Journal Title
XX(X):1–23
©The Author(s) 2022
Reprints and permission:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

USTC FLICAR: A Sensors Fusion Dataset of LiDAR-Inertial-Camera for Heavy-duty Autonomous Aerial Work Robots

Ziming Wang, Yujiang Liu, Yifan Duan, Xingchen Li, Xinran Zhang, Jianmin Ji, Erbao Dong and Yanyong Zhang

Abstract

In this paper, we present the *USTC FLICAR Dataset*, which is dedicated to the development of simultaneous localization and mapping and precise 3D reconstruction of the workspace for heavy-duty autonomous aerial work robots. In recent years, numerous public datasets have played significant roles in the advancement of autonomous cars and unmanned aerial vehicles (UAVs). However, these two platforms differ from aerial work robots: UAVs are limited in their payload capacity, while cars are restricted to two-dimensional movements. To fill this gap, we create the “Giraffe” mapping robot based on a bucket truck, which is equipped with a variety of well-calibrated and synchronized sensors: four 3D LiDARs, two stereo cameras, two monocular cameras, Inertial Measurement Units (IMUs), and a GNSS/INS system. A laser tracker is used to record the millimeter-level ground truth positions. We also make its ground twin, the “Okapi” mapping robot, to gather data for comparison. The proposed dataset extends the typical autonomous driving sensing suite to aerial scenes, demonstrating the potential of combining autonomous driving perception systems with bucket trucks to create a versatile autonomous aerial working platform. Moreover, based on the Segment Anything Model (SAM), we produce the Semantic FLICAR dataset, which provides fine-grained semantic segmentation annotations for multimodal continuous data in both temporal and spatial dimensions. The dataset is available for download at: https://ustc-flicar.github.io/.

Keywords

Dataset, aerial work, aerial robot, mobile robot, SLAM, semantic segmentation, computer vision, LiDAR, cameras, bucket truck

1 Introduction

Aerial work is crucial in both daily life and industrial or agricultural production, as illustrated in Figure 1. Despite its significance, the low efficiency and high risks, including complicated workflow, falling from high places, electrocution from overhead power lines, and being trapped or squeezed, lead to numerous worker injuries or fatalities each year. Replacing workers with robots in dangerous aerial work environments can greatly enhance efficiency and safety, and potentially save lives.

1.1 Motivation and Challenges

In August 2003, a massive blackout (Northeast Blackout of 2003) hit parts of the Northeastern and Midwestern United States and Ontario, Canada. The outage affected approximately 55 million people, caused $25-30 billion in economic losses, and caused nearly 100 people’s death. And it all started with just a tree branch falling on a power line in Ohio, and the incorrect handling triggered a huge chain reaction.

As evident from the Northeast Blackout incident, it is crucial to address the challenges faced by the power grid to ensure a reliable and uninterrupted electricity supply. One solution that holds immense potential is the utilization of autonomous aerial work robots for performing maintenance and inspection tasks in the power grid.

What tasks would an aerial work robot system face when working in the power grid environment? And what challenges do these tasks reflect in terms of aerial work autonomy?

Some typical accidents in the power grid scenario are shown in Figure 2, such as tree branches falling onto power lines and power line blowing, which can pose a risk of power outages. Additionally, dirty insulators and tilted insulators can cause a decline in insulation performance, leading to power overload.

Therefore, starting with the tasks of aerial work robots in the power grid environment, which include inspection of power lines and equipment, identification, and removal of vegetation encroachments, reconnection of interrupted

University of Science and Technology of China, 96 Jinzhai Road, Hefei, 230026, Anhui, China.

Corresponding author:
Erbao Dong, CAS Key Laboratory of Mechanical Behavior and Design of Materials, Department of Precision Machinery and Precision Instrumentation, University of Science and Technology of China, 96 Jinzhai Road, Hefei, Anhui Province,230026, China.
Email: {zimingwang, lyj0910, dyf0202, starlet, zxrr}@mail.ustc.edu.cn, {jianmin, ebdong*, yanyongz}@ustc.edu.cn

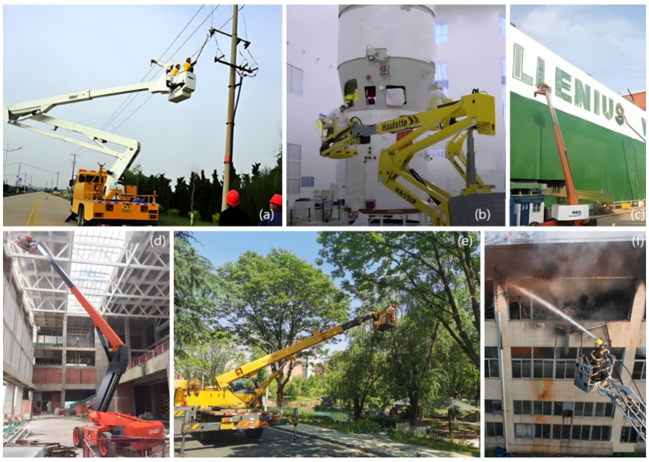


Figure 1. Typical aerial work scenes:
(a) repair and maintenance of electrical power facilities, (b) machinery manufacturing, (c) ship maintenance, (d) building construction,(e) tree trimming, and (f) aerial fire fighting and rescue

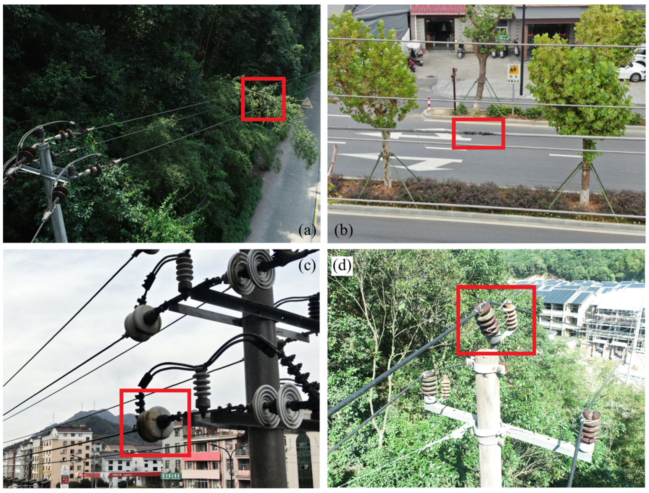


Figure 2. Typical power grid accidents:
(a) tree branch falling on the power line, (b) power line blown, (c) insulator contamination, (d) insulator tilt

transmission lines, and replacement of dirty or tilted insulators, etc. The significant challenge that these robots must overcome is how to effectively perceive and interact with the complex aerial work environment.

In order to navigate and perform tasks in this environment, aerial robots require accurate and real-time localization, as well as comprehensive 3D perception and reconstruction capabilities through their visual and inertial systems. These capabilities are crucial for tasks such as object recognition, trajectory planning and control, and scene understanding. However, the aerial work environment and tasks present specific challenges for visual and sensors fusion localization and mapping.

The presence of small-sized objects like tree branches, power lines, and trusses, combined with a lack of texture, makes their detection and reconstruction difficult. Unlike structured scenes such as warehouses or traffic, the aerial work environment is often cluttered and unstructured, making it challenging to rely on general structured features to improve algorithm performance. Furthermore, the aerial environment can be sparser than ground environments, making visual or LiDAR SLAM (Simultaneous Localization and Mapping) more challenging due to matching and loopback detection difficulties. Besides, with their increased degrees of freedom and exposure to sudden changes in motion, aerial work robots face challenges related to complex motion and attitude estimation.

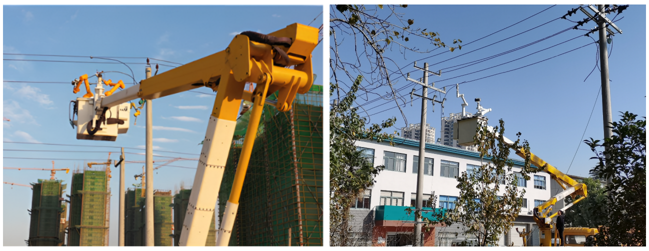


Figure 3. Aerial work robot for power grid tasks, equipped with two UR5 collaborative robot arms, lift into the air by bucket truck, developed by our lab.

Additionally, aerial work robots encounter similar difficulties as other outdoor robots, such as coping with a wide range of lighting conditions in different weather conditions. Direct sunlight or darkness at night can cause vision sensors to fail. To help overcome these challenges, the development of robust algorithms and the availability of appropriate datasets are crucial for advancing the capabilities of aerial work robots in the power grid environment. This is the motivation for us to make the USTC FLICAR dataset

1.2 Public Dataset

Public datasets play a critical role in the implementation of autonomous systems in new scenarios. They provide a standard benchmark for evaluating algorithms, allowing for the quick investigation, verification, and development of algorithms without the need for expensive hardware, complicated calibration, and time-consuming data preparation. This section will review relevant datasets to the USTC FLICAR, which can be divided into two categories: ground and aerial. The details are summarized in Table 1.

Ground scenarios, such as autonomous driving, have made significant progress in the past decade, partly due to the availability of diverse public datasets, such as the well-known KITTI dataset Geiger et al. [2013]. These datasets are remarkable for the abundance of sensors that can be incorporated because of the load-carrying capacity and space of ground vehicles. Ground scenarios rely on RTK-GPS or LiDAR SLAM for localization and mapping tasks, and generate ground truth values with centimeter-level accuracy.

Aerial autonomous systems have also progressed in recent years. The EuRoC dataset Burri et al. [2016] was the first to equip UAVs with synchronously triggered high frame rate stereo cameras and IMUs, enabling a tightly coupled visual-inertial system for aerial robot localization and attitude estimation. The most commonly used platform for aerial data collection is micro or small drones, which are usually equipped with a few cameras and inertial sensors due to payload constraints. In indoor or small scenes, motion capture systems are used to generate millimeter-level ground truth, while in outdoor or larger scenes, laser trackers are used.

However, it can be seen that there is a gap between ground and aerial datasets. The ground dataset comprises a large scene with rich sensors and provides centimeter-level accuracy for 2D motion ground truth. In contrast, the aerial dataset represents a smaller scene with limited sensors and offers millimeter-level accuracy for 3D motion ground truth. What if we combine the strengths of EuRoC and KITTI to create a new aerial dataset that incorporates both rich sensors and millimeter-level accuracy for 3D motion ground truth? This is where the USTC FLICAR dataset comes into play, bringing together the best aspects of both ground and aerial datasets.

Having rich sensors and highly accurate ground truth is essential for autonomous aerial working robots and systems. In the complex and unstructured aerial work environment, robots must be equipped with multiple complementary sensors to gather multi-modal data for perception. Moreover, as robots are expected to perform more precise tasks, such as precise interaction and manipulation of objects, millimeter-level accuracy ground truth becomes even more important.

1.3 Related Works: Bridge Ground-Aerial Gap

In the previous section, we compared two of the most prominent datasets in the ground and aerial domains: KITTI and EuRoC. We identified the gap between these datasets and proposed the creation of a new aerial dataset that features rich sensor data and 3D motion ground truth with millimeter-level accuracy. To achieve this goal, there are two options: adding sensors to a drone with a larger payload or enabling a car to "fly" for 3D movement.

The NTU VIRAL dataset Nguyen et al. [2022] chose the first option by equipping a larger UAV with two Ouster 16-beam 3D LiDARs, two cameras, and an IMU. It is the pioneer in applying LiDARs to drone scenarios.

The second option, which may seem like a fantasy, is the approach taken in the USTC FLICAR dataset. The USTC FLICAR dataset plans to extend the typical autonomous driving sensor suite to aerial scenes, with a large multi-sensor platform that includes four LiDARs (one Ouster 128-beam, two Velodyne 32-beam, one MEMS LiDAR), seven cameras (two stereo pairs), and three IMUs/INS. If we choose the first option like NTU VIRAL, only a few very expensive large drones, similar in size to helicopters, may have enough carrying capacity. However, it is nearly impossible and highly dangerous to fly such large aircraft equipped with a working robot at ultra-low altitudes (below 50m) in cities for aerial operations and data collection, due to current regulations and safety concerns. Luckily, we found the bucket truck.

A Bucket truck is a type of construction machinery with a high retention rate and is widely used in aerial work. The large pieces of machinery in various aerial work scenarios in Figure 1 are bucket trucks. They are heavy-duty construction vehicles equipped with an extendable hydraulic arm that carries a large bucket to elevate workers to elevated or inaccessible areas. These trucks have a strong payload capacity of around 200kg and can reach any target position in their 3D workspace through arm extension and joint rotation. With the help of a bucket truck, it is feasible to allow a heavy multi-sensor perception platform for autonomous driving to perform flexible 3D motion within a certain range in the air. With current technology, we cannot make a car fly, but we can start by making its sensing part fly.

When designing the sensor suite, we aimed to keep it as similar as possible to existing autonomous driving datasets. For example, our Bumblebee stereo camera is the same as those used in the Oxford RobotCar Maddern et al. [2017] and EU Long-term Yan et al. [2020] datasets. Our Velodyne HDL-32E horizontal 3D LiDAR is also the same as those used in the nuScenes Caesar et al. [2020], Oxford Radar RobotCar Barnes et al. [2019], EU Long-term, and NCLT Carlevaris et al. [2016] datasets. This makes it more convincing to compare algorithms using the same hardware between these datasets and the USTC FLICAR dataset. The USTC FLICAR dataset also shares similarities with existing aerial datasets but is geared toward more delicate and heavy-duty tasks. Millimeter-level outdoor ground truth was obtained using a laser tracker.

Looking back at Table 1, it is clear that USTC FLICAR has the most sensors among all aerial datasets. We believe that our dataset is a significant contribution that provides benchmarks for testing existing algorithms for autonomous systems and developing new ones that are more suited to the particularities of aerial work scenes.

In addition, the USTC FLICAR dataset and experiments will demonstrate that the innovative combination of an autonomous driving sensing suite and a bucket truck results in a versatile autonomous aerial platform with significant potential. This platform has exceptional perception capabilities that are comparable to those of self-driving cars and can be further equipped with a variety of working tools, such as robotic arms, to effectively carry out diverse and heavy-duty aerial working tasks. A combined prototype of the platform is illustrated in Figure 3.

1.4 Paper Roadmap

The remaining parts of the paper are organized as follows: In Section 2, we provide a detailed description of the various components of the data acquisition systems, as well as the parameters, characteristics, and functions of each sensor. Section 3 outlines the specific content of the dataset, the format of data storage, and the methods for accessing the data. Section 4 discusses the time synchronization, intrinsic and extrinsic calibration of sensors, and the generation of ground truth. We also present an evaluation of some state-of-the-art SLAM algorithms on the dataset as baselines and analyze the results in Section 5. Finally, Section 7 summarizes the paper and discusses future work.

2 Acquisition Systems and Sensor Setup

The data was acquired using two different systems, the "Giraffe" and "Okapi" systems, as depicted in Figure 4. The "Giraffe" system is an aerial platform consisting of a multi-sensor data collection platform (a), a laser tracker ground truth system (b), and a bucket truck (c). On the other hand, the "Okapi" system is a ground-based system similar to an autonomous vehicle, equipped with the same sensors (a) and a ground truth recording system (b), and mounted on a ground robot (d) for the acquisition of ground-level data for comparison with the data collected by the aerial system.

Table 1. Summary and comparison of notable public datasets for ground and aerial autonomous systems.

	Sensors					
Dataset	IMU	Camera	LiDAR	Ground Truth	Environment	Platform
MIT DARPA Huang et al. [2010]	N/A	5 PointGrey Firefly MV: 376×240×4/752×480	3D Velodyne HDL-64E 2D SICK LMS 291-S05 ×12	RTK GPS/INS	Outdoor (Urban)	Car
Ford Campus Pandey et al. [2011]	6 axis Xsens MTi-G	PointGrey LadyBug 3: 1600×600×6	3D Velodyne HDL-64E 2D Riegl LMS×2	RTK GPS/INS	Outdoor (Urban)	Car
KITTI Geiger et al. [2013]	6 axis OXTS RT3003	4 PointGrey FL2-14S3M/C-C : 1392×512×4	Velodyne HDL-64E	RTK GPS/INS	Outdoor (Urban)	Car
NCLT Carlevaris et al. [2016]	9 axis 3DM-GX3-45	PointGrey LadyBug 3: 1600×1200×6	3D Velodyne HDL-64E 2D Hokuyo×2	RTK GPS/ LiDAR SLAM	Outdoor (Campus)	Ground Robot
Oxford RobotCar Maddern et al. [2017]	6 axis NovAtel SPAN-CPT ALIGN	PointGrey Bumblebee XB3: 1280×960×3 3 Point Grey Grasshopper2: 1024×1024	2D SICK LMS-151×2 3D SICK LD-MRS	RTK GPS/INS	Outdoor (Urban)	Car
Oxford Radar RobotCar Barnes et al. [2019]	6 axis NovAtel SPAN-CPT ALIGN	PointGrey Bumblebee XB3: 1280×960×3 3 PointGrey Grasshopper2: 1024×1024×3	3D Velodyne HDL-32E ×2	RTK GPS/INS	Outdoor (Urban)	Car
Rosario Pire et al. [2019]	6 axis LSM6DS0	ZED stereo: 672×376×2	N/A	RTK GPS/INS	Outdoor (Agriculture)	Ground Robot
KAIST Urban Jeong et al. [2019]	9 axis Xsens MTi-G-300	FLIR FL3-U3-20E4C-C: 1280×560×2	3D Velodyne VLP-16C ×2	SLAM	Outdoor (Urban)	Car
EU Long-term Yan et al. [2020]	9 axis Xsens MTi-28A53G25	PointGrey Bumblebee XB2/3 2 Pixelink PL-B742F	3D Velodyne HDL-32E×2 2D SICK LMS	RTK-GPS	Outdoor (Urban)	Car
nuScenes Caesar et al. [2020]	9 axis Advanced Navigation Spatial	6 Basler acA1600-60gc: 1600×1200×6	3D Velodyne HDL-32E	RTK GPS/INS	Outdoor (Urban)	Car
EuRoC Burri et al. [2016]	6 axis ADIS16448	2 MT9V034: 752×480×2	N/A	6DOF MoCap 3D Laser Tracker	Indoor	UAV
Zurich Urban Majdik et al. [2017]	6 axis on PX4 autopilot board	GoPro Hero 4: 1920×1080	N/A	Aerial-Photogrammetry Visual SLAM	Outdoor (Urban)	UAV
UZH-FPV Delmerico et al. [2019]	6 axis IMU integrated with the camera	Snapdragon Fisheye Stereo: 640×480×2 mDAVIS Event: 346×260	N/A	3D Laser Tracker	Indoor Outdoor	UAV
NTU VIRAL Nguyen et al. [2022]	9 axis VectorNav VN100	2 uEye 1221 LE: 752×480×2	3D Ouster OS1-16×2	3D Laser Tracker	Outdoor (Campus)	UAV
USTC FLICAR	9 axis Xsens MTi-G-710	PointGrey Bumblebee XB3: 1280×960×3 PointGrey Bumblebee XB2: 1024×768×2 Hikvision MV-CB016-10GC 1440×1080 Hikvision MV-CE060-10UC 3072×2048	3D Velodyne HDL-32E 3D Velodyne VLP-32C 3D Ouster OS0-128 3D LiVOX Avia	3D Laser Tracker	Outdoor (Urban/ Aerial)	Bucket Truck/ Ground Robot

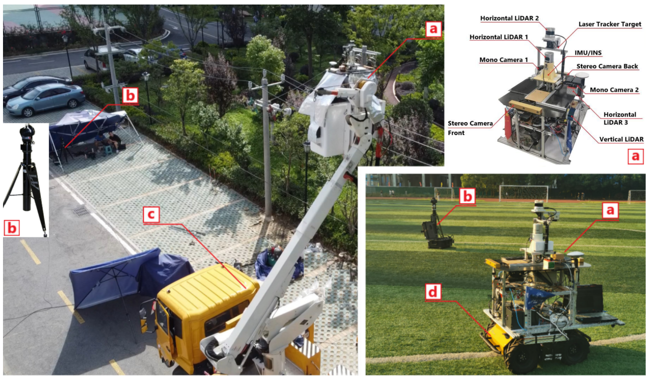


Figure 4. "Giraffe" and "Okapi" acquisition systems:
"Giraffe" aerial system: (a), (b) and (c).
"Okapi" ground system: (a), (b) and (d)
(a) multi-sensor data collection platform (Fig. 5), (b) laser tracker ground truth system, (c) bucket truck, (d) ground robot.

