Indoor Aerial 3D Mapping and Target Localization with a Custom-Built Multispectral Visual-Inertial Sensor System

Animesh K. Shastry*, Wei Cui*, Sydrak S. Abdi*, Srijal S. Poojari[†], Ahmed Ashry[‡], Qingwen Wei[§], Alec Luterman[¶], Vijay Ved[¶], and Derek A. Paley[‖]
University of Maryland (UMD), College Park, MD, 20742

This paper presents the systematic design of an advanced and feature-rich Uncrewed Aerial System (UAS) tailored for first responders in search and rescue operations. It details the creation and implementation of a 3D mapping, target detection, and localization framework alongside the specific hardware and software components essential for enhanced reliability. Utilizing a distributed computing approach, the UAS offloads intensive computations to an offboard computer, resulting in improved real-time mapping performance and system stability. The cost-effective UAS named *Intrigue*, participated in the National Institute of Standards and Technology (NIST) 2023 First Responder UAS 3D Mapping Challenge, securing third place overall, along with awards for Best-in-Class Bill of Materials Total Cost, Best-in-Class Map Data Acquisition Speed, and Best-in-Class Blue/Green UAS Capability. Competition results and independent testing demonstrate the system's reliable performance in diverse scenarios.

I. Introduction

In today's rapidly evolving landscape of emergency response, the demand for enhanced tools to support first responders in navigating hazardous indoor environments is more pressing than ever. Incidents such as fires, floods, and structural collapses frequently compromise the safety and practicality of traditional rescue operations, necessitating innovative solutions that can provide real-time situational awareness without exposing personnel to undue risk. Uncrewed Aerial Systems (UAS) have emerged as a pivotal technology in this regard, offering the capability to swiftly gather and relay critical information to incident command teams during high-stakes operations.

Despite significant advancements in robotics and UAS technology, the full potential of UAS as an invaluable tool for first responders remains largely untapped. Current applications have demonstrated the ability of UAS to operate in complex and dangerous environments, yet there is still a considerable gap in the widespread integration and utilization of these systems in everyday emergency response protocols. This manuscript presents a low-cost system for indoor 3D mapping and target localization designed to be used by first responders. This system competed in the National Institute of Standards and Technology (NIST) 2023 First Responder UAS 3D Mapping Challenge[1] and won third place overall, in addition to awards for Best-in-Class Bill of Materials Total Cost, Best-in-Class Map Data Acquisition Speed and Best-in-Class Blue/Green UAS Capable.

Several commercial UAS are currently used in the public safety and search and rescue industry. This paper compares the specifications of our solution against these commercial options. Many low-cost UAS lack real-time mapping, target detection, and target localization capabilities. While flagship and enterprise-level UAS in the industry meet some first responder requirements, they are generally larger and unsuitable for indoor environments. Additionally, their higher cost makes them less appealing to public safety departments with budget constraints.

In academia, various technical manuscripts describe or implement specific capabilities desirable in UAS for search and rescue applications. Typically, the focus is on individual software frameworks for mapping, object detection, object localization, communication design, or UAS hardware design. However, a systematic approach to integrating all these capabilities into a single UAS is lacking. Karam et al. [2] utilized a microdrone equipped with six laser rangefinders (1D scanners) and an optical sensor for mapping and positioning, employing graph SLAM for loop closure detection.

*Doctoral Graduate, Department of Aerospace Engineering, University of Maryland, College Park, MD, 20742. AIAA member

[†] Graduate Student, Department of Electrical and Computer Engineering, University of Maryland, College Park, MD, 20742

[‡] Graduate Student, Department of Aerospace Engineering, University of Maryland, College Park, MD, 20742

[§] Faculty Specialist, Department of Aerospace Engineering, University of Maryland, College Park, MD, 20742

[¶] Undergraduate Student, Department of Computer Science, University of Maryland, College Park, MD, 20742

[‖] Willis H. Young Jr. Professor of Aerospace Engineering Education, Department of Aerospace Engineering and Institute for Systems Research, University of Maryland, College Park, MD 20742. AIAA Senior Member

Despite this, the resulting maps are uncolored point clouds and textureless, limiting their usefulness for first responders. Caballero et al. [3] presented a mapping system for first responders that creates 2D maps using UAVs, but their design and implementation are geared towards outdoor use cases. Generally, indoor mapping and SLAM have recently seen numerous implementations. Otero et al. [4] provided an interesting analysis and comparison of the various mobile indoor mapping options available on the market. Their review includes handheld scanners, backpack devices, and trolley configurations, which are mostly suited for ground robots rather than UAS. Placed et al. [5] conducted a comprehensive survey of the state-of-the-art in active SLAM, highlighting how the disparity and lack of unification in the literature have hindered the development of cohesive frameworks, mature algorithms, and their transition to practical applications. Kolhatkar et al. [6] reviewed various techniques used in mapping and localization of mobile robots and the design of low-cost mobile platforms with sensors, focusing on LiDAR and RGB-D Camera technology—the latter utilized in this paper.

Additionally, LiDAR and IMU-based mapping technologies have seen widespread use, from pipe-inspecting UAVs to self-driving cars. Kumar et al. [7] introduced a method to efficiently determine the planar position of UAVs via a point-to-point scan matching algorithm, leveraging data from a horizontally scanning primary LiDAR. The UAV's altitude relative to the ground was estimated using a vertically scanning secondary LiDAR mounted orthogonally to the primary LiDAR, with a Kalman filter integrating data from both LiDARs to calculate the 3D position. However, the resulting map lacks texture, making it challenging for human interpretation, and finer details necessary for object classification are absent, diminishing its suitability for search and rescue mapping applications. Similarly, Opromolla et al. [8] presented an approach combining point clouds from LiDAR with inertial data. However, their implementation was confined to 2D environments, unsuitable for mapping intricate 3D spaces. Chan et al. [9] employed a LiDAR approach to demonstrate and assess state-of-the-art SLAM algorithms, LeGO-LOAM [10] and LIO-SAM [11], in simulated indoor environments using ground robots. Results revealed that even advanced LiDAR odometry and mapping methods can experience significant drift due to the absence of features. Cvisic et al. [12] introduced a stereo vision SLAM method that employs separate localization and mapping threads. Localization relies on visual feature matching, whereas mapping utilizes depth information generated using the semi-global matching algorithm [13]. However, while the approach excels in stereo localization, the resulting maps lack texture and fidelity, limiting their utility for first responders. Labbe et al. [14] presented a comprehensive mapping software package supporting both visual and LiDAR SLAM. Their graph-based SLAM approach is adopted here due to its versatility in accommodating various sensors, including unsynchronized RGB-D systems. Recent developments include generic multi-robot and multi-modal mapping frameworks, such as the one by Cramariuc et al. [15], capable of integrating multiple robots, visual landmarks, and LiDAR scans. Additionally, there's ongoing research in utilizing UAVs equipped with neural networks for mapping tasks [16, 17]. However, these advanced frameworks are yet to attain the technical readiness level required for reliable UAS mapping performance.

Recent research has been dedicated to exploring the potential of UAS in target-tracking missions, spanning both indoor and outdoor environments. Alhafnawi et al. [18] conducted a survey investigating UAV-based target tracking and monitoring across diverse settings. Their study delved into the deployment scenarios of these systems, offering detailed characterization and analysis. Cui et al. [19] introduced an end-to-end search and revisit framework designed for small UAVs to detect and localize targets using a single onboard camera module. This framework is well-suited for outdoor applications. In another study, Wang et al. [20] proposed a real-time multi-target localization scheme employing an electro-optical stabilized imaging system. However, the system's output of geodetic coordinates limits its applicability to outdoor settings. On a different note, Unlu et al. [21] introduced a comprehensive end-to-end framework for mapping, detecting, and extinguishing fire targets in indoor environments. However, the framework relied on a fiducial marker to simulate fire and was solely demonstrated in simulated environments.

This article makes several significant contributions: (1) it presents a comprehensive pipeline for real-time mapping and target detection in indoor environments with limited network infrastructure; (2) it offers a detailed description, comparison to alternatives, and evaluation of the hardware and software components comprising the system; (3) through experimental validation in diverse indoor settings, it demonstrates the efficacy of the proposed system in generating high-quality maps and detecting targets. These contributions are valuable as they pave the way for researchers and first responders to develop low-cost, highly capable, and reliable UAS using readily accessible open-source software tools and hardware designs.

The remainder of this article is structured as follows. Section 2 provides a detailed description of the problem statement that the UAS design aims to address, outlining the specific challenges and requirements that informed the design process. Section 3 offers a complete and comprehensive description of the UAS design, including the system architecture, hardware components, and software algorithms utilized. Section 4 presents the results from real-world testing of the developed UAS, discussing the performance, effectiveness, and any observed limitations during these trials. Finally, Section 5 summarizes the key findings of the article and discusses ongoing work aimed at further enhancing the UAS capabilities and addressing any identified issues.

II. Problem Statement

Consider a scenario where responders can access a real-time 3D map of an indoor environment, pinpointing obstacles, damaged structures, and the location of individuals in distress with high accuracy. Such capabilities would revolutionize the effectiveness and safety of emergency operations, providing responders with the insights needed to make split-second decisions in high-stakes situations. Unfortunately, the current landscape presents significant barriers to the widespread adoption of such advanced UAS solutions. While the technology exists in concept, commercial availability at a low cost remains elusive, posing challenges for public safety budgets and compliance with stringent national security requirements. Initiatives like the Defense Innovation Unit's Blue UAS Cleared List and the Association for Uncrewed Vehicle Systems International's Green UAS Cleared List are instrumental in setting standards and guidelines for security clearance. However, there is still much ground to cover in terms of affordability and accessibility for emergency response agencies.

This paper presents a systematic design of a highly capable and feature-rich UAS solution, integrating various valuable works that have been performed in academia in recent years. The aim is to bridge the gap between academia and industry by creating a comprehensive and adaptable UAS to meet the needs of first responders. This UAS will provide crucial assistance to Incident Command, facilitating rapid assessment of damage, hazard identification, and survivor location—essential components for an effective emergency response.

III. System Design

A. Autonomy Architecture

Achieving real-time performance of a 3D mapping framework, encompassing the intricate tasks of generating dense colored point-clouds and subsequent meshing with high-resolution texturing, presents a formidable challenge in the realm of UAV technology. This challenge is chiefly rooted in the monumental computational demands inherent to these tasks, rendering them impractical for smaller UAVs where stringent constraints on weight and size of onboard avionics reign supreme. Striking a delicate balance between computational power and the physical limitations of UAVs poses a significant obstacle, necessitating innovative solutions to optimize performance without compromising functionality or jeopardizing flight stability.

One such solution lies in the strategic implementation of distributed computing, offering a pathway to alleviate the computational load burdening the UAV while simultaneously enhancing runtime reliability and ushering advanced autonomy capabilities to lighter and smaller-sized UAVs. Through the orchestration of onboard and offboard computers, a symbiotic relationship is established wherein autonomy tasks are efficiently communicated and distributed. The onboard computer assumes the role of executing lightweight software modules tailored specifically for GPS-denied flight and obstacle avoidance, thereby ensuring swift and agile maneuvering in dynamic environments. Additionally, it assumes the responsibility of collecting data from onboard sensors and compressing the acquired data to optimize transmission efficiency before dispatching it to the offboard computer for further processing.

In contrast, the offboard computer serves as the computational powerhouse, tasked with executing resource-intensive processes integral to the generation of dense 3D maps. Leveraging its robust computing capabilities, the offboard computer meticulously crafts intricate point-clouds and meshes, while concurrently running neural network models for object detection and localization. Furthermore, it undertakes the arduous task of rendering detailed textured 3D maps in real-time, thereby facilitating swift decision-making and enhancing situational awareness during UAV operations. A block diagram of the full autonomy framework is shown in Figure 1. Additionally, a block diagram for the onboard part of the autonomy framework is shown in Figure 2.

Levels of Autonomy

In the design of Autonomy Frameworks for aerial vehicles, the incorporation of fallback options in the event of system failure emerges as a critical consideration. Drawing inspiration from the Autonomy Levels established for Self-Driving Vehicles, we have devised a structured hierarchy of aerial autonomy levels, shown in Figure 3. It is strategically engineered to minimize operator workload while maximizing operational efficiency. As the vehicle's

