+1} in Euclidean space, then the corresponding keyframe ${}_W F_{i+1}$ is scan-matched to the sub-keyframes $F_{2-u}, ..., F_2, ..., F_{2+u}$, where u is a user-defined search index and is correlated to the search distance. The relative transformation $\Delta T_{2,i+1}$ is obtained and added as a loop closure factor in the factor graph.

Chapter 5

Semantic Mapping

5.1 System Overview

Russell et al. [54] have developed a semantic mapping system that is able to generate a detailed global mapping of the environment that contains geometric, semantic, and temporal information to capture and localize regions of flammable materials in the forest environment. The semantic mapping system is designed to be modular so that it reads odometry or pose information from a SLAM system to generate the global map. Octomap is used in this system to discretize the semantic point cloud into voxels [27, 65]. This type of representation was chosen over other 3D representations since it is lightweight and computationally efficient to handle multiple dimensions of information. When a new local semantic point cloud is generated, it is registered into the world frame and used to update the global Octomap.

We build upon our semantic mapping system by improving the integration of our semantic mapping system with the SLAM system. Since our semantic mapping system reads pose information from the front-end of the SLAM system instead of backend poses and does not have the ability to optimize and correct for drift which may accumulate over time, this may lead to unwanted

artifacts propagating over to the semantic global map. Thus, our contribution in this area is a proof-of-concept offline approach that stores the history of poses and relays the corrected poses to the semantic mapping system. We have implemented the works of [29] in our contribution.

5.2 Semantic Segmentation

The primary objective of the semantic segmentation pipeline is to identify flammable regions, or fuel clusters, and localize these regions within the environment so that they can be ultimately removed by a UGV. To achieve this, we used semantic segmentation networks SegFormer [63] and SegNeXt [24].

The SegFormer network is based on a transformer architecture. This was chosen due to its robust performance on benchmark datasets and good generalization capabilities, especially since we have a relatively limited number of trained data from real-world images. Our model trained with SegFormer uses the default parameters in the MMSegmentation implementation [43]. We trained this model using 151 real images in the Porto region by manually labeling classes with polygons.

For a more recent set of data collected, we also trained with SegNeXt with three compressed classes (i.e., fuel, canopies, and background) since SegNeXt has shown to outperform SegFormer [24]. SegNeXt is a convolutional neural network architecture-based approach specifically for semantic segmentation and identifies the predicted class for each pixel in the image. SegNeXt is similarly able to generalize to a limited number of trained data from real-world images. Our model trained with SegNext was trained with 40 images, from the Pittsburgh and Coimbra regions, with pre-trained weights from the CityScapes datasets [15].

5.3 Integrating Semantics to Simultaneous Localization and Mapping

In this system, we used a SLAM system to generate pose estimates of the UAV that were then used to globally register the 3D mapping generated in the semantic mapping system.

The integration of the SLAM system with the semantic mapping system developed in [54] is shown in Figures 5.1. RGB images from the stereo cameras are used to produce the semantically segmented images. These images are then projected to the point cloud generated from LiDAR measurements through LiDAR-camera registration. This allows each point in the point cloud within the camera's field of view to store information visual information captured by the stereo cameras. To accomplish LiDAR-camera registration, two separate calibrations were performed between the LiDAR and IMU and the camera and IMU. We calibrated the sensors in this manner since the IMU is positioned to be as close to the center of mass of the payload and is assumed to be the origin of the UAV in our system, which is the body frame. Furthermore, having the IMU be the intermediary step allows the calibration process to be modular. In this way, the LiDAR-IMU and the camera-IMU calibration results can be used for separate systems. For example, LIO-SAM requires the LiDAR and IMU transform, while different SLAM systems that involve the camera and IMU will require the camera-IMU transform.

The camera-IMU calibration was accomplished using Kalibr [21]. In this manner, the camera/IMU calibration determines the relative rotation, translation, and time offset between the sensors. Further details on how this is accomplished can be found in [21]. The LiDAR-IMU calibration was accomplished by referring to the SolidWorks CAD file and determining the extrinsic values relating to the two components. With the results from the two, we can derive the transformation relating the camera to LiDAR.

Once the semantic point cloud is registered using the poses from the

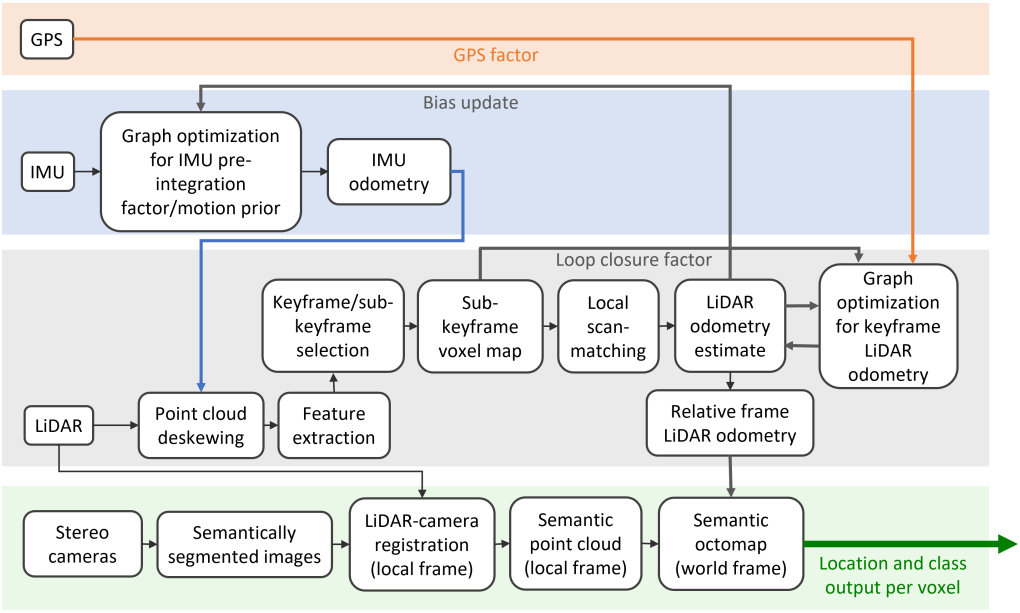


Figure 5.1: LIO-SAM integration with semantic mapping

SLAM system, which is in the world-frame, a 3D representation with semantic information is constructed using Octomap. Octomap is based on the octree data structure, where 3D space is recursively divided into eight partitions and each node in the octree data structure consists of eight children. Octomap can explicitly represent free and unknown areas and uses a probabilistic occupancy estimation. In our semantic mapping implementation, Octomap is used to discretize the semantic point cloud into voxels, where each voxel has a resolution of 0.05 m and contains the aggregated predicted classification information. When a new semantic point cloud is created as a result of the robot covering new areas of the environment, the global octomap is updated by registering the voxelized semantic point cloud using the pose information from SLAM. In this manner, we are using a Bayesian method where a probability distribution over all the classes considered is used. When a new observation is received, it is multiplied by the current distribution, renormalized and the most probably class is used to label the voxel.

5.4 Problem Formulation and Proposal

LIO-SAM is a keyframe-based SLAM system that has a frontend and a backend. As noted in the previous chapter, keyframes are defined as frames where the pose information exceeds a user-defined threshold when compared to the previous pose. Many SLAM algorithms adopt keyframes to preserve performance while reducing computation time. Poses given in the frontend are denoted as relative frame poses, which are not optimized by the factor graph scheme. These relative frame poses fall in between the keyframe poses. In this sense, keyframe poses are sparse in comparison to relative frame poses if the user-defined threshold is large. Keyframe poses are optimized in the backend in a factor graph scheme, which combines the IMU odometry factor, LiDAR odometry factor, loop closure factor, and, optionally, the GPS factor.

The semantic mapping system in [54] currently reads poses from the frontend of the SLAM system, which are called relative frame poses to register the semantically labeled point cloud and generate a global mapping of the environment. Registration using relative frame poses allows the global map to be dense since the relative frame poses are dense and high-frequency in comparison to the sparse and low-frequency optimized keyframe poses.

Since sensor measurements are prone to errors, over time, these errors can propagate over time. Furthermore, since our SLAM system, LIO-SAM, adopts a keyframe approach for computational efficiency, corrections do not propagate to the relative frames between keyframes. This can contribute to global map drifts or unwanted artifacts in the semantic mapping system since the non-optimized relative frame poses are used for registration. As it currently stands, our semantic mapping system does not have the ability to correct the relative frame poses used for registration.