Section 2 will provide more detailed information about the components of the data acquisition systems.

Both Systems are equipped with the following sensors:

2.1 Inertial Measurement Unit (IMU)

The main IMU in the acquisition system is an Xsens MTi-G-710 INS/GNSS module, which is installed at the center of the system. The body frame is aligned with the Xsens sensor frame.

- 1 × Xsens MTi-G-710 INS/GNSS, 9-axis, 400 Hz, accuracy: 0.2° in roll/pitch, 0.8° in heading.

The Xsens module outputs the three-axis acceleration and three-axis angular velocity in its own coordinate system, as well as the quaternion attitude in the North-East-Down (NED) coordinate system. The Xsens module is hardware-synchronized to the same extrinsic GPS clock source with the cameras and LiDARs in the system, forming visual-inertial and LiDAR-inertial sensor units together. In addition, two extra 6-axis IMUs are installed in the OS0-128 LiDAR and LiVOX Avia LiDAR, forming part of the LiDAR-Inertial sensor unit.

2.2 3D LiDARs

A LiDAR (Light Detection and Ranging) is a remote sensing method that uses light in the form of a pulsed laser to measure distances to objects and create a 3D map of the environment. LiDARs are essential for autonomous systems

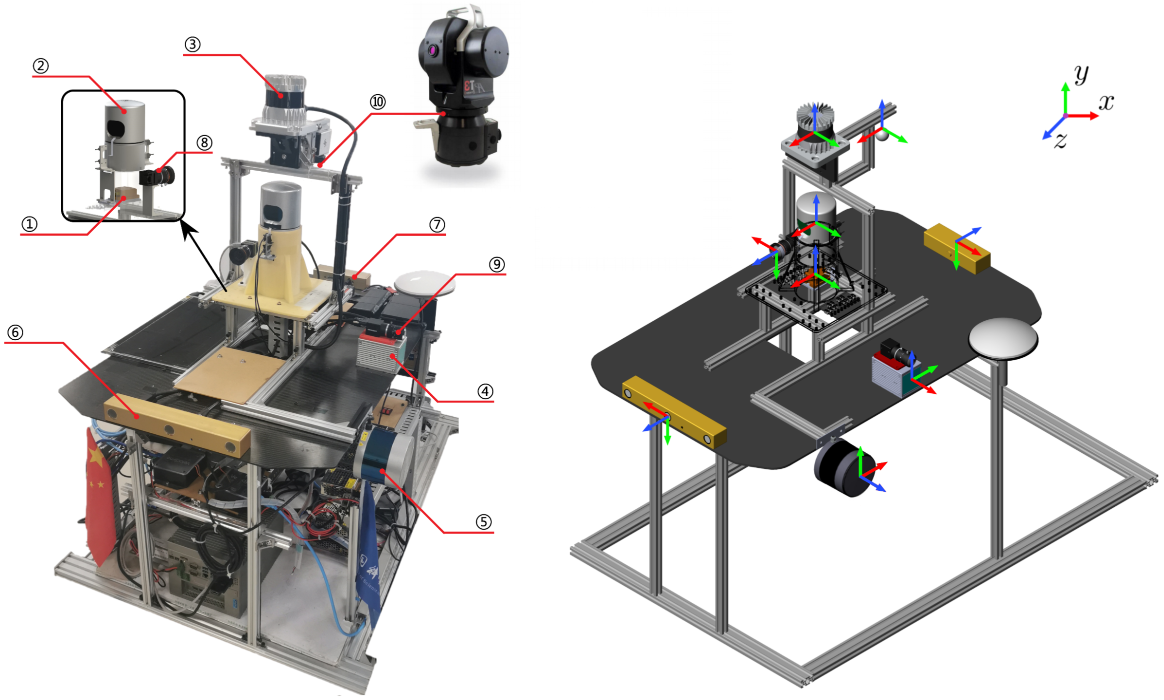


Figure 5. Sensor setup on a multi-sensor platform. The coordinate system shows the origin and orientation of each sensor mounted on the vehicle, with the following convention: X (red), Y (green), and Z (blue). Note that Velodyne LiDAR has two different coordinate system conventions. The coordinate system in the product manual is: Y: forward X: right Z: up; and the default coordinate system of the Velodyne ROS package is: X: forward Y: left Z: up. Here shows the second coordinate system of the ROS package. The number of the sensor in the figure corresponds to Table 2.

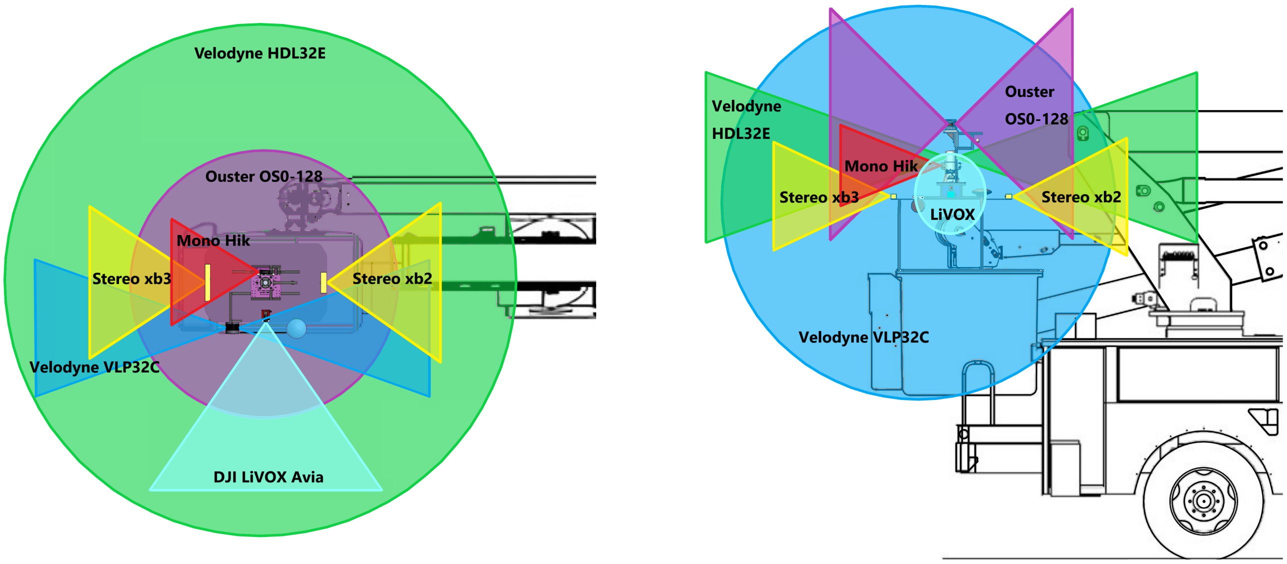


Figure 6. The visual scope of the acquisition system sensors, top view, and side view.

to accurately perceive and understand 3D scenes. The system in this work includes four different LiDARs: one digital LiDAR, two mechanical LiDARs, and one MEMS LiDAR.

• Digital LiDAR: 1 × Ouster OS0-128, 10 Hz, 128 beams, 0.7° angular resolution, $\pm$ 1.5 to $\pm$ 5 cm distance accuracy, collecting 2.62 million points/second, field of view: 360° HFoV, 90° VFoV ($\pm 45^{\circ}$), range:50 m

• Mechanical LiDAR: 1 × Velodyne HDL-32E, 5/10 Hz, 32 beams, 1.33° angular resolution, $\pm$ 2 cm distance accuracy, collecting 1.39 million points/second, the field of view: 360° HFoV, 41.3° VFoV ($+10.67^{\circ}$ to -30.67°), range: 100 m

• Mechanical LiDAR: 1 × Velodyne VLP-32C, 10 Hz, 32 beams, 0.33° angular resolution (non-linear distribution), ± 3 cm distance accuracy, collecting 1.20 million points/second, the field of view: 360° HFoV, 40° VFoV (-25° to $+15^{\circ}$), range: 200 m

• MEMS LiDAR: 1 × DJI LiVOX Avia, 10 Hz, 2 cm distance accuracy, collecting 0.24 million points/second, the

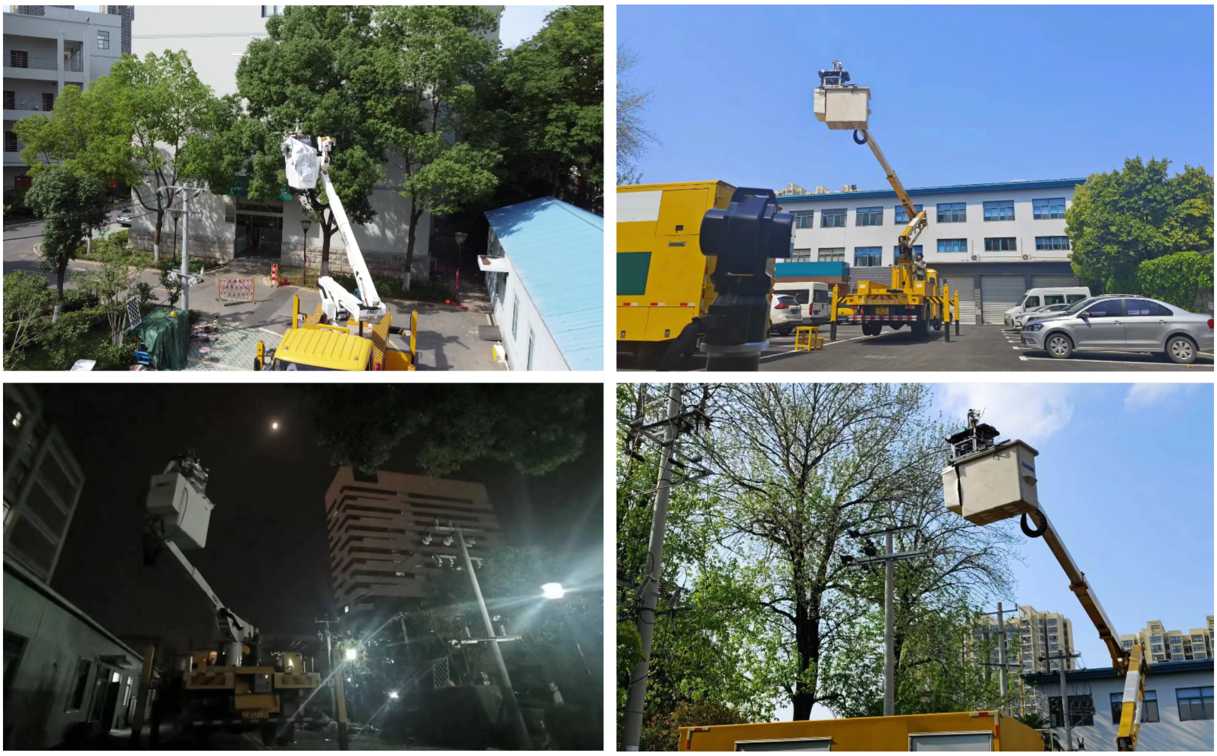


Figure 7. Images of several typical data acquisition sites. Left: HF0XX sites, Right: HN0XX sites.

field of view: 70.4° HFoV, 77.2° VFoV (Non-repetitive Scanning), range: 450 m

The system uses Ouster OS0-128 and Velodyne HDL-32E LiDARs as the main source of 3D environmental data, and Velodyne VLP-32C and DJI LiVOX Avia LiDARs to supplement the blind areas of vision. The Ouster OS0-128 is a digital LiDAR based on custom system-on-a-chip with single-photon avalanche diode detectors, which can output point clouds, depth images, and signal-intensity images of LiDAR and visible light spectrum (Figure 11 (c),(d)). Velodyne HDL-32E has the highest point cloud accuracy, while Velodyne VLP-32C and DJI LiVOX Avia LiDARs are mounted vertically and horizontally, respectively, to provide a 360-degree view in both horizontal and vertical directions. The point clouds from LiVOX LiDAR scans are uniformly accumulated on the map over time.

2.3 Monocular and Stereo Cameras

Cameras are an important part of autonomous system perception, as they capture high-resolution images of the surrounding environment, providing information about object shape, color, texture, and motion direction. And stereo cameras can effectively recover depth. We equipped the sensor platform with several cameras, including stereo cameras and monocular cameras:

• 1 x Point Grey Bumblebee XB3 (BBX3-13S2C-38) trinocular stereo camera, 1280 × 960 × 3, 10Hz, Sony ICX445 CCD, 1/3”, 3.75 µm, global shutter, 3.8mm lens, 66° HFoV, 12/24cm baseline, IEEE 1394B, 54 dB Signal To Noise Ratio (SNR).

• 1 x Point Grey Bumblebee XB2 (BBX2-08S2C-38) binocular stereo camera, 1024 x 768 × 2, 10-15Hz, Sony ICX204 CCD, 1/3”, 4.65 µm, global shutter, 3.8mm lens, 70° HFoV, 12cm baseline, IEEE 1394A, 60 dB SNR.

• 1 x Hikvision MV-CB016-10GC-C industrial monocular camera, 1440 × 1080, 20Hz, Sony IMX296 CCD, 1/2.9”, 3.45 µm, global shutter, 6mm lens (MVL-HF0628M-6MPE), 63.11° HFoV, GigE, 41 dB SNR.

• 1 x Hikvision MV-CE060-10UC industrial monocular camera, 3072 × 2048, 20Hz, Sony IMX178 CCD, 1/1.8”, 2.4 µm, global shutter, 6mm lens (MVL-HF0628M-6MP), 49.3° HFoV, USB 3.0, 41.3 dB SNR.

Stereo cameras use two or more cameras to capture the same scene from different viewpoints, allowing the system to estimate depth information. On the other hand, monocular cameras use a single camera and rely on other techniques like motion parallax and perspective to estimate depth. Both types of cameras have their advantages and disadvantages, and their usage depends on the specific application and requirements.

2.4 Laser Tracker

A laser tracker is a high-precision instrument used to accurately measure the position and orientation of objects in 3D space. It works by emitting a laser beam and tracking a target that reflects the laser light back to the tracker. The laser tracker measures the time it takes for the laser beam to travel to the target and back and uses this information to calculate the position and orientation of the target. The API T3 laser tracker we use is fixed horizontally on the ground and is the only sensor that is independent of the multi-sensor platform, as shown in Figure 4 (b). During the motion of the sensor platform, the laser tracker tracks the active target ball that is rigidly fixed on the platform body and outputs the three-dimensional space coordinates of the target point with millimeter precision in its own coordinate system.

• API T3 Laser Tracker, 50Hz, azimuth: ± 320° (640° end to end), angular resolution: ± 0.018 arc-seconds,

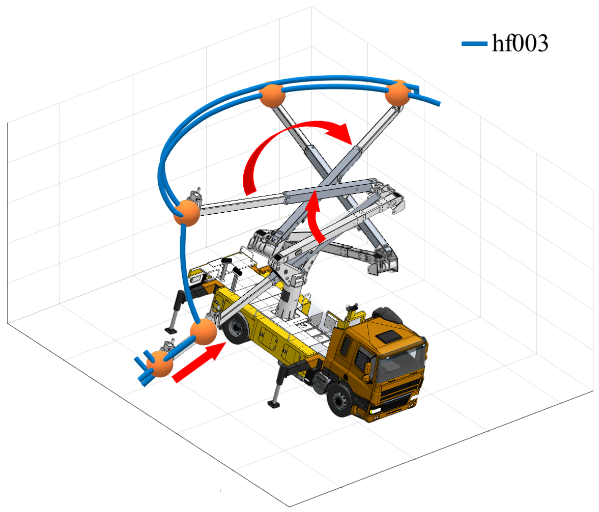


Figure 8. The joint action of the hydraulic arm of the bucket truck corresponds to the movement trajectory of the end sensor platform, taking the hf003 sequence as an example.