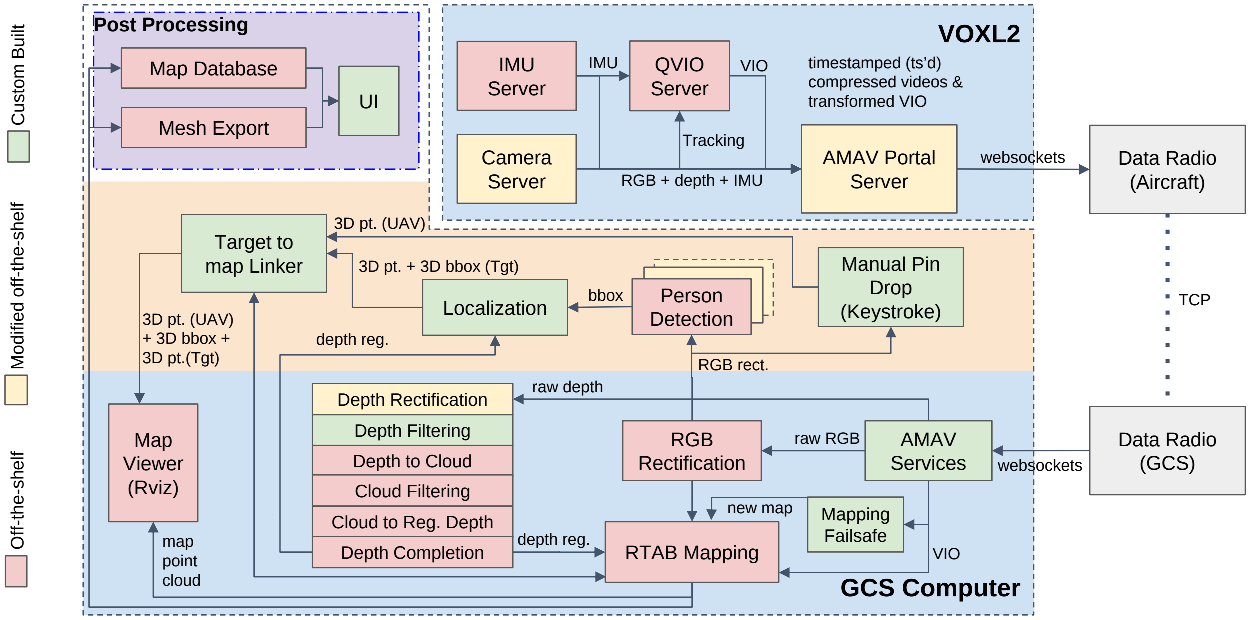


Fig. 1 Full autonomy block diagram

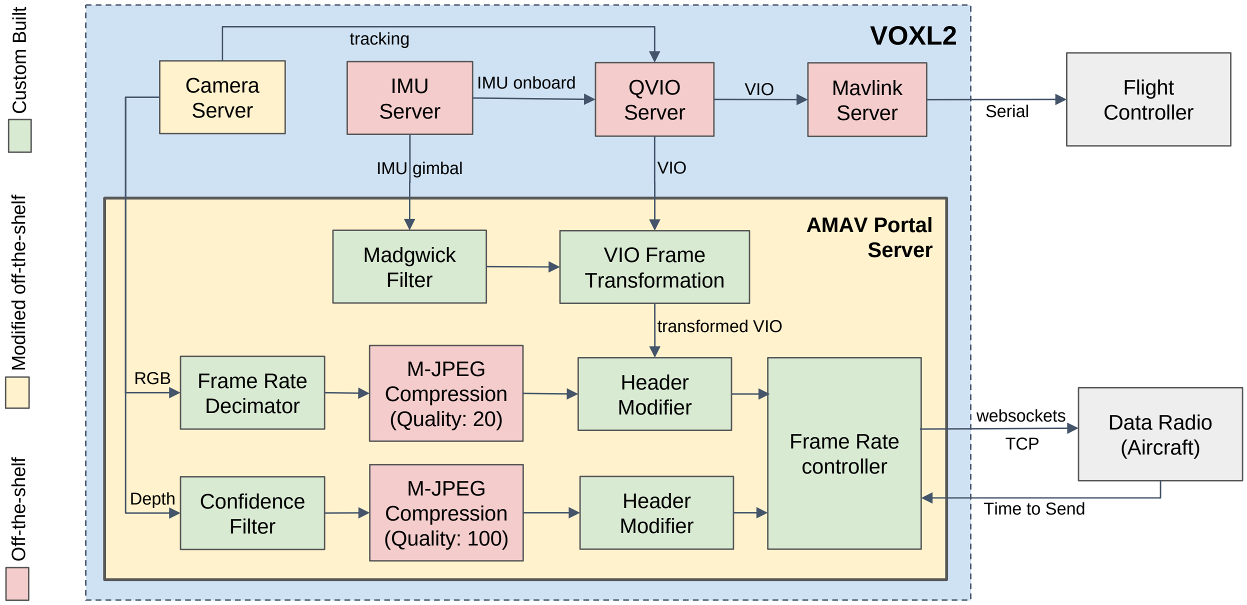


Fig. 2 Onboard autonomy block diagram

autonomy level ascends within this framework, the reliance on human intervention diminishes progressively.

Each autonomy level is meticulously defined, with specific capabilities and design requirements delineated to ensure seamless progression through the hierarchy. At the lowest autonomy levels, human operators retain primary control, with the system serving in a supplementary capacity to assist and augment decision-making. As autonomy levels advance, the aerial vehicle assumes greater responsibility for navigation, obstacle avoidance, and mission execution, thereby reducing the cognitive burden on operators. In accordance with this framework, stringent criteria are assigned to each autonomy level, encompassing factors such as sensor redundancy, fault tolerance, and real-time decision-making capabilities. By adhering to these predetermined standards, the autonomy framework not only fosters operational safety and reliability but also fosters innovation and advancement in autonomous aerial technologies. Moreover, the delineation of clear autonomy levels serves as a roadmap for the progressive development and integration of autonomy features, guiding the evolution of aerial vehicles towards increasingly autonomous operation while ensuring compatibility with existing regulatory frameworks and industry standards.

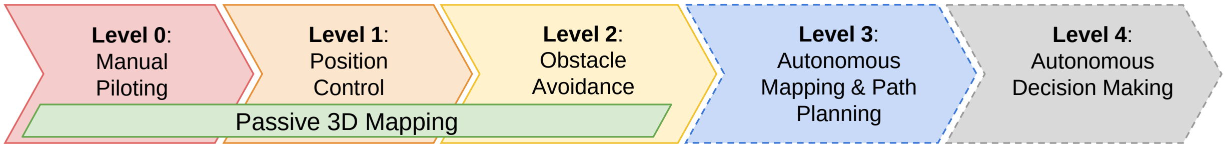

Level 0: Manual Piloting
Level 1: Position Control
Level 2: Obstacle Avoidance
Level 3: Autonomous Mapping & Path Planning
Level 4: Autonomous Decision Making
Passive 3D Mapping

Fig. 3 Proposed Levels of Indoor Aerial Autonomy

Onboard Autonomy

An onboard computer always has computational constraints due to the aircraft's size, weight, and power requirements. To achieve real-time performance through a distributed computing approach, the onboard autonomy is designed to be extremely lightweight, containing only software modules necessary for sustaining stable flight. A combination of off-the-shelf software tools and custom-built-from-scratch software modules forms the building blocks of the autonomy framework. Some off-the-shelf modules were also modified to make them more suitable for our requirements and to resolve existing bugs that were not addressed by the developers for the specific version.

Three cameras and two IMUs are used for localization and mapping needs. Two mapping cameras and one IMU are present on a gimbal; a localization/tracking camera and the other IMU are fixed to the aircraft's frame. The OV7251 global shutter camera with a fisheye lens is utilized as a tracking camera for obtaining features for performing Visual-Inertial Odometry with the help of an onboard ICM42688 IMU. The IMX412 low-light camera [22] is used as the RGB sensor for mapping purposes, capturing color images in a wide range of lighting conditions. The PMD-Tof camera [23] is used as the depth sensor for mapping, generating high-quality depth images for capturing scene structure. Finally, a secondary ICM2098 IMU on a gimbal provides inertial measurements to help in computing the gimbal orientation.

For Visual-Inertial-Odometry computation, a proprietary filter-based algorithm developed by Qualcomm is utilized. The odometry pose outputs are transformed to the gimbal frame by utilizing the gimbal's orientation estimate obtained from the gimbal IMU through the use of the Madgwick Filter. Both RGB and depth images undergo a preprocessing step. The RGB image frames are decimated to lower the frame rate to that of the depth image sensor before sending to a compression algorithm. The depth images are filtered by a confidence threshold so that lower confidence values are rejected. Confidence for depth is computed by inverting the estimated noise of the depth value of each pixel. The filtered depth frames are then sent to another instance of the compression algorithm with slightly different parameters. For compression, the M-JPEG intra-frame compression method is utilized as it is simple and fast to set up. The JPEG-quality parameter for depth is set to 100 to lower compression noise as much as possible so that the point cloud, which will be eventually computed from depth, has as few outliers as possible. Since the depth images are lower resolution 224x168 and are formatted with 8-bit monocular encoding, the resultant bandwidth consumption is low enough for reliable transmission. As for the RGB images, the JPEG-quality parameter is set to 20 in order to match the bandwidth consumption of the depth images and thereby match its transmission reliability as well. Since the bandwidth consumption of the RGB images is slightly lower than that of the depth images, the transformed VIO measurements are attached to its header along with the RGB image's acquisition timestamp. As for the depth images, only the acquisition timestamp is added to the M-JPEG frames. Both RGB and depth M-JPEG streams are then sent to a frame rate controller, which takes in the time to send each frame as feedback and switches between a low or a high frame-rate value in order to smoothly control the data packets that are being sent to the aircraft's radio through the use of WebSockets. We chose to use the Transmission Control Protocol (TCP) even though it has higher overhead since we needed the transmission to be reliable and image frames to be intact without data-loss artifacts that usually are associated with User Datagram Protocol (UDP). Using UDP will allow for lower transmission overhead, but it will destroy the map structure when there are data-loss artifacts in depth images; hence, it's important to use TCP, at least for the purpose of transmitting depth images.

Offboard Autonomy

An offboard computer, being external to the primary system, enjoys considerably looser computational constraints compared to its onboard counterpart, which is typically constrained by size, weight, and power limitations. This freedom from strict computational boundaries opens up a realm of possibilities for more complex and resource-intensive tasks. However, despite this flexibility, managing real-time performance can become a delicate balancing act, particularly when dealing with demanding processes such as running large neural networks and rendering extensive map databases concurrently. One of the primary challenges arises from the fact that while the offboard computer may have ample computational resources, ensuring seamless real-time performance remains paramount, especially in critical applications like autonomous navigation. This becomes particularly apparent when attempting to run tasks such as processing large-scale point-cloud or mesh data alongside intensive neural network computations. Additionally, the architecture of some processes may not be optimized to fully leverage the computational power of modern multi-core CPUs, further complicating the real-time performance optimization process. Therefore, when designing an offboard autonomy framework, it becomes imperative to not only harness the computational power available but also to carefully monitor and manage memory consumption, CPU utilization, and GPU loads. Failure to do so can result in performance bottlenecks, leading to significant delays in processing and potentially compromising the real-time mapping accuracy essential for autonomous navigation systems. Despite these challenges, the relaxed computational constraints of the offboard system offer unique opportunities for integration with advanced middleware solutions such as the Robot Operating System (ROS). ROS provides a flexible and modular framework equipped with a vast array of off-the-shelf software components, enabling seamless integration and interoperability with various hardware and software components.

To bridge the gap between the onboard and offboard systems, a custom script AMAV Services [24] was developed to facilitate the transmission of data via a datalink radio. This script employs WebSockets to efficiently read and package data into individual ROS messages, effectively bringing the data into the ROS ecosystem. Within this framework, RGB and Depth image frames undergo rectification to correct for distortions caused by the intrinsic properties of their respective camera lenses. However, rectification requires different interpolation methods for RGB and depth images due to the inherent characteristics of the data. RGB images are rectified using linear interpolation, while depth images utilize nearest-neighbor pixel interpolation techniques to ensure accurate sensor data transformation. Following rectification, the RGB images are directly utilized for mapping and object detection purposes. Meanwhile, the depth images undergo a comprehensive filtering process to eliminate compression noise and outliers, ensuring the integrity of the data before transformation into a point-cloud format. Despite the diligent filtering process, the asynchronous nature of the sensors often results in discrepancies between the timestamps of RGB and depth images. To address this, a motion compensation algorithm leveraging Visual-Inertial Odometry (VIO) data is employed to align the point-cloud data with the acquisition time of RGB frames, ensuring temporal coherence. Once aligned, the point cloud is projected onto the RGB sensor frame using meticulously calibrated extrinsic transformation matrices, effectively registering the depth image onto the RGB camera's frame. This registration process yields a registered depth image, which serves as a critical component for accurate spatial perception. Following registration, an unguided depth completion process utilizing linear spatial interpolation is employed to fill in the gaps in the registered depth image, further enhancing its completeness and accuracy. The rectified RGB images, registered depth images, and timestamped VIO data are then fed into the Real-Time Appearance-Based Mapping (RTAB) [14] framework. RTAB utilizes this rich dataset to generate and store a comprehensive map representation, leveraging graph-based optimization techniques to refine the map structure whenever loop closure is detected.

In addition to the mapping pipeline, an autonomous localization framework has been developed to identify and tag objects of interest, such as humans and markers, in the environment. To achieve this, we leveraged a 2D mapping target created by the National Institute of Standards and Technology (NIST), specifically designed for evaluating camera systems' performance on UAVs. This target comprises a series of Landolt rings utilized for measuring visual acuity. Consequently, an object detection model was trained to detect these Landolt rings alongside humans to facilitate comprehensive scene understanding. For training the object detection model, we employed the You Only Look Once v8 (YOLOv8) [25] architecture. A semi-custom dataset was curated, containing samples of Landolt rings and persons extracted from the Common Objects in Context (COCO) dataset, a widely used dataset for object detection tasks. Diverse augmentation techniques were applied to the dataset to enhance the model's robustness to various environmental conditions, including changes in lighting, exposure, orientation, and occlusion. During the training process, 15% of the images in the dataset were randomly selected for validation purposes. The validation subset served to monitor the model's performance and prevent overfitting by halting the training process when the validation loss ceased to improve. This approach ensured that the model generalized well to unseen data and maintained robustness across different scenarios. At runtime, the trained model outputs 2D bounding box coordinates, detection confidence scores, and class identifiers for each detected object. Concurrently, a Multi-Object Tracking algorithm, known as ByteTrack [26], operates to assign unique IDs to individual objects in the scene, enabling robust tracking capabilities. The combined output from the object detection model and the Multi-Object Tracking algorithm is fed into the Localization module. This module employs a confidence threshold to filter out false positives and ensures that only detections with valid track IDs are considered, reducing spurious detections. Subsequently, the registered depth image is utilized to calculate the relative position of each pixel within the bounding box of the detected object. A clustering algorithm is then applied to extract foreground pixels, representing the object of interest. The mean position of these foreground pixels serves as the estimate of the object's relative position in the 3D space. Furthermore, a 3D bounding box is constructed by determining the extreme points of the foreground pixel set, providing a comprehensive spatial representation of the detected object.

B. Communications and Reliability Considerations

Radio DataLinks Hardware Investigation

Robotic systems operating in long-distance or indoor environments frequently contend with bandwidth limitations. An illustrative example involves implementing robotic solutions utilizing WiFi Halow [27] to navigate complex indoor settings, where conventional 2.4GHz or 5GHz WiFi frequencies often face obstruction from glazed glass or concrete walls commonly encountered in buildings. Wi-Fi HaLow, operating in the sub-GHz band for industrial IoT frequency range, delivers enhanced coverage and superior ability to penetrate obstacles such as walls, presenting significant advantages over traditional WiFi technologies. However, it is worth noting that one limitation of WiFi HaLow is its relatively lower data transfer rates compared to higher frequency WiFi technologies like 2.4GHz and 5GHz. The best wireless device we have identified is Microhard [28], which offers a maximum bandwidth of 8Mhz, which translates to a maximum observed throughput of 16Mbps in ideal scenarios. In most situations, we have achieved 14Mbps through careful selection of antennas. The throughput is reduced due to obstacles, range limitations, interference, antenna orientations, and antenna polarity.