We implement an offline approach in [29], that involves storing the keyframe and relative frame poses, using the keyframe poses are constraints to correct the relative frame poses, and registering the semantically labeled

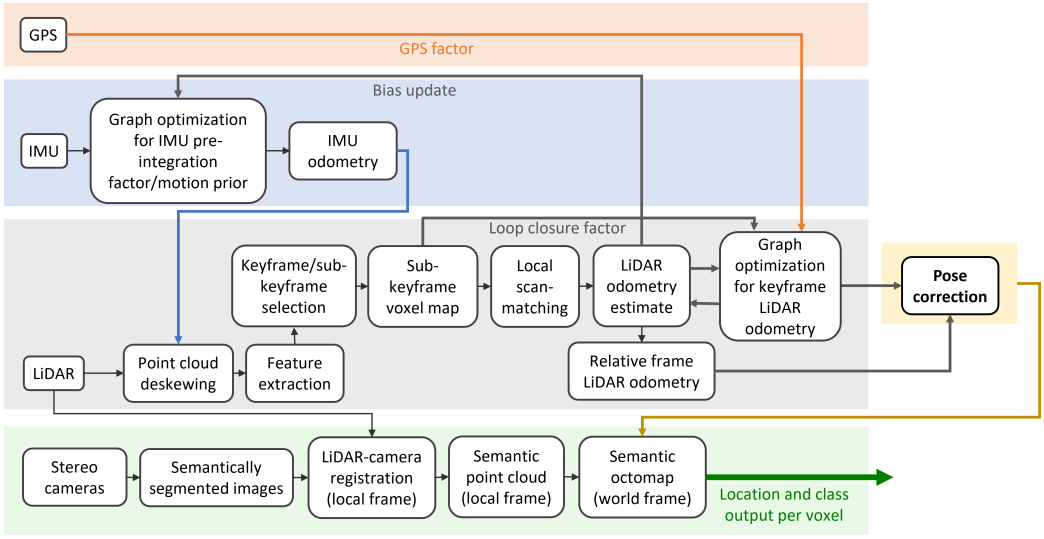


Figure 5.2: Framework of the pose correction module added

point cloud using the corrected relative frame poses. While our system is offline, it serves as a proof-of-concept to show improvement over the current implementation.

5.5 Semantic Mapping Pose Correction

Figure 5.3 shows an overview of the pose correction approach. Frames, in world-frame, are denoted f^W. Two consecutive keyframes with respect to the world-frame are denoted f_i^W and $f_{(i+1)}^W$, and the relative frame is denoted as f_j^W. The optimized consecutive keyframes and the corrected relative frame are denoted, f_i^{*W} and ${f_{(i+1)^*}}^W$, and $f_{j^*}^W$, respectively.

We denote the robot state or pose, defined by a 6 DoF vector, of the keyframe f_i^W as x_i^{KF}, and the poses correlated to the j^{th} relative frame as x_j^{RF}. We are assuming these variables are all in the world frame, so the superscript is dropped. The time associated with the poses x_i^{KF} and x_j^{RF} are denoted as t_i^{KF} and t_j^{RF}. In our approach, we are saving all the robot states from all keyframes and relative frames. Since we are doing this offline, the

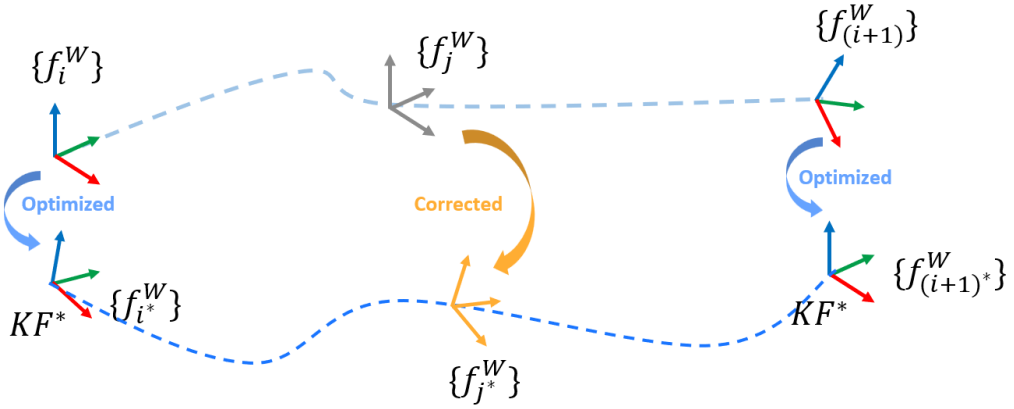


Figure 5.3: Overview of pose correction from [29]. ${f_i}^W$ and ${f_{(i+1)}}^W$ denotes two consecutive keyframes in world frame and ${f_i^*}^W$ and ${f_{(i+1)^*}}^W$ denote the corresponding optimized keyframes. f_j^W denotes the relative frame in world frame and $f_{j^*}^W$ is the resulting corrected relative frame.

poses associated with all of our keyframes are optimized and corrected after processing our data through the SLAM system. The poses associated with the relative frames are used for the semantic mapping system and are not optimized. The set of robot states from the keyframes and relative frames are denoted as $X^{KF} = \{x_i^{KF}, ..., x_{i+m}^{KF}\}$ and $X^{RF} = \{x_j^{RF}, ..., x_{j+n}^{RF}\}$, respectively. Since the keyframes are sparse, $m < n$. The set of times correlated to this are denoted $t^{KF} = \{t_i^{KF}, ..., t_{i+m}^{KF}\}$ for keyframes and $t^{RF} = \{t_j^{RF}, ..., t_{j+n}^{RF}\}$ for relative frames. We also note that $T^{KF} \subset T^{RF}$.

The semantic mapping system uses the poses from the relative frames to register the semantic point cloud. Hence, we implement an offline pose integration approach in [29], where the forward and backward keyframe poses are used as constraints to correct the poses in the relative frame. This implementation allows the optimized keyframe poses to be used for interpolation rather than solely using the uncorrected relative frames poses.

In a keyframe SLAM algorithm, the frontend provides the uncorrected poses, which consist of the original keyframe and relative poses in the world

frame. For two consecutive keyframes, i and $i+1$, and the relative frames $j, j+1, ..., j+n$, we can obtain $\{R_i, t_i\}$, $\{R_{(i+1)}, t_{(i+1)}\}$, $\{R_j, t_j\}$ and so forth from the frontend. The backend contains the graph optimization which corrects keyframe poses and relays some of these updated corrections over to improve the LiDAR odometry estimates. Note that all of the keyframe poses in the backend are continuously getting updated in the graph optimization, while the relative frame LiDAR poses are not. For two consecutive corrected keyframes, i^* and $(i+1)^*$, the backend provides the set of the rotation matrix and translation matrices $\{R_{i^*}, t_{i^*}\}$ and $\{R_{(i+1)^*}, t_{(i+1)^*}\}$. Without further elaboration, the asterisk (*) superscripts denoted corrected, and the keyframes i and $i+1$ are consecutive.

If we combine the information in the frontend, we can derive a transformation matrix of the form,

$$T = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \in SE(3).$$

Thus, we have the following transformations we can use from the frontend to relate the keyframes i and $(i+1)$ to the relative frame j.

$$T_{i,j} = T_i^{-1} T_j \tag{5.1}$$

$$T_{(i+1),j} = T_{(i+1)}^{-1} T_j \tag{5.2}$$

$$T_{i,(i+1)} = T_i^{-1} T_{(i+1)} \tag{5.3}$$

From the backend, we can derive the following relationships,

$$T_{i^*,(i+1)^*} = T_{i^*}^{-1} T_{(i+1)^*} \tag{5.4}$$

The proposed pose correction approach requires that the frame poses are

expressed in SE(3); however, since interpolating the rotation matrix has SO(3) constraints by definition ($\det(R) = +1$ and $R^T R = I$), it must be converted to a vector space in the form of quaternions, or Euler angles. LIO-SAM outputs rotation values with Euler angles, and for the convenience of our implementation, we have converted this to quaternions.

The pose correction approach derives a close-form solution constraint using the frontend and backend pose values, in which we will refer to [29] for more details and notes on derivations. These assumptions are based on the observation that the depth value of the visual feature increases as the translational difference between the corresponding keyframes to which the visual feature is observed increases, and that the homogeneous pixel coordinates of a visual feature in an image remain the same regardless of optimization updates from SLAM. The scaling factor, s_i which relates the corrected keyframe i translation and the corrected relative frame j to the original keyframe i and relative frame j translation, is found by assuming that the translation ratio and the depth ratio are equal. This scaling factor is given by,

$$s_i = \frac{||t_{i^*,(i+1)^*}||}{||t_{i,(i+1)}||} \tag{5.5}$$

In this equation, $t_{i^*,(i+1)^*}$ is the translation from the corrected keyframe i to the corrected keyframe $i+1$, which we can obtain from Equation 5.4. The term, $t_{i,(i+1)}$ can be calculated from Equation 5.3. Furthermore, we have the interpolation factor, $\alpha_{i,j}$, which is the ratio of the distance from the frame f_j to the frame f_i, and the distance from the frame f_j to the frame $f_{(i+1)}$. This was assigned based on the observation that the number of reliable edges increases as the distance between frames decreases since there is an increase in the number of shared features [29]. $\alpha_{i,j}$ is defined in the following.

$$\alpha_{i,j} = \frac{||t_{i,j}||}{||t_{(i+1),j}||} \tag{5.6}$$

In this expression, we have the translation from the original keyframe i to the original relative frame j, denoted $t_{i,j}$ derived from Equation 5.1, and the translation from the original keyframe $i+1$ to the original relative frame j, denoted $t_{(i+1),j}$ derived from Equation 5.2. Essentially, $\alpha_{i,j}$ is a ratio of the distance between the original keyframe i to the original relative frame j to the distance between the original keyframe $i+1$ to the original relative frame j.