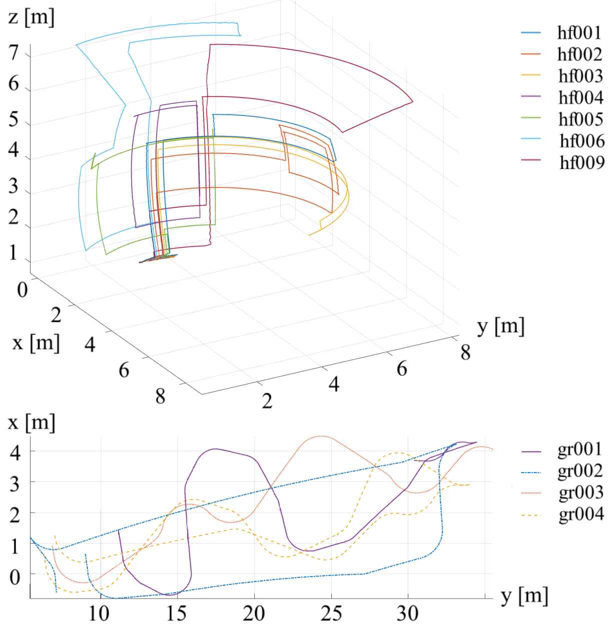


Figure 9. Representative aerial and ground trajectories in the dataset.

angular accuracy: 3.5 μm/meter, system resolution: 0.1 μm, maximum lateral target speed: 4 meters/sec, maximum acceleration: 2 g, internal level accuracy: ± 2 arc-second, linear range: 80 m.

3 Dataset

As shown in Figure 7, representative scenes of aerial work were selected to collect data. The surrounding objects include buildings, trees, power lines, roads, etc. At the same time, we collected data from morning to night and under different weather conditions, in order to ensure that the aerial work robot can work around the clock. All data is available here: `https://ustc-flicar.github.io/datasets/`.

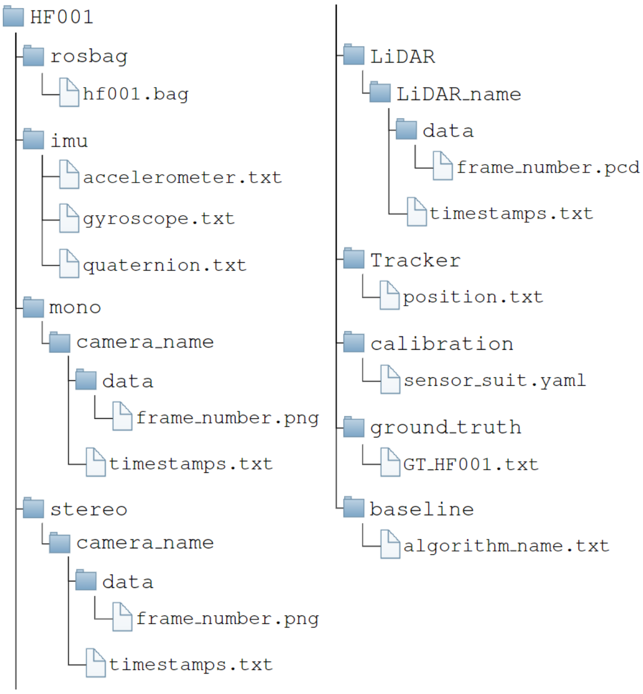


Figure 10. Dataset file structure.

The sensor platform completes various aerial motions through bucket truck, including movement in XYZ direction and large range rotations. Figure 8 shows one of these trajectories and the corresponding bucket truck movement, providing an intuitive explanation of how the motion trajectories in the dataset were generated.

To get an impression of the trajectories, some of the flight paths are shown in Figure 9. These paths were provided by an API T3 laser tracker and recorded on the base station as ground truth position measurements. The laser tracker measurements can be found under the `Tracker` folder in the file `position.txt`. A short summary of the paths is given in Table 4. The measurement data from the 9-axis Xsens IMU/INS is stored in the `imu` folder. The file `accelerometer.txt` contains acceleration data and `gyroscope.txt` contains angular velocity data, both in the IMU frame $\boldsymbol{I}$. The file `quaternion.txt` contains attitude data in the world frame $\boldsymbol{W}$. To facilitate the use of the dataset, we have provided the optimal estimate of the full pose $\in \mathbb{R}^3 \times \mathbb{H}$ (both position and attitude) in the body frame $\boldsymbol{B}$ as ground truth based on the original measurement. More details about the generation of ground truth can be found in section 4.8. The ground truth data is provided under the `ground_truth` folder in the file `GT_HF0XX.txt` in TUM format Sturm et al. [2012] `<timestamp,x,y,z,qx,qy,qz,qw>`.

The storage format of camera and LiDAR data in our dataset refers to the KITTI dataset. Images are stored with lossless compression using 8-bit PNG files. When collecting data, we only record the original images in Bayer format and do not perform parsing, compression, or filtering (such as Bayer demosaicing) in order to preserve the original information of the data to the greatest extent and improve data utility. For example, as shown in Figure 11 (a), the three views from the left, center, and right camera of Bumblebee-XB3 are encoded in the red, green, and blue channels,

Table 2. The sensor model specifications and data information in this dataset.

No	Sensor	Model	ROS Topic	Message type	Rate
1	IMU/INS	Xsens MTi-G-710	/imu/data	sensor_msgs/Imu	400Hz
2	Horizontal LiDAR 1	Velodyne HDL-32E	/velodyne_points_HDL32	sensor_msgs/PointCloud2	5/10Hz (rotate at 10Hz)
3	Horizontal LiDAR 2	Ouster OS0-128	/os_cloud_node/imu /os_cloud_node/points /img_node/reflect_image /img_node/signal_image	sensor_msgs/Imu sensor_msgs/PointCloud2 sensor_msgs/Image sensor_msgs/Image	100Hz 10Hz 10Hz 10Hz
4	Horizontal LiDAR 3	LiVOX Avia	/livox/lidar /livox/imu	livox_ros_driver/CustomMsg sensor_msgs/Imu	10Hz 200Hz
5	Vertical LiDAR 1	Velodyne VLP-32C	/velodyne_points_VLP32	sensor_msgs/PointCloud2	10Hz
6	Stereo Camera front	PointGrey Bumblebee xb3	/camera/left/image_raw /camera/center/image_raw /camera/right/image_raw	sensor_msgs/Image	10-16Hz 10-16Hz 10-16Hz
7	Stereo Camera back	PointGrey Bumblebee xb2	/cam_xb2/left/image_raw /cam_xb2/right/image_raw	sensor_msgs/Image	10-20 Hz 10-20 Hz
8	Mono Camera 1	Hikvision MV-CB016-10GC-C	/hik_camera/iamge_raw	sensor_msgs/Image	20Hz
9	Mono Camera 2	Hikvision MV-CE060-10UC	/right_camera/iamge	sensor_msgs/Image	20Hz

Table 3. Data sequences details and summary

Sequence	Time	Length (m)	Duration (s)	Size (GB)	Characteristic	Summary
HF001	2022-06-08 15:35	26.46	192	66.5	sun, semantic	Conducting data collection in a representative aerial work environment on the urban roadside, surrounding objects include utility poles, trees, buildings, etc. The collection environment involves significant lighting variations from daytime to dusk to nighttime. Additionally, the keyframes in the data are annotated with semantic segmentation. HF001-HF009 are the main sequences in this dataset.
HF002	2022-06-08 15:48	33.50	217	75.7	sun, semantic	
HF003	2022-06-08 16:14	34.26	217	83.2	sun, rotation, semantic	
HF004	2022-06-08 16:30	24.10	155	82.0	sun, semantic	
HF005	2022-06-08 17:20	22.82	260	90.3	sun, semantic	
HF006	2022-06-08 18:13	33.90	230	86.3	clouds, semantic	
HF007	2022-06-08 19:01	34.32	207	67.5	dusk, rotation,semantic	
HF008	2022-06-08 21:20	30.78	210	91.3	night, rotation, semantic	
HF009	2022-06-08 21:27	35.42	238	101.3	night, rotation, semantic	
HF010	2022-06-07 18:19	16.06	210	91.7	clouds	Additional Sequences Group I: Data collection environment same to the main sequence, except for the existing sensors, an extra down-looking camera mounted with a vertical lidar Velodyne VLP32 accessible via ROS topic "/down_camera/image"
HF011	2022-06-07 19:13	17.81	207	25.5	dusk	
HF012	2022-06-07 19:26	26.15	231	121.1	dusk	
HF013	2022-06-07 20:40	26.25	187	100.9	night	
HF014	2022-06-07 20:54	25.57	201	119.2	night	
HN001	2023-04-07 15:53	38.44	390	79.2	sun, rotation, semantic	Additional Sequences Group II: Collecting data in a new parking lot environment, which includes not only buildings, poles and trees but also numerous vehicles. An extra FLIR ADK infrared camera is installed, and it can be accessed via the ROS topic "/flir_boson/image_raw". Bumblebee XB3 and LiVOX are offline.
HN002	2023-04-07 12:12	44.97	395	56.1	sun, rotation, semantic	
HN003	2023-04-07 12:23	38.64	442	62.2	sun, rotation	
HN004	2023-04-07 13:02	42.50	417	59.1	sun, rotation	
GR001	2022-06-15 15:50	46.51	333	35.9	sun, turn	Additional Sequences Group III: Collecting ground data using multi-sensor platform mounted on a Husky chassis. The data collection involves capturing different motion patterns compared to aerial scenarios, such as 2D movements, turns, etc. Bumblebee XB3 and Ouster are offline.
GR002	2022-06-15 19:19	66.89	116	12.1	dusk, turn	
GR003	2022-06-15 19:24	40.59	190	19.5	dusk, turn	
GR004	2022-06-15 19:34	69.16	211	25.7	dusk, turn	

respectively, and logged to the disk as one Bayer format image. We perform operations such as view separation and Bayer demosaicing offline for the raw image, which strictly guarantees the timestamp and brightness consistency of stereo camera images because they were logged to the disk at exactly the same time in one image. Images of each view are provided separately in the `left/center/right` folder under the `xb3` folder.

The LiDAR scans are stored as PCD files which save the *(x,y,z)* coordinate of the point cloud. If you want to use additional information of LiDAR scans such as reflection intensity *intensity*, scanning lines *ring* corresponding to point cloud. The raw binary file we provide in rosbag contains these data.

For the convenience of users who use Robot Operation System (ROS), all sensor data is packaged and provided together in rosbag, and information about the topic name, topic type, frame rate, etc. of the data corresponding to the sensor is organized in Table 2.

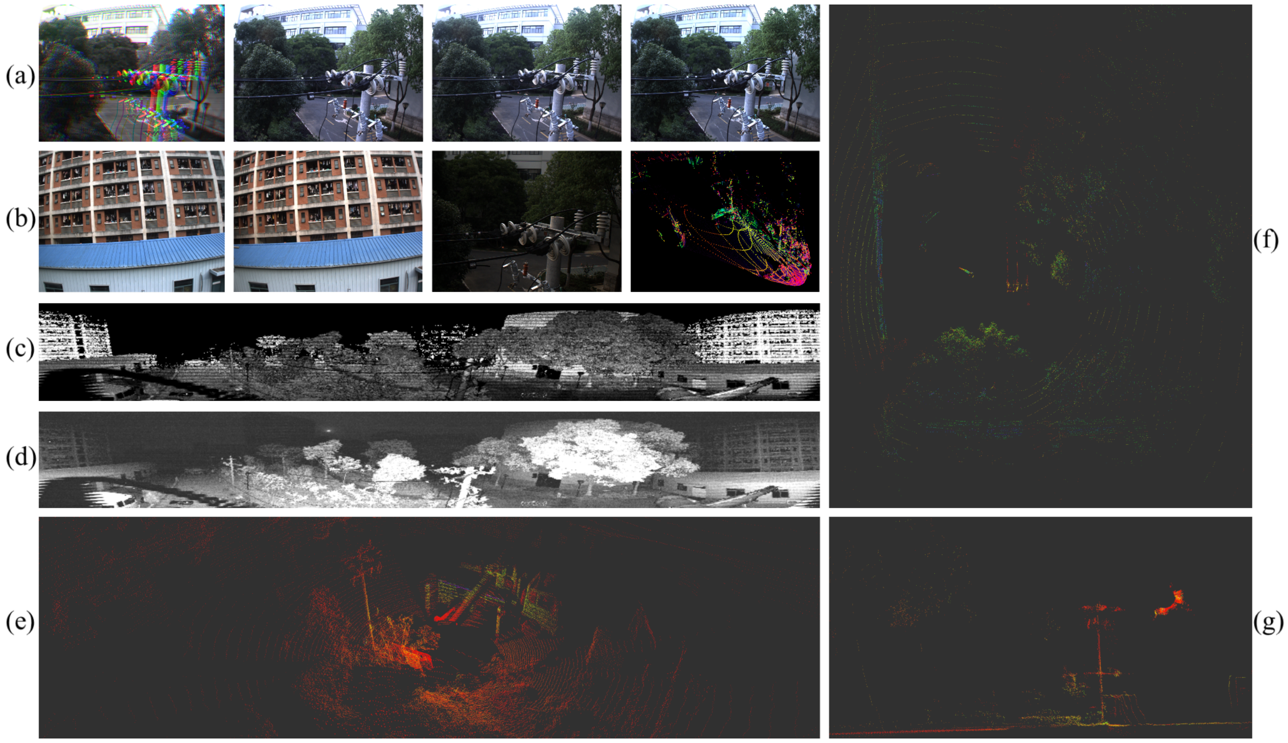


Figure 11. Visualization of multi-sensor data, at the same time. (a) Bumblebee-XB3 raw Bayer image, left, center, right views. (b) Bumblebee-XB2 left, right views; Hikvision camera; LiVOX Avia point cloud. (c) reflect image of Ouster OS0-128. (d) signal image of Ouster OS0-128. (e) Ouster OS0-128 point cloud. (f) Horizontal Velodyne HDL-32E point cloud. (g) Vertical Velodyne VLP-32C point cloud.

The calibration files of the sensors are provided under `calibration` folder in `sensor_suit.yaml`. The specific content of each `.yaml` file is related to the calibration method of each sensor or sensor suite. We will elaborate necessary information and definition in the next section Section 4. We ran some SOTA SLAM algorithms on the dataset baselines. These results are available in `algorithm_name.txt` under folder `baseline`. For more information, please refer to Section 5.

4 Sensor Synchronization and Calibration

Accurate time synchronization and spatial calibration of multiple sensors are necessary for sensor spatiotemporal fusion. In the system, sensors are securely mounted using aluminum profile brackets, 3D prints, and carbon fiber sheets. Time synchronization of multiple sensors and data acquisition computers is achieved using an FPGA-based hard trigger circuit and NTP synchronization network. Calibration data and methods can be obtained from this web page: https://ustc-flicar.github.io/calibration/.

4.1 Accurate Modeling of Multi-sensor Platform

As shown in Figure 12, we utilize the Photoneo PhoXi 3D Scanner L for precise modeling of the data acquisition multi-sensor platform. Its technical specifications are as follows:

- Scanner Type: structured light scanner
- Calibration Accuracy (1 σ): 0.200 mm
- Temporal Noise (1 σ): 0.190 mm
- Resolution: Up to 3.2 million 3D points
- Scanning Range: 870 - 2150 mm
- Optimal Scanning distance: 1239 mm
- Scanning Area (sweet spot): 1082 x 772 mm
- Point-to-Point Distance: 0.524 mm
- Scanning Time: 250 - 2750 ms

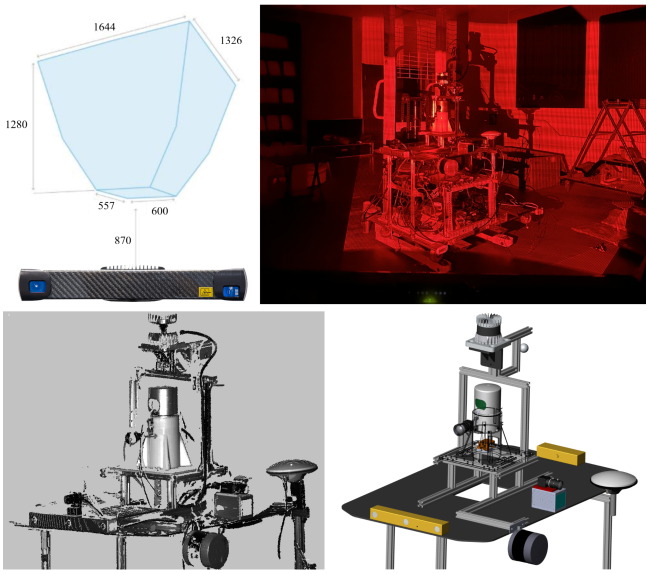


Figure 12. Precise modeling of the data acquisition multi-sensor platform using Photoneo PhoXi 3D Scanner. The scan result point cloud and the SolidWorks design model

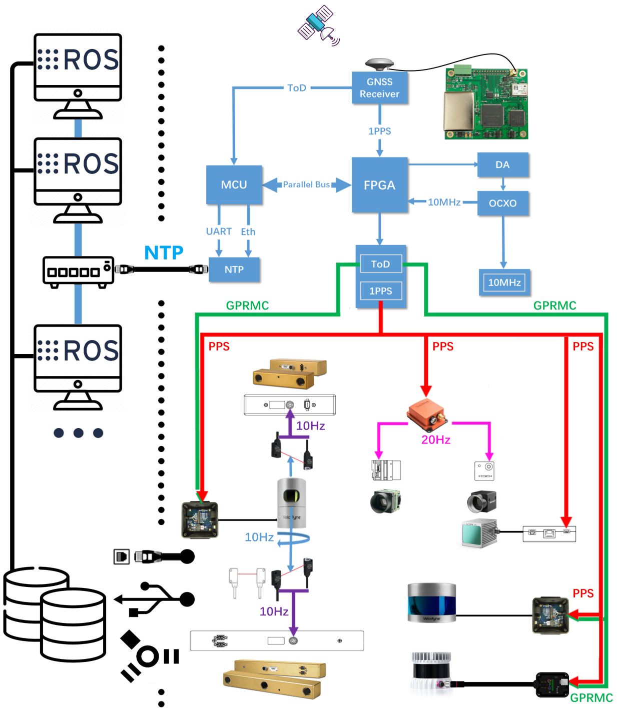


Figure 13. The multi-sensor time synchronization system structure, which is explained in detail in section 4.2.