Radio / Technology	Center Frequency	Bandwidth	Transmit Power	Data rate (81ft, LOS)	Data rate (341 ft, NLOS)	Cost (2 devices)
Wifi (802.11n)	2.4 GHz	20 MHz	20 dBm (0.1 W)	>20 Mbps	0 Mbps	<$100
Wifi Halow (802.11ah)	915 MHz	1-4 MHz	20 dBm (0.1 W)	1-3 Mbps	< 0.5 Mbps	$250
Doodle Mini Mesh Rider Radio	2.4 GHz	10-40 MHz	30 dBm (1 W)	>20 Mbps	0 Mbps	$3,800
Microhard Radio 2x2 MIMO pMDDL2450	2.4 GHz	8 MHz	30 dBm (1 W)	15-20 Mbps	0 Mbps	$1,400
Microhard Radio Dual-Band pDDL900	2.4 GHz	8 MHz	30 dBm (1 W)	10-15 Mbps	< 0.5 Mbps	$1,400
Microhard Radio Dual-Band pDDL900	915 MHz	8 MHz	30 dBm (1 W)	5 Mbps	2 Mbps	$1,400

Table 1 Performance of various data links tested in a semi-controlled environment. Non-line-of-sight (NLOS) testing involved obstruction by two glazed glass walls, which contributed to signal strength attenuation by about -60 dB.

Table 1 provides a comprehensive evaluation of six radio data links conducted within a semi-controlled environment. The testing setup involved placing the transmitter inside a room within a building while the receiver was positioned outdoors, replicating scenarios of Non-Line-of-Sight (NLOS) conditions typical for indoor Unmanned Aerial Systems (UAS). In this simulated environment, the challenge arose from the need for the transmitted signal to traverse two glazed glass walls upon exiting the building. These walls posed a significant obstacle, inducing higher attenuation compared to reinforced concrete, approximately -30 dB per wall, thus resulting in a cumulative -60 dB attenuation solely from the walls themselves.

Data throughput measurements were obtained from two distinct locations: one situated at an 81 ft distance from the transmitter, benefiting from line of sight through a single glazed wall, and the other positioned at a 341 ft distance from the transmitter, not in line of sight, and contending with two glazed glass walls alongside various sparse obstacles such as trees, poles, and pedestrians. Remarkably, observations revealed the inadequacy of 2.4GHz radios, including low-power Wifi, in maintaining a stable connection at the distant 341 ft NLOS testing location. Among these, only the Microhard pDDL900 operating at 2.4 GHz managed to transmit a limited amount of data, albeit at a minimal rate. This disparity in performance was attributed to the design of radios such as Doodle Mini and Microhard 2x2 MIMO, which employ dual antennas, consequently necessitating a reduction in RF power per antenna to adhere to FCC regulations stipulating a total Equivalent Isotropic Radiated Power (EIRP) of less than 1 Watt. In contrast, radios operating at 915 MHz, namely Wifi Halow and Microhard pDDL900, exhibited the capability to sustain a robust connection since they used single antennas only. However, Wifi Halow demonstrated significantly reduced throughput owing to its lower RF output power, which is ten times less than that of Microhard and its maximum frequency bandwidth being half of that of Microhard. Notably, the pDDL900 radio, equipped with stock antennas, achieved the highest throughput of 2Mbps among all radio data links at the challenging 341 ft distance in NLOS conditions, thereby solidifying its selection for integration into the UAS design. Subsequent enhancements were made to the pDDL900 by upgrading its antennas to a superior omnidirectional variant on the aircraft and a high-gain directional antenna on the Ground Unit Radio, both boasting superior Voltage Standing Wave Ratio (VSWR) characteristics compared to the stock antennas. Additionally, adjustments were made to reduce the RF power on both the Air and Ground Unit Radio Modules to ensure compliance with FCC requirements. Further experimentation revealed that the radios perform optimally at a 27 dBm power setting, yielding a maximum throughput of around 14Mbps under favorable environmental conditions. These findings underscore the importance of meticulous evaluation and strategic optimization in enhancing the performance and reliability of radio data links in challenging operational scenarios.

Software Communication Configuration

ROS has gained popularity in robotics for several reasons, such as open source, modularity, and large community support. Many common packages have been developed by the community. The other key feature of ROS is its communication infrastructure, which enables different components of a robotic system to exchange data seamlessly. However, the middleware layer in ROS introduces overhead compared to direct socket communication. The overhead can impact performance, especially for real-time or latency-sensitive applications. When operating in bandwidth-limited conditions, we have noted that ROS fails to transmit any messages from VOXL2, flashed with System Image 1.1 from Modal AI, when the network bandwidth drops below 8Mbps.

To facilitate ROS-based robotics operating in bandwidth-limited environments, our approach involves having UAS directly transmit data to the ground station using socket communication. The ROS messages are reconstructed on the ground station for ROS-based applications. Table 2 compares the pros and cons of the socket communication and ROS. Socket communication offers a range of advantages. Firstly, its versatility transcends specific application domains, making it adaptable for various purposes beyond robotics. Secondly, developers wield precise control over the communication process, managing aspects such as data encoding, transmission, and error handling. Additionally, its language-agnostic nature facilitates interoperability across systems coded in different programming languages. Lastly, due to its direct nature, socket communication tends to operate with lower overhead than higher-level frameworks like ROS, thus proving efficient for performance-critical scenarios. Socket communication also presents several challenges. Firstly, implementing it from scratch can be difficult, particularly when dealing with low-level intricacies like network protocols and data serialization. Moreover, socket communication lacks high-level abstractions for prevalent robotic tasks such as sensor processing, robot control, and message passing, leading to increased complexity in development. Furthermore, there is a dearth of tools and libraries tailored for socket-based communication compared to ROS, potentially elongating development timelines and intensifying the effort required. Lastly, managing edge cases and error conditions in socket communication necessitates meticulous programming to uphold robustness and reliability, posing a considerable risk for errors.

In environments where bandwidth is a precious resource, robotics finds a niche application in deploying indoor Unmanned Aerial Systems (UAS) for intricate 3D mapping tasks. These environments, often characterized by obstacles like glazed glass or concrete walls commonly found in buildings, present formidable challenges to establishing reliable communication networks required for mapping with an onboard and a remote ground station computer. The Microhard 915 MHz Data Link offers a promising solution in such communication-hostile environments. Operating within the sub-GHz frequency range, it provides extended coverage and improved penetration through obstructions, making it

	Socket Communication	**ROS**
Pros	Little overhead Low level control Efficiency	High-level abstraction Visualization and debugging features Community support
Cons	Lack of abstraction Limited features Error prone	Resource intensive Overhead Limited flexibility

Table 2 Performance comparison of socket communication and ROS

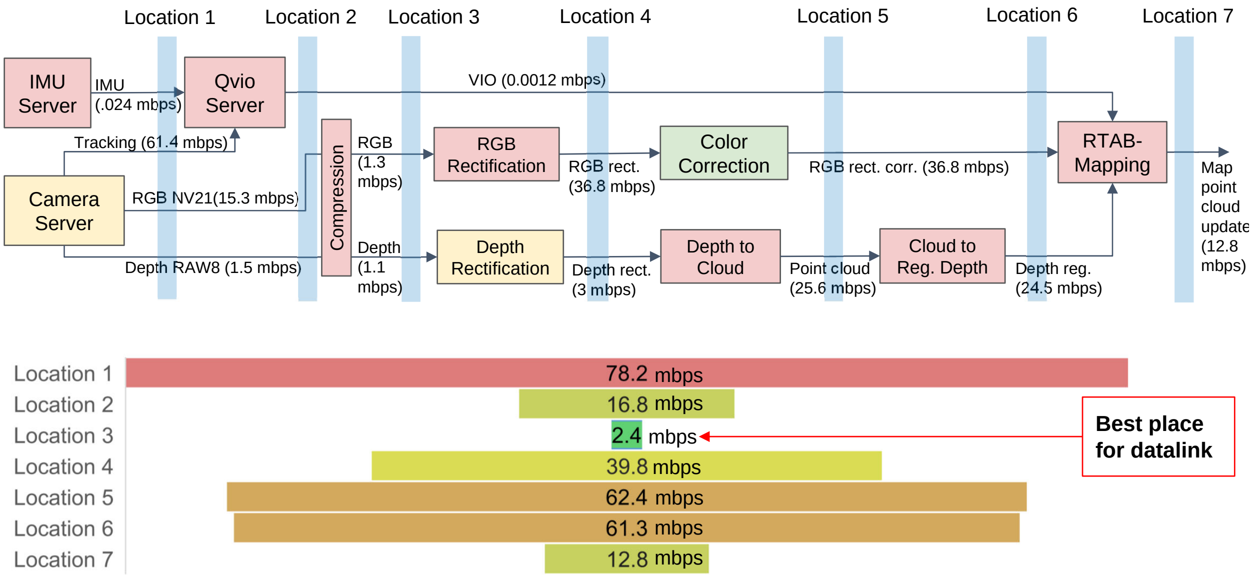


Fig. 4 Autonomy data rate diagram

particularly well-suited for navigating the complexities of indoor environments. Despite its advantages, the Microhard 915 MHz Data Link has restricted bandwidth. This limitation necessitates a careful and strategic approach in determining which types of data should be prioritized for transmission from the onboard computer to the offboard ground station computer. Every bit of data sent must be carefully considered to ensure that essential information is conveyed efficiently while minimizing unnecessary data transfer. To aid in this decision-making process, a flattened and simplified autonomy block diagram, as depicted in Figure 4, serves as a visual representation of the message data rates between various blocks in the system. By analyzing the aggregated data rates at seven different locations and plotting them, as demonstrated in the figure, we can pinpoint Location 2 as having the minimum data rate. This discovery highlights Location 2 as the optimal site for establishing data link communication, as it offers a substantial bandwidth margin compared to other locations. Additionally, it inadvertently also ensures low computational load on the aircraft's onboard computer, which is beneficial for long-term reliable operations. This strategic decision significantly enhances the reliability and robustness of the overall framework, enabling seamless operations even under challenging conditions such as extreme ranges and communication-hostile environments. It ensures that the UAS can effectively navigate through complex indoor spaces, capturing and transmitting essential data for 3D mapping purposes without being hampered by bandwidth constraints.

In our pursuit of accurate 3D mapping, we rely on RTAB-Map [29]. This powerful toolset requires the publication of critical data streams, including point cloud, RGB image, and camera pose data, via ROS topics, with a minimum frequency requirement of 1 Hz. The camera pose data, in particular, plays a crucial role in the mapping process. It is computed by applying a fixed transformation to the computed pose estimates from Visual-Inertial Odometry (VIO).

This fixed transformation, determined during the extrinsic calibration procedure, ensures the alignment of the camera's viewpoint with the UAS's position and orientation, facilitating accurate and consistent mapping results.

Table 3 provides a breakdown of the topics reconstructed on the ground station, along with the associated message sizes transmitted via sockets. For optical RGB imaging, we utilize the Starvis IMX412 sensor [22]. The RGB image resolution is 640 x 480 pixels, with a raw image size of 921.6 KB, which can be reduced using M-JPEG compression. The original point cloud consists of 40,000 points encoded with mono16, requiring 2 bytes per point, resulting in approximately 80 KB per message or a transmission rate of 640 Kbps when messages are sent at 1 Hz. Alternatively, we reconstruct the point cloud from depth images captured by the Modal AI VOXL Time of Flight (ToF) Depth Sensor [23]. The depth image resolution is 224 x 171 pixels, with a raw size of 38 KB. Pose data from Visual Inertial Odometry (VIO) includes position (x, y, z) and orientation in quaternions (qx, qy, qz, qw), with each double consuming 8 bytes, totaling 56 bytes. Additionally, VIO quality, represented as an integer between 0 and 100, is included, computed using the inverse of the largest diagonal element of the covariance matrix. To ensure optimal mapping quality, all topics must be synchronized. Therefore, when transferring data from VOXL2 to the ground station via sockets, timestamps (27 bytes) for VIO, RGB image, and depth image are included. This facilitates accurate reconstruction of ROS messages on the ground station. VIO data, VIO quality, and timestamps are concatenated with the images for transmission through the same socket. Socket communication allows precise selection and control over the transmitted data, facilitating efficient transmission for 3D mapping with less than 3 Mbps bandwidth. RGB images are compressed to less than 30 KB, and depth images to less than 35 KB. Depth images are less compressed to preserve their quality, improving the fidelity of reconstructed point clouds. The source code for serializing the data on VOXL2 can be found in the repository [30], while the repository for deserializing the data to reconstruct topics on the ground station is available [24]. Despite the memory-intensive nature of 3D mapping and the tendency of existing mapping apps on smartphones to crash due to memory shortages, our communication setup enables RTAB-Map to operate on a laptop with fewer memory constraints.

Topic	**Message Size**	**Bandwidth Usage (1 Hz)**	**Bandwidth Usage (5 Hz)**
RGB image	15 - 30 KB	120 - 240 Kbps	0.6 - 1.2 Mbps
Depth image	15 - 35 KB	120 - 270 Kbps	0.6 - 1.35 Mbps
Odometry	56 B	448 bps	2.24 Kbps
Odometry quality	2 B	16 bps	80 bps
Timestamp	27 B	216 bps	1.08 Kbps
Total	30 - 65 KB	240 - 510 Kbps	1.2 - 2.55 Mbps

Table 3 Size of messages per topic for 3D mapping. Raw images undergo M-JPEG compression for size reduction.

C. Target Detection and Localization

A custom object detection model was developed to detect both humans and Landolt rings. Figure 5 shows the training workflow used to develop the model. The YOLOv8 model was trained with a combined dataset consisting of a persons-only COCO dataset and a custom annotated Landolt rings dataset. We also included additional rotation, blur, and noise augmentation to the images. Without rotation augmentation, the trained model was biased towards detecting upright humans since the COCO person-only dataset mostly consists of humans in an upright posture. Hence, by applying rotation augmentation, the final trained model is more resilient and robust to the different orientations that the UAV might encounter a victim in during search and rescue scenarios. Post-training, we evaluated the model on a smaller test dataset as well as during live testing to ensure that the model detected persons and Landolt rings in various combinations of lighting, orientation, and environments.