Lastly, we also define the terms δR and δt, which account for the gap resulting from the measurement constraint assumptions. In this respect, δR and δt are defined below,

$$\begin{aligned} \delta R &= R^T_{i^*,j^*} R_{i^*,(i+1)^*} R_{(i+1)^*,j^*} \\ &= R^T_{i,j} R_{i^*,(i+1)^*} R_{(i+1),j} \qquad (5.7) \\ \delta t &= t_{i^*,j^*} + R_{j^*,i^*} t_{i^*(i+1)^*} + R_{j^*,(i+1)^*} t_{(i+1)^*,j^*} \\ &= R_{j^*,i^*}(t_{i^*,(i+1)^*} - s_i(t_{i,j} - R_{i^*,(i+1)^*} t_{(i+1,i)})) \qquad (5.8) \end{aligned}$$

Therefore, we can find the corrected $f^W_{j^*}$, through spherical and linear interpolation, given by,

$$R_{i^*,j^*} = \text{SLERP}(R_{i,j}, \delta R, \alpha_{i,j}) \tag{5.9}$$

$$t_{i^*,j^*} = \text{LERP}(t_{i^*,j^*}, R_{i^*,j^*}, \delta t, \alpha_{i,j}) \tag{5.10}$$

In this expression, SLERP is spherical linear interpolation, while LERP is linear interpolation. The interpolation parameter of SLERP is $\alpha_{i,j}$, which

must be constrained to a value between $(0, 1)$. Note that we would first need to compute R_{i^*,j^*} using known information from the frontend and backend processes, from Equations 5.9, 5.6, and 5.1 before computing δt. Furthermore, to register these values in world-frame, since for example, T_{i^*,j^*} is the transformation relating the corrected keyframe i to corrected relative frame j, we have,

$$T_{j^*} = T_{i^*}^{-1} T_{i^*,j^*} \tag{5.11}$$

Note that we have dropped the W superscript to denote the world-frame.

Chapter 6

Experiments

We have conducted a set of experiments to verify our contributions, namely, the performance of the sensing payload and the improved integration of the SLAM system and semantic mapping system. The objectives of our experiments are: (1) to verify that the custom-designed multi-sensing payload can withstand the conditions present during field testing and can record sensor measurements for post-processing; (2) to qualitatively demonstrate that the pose correction algorithm helps with artifacts resulting from the current integration of the semantic mapping system with the SLAM system.

6.1 Datasets

We tested and collected data using our multi-sensing payload mounted on the DJI M600 Pro and the Freefly Alta X. Data was collected in three primary locations. We had two locations in Portugal which were Porto (41°13'00.9"N 8°31'38.6"W), and Coimbra (40°11'05.9"N 8°24'50.5"W). The remaining location was in Pittsburgh, United States (40°27'24.6"N 79°47'22.1"W). Among the three locations tested, the Porto location contained the most dense and tall canopies. The Pittsburgh location was also rural but contained canopies

that were not as high. Lastly, the Coimbra test region was semi-urban and featured a more structured landscape which makes this a low-complexity case for our experiments. Both the Pittsburgh and Porto test locations have an unstructured and wild landscape, which makes it more challenging to perform extraction and subsequent scan matching of the employed SLAM approaches.

In these testing locations, we have conducted under-the-canopy surveys (in the Porto and Coimbra datasets) and over-the-canopy surveys (in the Pittsburgh dataset). Under-the-canopy surveys consist of a simple loop back to the start of the flight and were conducted by an experienced UAV pilot. Over-the-canopy surveys typically involved flying in a lawn-mower pattern and were done using UAV autopilot software. For each of the surveys in the Coimbra and Pittsburgh regions, the altitude, ground speed, and lateral distances were about 30 m, 3.0 m/s, and 20 m, respectively. For the surveys in Porto, the altitude, ground speed, and lateral distances were about 10 m, 1.0 m/s, and 50 m. The payload was oriented 30° for all flights.

We also have data collected using our past payload in [56] in the Porto (41°13'00.9"N 8°31'38.6"W) location. The UAV was flown under-the-canopy in this dataset at an approximate altitude, ground speed, and lateral distance of 10 m, 1.0 m/s, and 50 m. We will examine the pose correction algorithm on this dataset.

All the datasets were processed using a consumer-grade computer with an Intel Core i7-10870H CPU with 16M Cache, up to 5.00 GHz.

Figure 6.1: Testing locations: Porto, Portugal (left), Coimbra, Portugal (middle), and Pittsburgh, United States (right).

6.2 Benchmarking Results

We will rely on both qualitative and quantitative measures to evaluate our two objectives. For instance, to validate the performance of the multi-sensing payload, we require that the payload can be used multiple times for field testing and can reliably collect sensor measurements. Furthermore, we require that the sensor measurement from the sensing payload be processed using a state-of-the-art SLAM system and our semantic mapping system. We decided to test two state-of-the-art SLAM, LIO-SAM and FASTLIO-SC in datasets containing ground truth, such as the Coimbra and Pittsburgh datasets. We compare the pose estimations from LIO-SAM and FASTLIO-SC with the position measurements from a real-time kinematics (RTK) device that provides centimeter-level accuracy ground truth. The main metric we used to evaluate the system localization is the root mean square error (RMSE). The equation for computing the RMSE is shown below, where $X(i)$ is the i^{th} measurement, $\hat{X(i)}$ is the estimated i^{th} measurement, and N is the number of data points.

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} ||X(i) - X(\hat{i})||^2}{N}}$$

We will qualitatively show that the current integration of the semantic mapping system with LIO-SAM has drifting and can be corrected using our offline approach. This is done by comparing the point clouds processed without the offline approach with the offline approach with trajectory plots.

Chapter 7

Results

7.1 Sensing Payload Verification

We have successfully used our sensing payload to record data in the testing locations. We have data acquired using the DJI M600 Pro for the locations in Coimbra and Porto, and the Freefly Alta X for the location in Pittsburgh. Figures in 7.1 show the sensing payload attached to the DJI M600 Pro and the Freefly Alta X.

7.1.1 Implementing LiDAR-based SLAM in Forest Environment

We have implemented the Fast LiDAR Inertial Odometry with Scan Context (FASTLIO-SC) [33, 64], and LIO-SAM [55] both of which are state-of-the-art LiDAR-based SLAM systems. Tables 7.1 and 7.2 show an overview of the results of implementing this approach on the data collected in Coimbra and Pittsburgh, which contain ground truth measurements from the real-time kinematics (RTK) device. These results indicate that LIO-SAM performed better than FASTLIO-SC. A qualitative comparison of the two SLAM algorithms is

shown in Figure 7.3.

Figure 7.1: Sensing payload on UAVs: in Porto, Portugal (left), and in Pittsburgh, United States (right)

Table 7.1: Quantitative odometry results of LIO-SAM and FASTLIO-SC approaches on Coimbra dataset. Bold values denote the lowest RMSE [credits: Tito Arevalo-Ramirez] [7]

	Coimbra			
	Survey 1		**Survey 2**	
	LIO-SAM	**FASTLIO-SC**	**LIO-SAM**	**FASTLIO-SC**
X	0.26	1.79	0.67	42.51
Y	0.24	3.49	0.90	27.50
Z	0.33	4.13	0.63	7.88
Average	**0.28**	3.14	**0.73**	25.96

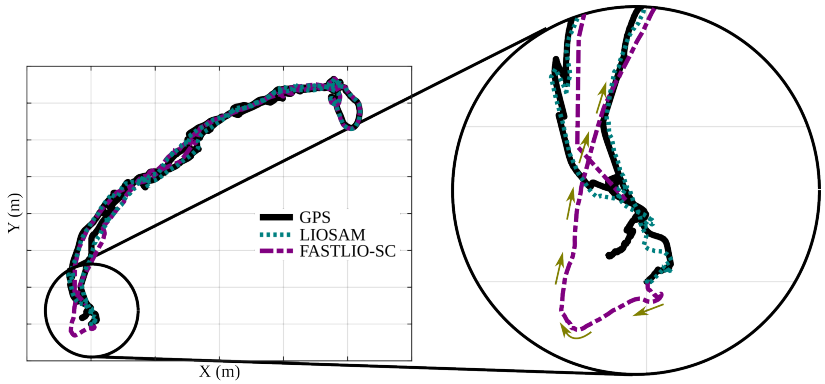
Since LIO-SAM has performed well, we have primarily implemented this SLAM system in our forest datasets. Figures 7.5 and 7.6 show the trajectory results of implementing the pose correction on relative frame odometry from LIO-SAM on under-the-canopy and over-the-canopy data, respectively. Note that the relative frame is denoted RF. Figure 7.4 shows the results from implementing LIO-SAM in the Porto location, which indicates that the point