These specifications exemplify the Photoneo PhoXi 3D Scanner L's capabilities, allowing for precise and detailed modeling for our multi-sensor platform.

Multiple-angle modeling pointcloud of data acquisition platform with the PhoXi 3D scanner, SolidWorks assembly models, and drawings are provided along with the dataset. Before users run the calibration program using the USTC FLICAR dataset, we recommend referring to the structured light scanning modeling data and SolidWorks design models to obtain reliable initial calibration values.

4.2 Time Synchronization

The time synchronization module performs the time synchronization of the camera, LiDAR, IMU, and the main control computing module. The first level of the time synchronization module is the GNSS receiving module, which obtains the UTC true time data with nanosecond precision through the satellite. The logic circuit processes the timing information, converts the GNSS signal into PPS and NMEA signals, and the LiDAR is connected to the two signals for time synchronization. At the same time, the PPS signal is connected to the IMU module and the frequency divider module. The PPS signal synchronizes the timestamp of the IMU's inertial data with the true UTC time, the frequency divider module is used to trigger the camera at the desired frame rate using the PPS signal. The camera trigger signal is aligned with the PPS signal at the edge of the whole second, and the delay between the two signals is within a few tens of ns. Therefore, the camera exposure image time is synchronized with the IMU data acquisition time. The time of each camera in a stereo camera is time-synchronized during device manufacture, so the trigger signal triggers all cameras on the serial trigger line simultaneously. The computing master accepts the NTP network data packets converted and sent by the FPGA and performs time synchronization through the NTP protocol.

For the horizontal LiDAR and the stereo camera on the axis, a photoelectric trigger sensor is designed. When the LiDAR rotates to coincide with the camera's field of view, the excitation switch is turned on and the camera is exposed to collect images to ensure the spatiotemporal synchronization of point clouds and images.

4.3 Mono and Stereo Cameras Calibration

In order to make full use of the metric information of 2D images for 3D tasks, we calibrate the intrinsic parameters of each camera and the extrinsic parameters between stereo cameras. The calibration approach we use is proposed by Zhang [2000]. A known size checkerboard is placed at different distances and attitudes relative to the cameras, and the cameras in the same stereo pair are triggered synchronously. They collect images of the checkerboard at a fixed frame rate as calibration data. The camera parameters are provided in the OpenCV format, which are stored in the `camera_name.yaml` calibration file.

The camera parameters are notated as:

- *image size* $\in \mathbb{N}^2$
- *camera_matrix* $\in \mathbb{R}^{3\times3}$
- *distortion_coefficients* $\in \mathbb{R}^5$
- *rectification_matrix* $\in \mathbb{R}^{3\times3}$
- *projection_matrix* $\in \mathbb{R}^{3\times4}$

Here, the *distortion_coefficients* vector is used to rectify the tangential and radial distortion of images, using the pinhole camera distortion model. The *rectification_matrix* is only applicable to stereo cameras, which is used to align the epipolar lines between two stereo images for 3D stereo vision geometry calculation. It is an identity matrix for monocular cameras.

The camera projection matrix is used to project objects in the 3D world to the camera's 2D image pixels:

$$P_{proj} = \begin{bmatrix} f_x & 0 & c_x & T_x \\ 0 & f_y & c_y & T_y \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{1}$$

The left 3×3 portion is the intrinsic *camera_matrix* for the rectified image. The fourth column $\begin{bmatrix} T_x & T_y & 0 \end{bmatrix}^T$ is to translate the optical center of the second camera to the position in the frame of the first camera. For monocular cameras, $T_x = T_y = 0$. The average calibration error of monocular cameras is around 0.08 pixels, while the average calibration error of stereo cameras is around 0.1 pixels.

4.4 Visual Inertial Calibration

The fusion of visual and inertial sensors will greatly improve the robustness of the visual-based SLAM system. The camera provides high-resolution measurements of the environment, while the IMU measures the internal ego-motion of the sensor platform.

The first task is to calibrate the intrinsic parameters of the IMU. The IMU sensor will drift over time, therefore it is necessary to add an error term into the motion model to correct the IMU raw data based on the IMU noise model. We fixed the IMU still on the anti-shake optical table for 4

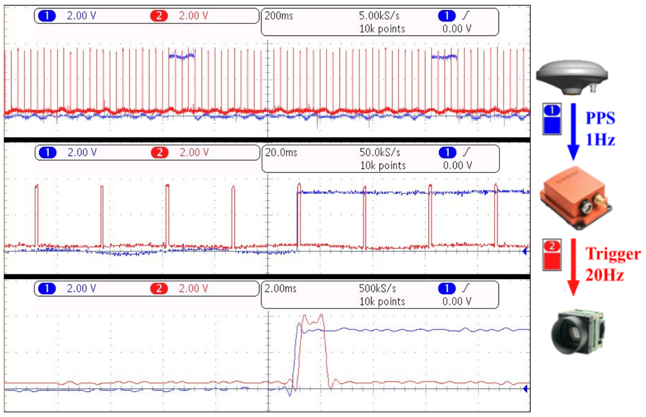


Figure 14. Time synchronization signals in the visual-inertial system. The time resolution of the abscissa, from top to bottom, is 200ms, 20ms, and 2ms, from a Tektronix MDO3024 oscilloscope. Blue: 1Hz PPS signal; Red: 20Hz camera trigger signal.

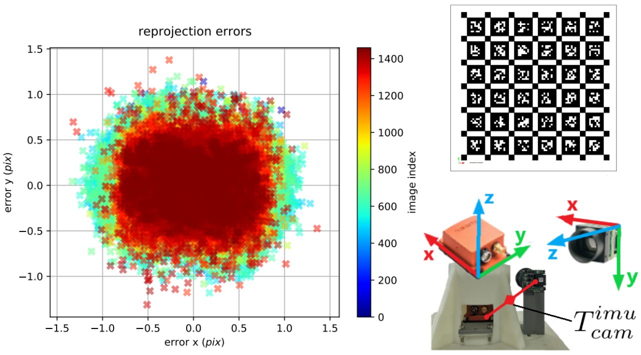


Figure 15. Reprojection error of camera-IMU extrinsics calibration using Kalibr.

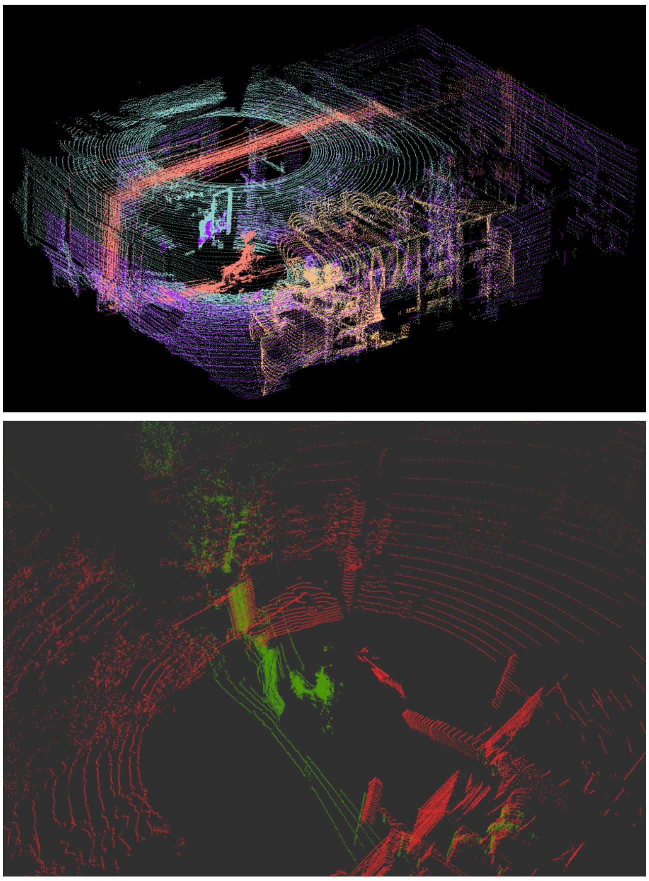


Figure 16. The point cloud was obtained from the calibration process. The image above shows point clouds in the indoor calibration room. The green point cloud is the Ouster OS0-128 LiDAR, the purple point cloud is the Velodyne HDL-32E LiDAR, the red point cloud is the Velodyne VLP-32C LiDAR, and the orange point cloud is the LiVOX Avia LiDAR.
The image below shows outdoor measurements. The red point cloud is the Velodyne HDL-32E LiDAR, and the green point cloud is the Velodyne VLP-32C LiDAR.

hours and recorded the data. The toolbox imu_utils is used for calibration.

IMU intrinsic parameters in the corresponding yaml file are as followed:

- σ_g — gyroscope white noise
- σ_a — accelerometer white noise
- σ_{bg} — gyroscope bias instability
- σ_{ba} — accelerometer bias instability

Note that calibration is done in a nearly ideal static setup. In a dynamic setting, the noise will be higher with other factors such as temperature changes. Therefore, it is beneficial to appropriately increase these parameters when using IMU data for camera-IMU extrinsic calibration or visual-inertial odometry.

The second step is to calibrate the extrinsic parameters between the IMU and the camera. The intrinsic parameters of the camera have already been calibrated in section 4.3. The time synchronization accuracy between the IMU and the monocular camera is shown in Figure 14. The time drift between the IMU clock reference PPS signal and the camera trigger signal is within 0.2 ms. The Kalibr Rehder et al. [2016] visual-inertial calibration toolbox is used to calibrate the relative spatial relationship between the IMU and the camera. The camera and IMU are rigidly fixed with the base bracket. The overall visual-inertial system performs translation along the XYZ three-axis and full rotation around each axis in front of an AprilTag Olson [2011] grid sequences with known size, and records the data for calibration.

camera-IMU extrinsics in the corresponding yaml file are as followed:

- *rotation matrix*: $\mathrm{R}^{imu}_{cam} \in SO(3) \subset \mathbb{R}^{3\times3}$
- *translation vector*: $\mathrm{t}^{imu}_{cam} \in \mathbb{R}^{1\times3}$

$$T^{imu}_{cam} = \begin{bmatrix} R^{imu}_{cam} & t^{imu}_{cam} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2}$$

The reprojection error of the Camera-IMU extrinsic parameter calibration is shown in Figure 15, for most images the reprojection error is within 1.0 pixels. The mean, median, and standard deviation of the reprojection error are 0.352 pixels, 0.321 pixels, and 0.197 pixels, respectively.

4.5 Multiple LiDAR Calibration

A single LiDAR has problems such as low information density and vertical blind spots. Therefore, we equip the aerial platform with LiDARs from different angles for environmental perception. Extrinsic parameter calibration between multiple LiDARs is a prerequisite for the fusion of LiDAR data.

Specifically, we refer to the Ouster OS0-128 as the base LiDAR on account of its biggest FOV. The extrinsic parameters between the remaining three LiDARs and the OS0-128 are derived through the Normal Distributions Transform (NDT), as proposed by Biber and Straßer [2003]. To minimize the adverse impact of the environment on the calibration accuracy, we choose a non-degenerate scene Zhang et al. [2016] characterized by a complex structure for the calibration task. Meanwhile, to improve the accuracy, a series of frames for each LiDAR were acquired from diverse positions and angles, which optimize the extrinsic parameters jointly.

As shown in Figure 16, according to the calibration results, the point clouds from the four LiDAR sensors (OS0-128, HDL-32, VLP-32, and LiVOX) are transformed and displayed in the same coordinate system.

The extrinsic parameters for multiple LiDARs in the corresponding yaml file are as follows:

- *rotation matrix*: $\mathrm{R}_{Lidar1}^{Lidar2} \in SO(3) \subset \mathbb{R}^{3\times3}$
- *translation vector*: $\mathrm{t}_{Lidar1}^{Lidar2} \in \mathbb{R}^{1\times3}$

$$T_{Lidar1}^{Lidar2} = \begin{bmatrix} R_{Lidar1}^{Lidar2} & t_{Lidar1}^{Lidar2} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{3}$$

4.6 LiDAR Inertial Calibration

We use the online method to calibrate the extrinsic parameters of the rotation between the Velodyne HDL-32E LiDAR and the Xsens MTi-G-710 IMU, which can be seen as a kind of hand-eye calibration. The attitude preintegration result of the IMU measurement value from time t_k to time t_{k+1} is denoted as $\boldsymbol{q}_{i_k}^{i_{k+1}}$. $\boldsymbol{q}_{L_k}^{L_{k+1}}$ is the attitude change of the LiDAR scan at time t_k relative to the LiDAR scan at time t_{k+1}, obtained by Generalized-ICP Segal et al. [2009]. $\boldsymbol{q}_L^i$ is the rotation transformation from LiDAR to IMU.

According to the properties of the rotation matrix, we can get:

$$\boldsymbol{q}_{i_k}^{i_{k+1}} = \boldsymbol{q}_L^i \otimes \boldsymbol{q}_{L_k}^{L_{k+1}} \otimes \boldsymbol{q}_i^L \tag{4}$$

$$\boldsymbol{q}_{i_k}^{i_{k+1}} \otimes \boldsymbol{q}_L^i = \boldsymbol{q}_L^i \otimes \boldsymbol{q}_{L_k}^{L_{k+1}} \tag{5}$$

According to the quaternion properties described in Sola [2012], we transform the above formula $\boldsymbol{q}$ into matrix representation $\boldsymbol{Q}$:

$$(\boldsymbol{Q}_{i_k}^{i_{k+1}}{}^{+} - \boldsymbol{Q}_{L_k}^{L_{k+1}}{}^{-})\boldsymbol{q}_L^i = 0 \tag{6}$$

We fully moved the LiDAR-Inertial system to collect N sets of measurement data. The final task is to solve the following overdetermined system:

$$\begin{bmatrix} \boldsymbol{Q}_{i_0}^{i_1}{}^{+} - \boldsymbol{Q}_{L_0}^{L_1}{}^{-} \\ \cdots \\ \boldsymbol{Q}_{i_{N-1}}^{i_N}{}^{+} - \boldsymbol{Q}_{L_{N-1}}^{L_N}{}^{-} \end{bmatrix} \boldsymbol{q}_L^i = \boldsymbol{A}_{4N\times4}\boldsymbol{q}_L^i = 0 \tag{7}$$

We use the SVD method to solve this overdetermined system, perform SVD decomposition on $\boldsymbol{A}_{4N\times4}$, and then take the eigenvector corresponding to the smallest singular value as the final result of $\boldsymbol{q}_L^i$.

The translation vector t_{Lidar}^{imu} between LiDAR and IMU is calibrated and estimated based on the initial value of the size of the 3D model. Since the connecting part is 3D printed, the accuracy of the calibration can be trusted.

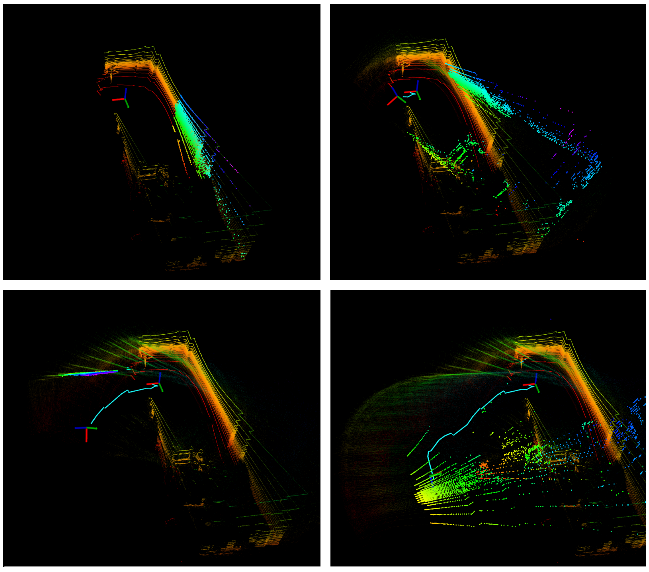


Figure 17. The failure scenario of direct calibration between Velodyne VLP-32C and Xsens MTi-G-710, when the sensor platform is in a slow movement during initialization, occurring significant drift in the calibration algorithm. LiDAR_IMU_Init Zhu et al. is used. Similar problems also exist in some other open-source and our own calibration methods.

LiDAR-IMU extrinsics in the corresponding yaml file are as followed:

- *rotation matrix*: $\mathrm{R}_{Lidar}^{imu} \in SO(3) \subset \mathbb{R}^{3\times3}$
- *translation vector*: $\mathrm{t}_{Lidar}^{imu} \in \mathbb{R}^{1\times3}$

$$T_{Lidar}^{imu} = \begin{bmatrix} R_{Lidar}^{imu} & t_{Lidar}^{imu} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{8}$$

Apart from the Velodyne HDL-32E LiDAR, there are three other LiDAR sensors. Among them, the LiVOX Avia LiDAR and Ouster OS0-128 LiDAR come with built-in IMUs. We recommend utilizing their built-in IMUs for algorithm development involving LiDAR-IMU coupling, such as SLAM. These IMUs are meticulously time-synchronized and spatially calibrated during the manufacturing process, ensuring precise alignment with the LiDAR sensor. For accurate extrinsic parameters, please refer to the manufacturer's manual.