The autonomous target localization system utilizes the custom YOLOv8 model trained to detect humans and Landolt rings as well as the registered depth map to find the 3D position and 3D bounding box of a desired target or person in the UAS's frame, respectively. Figure 6 shows an example of these images.

A rectified RGB image is passed through the custom YOLOv8 model, which outputs the provided RGB image with a bounding box around the detected objects as well as a detection message containing an identification number that corresponds to the class of the object, the coordinates for the corners of the bounding box, and a tracking number

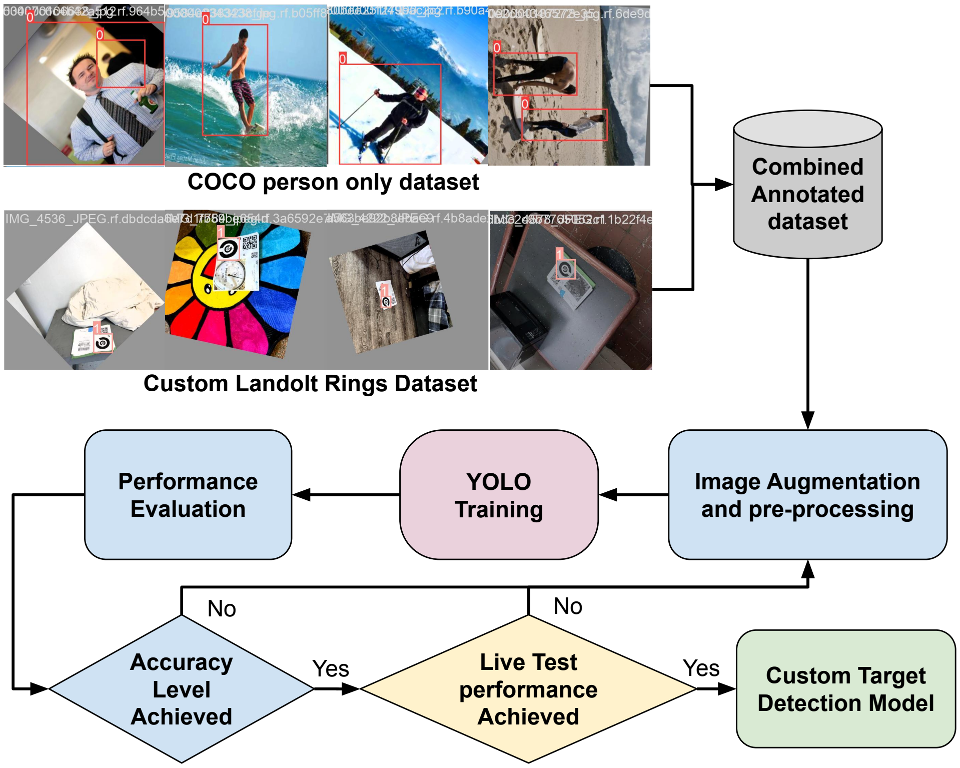
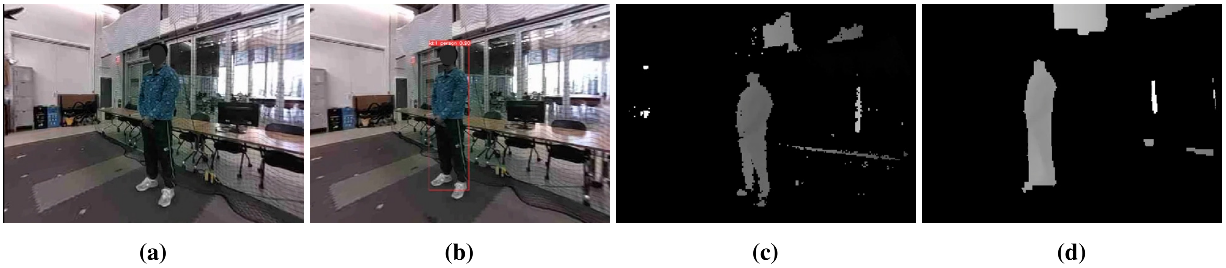

COCO person only dataset
Custom Landolt Rings Dataset
Combined Annotated dataset
Image Augmentation and pre-processing
YOLO Training
Performance Evaluation
Accuracy Level Achieved
No
Yes
Live Test performance Achieved
No
Yes
Custom Target Detection Model

Fig. 5 Workflow describing the training process for the custom YOLOv8 model.



(a) (b) (c) (d)

Fig. 6 (a) Rectified RGB image, (b) Custom YOLOv8 output, (c) Raw depth image, (d) Registered depth image

if the object has been tracked across multiple frames. Track numbers are used to remove spurious detections, thus only detected targets with track numbers are localized. The bounding box information from the detection message of the tracked target is then used to determine the pixels defining the target's location in the rectified image. These pixel locations, in conjunction with the pinhole camera model, are used to extrapolate the point cloud corresponding to the target's pixels from the registered depth map. The 3D bounding box is determined by finding the minimum and maximum values along the three axes of the point cloud. This process is repeated for each tracked target in each detection message. Figure 7 shows 3D bounding boxes generated from the localization pipeline during an experiment with 2 human actors in the environment.

D. User Interface Development

Although there are many free, open-source tools and proprietary software available for viewing mesh models generated by RTAB-Map, very few of them allow users to adjust map qualities. Additionally, very few map viewers include features for rendering target locations and detection images. We have developed a custom user interface in React

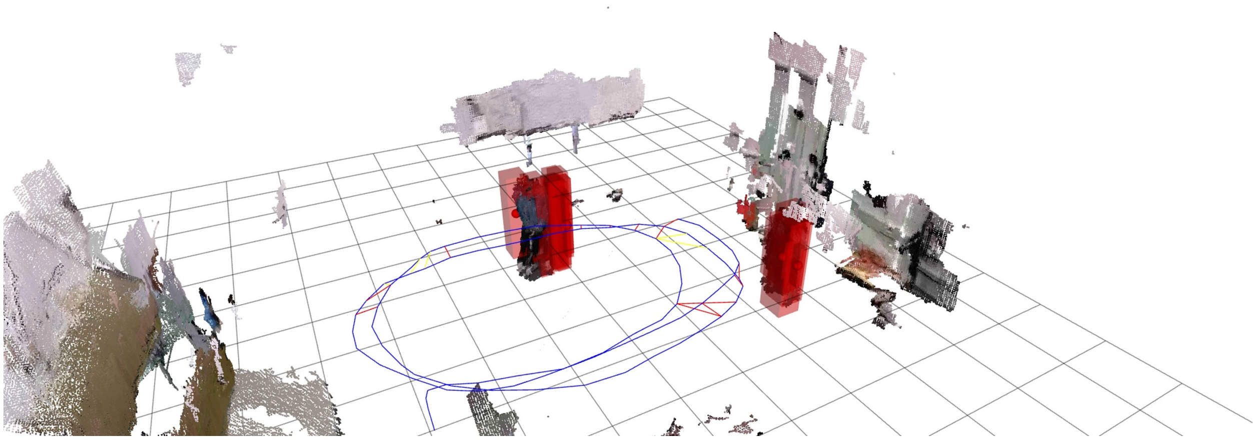


Fig. 7 3D bounding boxes generated from the target localization pipeline.

and Spring Boot to render high-quality maps with additional features, offering enhanced situational awareness to first responders. React, a popular open-source JavaScript library for building client-side components, enables developers to quickly create modular UI components [31]. Spring Boot, a framework for building Java-based enterprise applications, empowers developers to quickly set up and deploy HTTP endpoints [32].

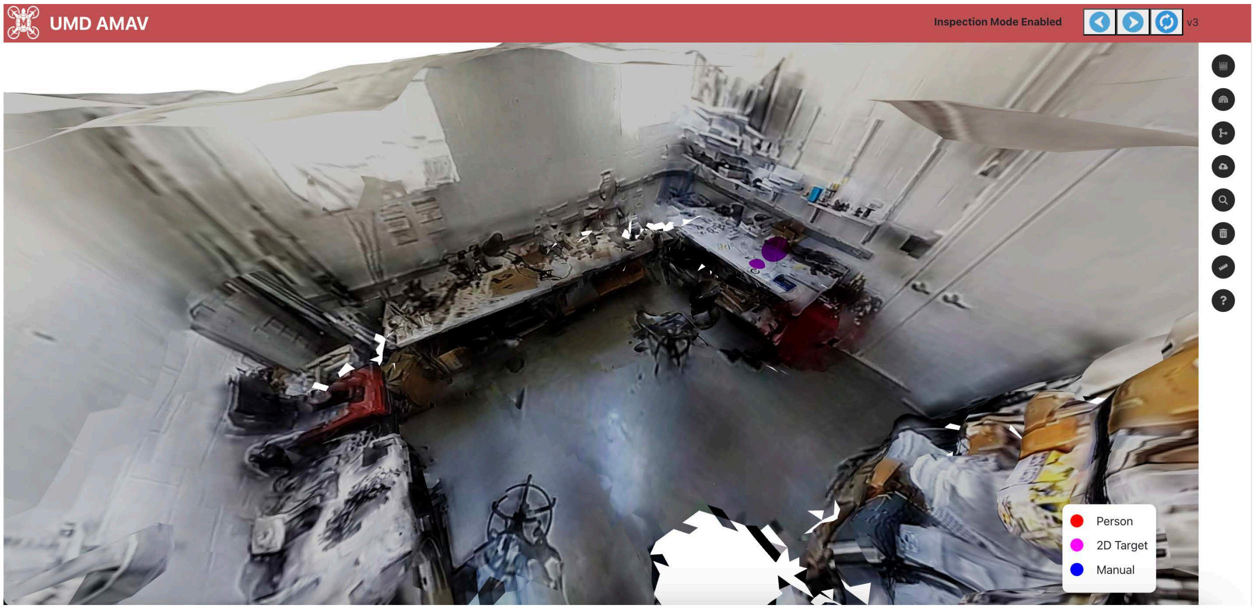
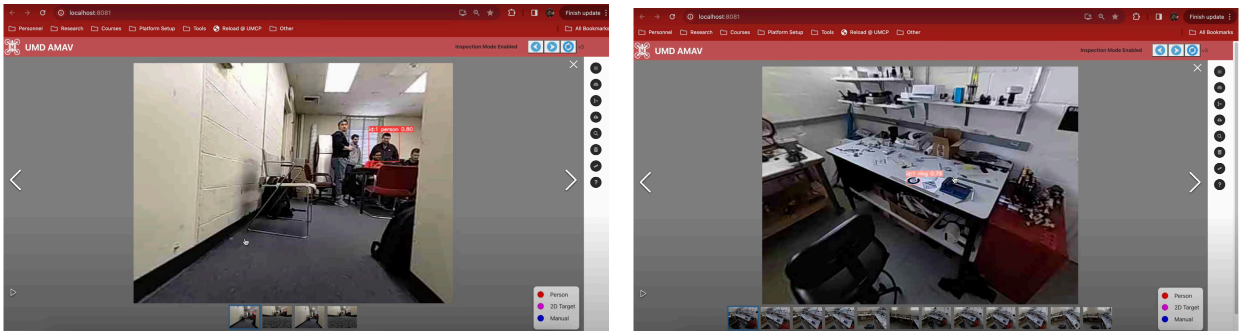


Fig. 8 3D mesh model with targets: person (red), Landolt "C" ring (magenta), and manual (blue)

Figure 8 showcases a post-processed mesh model generated by RTAB-Map in the UI, leveraging the data transmission facilitated by the inventive communication configuration. Target locations relative to the camera positions are saved in the RTAB-Map database via zlib compression during real-time mapping processes [33]. The locations are extracted and transformed to the inertial frame, rendering the colored dots that indicate target locations on the map. The detection images can be inspected by clicking on the dots as shown in Figures 9a and 9b.

Our custom user interface also supports mesh models generated by AliceVision for improved rendering quality. AliceVision is an open-source Photogrammetric Computer Vision framework that provides tools for creating 3D reconstructions from 2D images to generate high-quality realistic 3D models [34]. Figure 10 illustrates a comparison

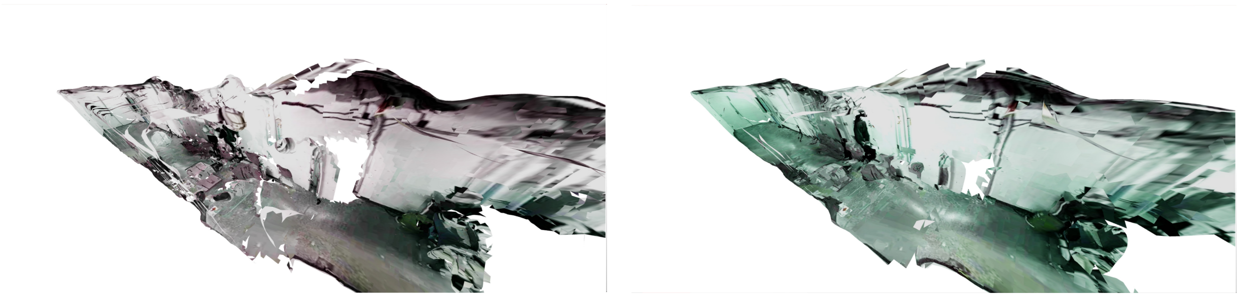


(a) Detected persons

(b) Detected Landolt "C" ring

Fig. 9 Target images can be displayed in a gallery view by clicking on the colored dots on the map

between models generated with and without AliceVision. In Figure 10a, noticeable features include larger gaps, less smooth texture, and misalignment of straight lines. Conversely, Figure 10b demonstrates a significantly enhanced model quality. Additionally, a person standing in the corridor is visible in the 3D map.



(a) Default model without AliceVision support.

(b) Improved model with AliceVision support.

Fig. 10 Comparison of mesh model generated with and without AliceVision support.