Figure 7.2: DJI M600 Pro flying with the sensing payload

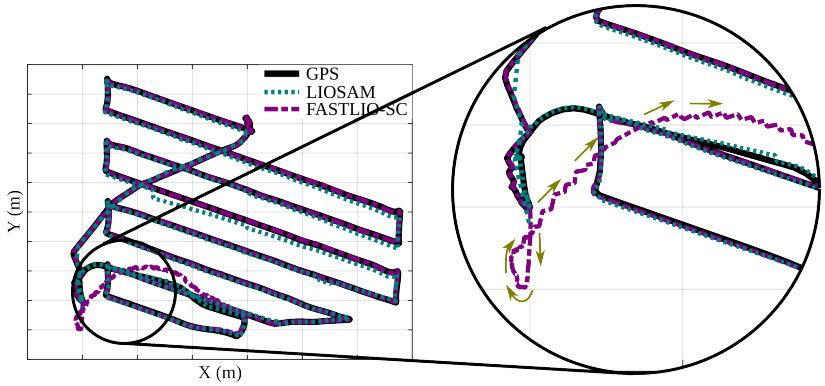
Table 7.2: Quantitative odometry results of LIO-SAM and FASTLIO-SC approaches on Pittsburgh dataset. Bold values denote the lowest RMSE [credits: Tito Arevalo-Ramirez] [7]

	Pittsburgh					
	Survey 1		**Survey 2**		**Survey 3**	
	LIO-SAM	**FASTLIO-SC**	**LIO-SAM**	**FASTLIO-SC**	**LIO-SAM**	**FASTLIO-SC**
X	1.33	2.67	0.73	0.75	0.72	1.80
Y	0.79	5.39	1.51	5.70	0.40	6.02
Z	1.60	4.64	3.53	3.00	1.91	5.88
Average	**1.24**	4.23	**1.92**	3.15	**1.01**	4.57

cloud generated can capture the shape of the canopies well. We have generated a global point cloud of the forest environment after implementing LIO-SAM, as shown in Figure 7.4. Figure 7.5 shows that the key pose trajectory (blue) aligns better with the GPS odometry (green). However, since this dataset was collected under the canopy, GPS may not be as reliable, as shown towards the bottom right which is not representative of the UAV's actual trajectory



(a)



(b)

Figure 7.3: Qualitative odometry results of LIO-SAM and FASTLIO-SC approaches. To avoid figure over-stacking, two representative cases of the study sites are shown. Figure 7.3a and 7.3b shows the odometry of survey 2 for the semi-urban area and survey 2 for the rural area, respectively. The golden arrows indicate the direction of travel of the post computed using FASTLIO-SC. [credits: Tito Arevalo-Ramirez] [7]

during flight. Figure 7.5 shows that the z values in the GPS odometry deviate significantly from the relative frame odometry and the keyframe odometry. Figure 7.6 contains the trajectory plots for over-the-canopy data. In this setting, the GPS signal is more reliable. These figures so that generally, the keyframe odometry align better with the GPS odometry.

These figures also show the corrected relative frame (RF) odometry com-

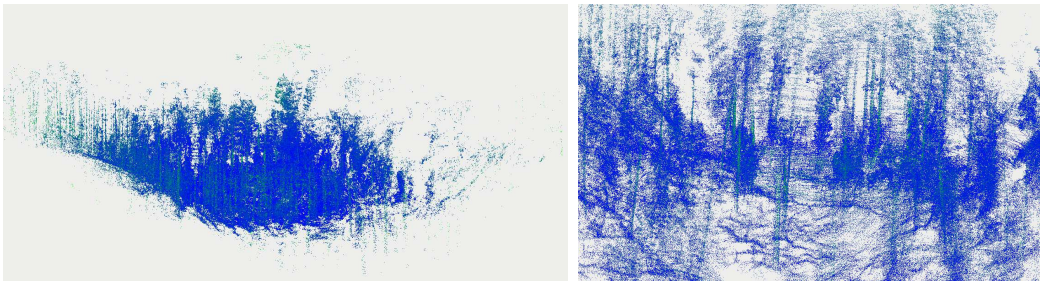


Figure 7.4: Global map of forest environment in Porto generated from under-the-canopy data

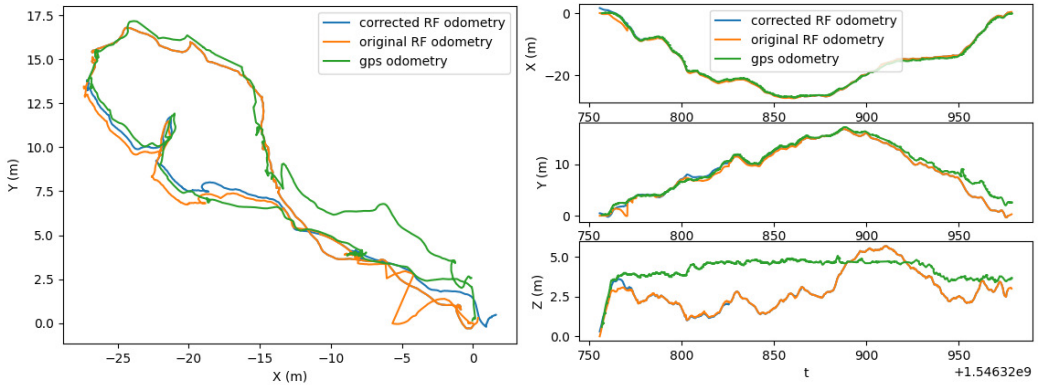


Figure 7.5: Trajectory results for under-the-canopy flight in Porto

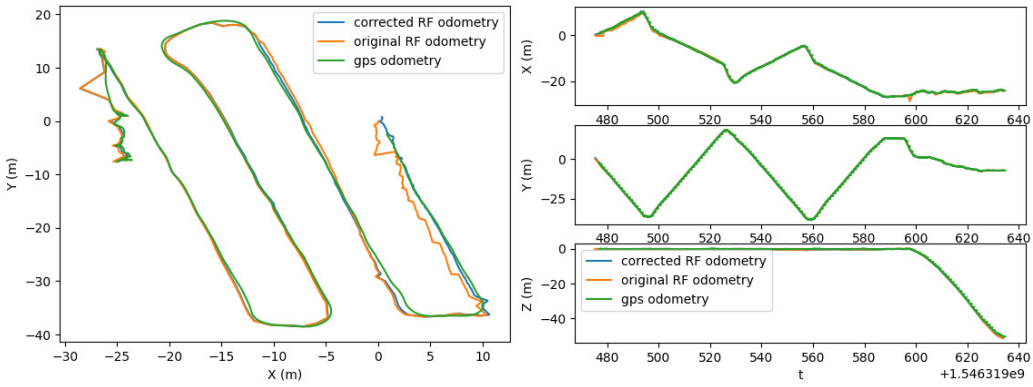


Figure 7.6: Trajectory results for over-the-canopy flight in Porto

pared with the original relative frame odometry. In Figure 7.5, it is apparent that the corrected RF odometry follows a smoother path, especially at the beginning of the trajectory where the corrected RF odometry is more representative of the actual trajectory. Furthermore, in Figure 7.6, it is evident that towards the beginning of the trajectory, the corrected KF odometry aligns better with the GPS odometry, which is reliable since this is over-the-canopy. We note that the semantic point clouds were not generated using this dataset due to timing, which is an area for future work.

7.2 Semantic Mapping Pose Correction

We implemented the semantic mapping pose correction on the data that we collected using a previous payload featured in the following work [56]. This payload also features stereo cameras, IMU and LiDAR. Figure 7.7 was generated by running LIO-SAM and the semantic mapping system on the data collected under the canopy in Porto from this payload. The corresponding trajectory plots of this dataset are shown in figure 7.8, which overlays the keyframe (KF) odometry with

Figure 7.13 shows an overlap of the original point cloud registered with the semantic point cloud that has had its pose corrected. Figure 7.8 shows the trajectory plot of the corresponding dataset with the pose corrected. The total distance traveled in this trajectory is approximately 50 m. The following figures 7.9, 7.10, 7.11, and 7.12 show close-up portions of the trajectory plot which shows that the corrected RF odometry generally aligns to the KF odometry more than the original RF odometry. In addition, we can see that in figure 7.10, the corrected RF odometry does not contain the artifact trajectory shown in the original RF odometry. The resulting semantic map generated using the original RF odometry for registration is compared with that from using the corrected RF odometry. Although it may be difficult to readily see obvious artifacts, we suspect that the corrected RF odometry will be helpful

for registration when the original RF odometry displays non-representative trajectories.