However, when it comes to the Velodyne VLP-32C LiDAR, we do not recommend directly calibrating it with Xsens. This is primarily due to the greater distance between the two devices and the VLP-32C's vertical installation, which leads to a sparse feature set in the captured point cloud. Consequently, the accuracy of calculating LiDAR odometry through frame-to-frame matching is compromised and the initial value estimation will be more difficult, making it challenging to obtain precise extrinsic parameters in the Xsens coordinate system through direct calibration using existing technology. Although we have explored various good open-source calibration methods for directly calibrating the VLP-32C and Xsens, such as LiDAR_IMU_Init Zhu et al. [2022], the results have proven unsatisfactory, as depicted in Figure 17.

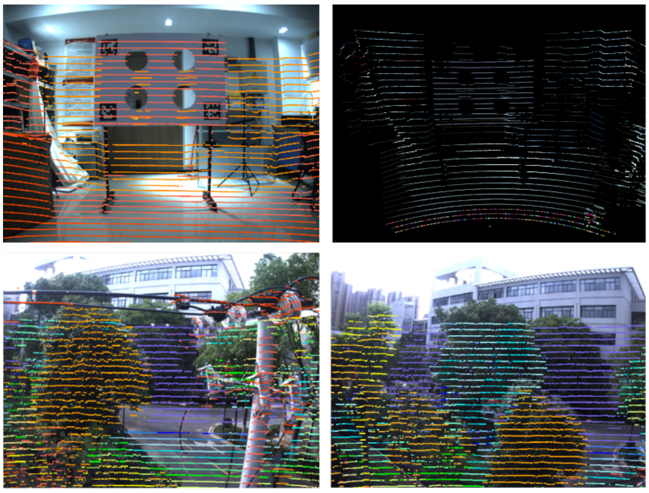


Figure 18. Velo2Cam Beltrán et al. camera-LiDAR calibration. *Up-Left*: special calibration board, project LiDAR point cloud to the image. *Up-Right*: colorize LiDAR point cloud with image. *Down*: Aerial scenes LiDAR points fusion with images. (Velodyne-HDL-32E and Bumblebee-xb3-center)

To address this issue, we propose a feasible solution. We recommend leveraging the multi-LiDAR calibration extrinsic parameters obtained in Section 4.5, indirectly calculated using the Velodyne HDL-32E LiDAR as an intermediary. Additionally, it is advisable to consult the detailed structured light modeling and Solidworks model described in Section 4.1. This approach will yield significantly higher accuracy in determining extrinsic parameters compared to direct calibration. Similarly, the LiVOX Avia and Ouster OS0-128 LiDARs can also obtain precise extrinsic parameter transformations to the Xsens Body frame by utilizing the Velodyne HDL-32E LiDAR as an intermediary.

On the other hand, the USTC FLICAR dataset also provides reliable baselines for the extrinsic parameters calibration of the LiDAR-IMU. We look forward to a LiDAR-IMU extrinsic parameters calibration algorithm with higher robustness, accuracy, and versatility.

4.7 LiDAR Camera Calibration

Image data has rich and dense object information, but lacks the depth information of the picture. The LiDAR data can just make up for this defect, giving accurate depth information and object structure information. In the process of 3D target detection, the fusion of image and LiDAR point cloud information can achieve higher accuracy.

Accurate camera-LiDAR calibration is a necessary condition for the fusion. We use the method proposed by Velo2cam Beltrán et al. [2022] to get the extrinsic parameters of LiDAR and cameras. Figure 18 illustrates the calibration scene and effect of Velo2cam. A special calibration board with four ArUco tags and four circular reference holes is placed in different positions as a calibration target. The 3D pose of each ArUco marker relative to the cameras is obtained by solving a classic perspective-n-point (PnP) problem to obtain the 3D position and orientation of the reference holes in space. Besides, automatic targetless calibration approach can also be used to get the extrinsic calibration between LiDARs and cameras. These methods are convenient and can be performed online. Our team conducted a careful survey Li et al. [2022] of automatic targetless camera-LiDAR calibration.

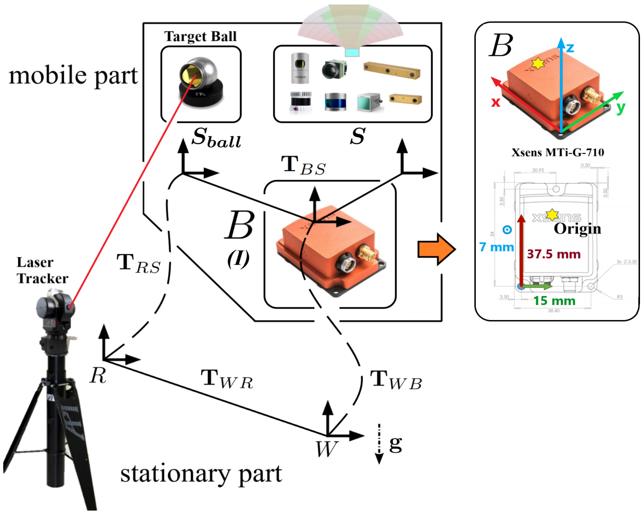


Figure 19. The sensor system used to capture the datasets consists of multiple sensors, each reporting measurements in its own reference frame $\boldsymbol{S}$. The datasets also include raw data from ground truth instruments, reported in the target ball frame $\boldsymbol{S_{ball}}$ and Laser tracker frame $\boldsymbol{R}$. The body frame $\boldsymbol{B}$ is aligned with the IMU sensor frame $\boldsymbol{I}$. Calibration information for all extrinsic parameters linking the sensors to the body frame $\boldsymbol{B}$ and intrinsic parameters is included in the dataset. The definition and transformation of the coordinate system of the data acquisition system will be further discussed in Section 4.8

camera-LiDAR extrinsics in the corresponding yaml file are as followed:

- *rotation matrix*: $\mathrm{R}_{velo}^{cam} \in SO(3) \subset \mathbb{R}^{3\times3}$
- *translation vector*: $\mathrm{t}_{velo}^{cam} \in \mathbb{R}^{1\times3}$

$$T_{velo}^{cam} = \begin{bmatrix} R_{velo}^{cam} & t_{velo}^{cam} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{9}$$

The commonly used camera-LiDAR multimodal data fusion schemes directly use LiDAR points as multimodal data aggregation points. LiDAR points are projected to the image plane as follows:

$$z_{cam}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = hP_{proj}T_{velo}^{cam}\begin{bmatrix} X_{velo} \\ Y_{velo} \\ Z_{velo} \\ 1 \end{bmatrix} \tag{10}$$

where X_{velo}, Y_{velo}, Z_{velo} denote 3D location of LiDAR point, x, y, z_{cam} denote its 2D position and projected depth on the image plane. P_{proj} got in section 4.3 denotes the camera intrinsic parameter.And h represents the scaling factor due to down-sampling. Examples of data fusion are shown in Figure 18.

4.8 Ground Truth Alignment

To provide useful and accurate ground truth, the measurements from the laser tracking system are spatiotemporally aligned with the sensor system (body frame $\boldsymbol{B}$ is defined at the IMU sensor frame $\boldsymbol{I}$). The basic information of the raw measurement data used to generate ground truth is as follows.

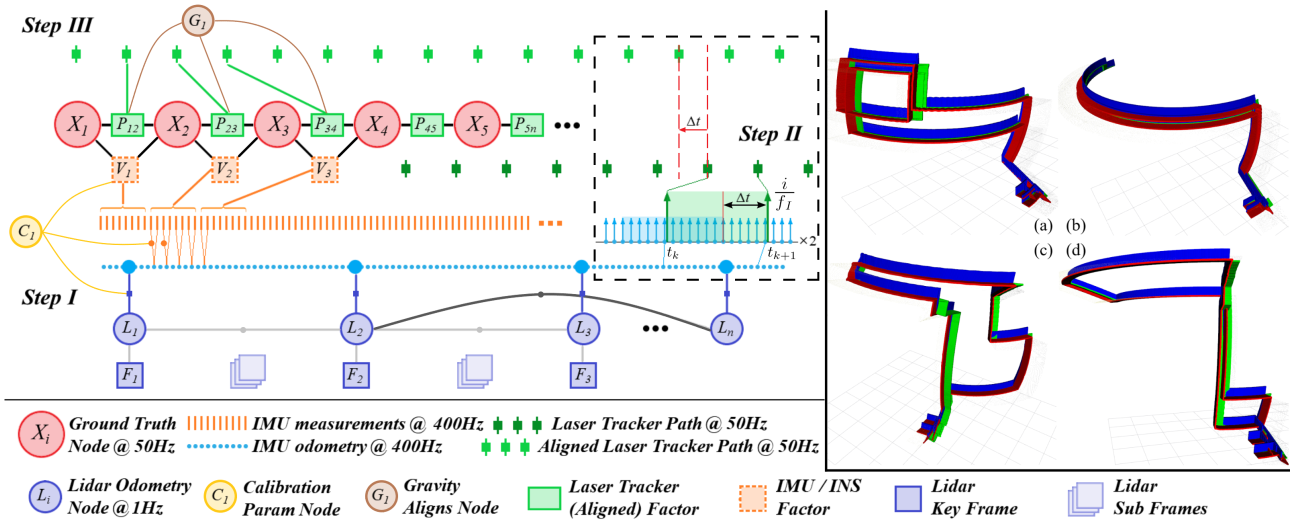


Figure 20. Example of a factor graph created by our system. The states to be estimated are represented by circles, and the measured values are represented by squares. For visualization, we group all calibration parameters into one node C_1, G_1 is the rotation from gravity alignment to laser tracking frame. On the right are the example ground truth trajectories generated on our dataset. On the right is the visualization of the ground truth of several sequences: (a) HF002, (b) HF003, (c) HF006, (d) HF009.

The definition of the coordinates below can be reviewed in Figure 19 :

- The API laser tracking system is mounted horizontally on the ground using a tripod. The target ball frame $\boldsymbol{S_{ball}}$ it tracks is rigidly fixed to the body of the sensor platform. The 3D position of the target point trajectory in the laser tracking frame $\boldsymbol{R}$, represented as $\boldsymbol{p^R_{S_{ball}}} \in \mathbb{R}^3$, is output at 50Hz (second highest). The time t_R is referenced to the intrinsic clock of the tracking system.

- The Xsens IMU/INS is rigidly fixed to the center of the sensor platform body with the body frame $\boldsymbol{B}$ defined at its own frame $\boldsymbol{I}$. Acceleration $\boldsymbol{a_{I(B)}} \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega_{I(B)}} \in \mathbb{R}^3$ are output under the IMU frame $\boldsymbol{I}$ at 400Hz (highest). The origin of the body frame $\boldsymbol{B}$ coincides with the origin of the accelerometer. At the same time, the AHRS (Attitude and Heading Reference System) system of Xsens outputs the attitude $\boldsymbol{q^W_B} \in \mathbb{H}$ of the sensor platform in world frame $\boldsymbol{W}$ (NED coordinates). $\boldsymbol{q_W}$ is a statistical optimal 3D orientation estimate computed by the Xsens Kalman Filter algorithm (XKF3) using signals of the rate gyroscopes, accelerometers, and magnetometers. Xsens is synchronized with the GPS clock, the time $t_{I(B)} = t_{gps}$.

- The Velodyne HDL-32E forms a LiDAR-Inertial system together with Xsens IMU. Here its data is not used directly but used as LiDAR odometry to constrain and optimize the trajectory obtained by IMU preintegration. It is synchronized with the same GPS clock as Xsens, therefore $t_{lidar} = t_{gps} = t_{I(B)}$.

In order to make full use of these data to generate reliable ground truth, three steps are performed as shown in Figure 20. Here comes the details of each step :

- **Step I. IMU Preintegration**

The purpose of Step I is to preintegrate the high-frequency data of the IMU to obtain trajectory data at the same frequency as the IMU (400Hz), which will serve as a benchmark for Step II. Based on the IMU noise model, the measurements of angular velocity and acceleration from IMU are defined as:

$$ {}^t\hat{\boldsymbol{\omega}}_{\boldsymbol{B}} = {}^t\boldsymbol{\omega}_{\boldsymbol{B}} {}^t\boldsymbol{b}_{\boldsymbol{\omega}} + {}^t\boldsymbol{n}_{\boldsymbol{\omega}} \tag{11} $$

$$ {}^t\hat{\boldsymbol{a}}_{\boldsymbol{B}} = \boldsymbol{R}^{\boldsymbol{B}}_{\boldsymbol{W}} \left({}^t\boldsymbol{a}_{\boldsymbol{B}} - \boldsymbol{g}_{\boldsymbol{W}}\right) + {}^t\boldsymbol{b}_{\boldsymbol{a}} + \boldsymbol{n}_{\boldsymbol{a}} \tag{12} $$

where $\boldsymbol{g_W}$ is the constant gravity vector in the world frame $\boldsymbol{W}$, and $\boldsymbol{R^B_W}$ is the rotation matrix from $\boldsymbol{W}$ to $\boldsymbol{B}$. Adding a slowly varying bias ${}^t\boldsymbol{b}$ and white noise $\boldsymbol{n}$ to the model corrects for ${}^t\boldsymbol{\omega_B}$ and ${}^t\boldsymbol{a_B}$. During time $t + \delta t$, velocity ${}^{t+\delta t}\boldsymbol{v_B} \in \mathbb{R}^3$, position ${}^{t+\delta t}\boldsymbol{p_B} \in \mathbb{R}^3$ and rotation ${}^{t+\delta t}\boldsymbol{q_B} \in \mathbb{H}$ are obtained by preintegrating (discrete calculus) the corrected acceleration and angular velocity measurements according to Newton's laws of motion.

The 1Hz LiDAR odometry is modified based on LOAM Zhang and Singh [2014] using Velodyne HDL-32E LiDAR data, which has good accuracy and robustness. Due to the long-term movement of the IMU will produce drift errors, the LiDAR odometry provides an initialization reference for the IMU preintegration trajectory per second, making the results more accurate. The transformation $T^{imu}_{lidar} \in SE(3)$ from $\boldsymbol{S_{lidar}}$ to $\boldsymbol{B}$ has been obtained by the method mentioned in Section 4.6.

The final result of the first step is a 400Hz IMU odometry, its final form is:

$$ \boldsymbol{x}_{\boldsymbol{B}_0} = \left[\boldsymbol{t}_{\boldsymbol{B}}\ \boldsymbol{p}_{\boldsymbol{B}_0}\ \boldsymbol{q}_{\boldsymbol{B}_0}\right]^T \tag{13} $$

- **Step II. Motion Correlation Analysis** The purpose of step II is to spatiotemporally align the laser tracker measurements with the body frame based on motion correlation analysis, as shown in figure 21. Our method is similar to Qiu et al. [2020]. In this step, we have to analyze two signals, the first is IMU odometry $\boldsymbol{x_{B_0}}$ gotten in step I, and the second is the measurement of laser tracking system $\boldsymbol{x_R} = \left[\boldsymbol{t_R}\ \boldsymbol{p^R_{S_{ball}}}\ \boldsymbol{po^B_{S_{ball}}}\right]^T$, $\boldsymbol{po^B_{S_{ball}}}$ indicates the initial transformation from $\boldsymbol{S_{ball}}$ to $\boldsymbol{B}$ estimated from the size of the mechanical installation. For the two signals, since the laser tracker and the bucket truck are kept horizontal to the ground using a tripod and hydraulic legs respectively

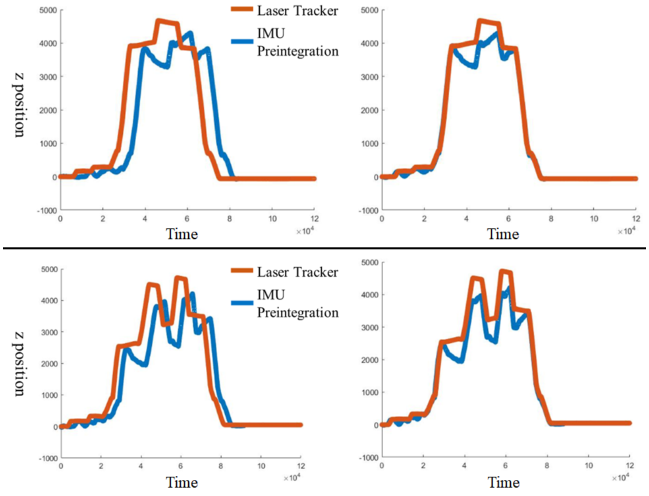


Figure 21. Temporal alignment of IMU preintegration trajectories with laser tracker trajectories via motion correlation analysis in Step II.

during the measurement process, the motion in the z direction already has a high correlation without rotation estimate. Therefore, the time delay analysis is performed by calculating the cross-correlation between $z_{B_0}(t_B)$ and $z_R(t_R)$. We enumerate the time offset, and the maximum of the cross-correlation function indicates the point in time where the signals are best aligned:

$$\tau_{\text{delay}} = \underset{t_d \in \mathbb{R}}{\arg\max}((z_{B_0} \star z_R)(t_d)) \tag{14}$$

After finishing this step state $\boldsymbol{x_R}$ is aligned to $\boldsymbol{t_B}$. Here we get:

$$\boldsymbol{x_{B_1}} = \left[\boldsymbol{t_B}\ \boldsymbol{p^{R}_{S_{ball}}}\ \boldsymbol{po^{B}_{S_{ball}}}\right]^T \tag{15}$$

Finally, the raw data and states prepared for step III is as followed $(\boldsymbol{S_b} = \boldsymbol{S_{ball}})$:

$$\boldsymbol{x_{raw}} = \left[\boldsymbol{t_B}\ \boldsymbol{p^{R}_{S_b}}\ \boldsymbol{po^{B}_{S_b}}\ \boldsymbol{q^{W}_{B}}\ \boldsymbol{g_W}\ \boldsymbol{a_B}\ \boldsymbol{\omega_B}\ \boldsymbol{b_{a,\omega}}\ \boldsymbol{n_{a,\omega}}\right]^T \tag{16}$$

- **Step III. Factor Graph Optimization**

The final step is to use the known measurements and states to generate an optimal estimate of all states of the ground truth in the body frame $\boldsymbol{B}$. Our method is modified based on the vicon2gt Geneva and Huang [2020] toolbox. Specifically, the following states are estimated:

$$\boldsymbol{x} = \left[\boldsymbol{x_{B_1}} \ldots \boldsymbol{x_{B_N}}\ \boldsymbol{x_C}\ \boldsymbol{\bar{q}^{R}_{W}}\ \right]^T \tag{17}$$

$$\boldsymbol{x_{B_i}} = \left[\boldsymbol{\bar{q}^{B_i}_{R}}\ \boldsymbol{p^{R}_{B_i}}\ \boldsymbol{v^{R}_{B_i}}\ \boldsymbol{b_{a,i}}\ \boldsymbol{b_{\omega,i}}\ \right]^T \tag{18}$$

$$\boldsymbol{x_C} = \left[\boldsymbol{\bar{q}^{B}_{S_b}}\ \boldsymbol{p^{B}_{S_b}}\ \Delta t^R_B\right]^T \tag{19}$$

Here we are estimating N inertial states at laser tracker frequency 50Hz, along with a calibration state $\boldsymbol{x_C}$ containing the spatial-temporal parameters between the target ball frame $\boldsymbol{S_b}$ and IMU frame $\boldsymbol{B}$, and $\boldsymbol{\bar{q}^{R}_{W}}$ the rotation between the global laser tracking frame $\boldsymbol{R}$ and global inertial frame $\boldsymbol{W}$. $\boldsymbol{\bar{q}^{B_i}_{R}}$ is the unit quaternion parameterizing the rotation from the global laser tracking frame of reference $\boldsymbol{R}$ to the IMU local frame $\boldsymbol{B_i}$ at time t_i. $\boldsymbol{p^{R}_{B_i}}$ and $\boldsymbol{v^{R}_{B_i}}$ are the position and velocity of the IMU body frame $\boldsymbol{B}$ expressed in the global laser tracking frame $\boldsymbol{R}$, respectively. $\boldsymbol{b_{a,i}}$ and $\boldsymbol{b_{\omega,i}}$ are the biases of accelerometer and gyroscope. Δt^R_B is the time offset between the laser tracking system and the IMU body frame that we further estimate on the basis of step II. The inertial state $\boldsymbol{x_{B_i}}$ lies on the manifold defined by the product of the unit quaternions $\mathbb{H}$ with the vector space $\mathbb{R}^{12}$ (i.e. $\mathcal{M} = \mathbb{H} \times \mathbb{R}^{12}$) and has 15 DoF.