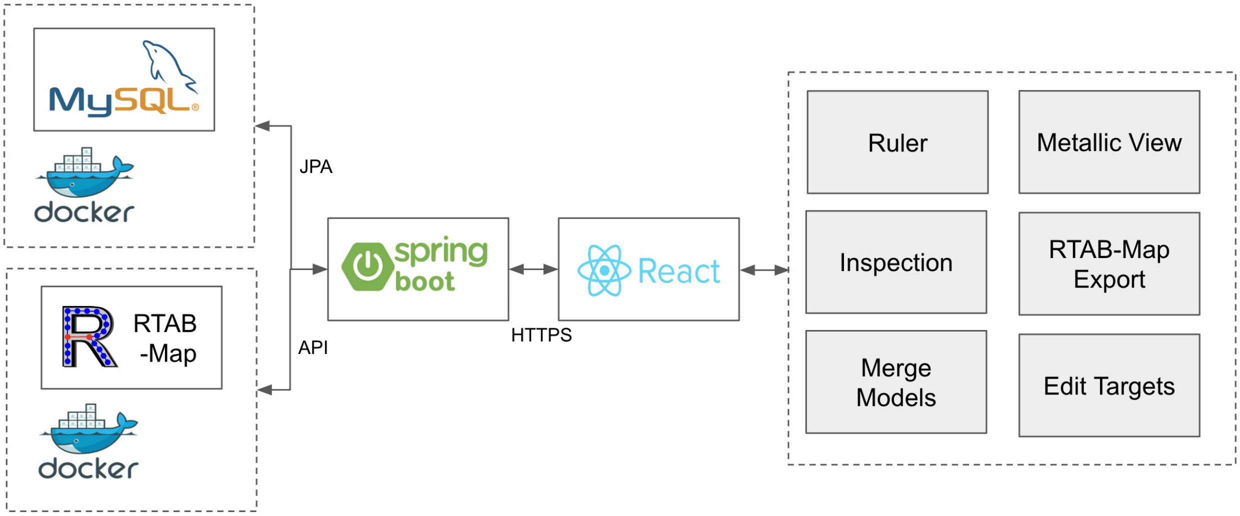


Fig. 11 Technologies utilized and features developed in custom UI

In Figure 11, the gray boxes are the features developed in the UI. Additionally, Hypertext Transfer Protocol Secure (HTTPS) is an extension of the HTTP protocol used for secure communication between the React front end and Spring

Boot back end. Docker is a popular technology used for containerization, enabling developers to package applications and their dependencies into isolated, lightweight containers that can run consistently across various environments. Both MySQL and RTAB-Map are dockerized to enhance consistency, enabling fast development and testing. The Spring Boot back end utilizes the Java Persistence API (JPA) to communicate with MySQL, enabling developers to interact with relational databases using Java objects. Additionally, the back end accesses RTAB-Map and its associated database via API calls.

E. Hardware Design

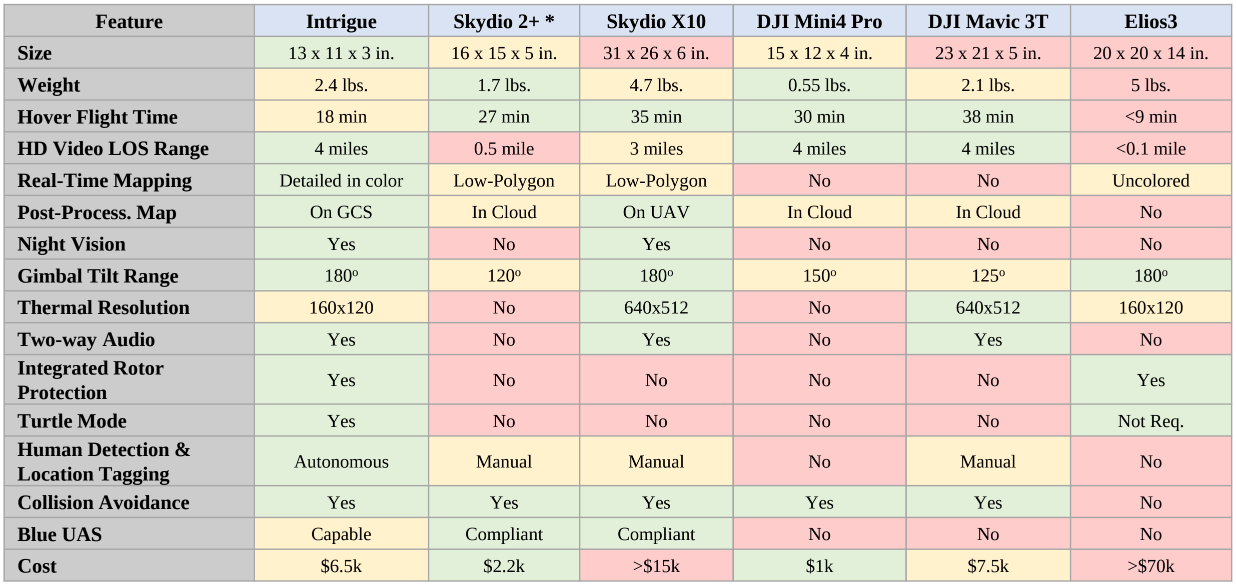
UAS Design Objectives

The UAS was designed and manufactured with a comprehensive vision: not just to validate a newly formulated mapping framework but to elevate its utility to the level of established commercial solutions for indoor Search and Rescue (SAR) operations. This ambitious goal necessitated meticulous attention to several key factors. Foremost among these was the importance of achieving a high level of reliability and real-time performance for both the mapping framework and the detection and localization of targets within indoor environments. Concurrently, the UAS has to adhere to strict size and weight limitations, ensuring nimble maneuverability within confined spaces. Complementing these core objectives was the integration of a diverse suite of onboard sensors tailored to the unique demands of indoor SAR missions. The inclusion of high-definition (HD) First Person View (FPV) video capability provides operators with clear, immersive visuals essential for navigating complex indoor layouts with precision and confidence. Meanwhile, a Night Vision-capable camera system provides critical support for operations in environments with limited or zero illumination, ensuring SAR efforts can continue unhindered by darkness. Additionally, thermal imaging technology assists in detecting human body heat signatures, facilitating swift localization of individuals in need of rescue, and aiding in the identification and characterization of fires, a common hazard in emergency situations. Beyond sensor technology, the UAS boasts a suite of advanced functionalities expressly designed to enhance its effectiveness in SAR missions. A two-way audio system enables direct communication with victims, fostering a vital lifeline of reassurance and information exchange in the midst of a crisis. The inclusion of a full 180° pitch-range gimbal augments situational awareness by affording comprehensive views of the surrounding environment, enabling operators to swiftly assess and respond to dynamic circumstances. Furthermore, the incorporation of turtle/auto-flip mode ensures mission continuity in the event of an unforeseen mishap, minimizing downtime and maximizing operational efficiency. Moreover, to safeguard both the UAS and its surroundings, a simple collision avoidance capability is useful, serving as a critical safety net against accidents that could jeopardize rescue efforts. Finally, it's important to keep the total cost low and affordable for first responders and to also comply with Blue UAS requirements to make the UAS available to DoD and Federal Government partners. Keeping these objectives in focus, we developed a competitive UAS named *Intrigue*, and its specifications are presented in Figure 12 and are compared with those of existing top-tier commercial SAR UAVs.

UAS Overview

An external overview of the final design iteration of the Intrigue UAS is shown in Figure 13. Locations of major individual hardware components are labeled with red arrows. The UAS structure comprises three CNC-milled carbon fiber plates connected together using aluminium standoffs and steel fasteners, creating a rigid, strong, and durable frame while keeping material and manufacturing costs low. The bottom and middle plates are 4 mm thick since they are the primary load-bearing structure, and the top plate is 3 mm thick to aid in the airframe's rigidity. Additionally, covers to protect the electronics and also to aid in cooling were 3D printed using ABS (acrylonitrile butadiene styrene) material. The gimbal structure was also 3D printed using the same material. Appropriate mounting holes were strategically drilled on the carbon fiber plates to securely mount the avionics components. Vibration isolation standoffs were used to mount the flight controller and the onboard computer. Antennas were strategically mounted to place them as far away from the conductive frame as possible without adding too much weight from the co-axial extension cables/rods. Additionally, 35V 1000uF and 10V 220uF capacitors were integrated into the power source for ESC and Microhard radio, respectively. Capacitors attenuate voltage spikes and electrical noise generated from rapid motor acceleration, active motor braking, and switching regulators.

A block diagram of the complete set of avionics hardware on the Intrigue UAS is presented in Figure 14. The arrows represent wired connections between the components and the direction where the majority of the data goes from one component to another. The dashed lines and a wireless icon on a block represent wireless communication. The colors represent individual avionics sub-groups, i.e., mapping, flight control, FPV video, control radio, gimbal, audio, thermal,



Feature	Intrigue	Skydio 2+ *	Skydio X10	DJI Mini4 Pro	DJI Mavic 3T	Elios3
Size	13 x 11 x 3 in.	16 x 15 x 5 in.	31 x 26 x 6 in.	15 x 12 x 4 in.	23 x 21 x 5 in.	20 x 20 x 14 in.
Weight	2.4 lbs.	1.7 lbs.	4.7 lbs.	0.55 lbs.	2.1 lbs.	5 lbs.
Hover Flight Time	18 min	27 min	35 min	30 min	38 min	<9 min
HD Video LOS Range	4 miles	0.5 mile	3 miles	4 miles	4 miles	<0.1 mile
Real-Time Mapping	Detailed in color	Low-Polygon	Low-Polygon	No	No	Uncolored
Post-Process. Map	On GCS	In Cloud	On UAV	In Cloud	In Cloud	No
Night Vision	Yes	No	Yes	No	No	No
Gimbal Tilt Range	180^{o}	120^{o}	180^{o}	150^{o}	125^{o}	180^{o}
Thermal Resolution	160x120	No	640x512	No	640x512	160x120
Two-way Audio	Yes	No	Yes	No	Yes	No
Integrated Rotor Protection	Yes	No	No	No	No	Yes
Turtle Mode	Yes	No	No	No	No	Not Req.
Human Detection & Location Tagging	Autonomous	Manual	Manual	No	Manual	No
Collision Avoidance	Yes	Yes	Yes	Yes	Yes	No
Blue UAS	Capable	Compliant	Compliant	No	No	No
Cost	$6.5k	$2.2k	>$15k	$1k	$7.5k	>$70k

* No longer in Production

Fig. 12 Comparision of Intrigue UAS with top-tier commercial search and rescue UAS

and illumination systems. The Ground Station Laptop can be any generic laptop running the Ubuntu 20.04 Operating System configured with ROS noetic, the latest stable Rtabmap, and compatible AliceVision software packages. We have successfully tested the software framework on multiple laptops, but for the purpose of this manuscript, we used the Gigabyte Aero 15 XC Laptop equipped with a 10th Gen Intel Core i7-10870H (2.2GHz) CPU, 16 GB DDR4 RAM, and NVIDIA GeForce RTX 3070 Laptop GPU with 8 GB GDDR6 VRAM.

Multispectral Visual-Inertial Sensor System

A gimbaled multispectral visual-inertial sensor system was developed to provide complete freedom for performing mapping from any desired pose. The gimbal provides pitch axis control and houses the FPV low-light camera, RGB camera, ToF camera's IR emitter and receiver system, and two IMUs. One of the IMUs is used by the gimbal controller to stabilize and control the gimbal's pitch angle, whereas the other IMU is used by the onboard computer to calculate the poses of individual sensors in the camera system.

Avionics Cooling Considerations

Some avionics components inside the UAS generate a considerable amount of heat, and hence, it is imperative to integrate proper cooling measures in the design to prevent damage and improve runtime performance and stability. Appropriately sized aluminum heat sinks were installed on the Microhard Radio, VOXL2, and the FPV video transmitter. Ambient air is also forced inside the avionics bay with the help of a 40 mm and a 20 mm fan mounted on the back and top side of the UAS respectively. Without active cooling from fans, we observed a maximum temperature of 93°C on the VTX, 79°C on the Microhard Radio, 71°C on VOXL2, and 45°C on the Gimbal motor during benchtop testing. With active cooling from the dual fans, the temperatures were brought down to 70°C on the VTX, 42°C on the Microhard Radio, 44°C on VOXL2, and 34°C on the Gimbal motor. Figure 15 shows a side view of the active vs cooling performance of the UAV with a thermal heatmap overlaid. The heat from the internal avionics heats up the 3D-printed side walls; the outer surface temperature is plotted in the images.