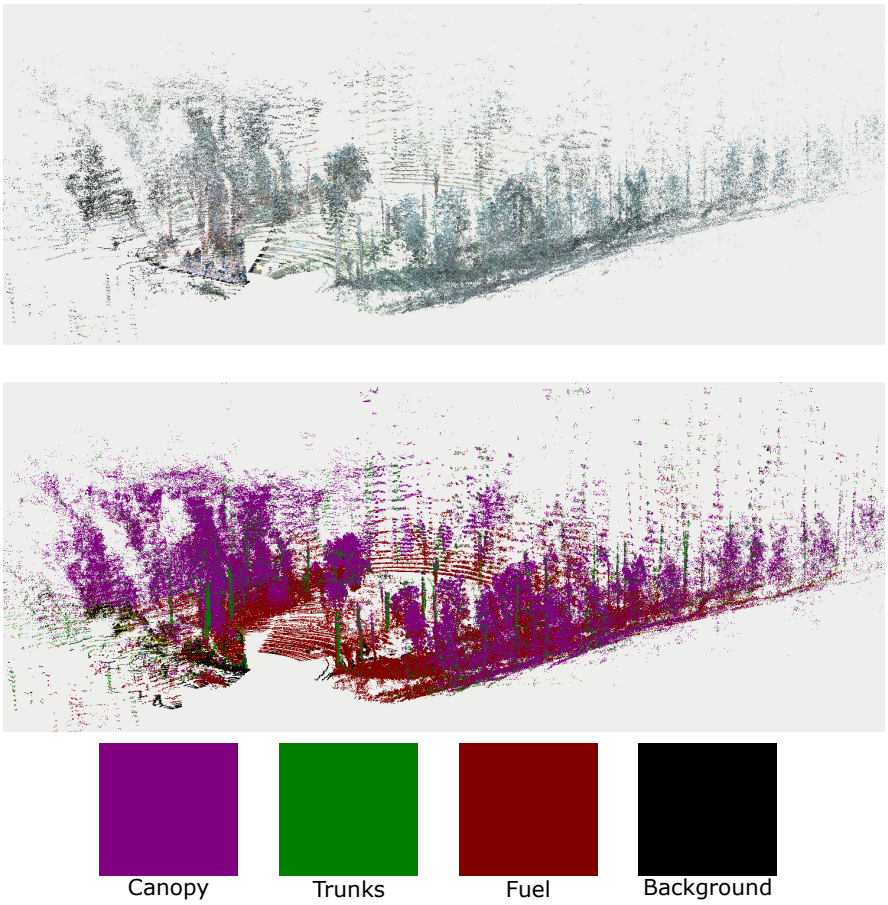


Figure 7.7: RGB point cloud (top) and semantic point cloud (bottom)

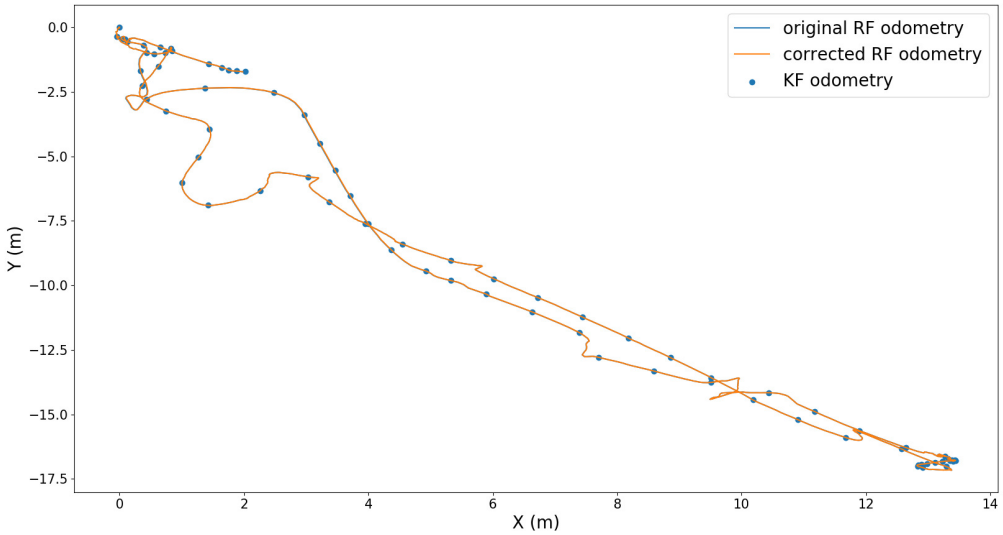


Figure 7.8: Trajectory plot of Porto dataset with a comparison of original, corrected, and keyframe odometry

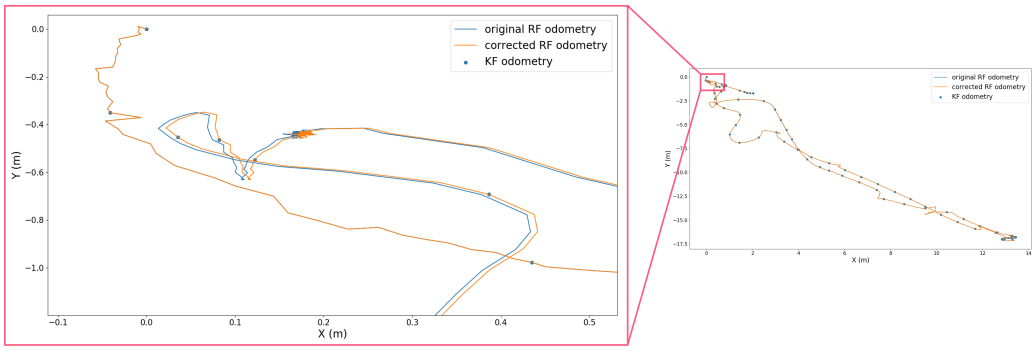


Figure 7.9: Close-up view of the top left portion towards the start of the trajectory

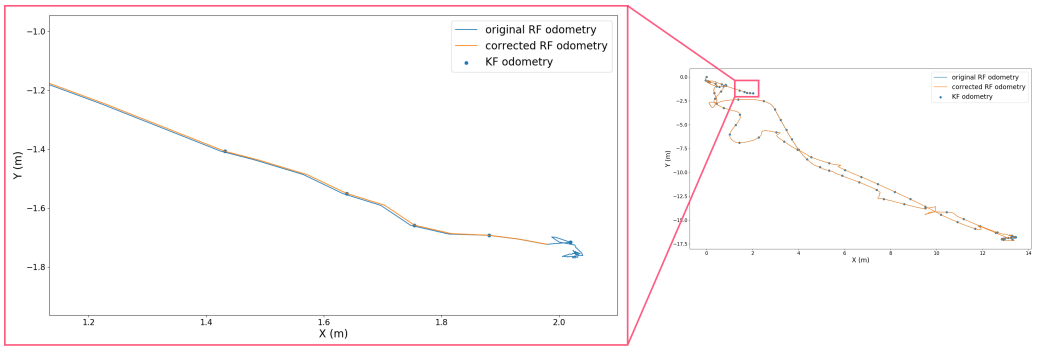


Figure 7.10: Close-up view of the top left portion towards the end of the trajectory

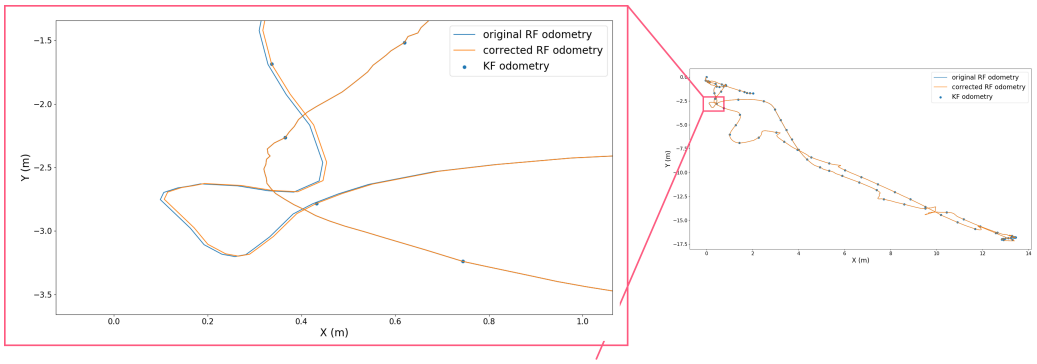


Figure 7.11: Close-up view of the top left portion of the trajectory after the UAV has departed from the starting point

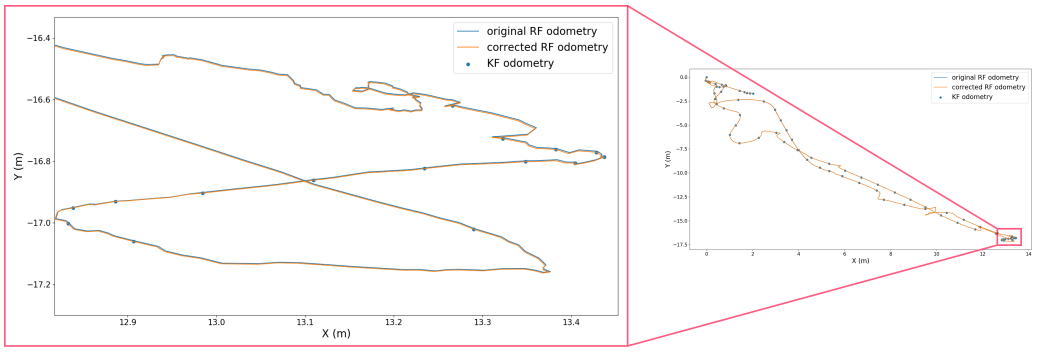


Figure 7.12: Close-up view of the bottom right portion of the trajectory as the UAV reaches the span of the trajectory