An overview of the nonlinear factor graph we solved is shown in Figure 20.

The final ground truth is available in TUM format Sturm et al. [2012]:

```
timestamp tx ty tz qx qy qz qw
1654673708.251979 -0.000063 0.000072 -0.000068
-0.050800 -0.025895 0.001970 0.998371
...
```

5 Evaluation and Baselines

We run some state-of-the-art baselines on several sensor suites and data sequences to illustrate the characteristics and challenges of our dataset. The absolute trajectory error (ATE, as defined in Sturm et al. [2012]) is used as the indicator to measure the effect of the SLAM algorithms. To ensure fairness, we carefully tuned the parameters for the algorithms evaluated on each data sequence to make the results of each algorithm close to their best. And parameters of the algorithms using the same sensor suite are set to be exactly the same, the online parameter estimation of some algorithms will not be enabled. Table 4 summarizes necessary information and the corresponding results. The algorithms evaluated are run on a PC with Ubuntu 18.04 operating system, ROS melodic, Intel® Core™ i7-8750H CPU @ 2.20GHz, and 16GB RAM.

For the evaluation of visual SLAM, we have tested several state-of-the-art algorithms on different sensor suites and data sequences. These include ORB-SLAM3 Campos et al. [2021] on two monocular cameras, ORB-SLAM3 and VINS-Mono Qin et al. [2018] on two monocular-inertial systems, and ORB-SLAM3 and VINS-Fusion on a stereo and stereo-inertial system. Our stereo camera consists of three cameras, resulting in three stereo pairs in total. In this case, we have chosen the left and right cameras with the longest baseline of 24cm.

As shown in the table, we have tested the same algorithm on two different monocular cameras because they are complementary to each other. As shown in Figure 22, the Bumblebee-XB3 camera has a larger field of view and higher sensitivity to light compared to the Hikvision camera and can capture more environmental information in the darkness, but it may produce glare in strong light. The Hikvision camera has the opposite strengths and weaknesses. We hope that at least one camera can provide a good environment perception for visual SLAM in each working environment. At the same time, we are analyzing why the accuracy of visual SLAM decreases in some sequences by comparing the results on the two cameras. For example, in the hf005 sequence, the accuracy of ORB-SLAM3 on the Bumblebee-XB3 camera is significantly lower than on the Hikvision camera and fails halfway, which is probably due to glare from the xb3

Table 4. ATE (m) of state-of-the-art SLAM methods over USTC FLICAR datasets. * points out that runs have not been successful on less than 1/2 of a sequence. × points out that runs have not been successful on more than 1/2 of a sequence. — points out that data is not available in a sequence. The ATE in this table is calculated using evo Grupp [2017].

Sequence	Sensor Suite	hf001	hf002	hf003	hf004	hf005	hf006	hf007	hf008	hf009
Time		15:35	15:48	16:14	16:30	17:20	18:13	19:01	21:20	21:27
ORB-SLAM3	Hikcam	0.097	0.028	0.093	0.081	0.163	0.144*	0.084	×	×
(Visual)	Xb3-C	0.144	0.182	0.106	0.113	0.417*	0.178	0.126	×	×
ORB-SLAM3	Hikcam+Xsens	0.017	0.181	0.090	0.082	0.086*	0.118	0.078	×	×
(V-Mono-Inertial)	Xb3-C+Xsens	0.150	0.276	0.156	0.116	0.159*	0.184*	0.407	×	×
VINS-Mono	Hikcam+Xsens	0.146	0.291	0.332	0.092	0.204	0.232	0.131	1.381	1.794
(V-Mono-Inertial)	Xb3-C+Xsens	0.108	0.166	0.163	0.089	0.156	0.150	0.125	0.074	0.116
ORB-SLAM3 (Stereo)	Xb3-L/R	0.150	0.184	0.135	0.117	0.215	0.200	0.137	×	×
VINS-Fusion (Stereo)	Xb3-L/R	0.155	0.211	0.147	0.097	0.213	0.250	0.131	0.154	1.277
ORB-SLAM3 (V-Stereo-Inertial)	Xb3-L/R + Xsens	0.406	0.184	0.205	0.136	0.226	0.282	0.218	0.568*	0.551*
VINS-Fusion (V-Stereo-Inertial)	Xb3-L/R + Xsens	0.116	0.192	0.096	0.089	0.164	0.161	0.166	0.059	0.136
A-LOAM (Horizontal-LiDAR)	Velo-HDL32	0.089	0.118	0.091	0.078	0.171	0.115	0.078	0.053	0.062
A-LOAM (Vertical-LiDAR)	Velo-VLP32	0.265	0.278	0.526	—	0.423	0.481	0.489	0.507	0.644
LeGO-LOAM (Horizontal-LiDAR)	Velo-HDL32	0.094	0.121	0.075	0.081	0.165	0.116	0.080	0.046	0.059
LeGO-LOAM (Vertical-LiDAR)	Velo-VLP32	0.698	1.528	1.047	—	0.771	0.484	1.214	1.962	1.325
LIO-SAM (H-LiDAR-Inertial)	Velo-HDL32 + Xsens	0.086	0.114	0.075	0.079	0.161	0.111	0.073	0.042	0.054
FAST-LIO (H-LiDAR-Inertial)	Velo-HDL32 + Xsens	0.088	0.115	0.081	0.078	0.168	0.117	0.075	0.052	0.059
FAST-LIO (H-MEMS-Inertial)	LiVOX-Avia + Internal IMU	0.063	0.069	0.111	0.079	0.147	0.119	0.082	0.050	0.060
Duration (s)		192.5	217.8	217.1	155.9	260.4	230.6	207.6	210.6	238.7
Length (m)		26.46	33.50	34.26	24.10	22.82	33.90	34.32	30.78	35.42
Avg. Vel./ (m/s)		0.137	0.154	0.158	0.155	0.088	0.147	0.165	0.146	0.148
Ang. Vel. (°/s)		0.537	0.529	0.503	0.660	0.556	0.614	0.612	0.574	0.148

camera. Similarly, in the hf009 sequence, the accuracy of VINS-Mono on the Bumblebee-XB3 camera is significantly higher than on the Hikvision camera, which is because the Bumblebee-XB3 camera provides more environmental information in the darkness.

Overall, the accuracy of the ORB-SLAM3 method is slightly higher than the VINS-based SLAM method in sequences with good lighting conditions (no glare, no darkness), consistent with the experimental results reported in their paper, which is based on the EuRoC dataset. However, the robustness of the VINS-based SLAM method is significantly higher in extreme light and dark environments, and both can maintain the same accuracy as in good lighting conditions. For example, in the hf008 and hf009 night sequences, we used the Bumblebee-XB3 camera which can image effectively in the darkness. However, the ORB-SLAM3 algorithm still failed due to its inability to extract features. On the other hand, both VINS-Mono and VINS-Fusion were able to maintain the same accuracy as the situation in good lighting conditions.

Besides, the accuracy and robustness of visual SLAM under semi-failures can be improved by tightly coupling a well-calibrated IMU. For example, in hf009 night sequence with a wide range of rotation, the ATE of VINS-Fusion that only uses a stereo camera is 1.277 m. If IMU data is used, the ATE of VINS-Fusion will decrease to 0.136 m.

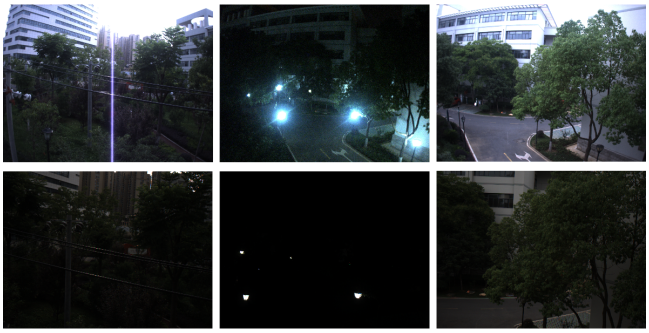


Figure 22. The different performances of the two cameras under complex lighting conditions. The upper part is the Bumblebee XB3 center camera, and the lower part is the hikvison camera. The upper and lower pairs are at the same moment.

For the evaluation of LiDAR SLAM, we tested A-LOAM (an implementation of LOAM Zhang and Singh [2014] modified by Qin et al.) and LeGO-LOAM Shan and Englot [2018] on two 32-beam Velodyne LiDARs, one horizontal and one vertical. We also used a LiDAR-Inertial system consisting of a horizontal Velodyne HDL-32E LiDAR and an

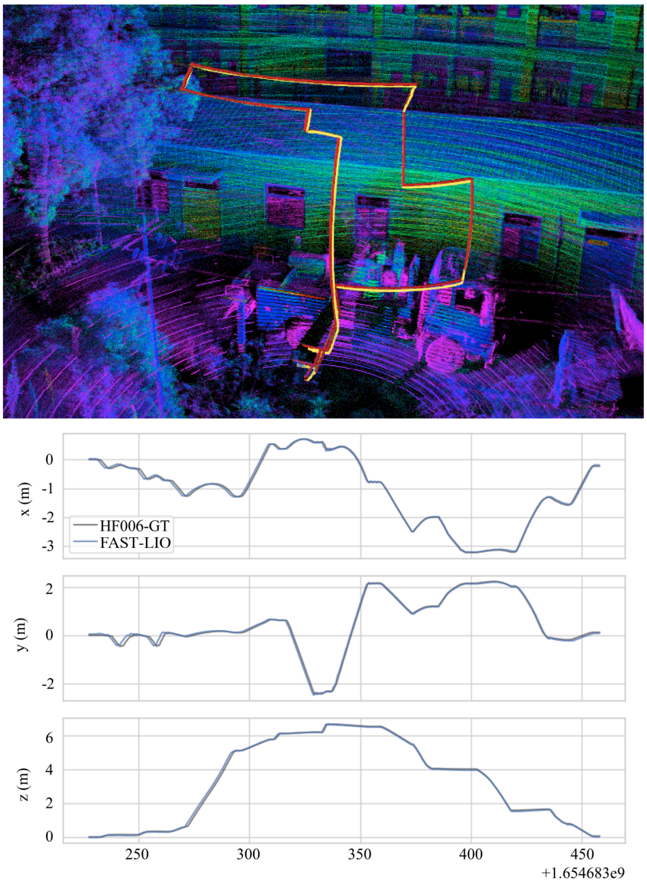


Figure 23. The ground truth is marked in red and the SLAM trajectory is marked in yellow. The figure shows the running results of Fast-LIO on the hf006 sequence.

Xsens IMU to test LIO-SAM Shan et al. [2020] and FAST-LIO Xu and Zhang [2021]. FAST-LIO was also tested on a LiVOX MEMS LiDAR.

The LiDAR SLAM algorithms that were tested on data sequences from a horizontal LiDAR achieved good results, with the ATE of approximately 0.1 meters. The results are reported in Figure 23 Additionally, the accuracy of LiDAR SLAM can be further improved by tightly coupling a well-calibrated IMU. This suggests that among current sensing technologies, LiDAR is relatively reliable for the localization and mapping of aerial work robots.

However, LiDAR is not without its flaws. Issues have been observed when using the vertical LiDAR. It is necessary for aerial work robots to have 360-degree perception in the vertical direction, as obstacles may come from any direction while in motion. Merely perceiving in the horizontal direction is not sufficient. In this comparison, we have chosen algorithms that use a single LiDAR to minimize the influence of calibration errors. As shown in the table, the ATE of A-LOAM on the vertical LiDAR is 3 to 10 times higher than on the horizontal LiDAR in the same sequence, despite slight differences in the LiDAR models. This experimental result is convincing enough. When the field of view rotates 180 degrees, the world seen by the LiDAR is very different. For the lower half of the LiDAR, the ground and walls occupy most of the point cloud; for the upper half of the LiDAR, only scattered small objects such as wires and branches in the air can reflect LiDAR echoes, and most of the lasers disappear in the sky. This is a typical scenario where LiDAR odometry fails — low texture. Besides, due to the large range of rotations of the aerial platform, the overlap between adjacent LiDAR frames is small during rotations, which is not conducive to the continuous tracking of features. This is evident in sequences hf003, hf007, hf008, and hf0009, which all have large rotations and therefore larger ATE values compared to other sequences. The aerial platform itself can also act as a dynamic object and cause interference. Among the algorithms tested, LeGO-LOAM tested on the vertical LiDAR performed the worst because it is difficult to extract the ground plane from the point cloud during aerial motion. In this case, optimization for ground motion deteriorates the accuracy of the algorithm.

Overall, the results of the evaluations show that LiDAR-based SLAM algorithms tend to have good accuracy, with an ATE of approximately 0.1 meters when using a horizontal LiDAR. The accuracy of visual SLAM algorithms varied depending on the sensor and the lighting conditions, with ORB-SLAM3 generally performing better in good lighting conditions and VINS-based algorithms being more robust in extreme lighting conditions. The accuracy of visual SLAM can also be improved by tightly coupling a well-calibrated IMU. LiDAR-based SLAM algorithms tended to have difficulty in scenarios with low texture or large rotations, with the ATE increasing significantly in these cases.

We did not aim to intentionally make it difficult for these algorithms. To verify whether the accuracy of the algorithms decreases under faster or more aggressive movements, you can try downsampling the data and running the algorithms to simulate faster speeds. We have only evaluated a select number of classic SLAM methods in this study. More recent SLAM methods, such as LVI-SAM Shan et al. [2021], which combines VINS-Mono and LIO-SAM, may improve upon these classic methods.

6 Semantic Annotations

In this section, we will present the Semantic FLICAR dataset, including its generation process, content, and distinctive features. This dataset can be downloaded from the following website:https://ustc-flicar.github.io/semantic/.

6.1 Annotation Process

The rapid progress in artificial intelligence (AI) and the foundation AI Large Vision Models (LVMs) has opened up new possibilities for collaborative data annotation between human annotators and AI systems, resulting in significant improvements in data annotation and production efficiency.

Building upon the capabilities of the Segment Anything Model (SAM) Kirillov et al. [2023], we present the Semantic FLICAR dataset, which provides fine-grained image semantic segmentation and 128-beam LiDAR ring signal scene segmentation for the USTC FLICAR dataset. The Semantic FLICAR dataset offers detailed annotations that greatly enhance our understanding of object boundaries and semantic information in aerial work scenes, making it an

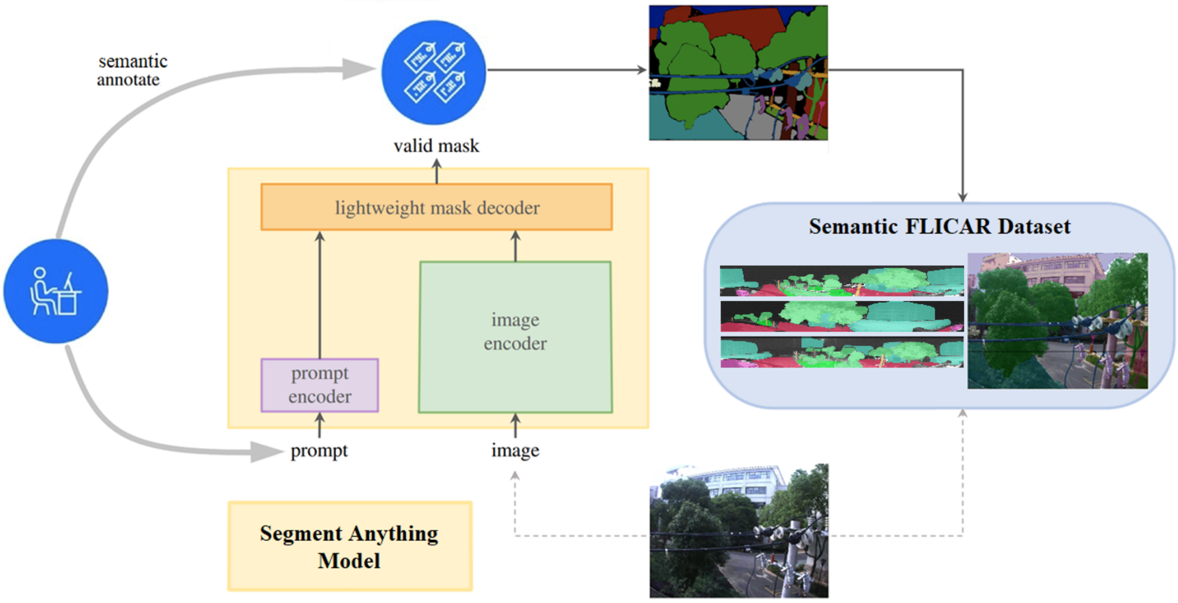


Figure 24. Semantic annotation data engine based on SAM: Semantic FLICAR dataset involves the initial generation of global mask segmentations by the Segment Anything Model (SAM), followed by collaborative refinement by human annotators through interactive prompts for regions of interest, resulting in precise and detailed annotations.