IV. Experimental Testing Results

RTAB-Map outputs map data into a database file (.db) format, which requires post-processing to generate the final textured map files. This includes the conversion to a wavefront geometry (.obj) file, material library (.mtl) file, and texture image (.jpg) file. Figure 16 illustrates the two primary post-processing steps employed to enhance the fidelity of

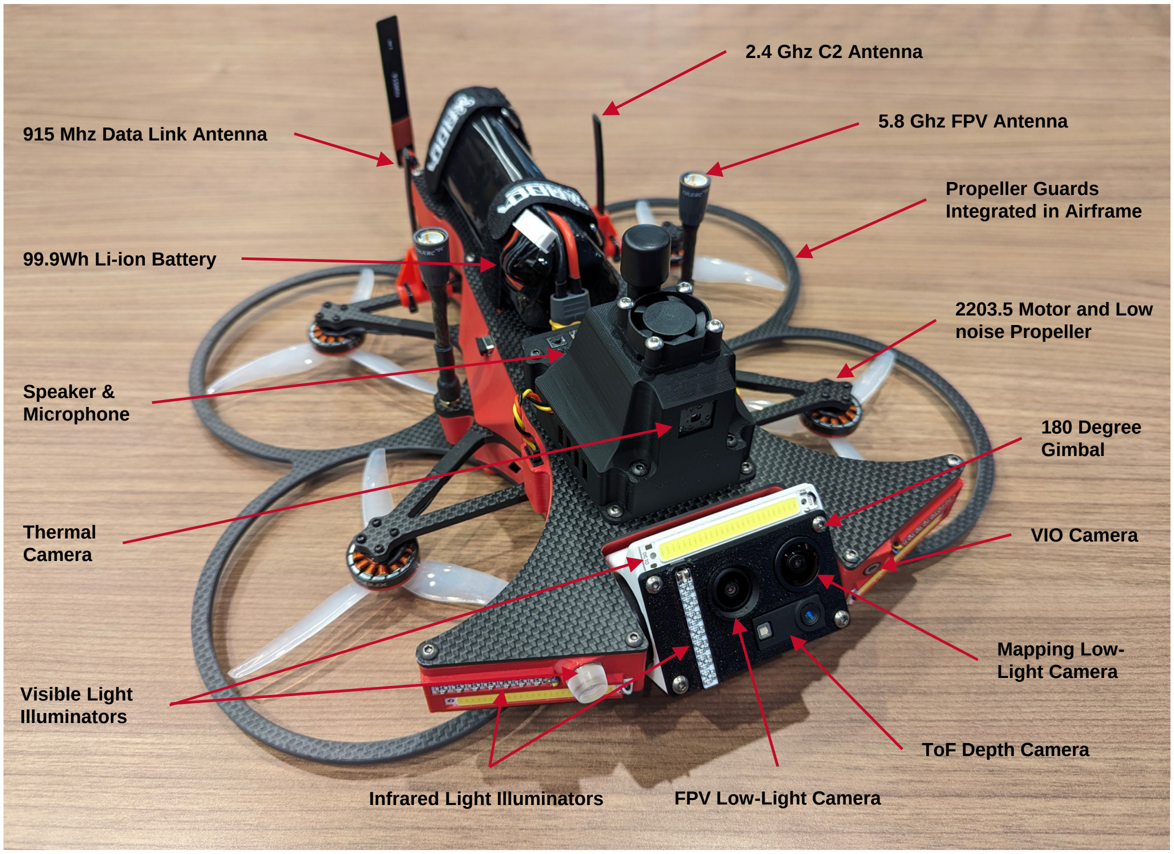


Fig. 13 Intrigue UAS overview

the mesh map. For the experiment, the Intrigue UAS flew in a straight line within a narrow corridor designated for NIST's UAS 3D Mapping Lane Test [35]. The corridor featured fiducial foam structures for measuring map dimensional accuracy, foam cubes on walls and floors to assess captured shape details, and NIST ring targets on letter paper affixed to the wall to evaluate visual acuity. Figure 16a displays the original mesh map without any post-processing. In contrast, Figure 16b exhibits the mesh map after the bundle adjustment procedure. Bundle adjustment refines the map's shape and texture by minimizing the re-projection error of landmark features across the camera pose-graph data structures. This refinement enhances overall accuracy. The final step, depicted in Figure 16c, involves Multiband Blending, a feature provided by AliceVision [36], an open-source Photogrammetric Computer Vision tool. This process eliminates edges between adjacent texture patches and corrects texture matching to individual mesh cells, resulting in increased sharpness and resolution across the mesh map.

The development of the Intrigue UAS underwent meticulous testing across various simulated indoor missions to ensure optimal performance of both its hardware and software components in real-world scenarios. These missions served as vital proving grounds, enabling the team to assess the UAS's capabilities and refine its functionalities. A primary focus during these missions was achieving real-time mapping, target detection, and localization visualizations. To accomplish this, tools like Rviz and Rtabmapviz were utilized, providing valuable insights into the UAV's surroundings for swift decision-making and course corrections as needed. Furthermore, a custom-designed user interface (UI) was developed to comprehensively analyze mission outcomes, as detailed in Section III.D. This UI served as a platform for post-processing analysis, presenting collected data in a gallery-style format. Through this interface, team members could review detailed mesh representations, pinpoint target locations, and assess the accuracy of target detections. A critical aspect of the evaluation process involved comparing different rendering techniques. Figure 17 illustrates one such comparison, focusing on real-time map rendering details. By presenting the same scene side by side, one with

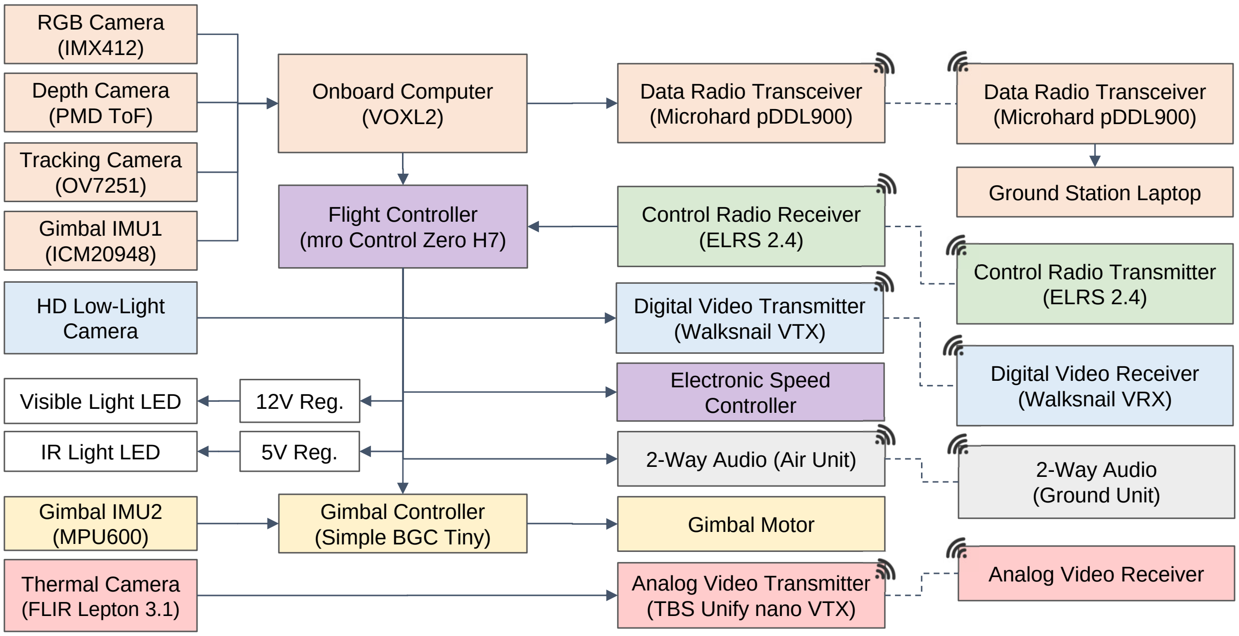
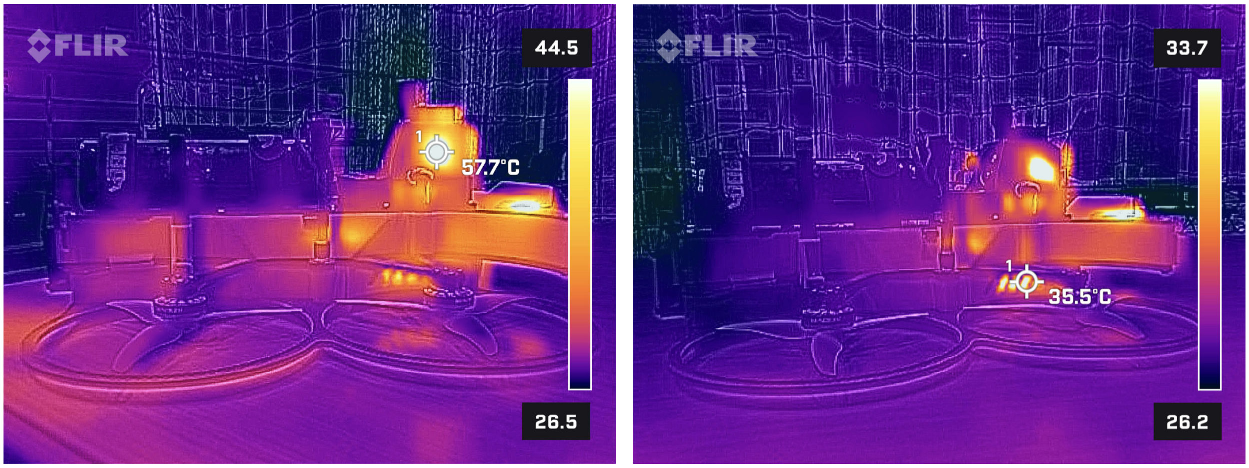

RGB Camera (IMX412)
Depth Camera (PMD ToF)
Tracking Camera (OV7251)
Gimbal IMU1 (ICM20948)
HD Low-Light Camera
Visible Light LED
IR Light LED
Gimbal IMU2 (MPU600)
Thermal Camera (FLIR Lepton 3.1)
Onboard Computer (VOXL2)
Flight Controller (mro Control Zero H7)
12V Reg.
5V Reg.
Gimbal Controller (Simple BGC Tiny)
Data Radio Transceiver (Microhard pDDL900)
Control Radio Receiver (ELRS 2.4)
Digital Video Transmitter (Walksnail VTX)
Electronic Speed Controller
2-Way Audio (Air Unit)
Gimbal Motor
Analog Video Transmitter (TBS Unify nano VTX)
Data Radio Transceiver (Microhard pDDL900)
Ground Station Laptop
Control Radio Transmitter (ELRS 2.4)
Digital Video Receiver (Walksnail VRX)
2-Way Audio (Ground Unit)
Analog Video Receiver

Fig. 14 UAS avionics hardware block diagram



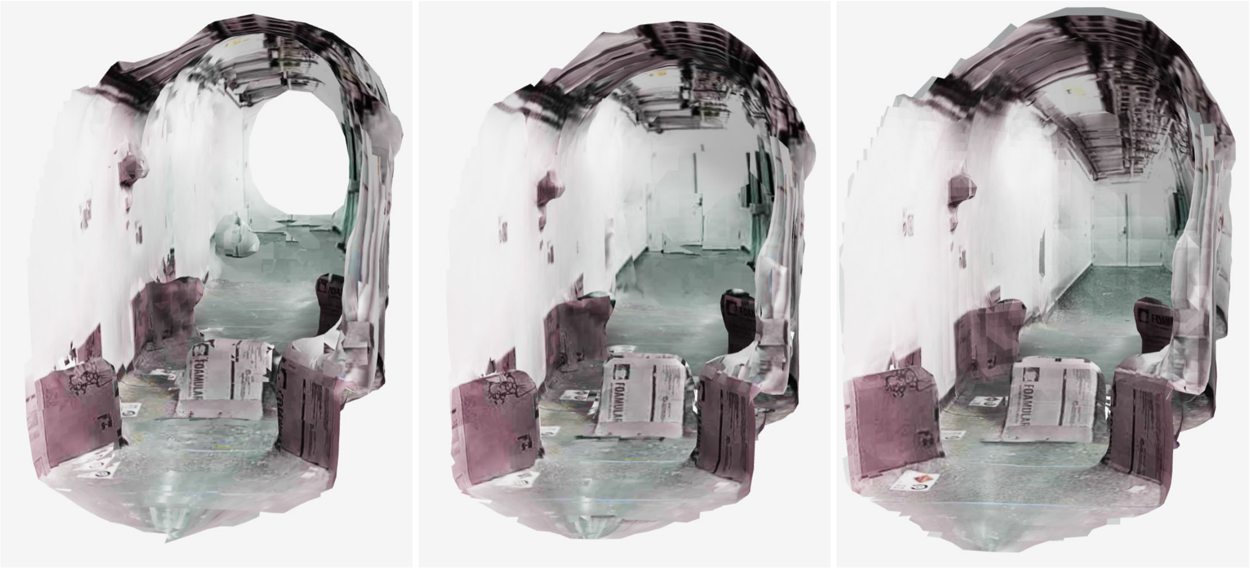
(a) Passive cooling

(b) Active and passive cooling

Fig. 15 Avionics cooling performance during benchtop tests

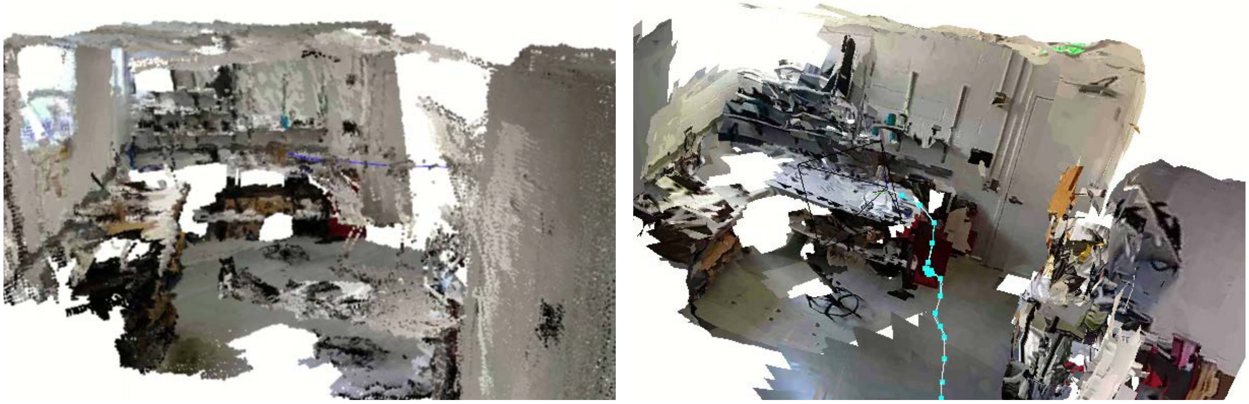
the display of point-cloud data and the other with the application of the FastMesh algorithm, the figure highlights the significant enhancement in textural detail achieved through the latter approach. The FastMesh procedure, detailed in FastMesh [37], involves connecting neighboring points (pixels in depth image space) to construct a quad mesh. This process enhances the visual fidelity of the mapped environment, providing a more detailed picture of the surroundings and objects within it. However, in our current software state, localized targets and objects in the map cannot be displayed in RTAB-Map Viz. Therefore, we utilize RViz to show the map point-cloud and target locations in real-time, while the custom-developed UI displays the final post-processed mesh map along with the target locations.