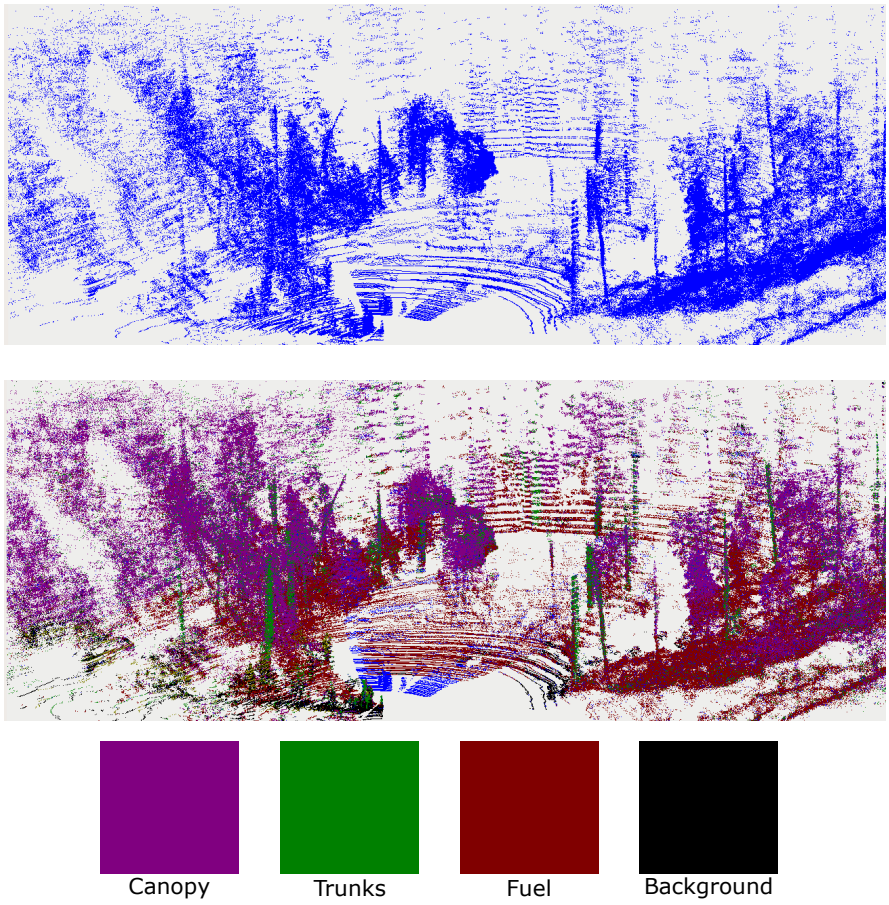


Figure 7.13: Original pose registered (top) and corrected pose registered with original pose registered overlayed (bottom)

Chapter 8

Conclusions

In our work, we have developed a modular multi-sensor payload that features LiDAR, inertial, and visual modalities. We have tested such a system in real forest environments in Porto, Portugal, and Pittsburgh, United States, and have verified the performance of the payload. Furthermore, we have implemented a pose correction method that uses the key poses as constraints and interpolates the relative frames in between the keyframe poses from SLAM. This pose correction method will be beneficial as we integrate state-of-the-art SLAM systems into our existing semantic mapping system.

8.1 Future Work

Future directions of this work include developing a visual-inertial-LiDAR SLAM system and fusing the SLAM and semantic mapping system with pose correction updates in real-time. Furthermore, there could be opportunities to utilize other sensors in our payload or add additional sensors to improve the performance of the SLAM system or semantic mapping system in the forest environment. In this sense, the multispectral camera could possibly be used to improve visual odometry since it may not be as sensitive to visible

lighting changes as a standard RGB camera. Multispectral cameras could also provide semantic information and be used to better aid the classification of fuel clusters instead of relying solely on RGB values since dried vegetation can fall under a specific vegetation index.

We note the following areas of improvement in the mechanical design and pose correction integration:

1. **Mechanical design improvements**: We acknowledge that vibrations are present during flight and a thorough vibration analysis would be beneficial to make the payload design more robust. Padding connections with a dampening medium can also warrant the payload and sensor measurements from unwanted vibrations present during flight. As the payload design currently stands, there could be some flexing present in the four carbon fiber tubes connecting the mounting plate to the base of the brackets on the side of the payload. A different mounting style for the stereo cameras can also be implemented: instead of mounting the stereo cameras directly onto the main payload base, the stereo cameras should be mounted on separate detachable plates, similar to how the LiDAR and NUC were mounted. This can also apply to other components. Although we have not encountered significant issues resulting from this during all of our field tests with the payload so far, improving the payload in this area could bolster its performance and prevent issues that may propagate over to the sensors.
2. **Semantic mapping integration improvements**: The semantic mapping integration can be improved in two aspects. The first aspect notes improving the datasets tested. Although it was difficult to readily identify obvious corrections in the semantic point cloud if we tested a dataset with more loop closures and significant updates, the improvements may possibly be more apparent. From the previous section, we note that a robust calibration with the current sensor payload is the next immediate step. A demonstration of the pose correction implemented in the seman-

tic mapping system can be further demonstrated by obtaining datasets with ground truth point clouds and trajectories. This can be achieved by using precise measurement devices such as the FARO scanner or an RTK system similar to the one we used for the Pittsburgh, PA dataset. The second aspect notes implementing the SLAM, semantic mapping, and pose correction online rather than offline. This would allow our system to be implemented on our UAVs in real-time in the field.

Appendix A

Additional Information

Table A.1: Material properties of carbon fiber plate

Property	**Value**
Elastic modulus	$2.275269908 \times 10^{11}$ N/m^2
Poisson's ratio	0.27
Shear modulus	$3.159998269 \times 10^{3}$ N/m^2
Mass density	1605.434926 kg/m^3
Tensile strength	$3.447378648 \times 10^{9}$ N/m^2
Compressive strength	-
Yield strength	-

Table A.2: Material properties of carbon fiber tube

Property	**Value**
Elastic modulus	$1.349999985 \times 10^{11}$ N/m^2
Poisson's ratio	0.3
Shear modulus	1.0×10^{10} N/m^2
Mass density	1600 kg/m^3
Tensile strength	1.5×10^{9} N/m^2
Compressive strength	-
Yield strength	1.2×10^{9} N/m^2

Table A.3: Material properties of 1060 aluminum alloy

Property	**Value**
Elastic modulus	6.9×10^{10} N/m^2
Poisson's ratio	0.33
Shear modulus	2.7×10^{10} N/m^2
Mass density	2700 kg/m^3
Tensile strength	6.89356×10^7 N/m^2
Compressive strength	-
Yield strength	2.75742×10^7 N/m^2

Table A.4: Material properties of alloy steel

Property	**Value**
Elastic modulus	2.1×10^{11} N/m^2
Poisson's ratio	0.28
Shear modulus	7.9×10^10 N/m^2
Mass density	7700 kg/m^3
Tensile strength	7.238256×10^8 N/m^2
Compressive strength	-
Yield strength	6.20422×10^8 N/m^2

Table A.5: Material properties of nylon 101

Property	**Value**
Elastic modulus	1.0×10^9 N/m^2
Poisson's ratio	0.3
Shear modulus	-
Mass density	1150 kg/m^3
Tensile strength	7.9289709×10^7 N/m^2
Compressive strength	-
Yield strength	6.0×10^7 N/m^2

Bibliography

[1] Alvium usb cameras user guide v4.2.3. URL https://cdn.alliedvision.com/fileadmin/content/documents/products/cameras/Alvium_USB/techman/Alvium-USB-Cameras_User-Guide.pdf. (document), 3.9

[2] Dji matrice 600 pro, 2017. URL https://www.uav-plus.com/dji-matrice-600-pro-industry-enterprise-professional-drone/. (document), 3.3

[3] Alta x, 2023. URL https://freeflysystems.com/alta-x. (document), 3.4

[4] André Silva Aguiar, Filipe Neves dos Santos, José Boaventura Cunha, Héber Sobreira, and Armando Jorge Sousa. Localization and Mapping for Robots in Agriculture and Forestry: A Survey. *Robotics*, 9(4):97, 2020. Publisher: Multidisciplinary Digital Publishing Institute. 2.1

[5] Abdulla Al-Kaff, Ángel Madridano, Sergio Campos, Fernando García, David Martín, and Arturo de la Escalera. Emergency support unmanned aerial vehicle for forest fire surveillance. *Electronics*, 9(2), 2020. ISSN 2079-9292. doi: 10.3390/electronics9020260. URL https://www.mdpi.com/2079-9292/9/2/260. 2.1

[6] M. Eduarda Andrada, Joao F. Ferreira, David Portugal, and Micael S. Couceiro. Integration of an artificial perception system for identification of live flammable material in forestry robotics. In *2022 IEEE/SICE International Symposium on System Integration (SII)*, pages 103–108, 2022. doi: 10.1109/SII52469.2022.9708734. 1, 2.1

[7] Maria Eduarda Andrada, David Russell, Tito Arevalo-Ramirez, Winnie Kuang, George Kantor, and Francisco Yandun. Mapping of potential fuel regions using uncrewed aerial vehicles for wildfire prevention. *Forests*,

14(8), 2023. ISSN 1999-4907. doi: 10.3390/f14081601. URL https://www.mdpi.com/1999-4907/14/8/1601. (document), 4.1, 7.1, 7.2, 7.3

[8] Rana Azzam, Tarek Taha, Shoudong Huang, and Yahya Zweiri. Feature-based visual simultaneous localization and mapping: A survey. *SN Applied Sciences*, 2:1–24, 2020. 2.2

[9] Panagiotis Barmpoutis, Periklis Papaioannou, Kosmas Dimitropoulos, and Nikos Grammalidis. A review on early forest fire detection systems using optical remote sensing. *Sensors*, 20(22):6442, 2020. 1