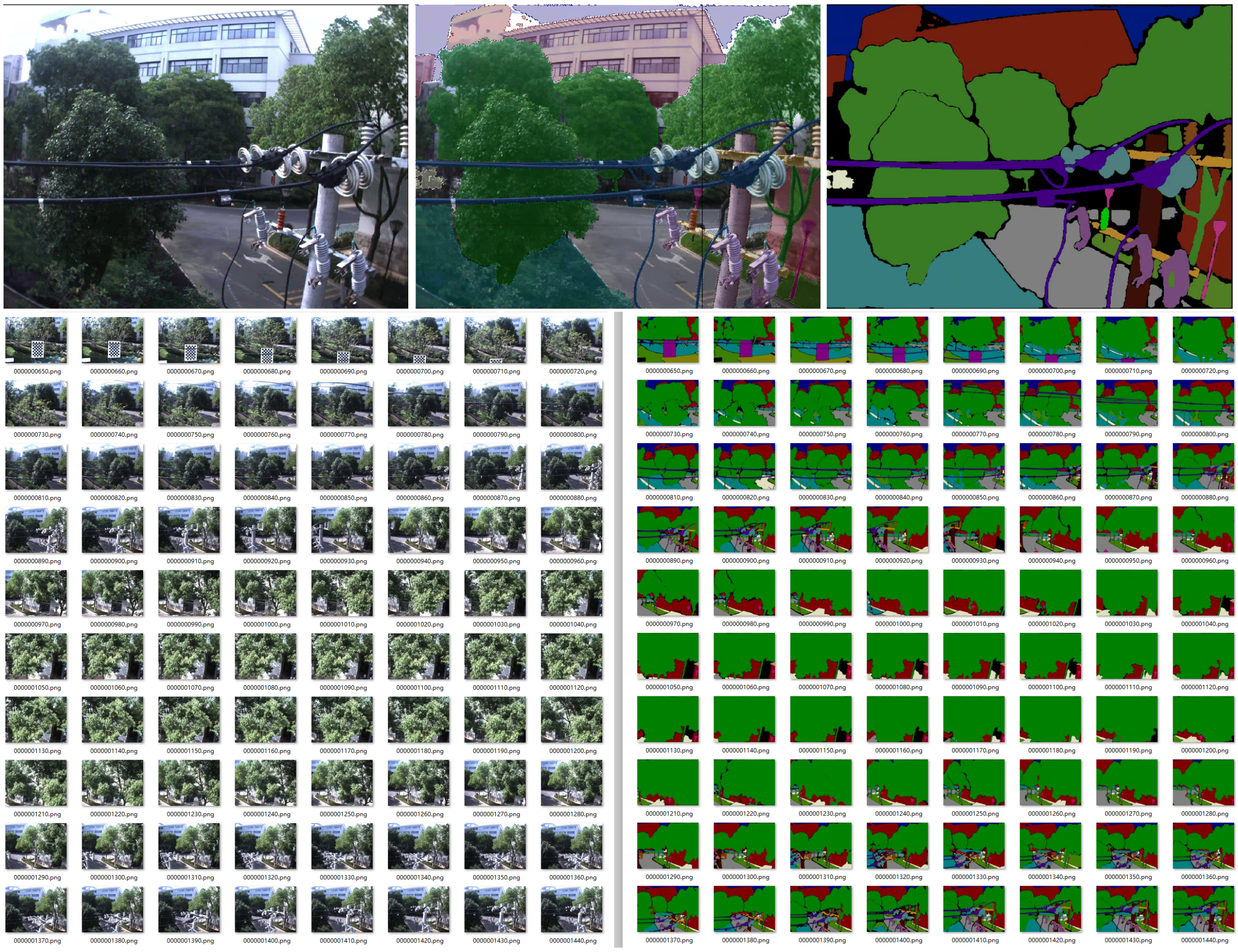
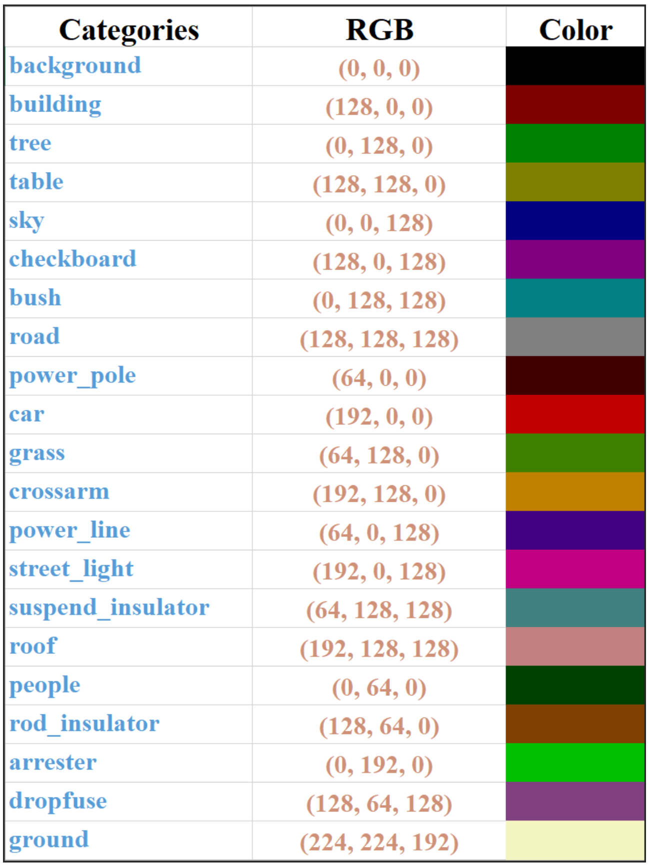


Figure 25. Bumblebee XB3 center image semantic segmentation preview in HF001 sequence

invaluable resource for various aerial work computer vision applications.

The data annotation process for the Semantic FLICAR dataset is illustrated in Figure 24. Initially, the USTC FLICAR dataset is input into the Segment Anything



Categories	RGB	Color
background	(0, 0, 0)	⬛
building	(128, 0, 0)	🟥
tree	(0, 128, 0)	🟩
table	(128, 128, 0)	🟨
sky	(0, 0, 128)	🟦
checkboard	(128, 0, 128)	🟪
bush	(0, 128, 128)	🟦
road	(128, 128, 128)	⬜
power_pole	(64, 0, 0)	🟫
car	(192, 0, 0)	🟥
grass	(64, 128, 0)	🟩
crossarm	(192, 128, 0)	🟧
power_line	(64, 0, 128)	🟪
street_light	(192, 0, 128)	🟪
suspend_insulator	(64, 128, 128)	🟦
roof	(192, 128, 128)	🟫
people	(0, 64, 0)	🟩
rod_insulator	(128, 64, 0)	🟫
arrester	(0, 192, 0)	🟩
dropfuse	(128, 64, 128)	🟪
ground	(224, 224, 192)	🟨

Figure 26. Semantic label categories for Bumblebee XB3 center image data

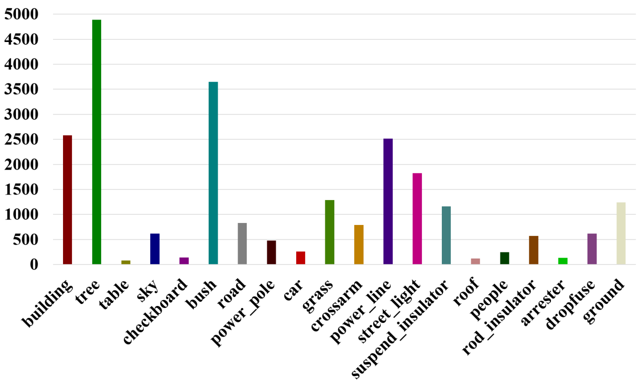
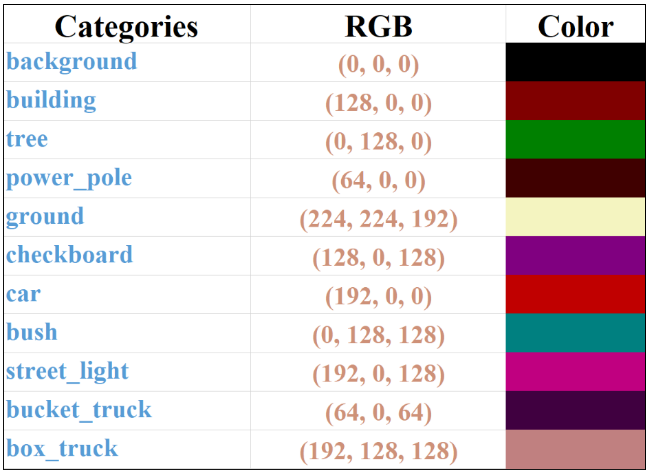


Figure 27. Semantic maskes annotation statistics for Bumblebee xb3 center image data

Model (SAM), automatically generating global mask segmentations. This step leverages SAM's efficiency and promptability to produce initial annotations quickly. Besides, human annotators actively participate in the process to further improve the quality and precision of the annotations. Human annotators interact with the initial masks generated by SAM and provide prompts on regions of interest. This collaborative approach allows the annotators to refine the boundaries and assign accurate semantic information, resulting in more precise annotations. The combination of AI-generated masks and human expertise ensures that the Semantic FLICAR dataset achieves a high level of accuracy and detail.



Categories	RGB	Color
background	(0, 0, 0)	⬛
building	(128, 0, 0)	🟥
tree	(0, 128, 0)	🟩
power_pole	(64, 0, 0)	🟫
ground	(224, 224, 192)	🟨
checkboard	(128, 0, 128)	🟪
car	(192, 0, 0)	🟥
bush	(0, 128, 128)	🟦
street_light	(192, 0, 128)	🟪
bucket_truck	(64, 0, 64)	🟪
box_truck	(192, 128, 128)	🟫

Figure 28. Semantic label categories for Ouster OS0-128 signal image data

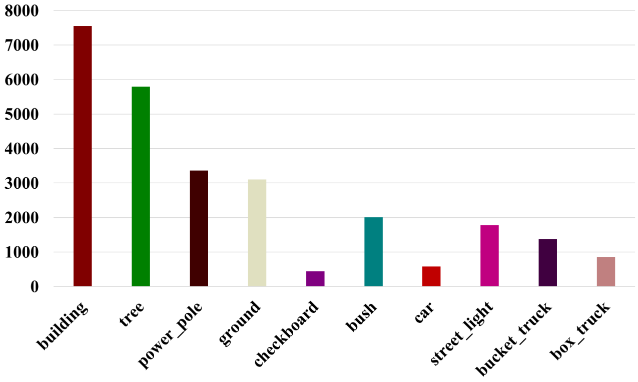


Figure 29. Semantic maskes annotation statistics for Ouster OS0-128 signal image data

6.2 Semantic FLICAR Dataset

The semantic segmentation data of Semantic FLICAR is provided in the format of the VOC segmentation dataset. In total, 1,774 camera keyframes and 1,426 LiDAR keyframes were annotated at 1 Hz, resulting in 56,889 unique semantic masks.

When annotating camera images, the higher resolution allows for the definition of more diverse and fine-grained object categories. As depicted in Figure 26, the annotation of the Bumblebee xb3 center image in the HF0XX sequence includes 20 distinct object categories, covering various objects commonly found in aerial work scenarios. The statistical charts for each category label can be found in Figure 27. For a preview of the annotated camera dataset, please refer to Figure 25.

Keyframe annotation for Ouster LiDAR is typically paired with camera data. This means that the closest timestamped pair of camera and LiDAR data will be annotated. Unlike camera images, LiDAR has a lower resolution but a wide field of view. In our system, we choose to annotate using the highest-resolution Ouster OS0-128 LiDAR. The annotation is performed on the signal image data, which is a panoramic image with a resolution of 1024×128. This signal image is derived from the near-infrared signals and environmental data captured by the Ouster LiDAR. It is spatially aligned with other data layers of the Ouster LiDAR, such as point

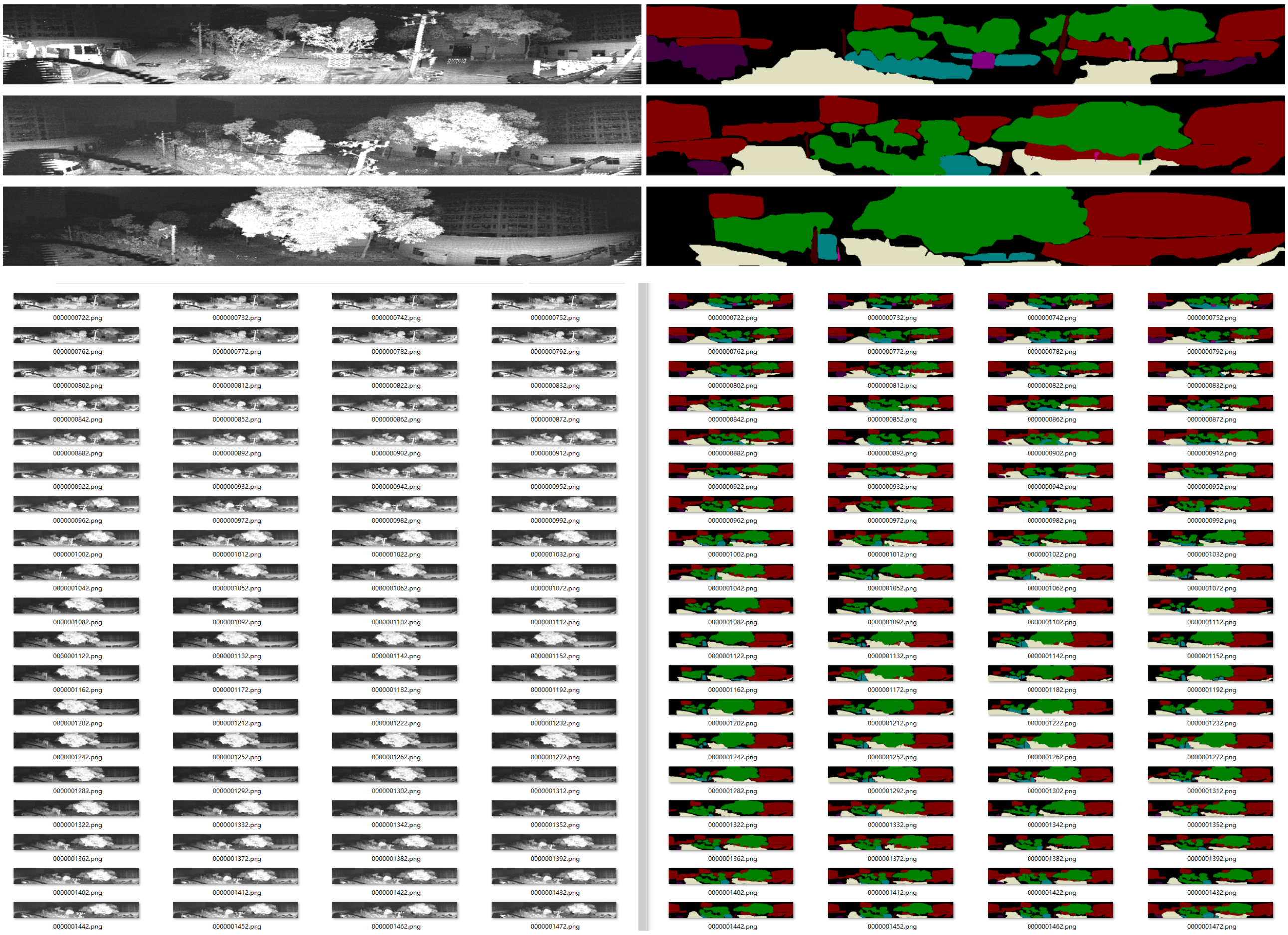


Figure 30. Ouster OS0-128 signal image semantic segmentation preview in HF001 sequence

clouds, which is perfectly spatially correlated, with zero temporal mismatch or shutter effects, and has 16 bits per pixel and linear photo response. As depicted in Figure 28, the annotation of the Ouster OS0-128 signal image includes 10 distinct object categories. The statistical charts for each category label can be found in Figure 29. For a preview of the annotated LiDAR dataset, please refer to Figure 30.

6.3 Data Association and Augmentation

Unlike datasets that focus only on semantic segmentation tasks, the Semantic FLICAR dataset is built upon the USTC FLICAR dataset. Therefore, the previously mentioned sensor calibration and SLAM-related content will play a crucial role in data association and augmentation for the Semantic FLICAR dataset.

As depicted in Figure 31, the segmentation annotations of the Bumblebee XB3 center image can be mapped to the left image and right image using the calibration results of the stereo camera. This mapping results in two additional stereo perspectives, effectively tripling the number of labels. In Figure 32, the segmentation annotations of the Bumblebee XB3 center image can be mapped to the Velodyne HDL32 point cloud using the joint calibration results of the laser radar and the camera. This mapping process assists in incorporating semantic information into the low-resolution 32-beam LiDAR system.