The Intrigue UAS underwent a rigorous evaluation in a simulated indoor search and rescue mission orchestrated by the National Institute of Standards and Technology (NIST) as part of the NIST UAS 5.0 First Responder UAS 3D Mapping Challenge. This mission was designed to assess the UAS's mapping capabilities as well as detecting



(a) Before post-processing (b) After bundle adjustment (c) After multiband blending

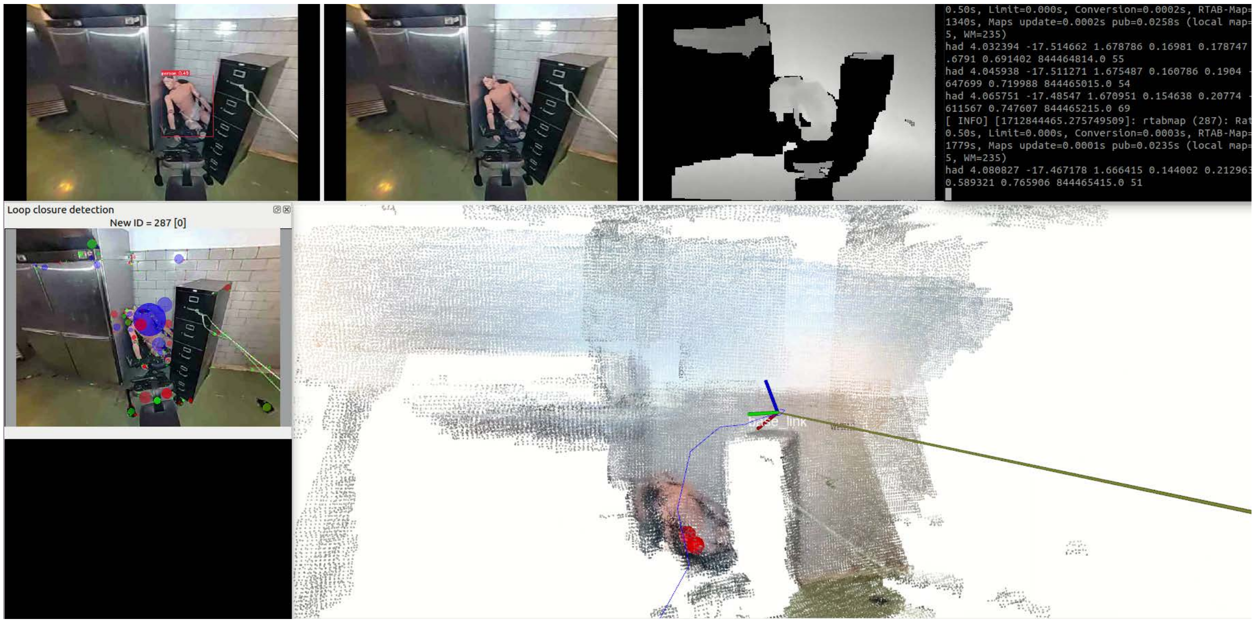
Fig. 16 Post-processed output map after each stage.



(a) Real-time map rendered as a point cloud by RViz. (b) Real-time FastMesh generation in RTAB-Map Viz.

Fig. 17 Comparison of real-time point-cloud and real-time mesh map rendering.

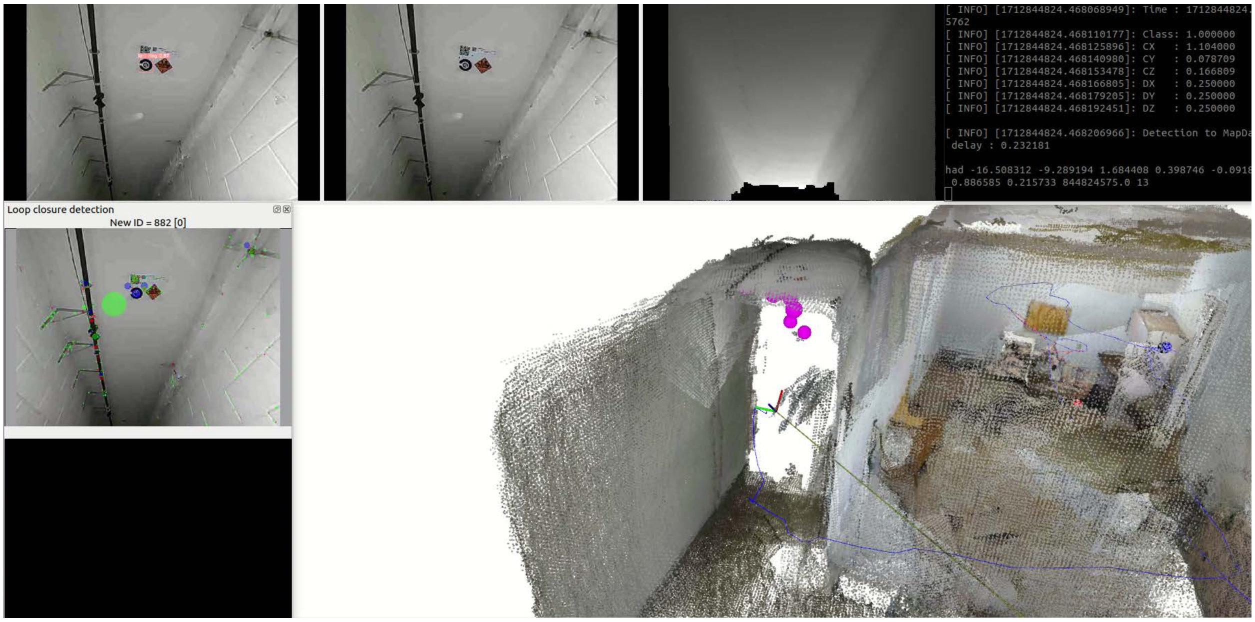
and localizing various objects within a highly cluttered and hostile environment. In Figure 18, a snapshot from the RViz visualization tool showcases the UAS's autonomous detection of a person mannequin in the RGB image. The system localized the detected individual within the point-cloud map, represented by distinctive red spheres. Similarly, Figure 19 presents another instance of successful detection and localization, this time featuring NIST ring targets. The corresponding pink spheres in the point-cloud map precisely indicate the locations of these targets, demonstrating the UAS's ability to identify and map objects of interest in real-time. Despite encountering challenges during the mission, such as premature crashes leading to missed loop closures and odometry drift errors, the resulting map, showcased in Figure 20, remained remarkably useful. Despite these setbacks, the map retained numerous intricate details, as highlighted in Figure 21. For instance, Figure 21a offers a detailed view of a mapped room containing a hidden person behind a couch. The fidelity of the mapping captured the room's shape and textures in great detail, underscoring the UAS's capability to provide detailed reconstructions of complex environments. Additionally, Figure 21b showcases another significant observation, illustrating a narrow room featuring a NIST ring target positioned on the ceiling. The flexibility afforded by the mapping sensors mounted on a gimbal proved invaluable in capturing intricate details, particularly in scenarios requiring upward or downward detection. A combined recorded video of the GCS laptop screen and the FPV video feed for the NIST simulated indoor mission can be found at [38]. Overall, despite encountering operational challenges, the Intrigue UAS demonstrated significant capabilities in autonomous detection, localization, and detailed mapping of complex indoor environments. These capabilities hold considerable promise for applications in search and rescue operations, disaster response, and various other critical scenarios where precise spatial awareness is paramount.



```
0.50s, Limit=0.000s, Conversion=0.0002s, RTAB-Map=
1340s, Maps update=0.0002s pub=0.0258s (local map=
5, WM=235)
had 4.032394 -17.514662 1.678786 0.16981 0.178747
.6791 0.691402 844464814.0 55
had 4.045938 -17.511271 1.675487 0.160786 0.1904 -
647699 0.719988 844465015.0 54
had 4.065751 -17.48547 1.670951 0.154638 0.20774 -
611567 0.747607 844465215.0 69
[ INFO] [1712844465.275749509]: rtabmap (287): Rat
0.50s, Limit=0.000s, Conversion=0.0003s, RTAB-Map=
1779s, Maps update=0.0001s pub=0.0235s (local map=
5, WM=235)
had 4.080827 -17.467178 1.666415 0.144002 0.21296
0.589321 0.765906 844465415.0 51
```

Fig. 18 Real-time detection and localization of a person manikin

In addition to its evaluation in simulated search and rescue scenarios, the Intrigue UAS underwent testing within a building located at the University of Maryland (UMD), where it encountered a controlled and cluttered environment containing manikins. Figure 22 offers a glimpse into the post-processed map generated from a successful flight of the Intrigue UAS within this building. Notably, the structure of the building comprises a series of cluttered interconnected rooms along with a narrower corridor featuring considerably fewer distinctive features. During the flight, the UAS demonstrated its robustness by executing successful loop closures, a vital process that helped maintain the integrity of the map structure by rectifying Visual-Inertial Odometry (VIO) drift errors. This ensured the accuracy and reliability of the mapped environment, even in complex and cluttered indoor settings. A closer examination of the map, as depicted in Figure 23, reveals the textural details that enhance the fidelity of the reconstruction. Noteworthy features such as objects positioned on desks, labels on trash cans, and even writing on a whiteboard are discernible within the map. This level of detail serves as a testament to the efficacy of both the custom-developed low-cost multispectral mapping sensor system and the associated mapping framework deployed on the Intrigue UAS. The successful capture and representation



```
[ INFO] [1712844824.468068949]: Time : 1712844824.
5762
[ INFO] [1712844824.468110177]: Class: 1.000000
[ INFO] [1712844824.468125896]: CX   : 1.104000
[ INFO] [1712844824.468140980]: CY   : 0.078709
[ INFO] [1712844824.468153478]: CZ   : 0.166809
[ INFO] [1712844824.468166805]: DX   : 0.250000
[ INFO] [1712844824.468179205]: DY   : 0.250000
[ INFO] [1712844824.468192451]: DZ   : 0.250000

[ INFO] [1712844824.468206966]: Detection to MapDa
 delay : 0.232181

had -16.508312 -9.289194 1.684408 0.398746 -0.0918
 0.886585 0.215733 844824575.0 13
```

Fig. 19 Real-time detection and localization of a NIST ring target

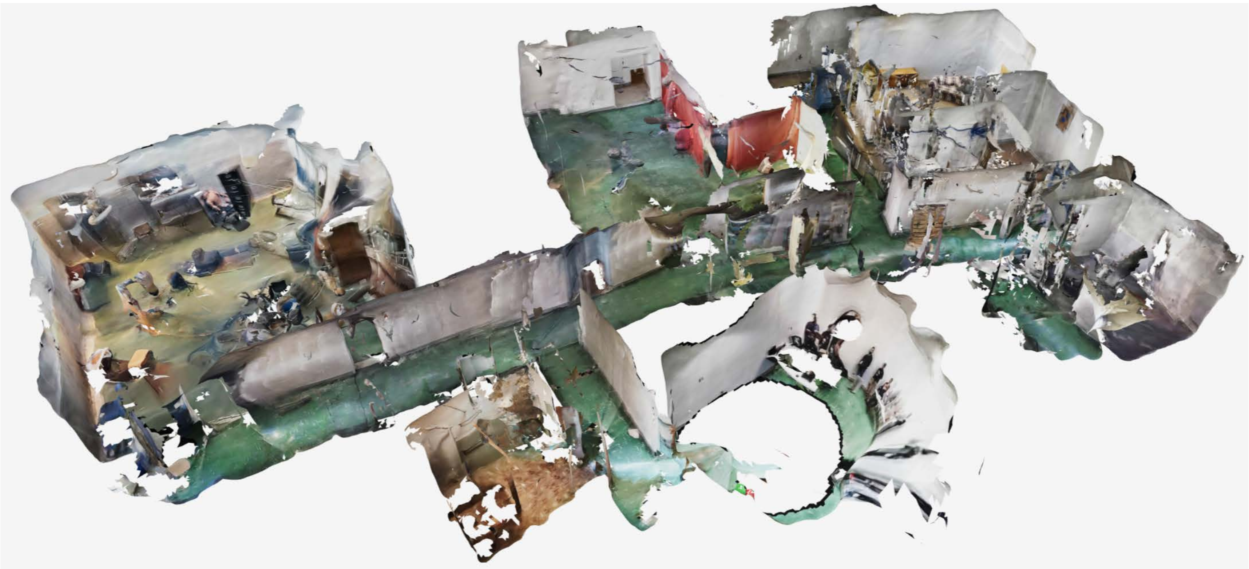
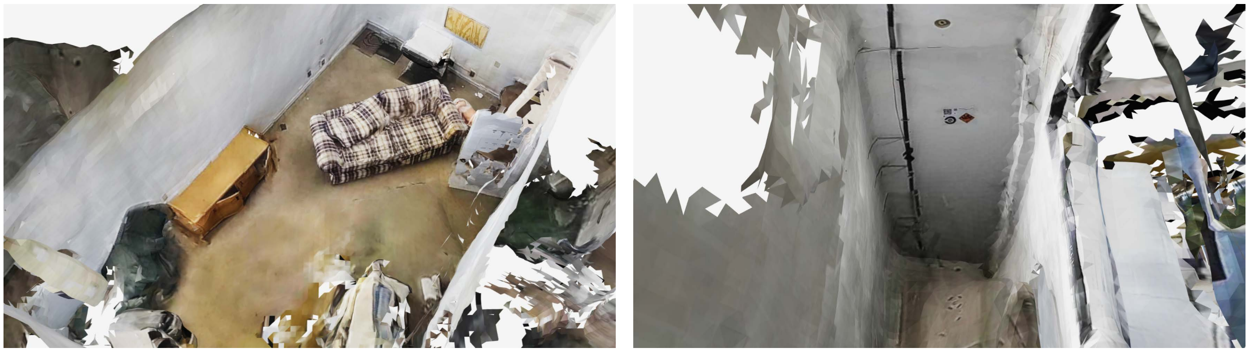


Fig. 20 The final post-processed map generated during the NIST's simulated indoor mission

of such intricate details underscore the system's capabilities in accurately mapping indoor environments with diverse textures and objects. This achievement is particularly noteworthy given the challenges posed by cluttered spaces and varying lighting conditions often encountered in indoor settings. A combined recorded video of the GCS laptop screen and the FPV video feed for this controlled indoor mapping experiment can be found at [39]. Overall, the performance of the Intrigue UAS within the UMD building highlights its potential for a wide range of applications, including indoor surveillance, mapping, and reconnaissance tasks. Its ability to navigate and map complex indoor environments while preserving detailed textural information positions it as a valuable tool for various domains, including urban planning, facility management, and emergency response.



(a) A person hidden behind the couch

(b) A NIST ring target on the ceiling of an unlit room

Fig. 21 Interesting viewpoints in the post-processed mesh map.