[10] Panagiotis Barmpoutis, Periklis Papaioannou, Kosmas Dimitropoulos, and Nikos Grammalidis. A review on early forest fire detection systems using optical remote sensing. *Sensors*, 20(22), 2020. ISSN 1424-8220. doi: 10.3390/s20226442. URL https://www.mdpi.com/1424-8220/20/22/6442. 1

[11] Benjamin C Bright, Andrew T Hudak, T Ryan McCarley, Alexander Spannuth, Nuria Sánchez-López, Roger D Ottmar, and Amber J Soja. Multitemporal lidar captures heterogeneity in fuel loads and consumption on the kaibab plateau. *Fire Ecology*, 18(1):18, 2022. 1

[12] Yanhong Chen, Youmin Zhang, Jing Xin, Yingmin Yi, Ding Liu, and Han Liu. A uav-based forest fire detection algorithm using convolutional neural network. In *2018 37th Chinese Control Conference (CCC)*, pages 10305–10310, 2018. doi: 10.23919/ChiCC.2018.8484035. 1

[13] Emilio Chuvieco, Inmaculada Aguado, Marta Yebra, Héctor Nieto, Javier Salas, M Pilar Martín, Lara Vilar, Javier Martínez, Susana Martín, Paloma Ibarra, et al. Development of a framework for fire risk assessment using remote sensing and geographic information system technologies. *Ecological modelling*, 221(1):46–58, 2010. 1

[14] Emilio Chuvieco, Florent Mouillot, Guido R. van der Werf, Jesús San Miguel, Mihai Tanase, Nikos Koutsias, Mariano García, Marta Yebra, Marc Padilla, Ioannis Gitas, Angelika Heil, Todd J. Hawbaker, and Louis Giglio. Historical background and current developments for mapping burned area from satellite earth observation. *Remote Sensing of Environment*, 225:45–64, 2019. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2019.02.013. URL https://www.sciencedirect.com/science/article/pii/S0034425719300689. 1

[15] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld,

Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proc. of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016. 5.2

[16] Micael S. Couceiro, David Portugal, João F. Ferreira, and Rui P. Rocha. Semfire: Towards a new generation of forestry maintenance multi-robot systems. In *2019 IEEE/SICE International Symposium on System Integration (SII)*, pages 270–276, 2019. doi: 10.1109/SII.2019.8700403. 1

[17] Micael S. Couceiro, David Portugal, João F. Ferreira, and Rui P. Rocha. Semfire: Towards a new generation of forestry maintenance multi-robot systems. In *2019 IEEE/SICE International Symposium on System Integration (SII)*, pages 270–276, 2019. doi: 10.1109/SII.2019.8700403. 2.1

[18] César Debeunne and Damien Vivet. A review of visual-lidar fusion based simultaneous localization and mapping. *Sensors*, 20(7):2068, 2020. 2.2

[19] Todd M Ellis, David MJS Bowman, Piyush Jain, Mike D Flannigan, and Grant J Williamson. Global increase in wildfire risk due to climate-driven declines in fuel moisture. *Global change biology*, 28(4):1544–1559, 2022. 1

[20] Christian Forster, Luca Carlone, Frank Dellaert, and Davide Scaramuzza. Imu preintegration on manifold for efficient visual-inertial maximum-a-posteriori estimation. Georgia Institute of Technology, 2015. 4.2.1, 4.2.1

[21] Paul Furgale, Timothy D. Barfoot, and Gabe Sibley. Continuous-time batch estimation using temporal basis functions. In *2012 IEEE International Conference on Robotics and Automation*, pages 2088–2095, 2012. doi: 10.1109/ICRA.2012.6225005. 5.3

[22] V M Gedjo, V K Dubovyi, and A A Leonovich. New ways and means of localization and extinguishing surface forest fires. *IOP Conference Series: Earth and Environmental Science*, 316(1):012005, aug 2019. doi: 10.1088/1755-1315/316/1/012005. URL https://dx.doi.org/10.1088/1755-1315/316/1/012005. 1

[23] Alessandro Giusti, Jérôme Guzzi, Dan C. Cireşan, Fang-Lin He, Juan P. Rodríguez, Flavio Fontana, Matthias Faessler, Christian Forster, Jürgen Schmidhuber, Gianni Di Caro, Davide Scaramuzza, and Luca M. Gam-

bardella. A machine learning approach to visual perception of forest trails for mobile robots. *IEEE Robotics and Automation Letters*, 1(2): 661–667, 2016. doi: 10.1109/LRA.2015.2509024. 2.1

[24] Meng-Hao Guo, Cheng-Ze Lu, Qibin Hou, Zhengning Liu, Ming-Ming Cheng, and Shi-Min Hu. Segnext: Rethinking convolutional attention design for semantic segmentation, 2022. 5.2

[25] Yinan He, Gang Chen, Christopher Potter, and Ross K Meentemeyer. Integrating multi-sensor remote sensing and species distribution modeling to map the spread of emerging forest disease and tree mortality. *Remote sensing of Environment*, 231:111238, 2019. 1

[26] M. Herbert, C. Caillas, E. Krotkov, I.S. Kweon, and T. Kanade. Terrain mapping for a roving planetary explorer. In *Proceedings, 1989 International Conference on Robotics and Automation*, volume 2, pages 997–1002, 1989. doi: 10.1109/ROBOT.1989.100111. 2.3

[27] Armin Hornung, Kai M. Wurm, Maren Bennewitz, Cyrill Stachniss, and Wolfram Burgard. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Autonomous Robots*, 2013. doi: 10.1007/s10514-012-9321-0. URL `https://octomap.github.io`. Software available at `https://octomap.github.io`. 2.3, 5.1

[28] Piyush Jain, Dante Castellanos-Acuna, and et al. Coogan, Sean C. P. Observed increases in extreme fire weather driven by atmospheric humidity and temperature. *Nature Climate Change*, 12:63 – 70, January 2022. doi: 10.1038/s41558-021-01224-1. 1

[29] Youngseok Jang, Hojoon Shin, and H Jin Kim. Pose correction algorithm for relative frames between keyframes in slam. In *Proceedings of the Asian Conference on Computer Vision*, 2020. (document), 5.1, 5.4, 5.3, 5.5, 5.5, 5.5

[30] Zhentian Jiao, Youmin Zhang, Jing Xin, Lingxia Mu, Yingmin Yi, Han Liu, and Ding Liu. A deep learning based forest fire detection approach using uav and yolov3. In *2019 1st International Conference on Industrial Artificial Intelligence (IAI)*, pages 1–5, 2019. doi: 10.1109/ICIAI.2019.8850815. 2.1

[31] W. Matt Jolly, Mark A. Cochrane, Patrick H. Freeborn, Zachary A. Holden, Timothy J. Brown, Grant J. Williamson, and David M. J. S. Bowman. Climate-induced variations in global wildfire danger from 1979

to 2013. *Nature Communications*, 6(7537):7537, 2015. ISSN 2041-1723. doi: 10.1038/ncomms8537. 1

[32] Thomas Katagis, Ioannis Z. Gitas, Pericles Toukiloglou, Sander Veraverbeke, and Rudi Goossens. Trend analysis of medium- and coarse-resolution time series image data for burned area mapping in a mediterranean ecosystem. *International Journal of Wildland Fire*, 23(5):668 – 677, 2014. doi: 10.1071/WF12055. Cited by: 18. 1

[33] Giseop Kim and Ayoung Kim. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809. IEEE, 2018. 4.1, 7.1.1

[34] Eldar Kurbanov, Oleg Vorobev, Sergey Lezhnin, Jinming Sha, Jinliang Wang, Xiaomei Li, Janine Cole, Denis Dergunov, and Yibo Wang. Remote sensing of forest burnt area, burn severity, and post-fire recovery: a review. *Remote Sensing*, 14(19):4714, 2022. 1

[35] Westerling Anthony LeRoy. Increasing western us forest wildfire activity: sensitivity to changes in the timing of spring. *Philos. Trans. R. Soc. B Biol. Sci.*, 371, 2016. ISSN 1696. doi: https://doi.org/10.1098/rstb.2015.0178. 2.1

[36] Eva Marino, Pedro Ranz, José Luis Tomé, Miguel Ángel Noriega, Jessica Esteban, and Javier Madrigal. Generation of high-resolution fuel model maps from discrete airborne laser scanner and landsat-8 oli: A low-cost and highly updated methodology for large areas. *Remote Sensing of Environment*, 187:267–280, 2016. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2016.10.020. URL https://www.sciencedirect.com/science/article/pii/S0034425716303923. 1

[37] Donald Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 19(2):129–147, 1982. ISSN 0146-664X. doi: https://doi.org/10.1016/0146-664X(82)90104-6. URL https://www.sciencedirect.com/science/article/pii/0146664X82901046. 2.3

[38] George H Mitri and Ioannis Z Gitas. Fire type mapping using object-based classification of ikonos imagery. *International journal of wildland Fire*, 15(4):457–462, 2006. 1

[39] Hans P. Moravec. Robot spatial perception by stereoscopic vision and 3d evidence grids. 1996. 2.3