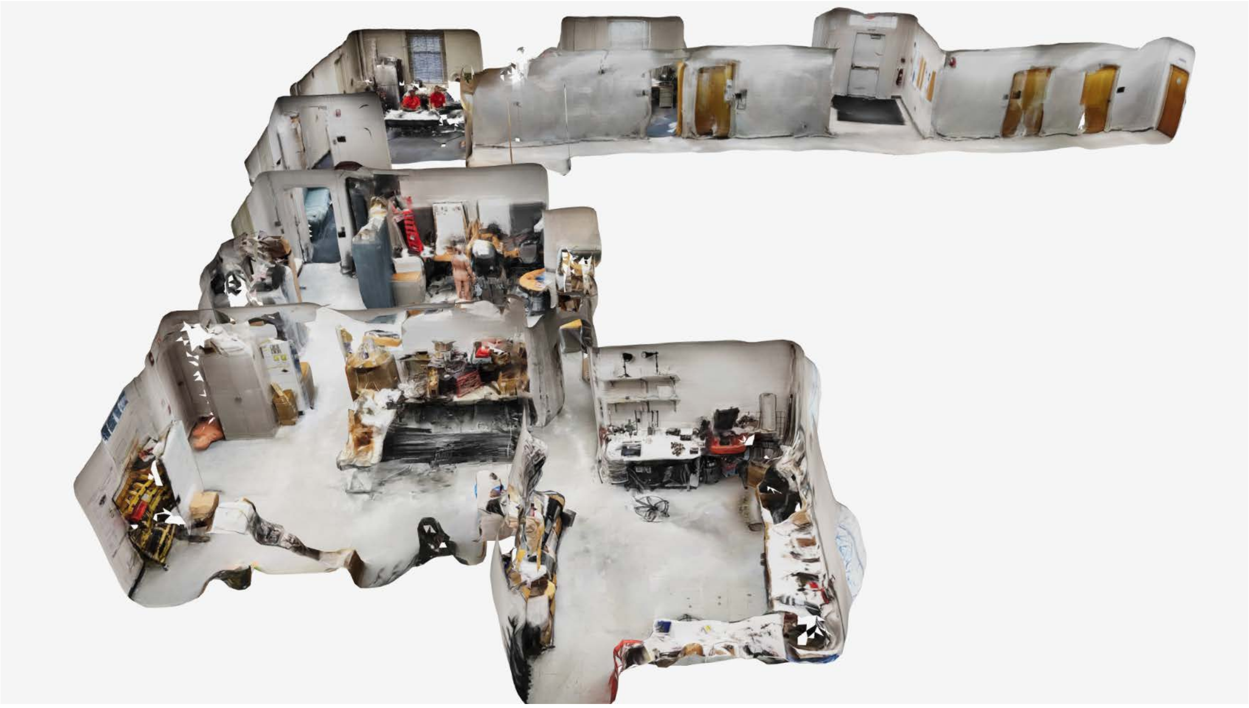


Fig. 22 Detailed post-processed map of a set of cluttered interconnected rooms and a narrow corridor of a building at UMD.

V. Conclusion

This paper details the design and implementation of a robust and cost-effective Unmanned Aerial System (UAS) specifically engineered for first responder use in search and rescue operations. The innovative 3D mapping, target detection, and localization framework, combined with a distributed computing approach, has demonstrated significant improvements in data acquisition speed and system stability. The system's performance was validated through its success in the National Institute of Standards and Technology (NIST) 2023 First Responder UAS 3D Mapping Challenge, where it earned notable accolades, including Third Place Overall and multiple Best-in-Class awards. These achievements, along with independent testing results, underscore the UAS's reliability and effectiveness in real-world scenarios. Ongoing work focuses on further enhancing the system's capabilities, including autonomous navigation and exploration, to provide even greater support for first responders in critical situations.

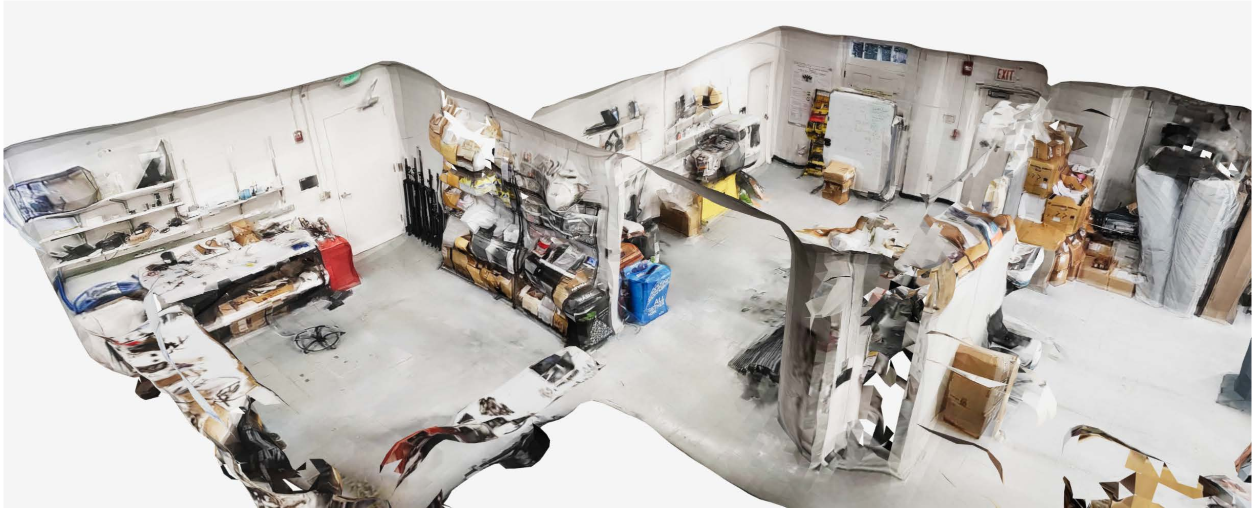


Fig. 23 Another view of the map showing the quality of textural details captured by the custom-developed low-cost multispectral mapping sensor system.

Acknowledgments

A. Shastry was supported in part by Army Grant No. W911W62120003. S. Abdi and S. Poojari were supported in part by Army Cooperative Agreement W911NF2120076. A. Ashry was supported on a Fulbright Fellowship. Special thanks to Grant Williams, lead UAS Pilot at the UAS Research and Operations Center, for piloting the Intrigue UAS.

References

[1] NIST, "NIST First Responder UAS Challenge Home," `https://firstresponderuaschallenge.org/uas5/`, 2023. Online; accessed May 2024.

[2] Karam, S., Nex, F., Chidura, B. T., and Kerle, N., "Microdrone-based indoor mapping with graph SLAM," *Drones*, Vol. 6, No. 11, 2022.

[3] Caballero, E., Madridano, A., Sainidis, D., Konstantoudakis, K., Daras, P., and Flores, P., "An automated UAV-assisted 2D mapping system for first responders," *18th International Conference on Information Systems for Crisis Response and Management. Virginia Tech, Blacksburg, VA*, 2021, pp. 890–902.

[4] Otero, R., Lagüela, S., Garrido, I., and Arias, P., "Mobile indoor mapping technologies: A review," *Automation in Construction*, Vol. 120, 2020, p. 103399.

[5] Placed, J. A., Strader, J., Carrillo, H., Atanasov, N., Indelman, V., Carlone, L., and Castellanos, J. A., "A survey on active simultaneous localization and mapping: State of the art and new frontiers," *IEEE Transactions on Robotics*, 2023.

[6] Kolhatkar, C., and Wagle, K., "Review of SLAM algorithms for indoor mobile robot with LIDAR and RGB-D camera technology," *Innovations in Electrical and Electronic Engineering*, Springer Singapore, 2021, pp. 397–409.

[7] Kumar, G. A., Patil, A. K., Patil, R., Park, S. S., and Chai, Y. H., "A LiDAR and IMU integrated indoor navigation system for UAVs and Its Application in Real-Time Pipeline Classification," *Sensors*, Vol. 17, No. 6, 2017.

[8] Opromolla, R., Fasano, G., Rufino, G., Grassi, M., and Savvaris, A., "LIDAR-inertial integration for UAV localization and mapping in complex environments," *2016 International Conference on Unmanned Aircraft Systems*, 2016, pp. 649–656.

[9] Chan, T. H., Hesse, H., and Ho, S. G., "LiDAR-based 3D SLAM for indoor mapping," *2021 7th International Conference on Control, Automation and Robotics*, 2021, pp. 285–289.

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

[11] Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C., and Rus, D., "Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping," *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2020, pp. 5135–5142.

[12] Cvišić, I., Ćesić, J., Marković, I., and Petrović, I., "SOFT-SLAM: Computationally efficient stereo visual simultaneous localization and mapping for autonomous unmanned aerial vehicles," *Journal of Field Robotics*, Vol. 35, No. 4, 2018, pp. 578–595.

[13] Hirschmüller, H., "Semi-global matching-motivation, developments and applications," *Photogrammetric Week 11*, 2011, pp. 173–184.

[14] Labbé, M., and Michaud, F., "RTAB-Map as an open-source lidar and visual simultaneous localization and mapping library for large-scale and long-term online operation," *Journal of Field Robotics*, Vol. 36, No. 2, 2019, pp. 416–446.

[15] Cramariuc, A., Bernreiter, L., Tschopp, F., Fehr, M., Reijgwart, V., Nieto, J., Siegwart, R., and Cadena, C., "maplab 2.0 – A Modular and Multi-Modal Mapping Framework," *IEEE Robotics and Automation Letters*, Vol. 8, No. 2, 2023, pp. 520–527.

[16] Steenbeek, A., and Nex, F., "CNN-based dense monocular visual SLAM for real-time UAV exploration in emergency conditions," *Drones*, Vol. 6, No. 3, 2022, p. 79.

[17] Chhikara, P., Tekchandani, R., Kumar, N., Chamola, V., and Guizani, M., "DCNN-GA: A deep neural net architecture for navigation of UAV in indoor environment," *IEEE Internet of Things Journal*, Vol. 8, No. 6, 2020, pp. 4448–4460.

[18] Alhafnawi, M., Bany Salameh, H. A., Masadeh, A., Al-Obiedollah, H., Ayyash, M., El-Khazali, R., and Elgala, H., "A Survey of indoor and outdoor UAV-based target tracking systems: Current status, challenges, technologies, and future directions," *IEEE Access*, Vol. 11, 2023, pp. 68324–68339.

[19] Cui, W., Shastry, A., Paley, D., and Nogar, S., "Autonomous aerial search and revisit behavior for communication limited environments," 2024, pp. 1–12.

[20] Wang, X., Liu, J., and Zhou, Q., "Real-time multi-target localization from Unmanned Aerial Vehicles," *Sensors*, Vol. 17, No. 1, 2017.

[21] Unlu, H. U., Chaikalis, D., Tsoukalas, A., and Tzes, A., "UAV indoor exploration for fire-target detection and extinguishing," *Journal of Intelligent & Robotic Systems*, Vol. 108, No. 3, 2023, p. 54.

[22] Modal AI, "Image sensor 4k high-resolution, low-light sensor for VOXL (Starvis IMX412 w/ M12-style Lens)," `https://www.modalai.com/products/mdk-m0061-1`, 2023. Online; accessed Apr 2024.

[23] Modal AI, "VOXL Time of Flight (ToF) depth sensor," `https://www.modalai.com/products/voxl-dk-tof`, 2011. Online; accessed Apr 2024.

[24] UMD AMAV Team, "AMAV Services," `https://github.com/UMD-AMAV/amav_services_nist5`, 2024. Online; accessed Apr 2024.

[25] Jocher, G., Chaurasia, A., and Qiu, J., "Ultralytics YOLO," , 2023. URL `https://github.com/ultralytics/ultralytics`.

[26] Zhang, Y., Sun, P., Jiang, Y., Yu, D., Weng, F., Yuan, Z., Luo, P., Liu, W., and Wang, X., "Bytetrack: Multi-object tracking by associating every detection box," *European Conference on Computer Vision*, Springer, 2022, pp. 1–21.

[27] ALFA Network, "HaLow-U," `https://www.alfa.com.tw/products/halow-u?variant=39467228758088`, 2020. Online; accessed May 2024.

[28] Microhard, "pDDL900 - Dual Frequency OEM Ethernet Serial Digital Data Link," `https://www.microhardcorp.com/pDDL900.php`, 2020. Online; accessed Apr 2024.

[29] Interdisciplinary Robot Lab, Université de Sherbrooke, "RTAB-Map," `https://github.com/introlab/rtabmap`, 2011. Online; accessed Apr 2024.

[30] UMD AMAV Team and Modal AI, "AMAV Portal," `https://github.com/UMD-AMAV/amav-portal`, 2024. Online; accessed Apr 2024.

[31] Meta, "React," `https://react.dev/`, 2021. Online; accessed Apr 2023.

[32] VMWare, "Spring Boot," `https://spring.io/projects/spring-boot`, 2021. Online; accessed Apr 2023.

[33] Open Source Community, "zlib," `https://github.com/madler/zlib`, 1995. Online; accessed Apr 2024.

[34] Open Source Community, "AliceVision - Photogrammetric Computer Vision Framework," `https://alicevision.org/`, 2020. Online; accessed Apr 2024.

[35] NIST, "NIST 3D Mapping Test Lane Guidebook for First Responder UAS 3D Mapping Challenge," https://firstresponderuaschallenge.org/wp-content/uploads/sites/5/2023/10/UAS5-NIST-5.0-3D-Test-Lane-Instructions-Final.pdf, 2023. Online; accessed Nov 2024.

[36] Griwodz, C., Gasparini, S., Calvet, L., Gurdjos, P., Castan, F., Maujean, B., Lillo, G. D., and Lanthony, Y., "AliceVision Meshroom: An open-source 3D reconstruction pipeline," *Proceedings of the 12th ACM Multimedia Systems Conference - MMSys '21*, ACM Press, 2021.

[37] Holz, D., and Behnke, S., "Fast range image segmentation and smoothing using approximate surface reconstruction and region growing," *Intelligent Autonomous Systems 12: Volume 2 Proceedings of the 12th International Conference*, Springer, 2013, pp. 61–73.

[38] Animesh Shastry, "Real-time mapping NIST indoor simulated mission," https://youtu.be/LKVjx_fZacU, 2024. Online; accessed May 2024.

[39] Animesh Shastry, "Real-time mapping at UMD engineering annex," https://youtu.be/6Ib0U-VxAXU, 2024. Online; accessed May 2024.