[40] Penelope Morgan, Robert E Keane, Gregory K Dillon, Theresa B Jain, Andrew T Hudak, Eva C Karau, Pamela G Sikkink, Zachary A Holden, and Eva K Strand. Challenges of assessing fire and burn severity using field measures, remote sensing and modelling. *International Journal of Wildland Fire*, 23(8):1045–1060, 2014. 1

[41] Mohsen Naderpour, Hossein Mojaddadi Rizeei, Nima Khakzad, and Biswajeet Pradhan. Forest fire induced natech risk assessment: A survey of geospatial technologies. *Reliability Engineering System Safety*, 191:106558, 2019. ISSN 0951-8320. doi: https://doi.org/10.1016/j.ress.2019.106558. URL https://www.sciencedirect.com/science/article/pii/S095183201831202X. 1

[42] Luiz F. P. Oliveira, António P. Moreira, and Manuel F. Silva. Advances in forest robotics: A state-of-the-art survey. *Robotics*, 10(2), 2021. ISSN 2218-6581. doi: 10.3390/robotics10020053. URL https://www.mdpi.com/2218-6581/10/2/53. 2.1

[43] Open MMLab. Mmsegmentation. URL https://github.com/open-mmlab/mmsegmentation. 5.2

[44] Mehtap Ozenen Kavlak, Saye Nihan Cabuk, and Mehmet Cetin. Development of forest fire risk map using geographical information systems and remote sensing capabilities: Ören case. *Environmental Science and Pollution Research*, 28(25):33265–33291, 2021. 1

[45] Juli G Pausas and Jon E Keeley. Wildfires and global change. *Frontiers in Ecology and the Environment*, 19(7):387–395, 2021. doi: https://doi.org/10.1002/fee.2359. URL https://esajournals.onlinelibrary.wiley.com/doi/abs/10.1002/fee.2359. 1

[46] Fernando Pérez-Cabello, Raquel Montorio, and Daniel Borini Alves. Remote sensing techniques to assess post-fire vegetation recovery. *Current Opinion in Environmental Science & Health*, 21:100251, 2021. 1

[47] Seth H. Peterson, Janet Franklin, Dar A. Roberts, and Jan W. van Wagtendonk. Mapping fuels in yosemite national park. *Canadian Journal of Forest Research*, 43(1):7–17, 2013. doi: 10.1139/cjfr-2012-0213. URL https://doi.org/10.1139/cjfr-2012-0213. 1

[48] Patrick Pfaff, Rudolph Triebel, and Wolfram Burgard. n efficient extension to elevation maps for outdoor terrain mapping and loop closing. *The International Journal of Robotics Research*, 26(2):217–230, 2007. doi:

10.1177/0278364906075165. 2.3

[49] Marek Pierzchała, Philippe Giguère, and Rasmus Astrup. Mapping forests using an unmanned ground vehicle with 3d lidar and graph-slam. *Computers and Electronics in Agriculture*, 145:217–225, 2018. 2.2

[50] David Portugal, Maria Eduarda Andrada, André G. Araújo, Micael S. Couceiro, and João Filipe Ferreira. *ROS Integration of an Instrumented Bobcat T190 for the SEMFIRE Project*, pages 87–119. Springer International Publishing, Cham, 2021. ISBN 978-3-030-75472-3. doi: 10.1007/978-3-030-75472-3_3. URL https://doi.org/10.1007/978-3-030-75472-3_3. 2.1

[51] Marcos Rodrigues, Àngel Cunill Camprubí, Rodrigo Balaguer-Romano, Celso J. Coco Megía, Francisco Castañares, Julien Ruffault, Paulo M. Fernandes, and Víctor Resco de Dios. Drivers and implications of the extreme 2022 wildfire season in southwest europe. *Science of The Total Environment*, 859:160320, 2023. ISSN 0048-9697. doi: https://doi.org/10.1016/j.scitotenv.2022.160320. URL https://www.sciencedirect.com/science/article/pii/S0048969722074204. 2.1

[52] Juan Jesús Roldán-Gómez, Eduardo González-Gironda, and Antonio Barrientos. A survey on robotic technologies for forest firefighting: Applying drone swarms to improve firefighters' efficiency and safety. *Applied Sciences*, 11(1), 2021. ISSN 2076-3417. doi: 10.3390/app11010363. URL https://www.mdpi.com/2076-3417/11/1/363. 2.1

[53] Juan Jesús Roldán-Gómez, Eduardo González-Gironda, and Antonio Barrientos. A survey on robotic technologies for forest firefighting: Applying drone swarms to improve firefighters' efficiency and safety. *Applied Sciences*, 11(1), 2021. ISSN 2076-3417. doi: 10.3390/app11010363. URL https://www.mdpi.com/2076-3417/11/1/363. 1

[54] David Jacob Russell, Tito Arevalo-Ramirez, Chinmay Garg, Winnie Kuang, Francisco Yandun, David Wettergreen, and George Kantor. UAV Mapping with Semantic and Traversability Metrics for Forest Fire Mitigation. In *ICRA 2022 Workshop in Innovation in Forestry Robotics: Research and Industry Adoption*, 2022. 1, 2.1, 4.1, 5.1, 5.3, 5.4

[55] Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Rus Daniela. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142. IEEE, 2020.

2.2, 3.1, 4.1, 7.1.1

[56] Weizhao Shao, Srinivasan Vijayarangan, Cong Li, and George Kantor. Stereo visual inertial lidar simultaneous localization and mapping. *CoRR*, abs/1902.10741, 2019. URL http://arxiv.org/abs/1902.10741. 2.2, 3.1, 6.1, 7.2

[57] Ilan Stavi. Wildfires in grasslands and shrublands: A review of impacts on vegetation, soil, hydrology, and geomorphology. *Water*, 11(5), 2019. ISSN 2073-4441. doi: 10.3390/w11051042. URL https://www.mdpi.com/2073-4441/11/5/1042. 2.1

[58] S. Sudhakar, V. Vijayakumar, C. Sathiya Kumar, V. Priya, Logesh Ravi, and V. Subramaniyaswamy. Unmanned aerial vehicle (uav) based forest fire detection and monitoring for reducing false alarms in forest-fires. *Computer Communications*, 149:1–16, 2020. ISSN 0140-3664. doi: https://doi.org/10.1016/j.comcom.2019.10.007. URL https://www.sciencedirect.com/science/article/pii/S0140366419308655. 2.1

[59] Tom Toulouse, Lucile Rossi, Antoine Campana, Turgay Celik, and Moulay A. Akhloufi. Computer vision for wildfire research: An evolving image dataset for processing and analysis. *Fire Safety Journal*, 92:188–194, 2017. ISSN 0379-7112. doi: https://doi.org/10.1016/j.firesaf.2017.06.012. URL https://www.sciencedirect.com/science/article/pii/S0379711217302114. 1

[60] Rudolph Triebel, Patrick Pfaff, and Wolfram Burgard. Multi-level surface maps for outdoor terrain mapping and loop closing. In *2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2276–2282, 2006. doi: 10.1109/IROS.2006.282632. 2.3

[61] Carlos Viegas, Babak Chehreh, José Andrade, and João Lourenço. Tethered uav with combined multi-rotor and water jet propulsion for forest fire fighting. *Journal of Intelligent & Robotic Systems*, 104(2):21, 2022. 1

[62] Martin J. Wooster, Gareth J. Roberts, Louis Giglio, David P. Roy, Patrick H Freeborn, Luigi Boschetti, Chris Justice, Charles Ichoku, Wilfrid Schroeder, Diane Davies, Alistair M.S. Smith, Alberto Setzer, Ivan Csiszar, Tercia Strydom, Philip Frost, Tianran Zhang, Weidong Xu, Mark C de Jong, Joshua M. Johnston, Luke Ellison, Krishna Vadrevu, Aaron M. Sparks, Hannah Nguyen, Jessica McCarty, Veerachai Tanpipat, Chris Schmidt, and Jesus San-Miguel-Ayanz. Satellite remote sensing of active fires: History and current status, applications and future require-

ments. *Remote Sensing of Environment*, 267:112694, 2021. ISSN 0034-4257. doi: https://doi.org/10.1016/j.rse.2021.112694. URL https://www.sciencedirect.com/science/article/pii/S0034425721004144. 1

[63] Enze Xie, Wenhai Wang, Zhiding Yu, Anima Anandkumar, Jose M. Alvarez, and Ping Luo. SegFormer: Simple and Efficient Design for Semantic Segmentation with Transformers. pages 1–18, 2021. URL http://arxiv.org/abs/2105.15203. 5.2

[64] Wei Xu, Yixi Cai, Dongjiao He, Jiarong Lin, and Fu Zhang. Fast-lio2: Fast direct lidar-inertial odometry. *IEEE Transactions on Robotics*, 38 (4):2053–2073, 2022. 4.1, 7.1.1

[65] Zhang Xuan and Filliat David. Real-time voxel based 3d semantic mapping with a hand held rgb-d camera. https://github.com/floatlazer/semantic_slam, 2018. 5.1

[66] Chi Yuan, Zhixiang Liu, and Youmin Zhang. Uav-based forest fire detection and tracking using image processing techniques. In *2015 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 639–643, 2015. doi: 10.1109/ICUAS.2015.7152345. 1

[67] Ji Zhang and Sanjiv Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, volume 2, pages 1–9. Berkeley, CA, 2014. 2.2