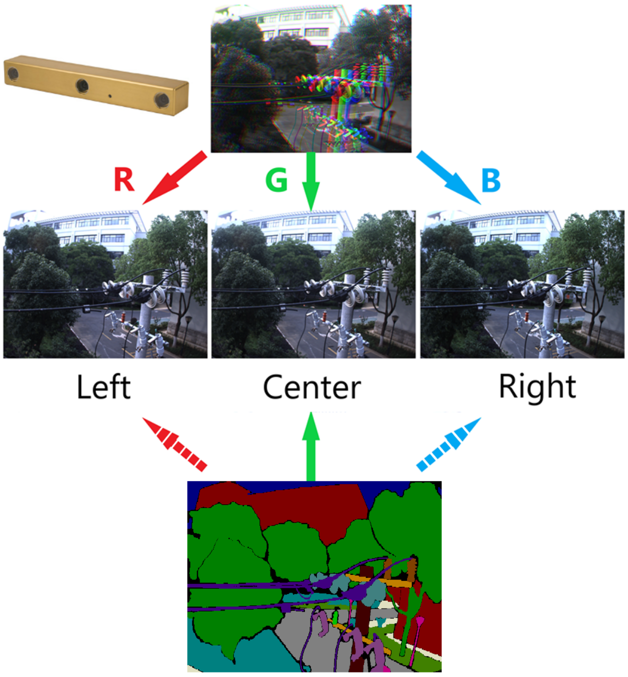


Figure 31. Data association and augmentation based on stereo camera calibration results in section 4.3

From the perspective of data association, SLAM is capable of linking together various elements such as scenes, objects, locations, and timestamps during the process of simultaneous

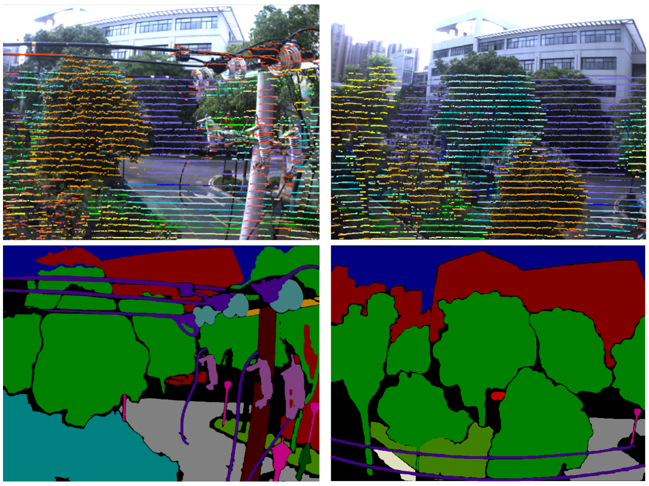


Figure 32. Data association and augmentation based on LiDAR camera calibration results in section 4.7

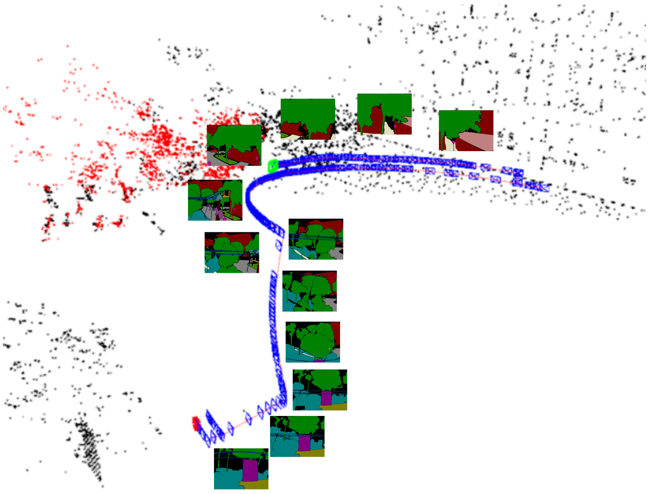


Figure 33. Data association and augmentation based on SLAM results in section 5

localization and mapping. As shown in Figure 33, when employing the Bumblebee XB3 center camera for ORB-SLAM3, image frames are arranged according to their corresponding time and location. Therefore, for all image data collected by the Bumblebee camera at a rate of 10Hz, it can be associated based on the results of SLAM, allowing for annotations to be inserted between keyframes marked at a rate of 1Hz. This slightly compromises fidelity but greatly doubles the number of annotated frames.

The Semantic FLICAR dataset encompasses a significant number of annotated images, with comprehensive fine-grained semantic segmentation and 128-beam Ouster LiDAR ring signal image segmentation. This dataset offers researchers a rich resource for training, evaluating, and advancing algorithms and models in various computer vision tasks, including object recognition, scene segmentation and understanding, and autonomous aerial work applications. The impact of the Semantic FLICAR dataset extends to practical applications as well. The meticulous annotations within this dataset can contribute to advancements in autonomous aerial work systems, robotics, and other domains that rely on precise scene analysis and understanding. By fostering research in these areas, the Semantic FLICAR dataset plays a crucial role in the development of safer and more intelligent AI systems for aerial work tasks.

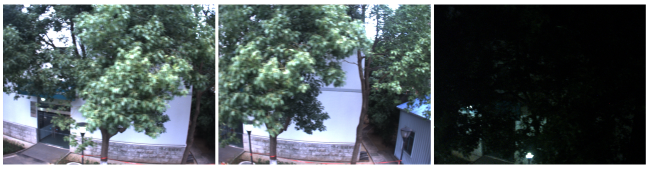


Figure 34. Motion blur in poor lighting conditions, data form HF007 and HF008

7 Known Issues

Despite careful design and execution of the data collection experiments, we are aware of various issues that present additional challenges in processing and limit the achievable accuracy when compared to ground truth.

These are the issues that users will need to be aware of:

- Camera overexposure in direct sunlight and insufficient lighting at night, as shown in Figure 22: In certain conditions, such as direct sunlight, the camera may experience overexposure, resulting in washed-out or overly bright images. Conversely, during nighttime or low-light situations, the lighting may be inadequate, leading to underexposed images with limited visibility. Additionally, there might be the presence of lens flares or halos, which can occur when strong light sources are present in the frame.
- Motion blur in poor lighting conditions, as shown in Figure 34: In challenging lighting conditions where the illumination is insufficient, the camera extends the exposure time to capture more light. As a result, when there is fast motion, such as large-scale rotations, motion blur may occur in the captured images. This motion blur can lead to reduced image clarity and potentially impact the accuracy of certain computer vision tasks.
- Attention to timestamps in Ouster and LiVOX LiDAR rostopics: Please pay attention to the timestamps recorded within the Ouster and LiVOX LiDAR rostopics. During the entire experimental process, the availability of GPS signals cannot be guaranteed at all times. In situations where the GPS signal is interrupted, the driver programs of these two LiDAR sensors may record the time elapsed since startup as timestamps instead of UTC time. To ensure accurate UTC timestamp synchronization, it is recommended to use the timestamps provided by the rostopic as a reference, as they record the computer system time. Both the computers that record Ouster and LiVOX LiDAR sensor data synchronize their time using the same server through NTP with others.
- Attention to the accuracy of semantic segmentation label edges: Please pay attention to the accuracy of the semantic segmentation labels, particularly at the edges. The semantic labels in our Semantic FLICAR dataset were generated by prompting SAM. As a result, the semantic segmentation labels may not achieve pixel-level accuracy, especially at the edges of objects. Additionally, the pixel accuracy of the labels in the Ouster OS0-128 signal images may be lower compared to the visible light images due to the lower resolution of the Ouster LIDAR sensor.

8 Summary and future work

The USTC FLICAR is a unique dataset that focuses on the task of heavy-duty aerial work, featuring a special aerial platform, the bucket truck, which allows for greater payload capacity and stationarity compared to traditional drones. It is also the most sensor-rich aerial dataset to date, with a wide range of sensors including seven cameras, four 3D LiDARs, and three IMUs, which covers 360 degrees of horizontal and vertical views. This dataset is designed to enable aerial work robots to effectively interact with complex aerial work environments, with millimeter-level outdoor ground truth obtained using a laser tracker. It is our hope that this dataset will serve as a valuable benchmark for evaluating the performance of various algorithms in this field, and inspire researchers to design sensor suites specifically tailored for autonomous aerial work systems. Moreover, the experimental results on our dataset also demonstrate that the novel combination of an autonomous driving sensing kit and bucket truck is a general autonomous aerial platform with high potential for various aerial work tasks.

Looking ahead to the future, our goal is to expand into multiple domains and further advance the practical application of aerial robotics. One key aspect is to enrich and refine the semantic annotations based on existing labeled data, which will enable the aerial robots to have a more accurate and detailed understanding of the environment and its objects. Additionally, we plan to incorporate more sensors and explore new sensor configurations. For instance, our aim is to include infrared night vision cameras and downward-facing cameras to enhance the dataset's capabilities. These additional features will significantly improve the overall abilities of the aerial robots. Furthermore, we intend to continue collecting new data to cover a broader range of scenarios and environments. This approach allows us to enhance the robustness and generality of our algorithms and models. By continuously expanding our dataset, we can improve the effectiveness and reliability of aerial robots in various real-world situations. In conclusion, we firmly believe that the USTC FLICAR dataset represents an important milestone in making aerial work safer, more efficient, and more accessible. Through our continuous efforts, our goal is to push the boundaries of innovation and contribute to the ongoing progress of aerial robotics technology.

Acknowledgements

The authors would like to acknowledge Yonggang Gu at the Experimental Center of Engineering and Material Sciences of USTC for his assistance in training and providing the laser tracking equipment. The authors would like to express our gratitude to STATE GRID ANHUI ELECTRIC POWER CO. LTD. for their support in providing us with a bucket truck. The author would like to express gratitude to Meta AI Research, FAIR for generously open-sourcing the Segment Anything project. This invaluable contribution played a pivotal role in enabling us to successfully construct the Semantic FLICAR dataset. The authors gratefully acknowledge support from the National Key R&D Program of China (No. 2018YFB1307403), the Fundamental Research Funds for the Central Universities and the Cyrus Tang Foundation.

References

Barnes, D., Gadd, M., Murcutt, P., Newman, P. and Posner, I. [2019], 'The Oxford Radar RobotCar Dataset: A Radar Extension to the Oxford RobotCar Dataset', *arXiv preprint arXiv: 1909.01300* .
URL: *https://arxiv.org/pdf/1909.01300*

Beltrán, J., Guindel, C., de la Escalera, A. and García, F. [2022], 'Automatic Extrinsic Calibration Method for LiDAR and Camera Sensor Setups', *IEEE Transactions on Intelligent Transportation Systems* .

Biber, P. and Straßer, W. [2003], The normal distributions transform: A new approach to laser scan matching, *in* 'Proceedings 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003)(Cat. No. 03CH37453)', Vol. 3, IEEE, pp. 2743–2748.

Burri, M., Nikolic, J., Gohl, P., Schneider, T., Rehder, J., Omari, S., Achtelik, M. W. and Siegwart, R. [2016], 'The EuRoC micro aerial vehicle datasets', *The International Journal of Robotics Research* **35**(10), 1157–1163.

Caesar, H., Bankiti, V., Lang, A. H., Vora, S., Liong, V. E., Xu, Q., Krishnan, A., Pan, Y., Baldan, G. and Beijbom, O. [2020], nuScenes: A multimodal dataset for autonomous driving, *in* 'Proceedings of the IEEE/CVF conference on computer vision and pattern recognition', pp. 11621–11631.

Campos, C., Elvira, R., Rodríguez, J. J. G., Montiel, J. M. and Tardós, J. D. [2021], 'ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual–Inertial, and Multimap SLAM', *IEEE Transactions on Robotics* **37**(6), 1874–1890.

Carlevaris, N., Ushani, A. K. and Eustice, R. M. [2016], 'University of Michigan North Campus long-term vision and lidar dataset', *The International Journal of Robotics Research* **35**(9), 1023–1035.

Delmerico, J., Cieslewski, T., Rebecq, H., Faessler, M. and Scaramuzza, D. [2019], Are we ready for autonomous drone racing? the UZH-FPV drone racing dataset, *in* '2019 International Conference on Robotics and Automation (ICRA)', IEEE, pp. 6713–6719.

Geiger, A., Lenz, P., Stiller, C. and Urtasun, R. [2013], 'Vision meets Robotics: The KITTI Dataset', *International Journal of Robotics Research (IJRR)* .

Geneva, P. and Huang, G. [2020], vicon2gt: Derivations and Analysis, Technical Report RPNG-2020-VICON2GT, University of Delaware. Available: http://udel.edu/~ghuang/papers/tr_vicon2gt.pdf.

Grupp, M. [2017], 'evo: Python package for the evaluation of odometry and SLAM.', https://github.com/MichaelGrupp/evo.

Huang, A. S., Antone, M., Olson, E., Fletcher, L., Moore, D., Teller, S. and Leonard, J. [2010], 'A high-rate, heterogeneous data set from the darpa urban challenge', *The International Journal of Robotics Research* **29**(13), 1595–1601.

Jeong, J., Cho, Y., Shin, Y.-S., Roh, H. and Kim, A. [2019], 'Complex urban dataset with multi-level sensors from highly diverse urban environments', *The International Journal of Robotics Research* **38**(6), 642–657.

Kirillov, A., Mintun, E., Ravi, N., Mao, H., Rolland, C., Gustafson, L., Xiao, T., Whitehead, S., Berg, A. C., Lo, W.-Y. et al. [2023], 'Segment anything', *arXiv preprint arXiv:2304.02643* .

Li, X., Xiao, Y., Wang, B., Ren, H., Zhang, Y. and Ji, J. [2022], 'Automatic targetless LiDAR–camera calibration: a survey', *Artificial Intelligence Review* pp. 1–39.

Maddern, W., Pascoe, G., Linegar, C. and Newman, P. [2017], '1 year, 1000 km: The Oxford RobotCar dataset', *The International Journal of Robotics Research* **36**(1), 3–15.

Majdik, A. L., Till, C. and Scaramuzza, D. [2017], 'The Zurich urban micro aerial vehicle dataset', *The International Journal of Robotics Research* **36**(3), 269–273.

Nguyen, T.-M., Yuan, S., Cao, M., Lyu, Y., Nguyen, T. H. and Xie, L. [2022], 'Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint', *The International Journal of Robotics Research* **41**(3), 270–280.

Olson, E. [2011], AprilTag: A robust and flexible visual fiducial system, *in* '2011 IEEE international conference on robotics and automation', IEEE, pp. 3400–3407.

Pandey, G., McBride, J. R. and Eustice, R. M. [2011], 'Ford campus vision and lidar data set', *The International Journal of Robotics Research* **30**(13), 1543–1552.

Pire, T., Mujica, M., Civera, J. and Kofman, E. [2019], 'The Rosario dataset: Multisensor data for localization and mapping in agricultural environments', *The International Journal of Robotics Research* **38**(6), 633–641.

Qin, T., Li, P. and Shen, S. [2018], 'VINS-Mono: A Robust and Versatile Monocular Visual-Inertial State Estimator', *IEEE Transactions on Robotics* **34**(4), 1004–1020.

Qiu, K., Qin, T., Pan, J., Liu, S. and Shen, S. [2020], 'Real-time temporal and rotational calibration of heterogeneous sensors using motion correlation analysis', *IEEE Transactions on Robotics* **37**(2), 587–602.

Rehder, J., Nikolic, J., Schneider, T., Hinzmann, T. and Siegwart, R. [2016], Extending kalibr: Calibrating the extrinsics of multiple IMUs and of individual axes, *in* '2016 IEEE International Conference on Robotics and Automation (ICRA)', IEEE, pp. 4304–4311.

Segal, A., Haehnel, D. and Thrun, S. [2009], Generalized-ICP, *in* 'Robotics: science and systems', Vol. 2, Seattle, WA, p. 435.

Shan, T. and Englot, B. [2018], LeGO-LOAM: Lightweight and Ground-Optimized Lidar Odometry and Mapping on Variable Terrain, *in* '2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', IEEE, pp. 4758–4765.

Shan, T., Englot, B., Meyers, D., Wang, W., Ratti, C. and Rus, D. [2020], LIO-SAM: Tightly-coupled Lidar Inertial Odometry via Smoothing and Mapping, *in* '2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)', IEEE, pp. 5135–5142.

Shan, T., Englot, B., Ratti, C. and Rus, D. [2021], LVI-SAM: Tightly-coupled Lidar-Visual-Inertial Odometry via Smoothing and Mapping, *in* '2021 IEEE international conference on robotics and automation (ICRA)', IEEE, pp. 5692–5698.

Sola, J. [2012], 'Quaternion kinematics for the error-state KF', *Laboratoire dAnalyse et dArchitecture des Systemes-Centre national de la recherche scientifique (LAAS-CNRS), Toulouse, France, Tech. Rep* .

Sturm, J., Engelhard, N., Endres, F., Burgard, W. and Cremers, D. [2012], A benchmark for the evaluation of RGB-D SLAM systems, *in* '2012 IEEE/RSJ international conference on intelligent robots and systems', IEEE, pp. 573–580.

Xu, W. and Zhang, F. [2021], 'FAST-LIO: A Fast, Robust LiDAR-Inertial Odometry Package by Tightly-Coupled Iterated Kalman Filter', *IEEE Robotics and Automation Letters* **6**(2), 3317–3324.

Yan, Z., Sun, L., Krajník, T. and Ruichek, Y. [2020], EU long-term dataset with multiple sensors for autonomous driving, *in* '2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', IEEE, pp. 10697–10704.

Zhang, J., Kaess, M. and Singh, S. [2016], On degeneracy of optimization-based state estimation problems, *in* '2016 IEEE International Conference on Robotics and Automation (ICRA)', IEEE, pp. 809–816.

Zhang, J. and Singh, S. [2014], LOAM: Lidar Odometry and Mapping in Real-time, *in* 'Robotics: Science and Systems', Vol. 2, Berkeley, CA, pp. 1–9.

Zhang, Z. [2000], 'A flexible new technique for camera calibration', *IEEE Transactions on pattern analysis and machine intelligence* **22**(11), 1330–1334.

Zhu, F., Ren, Y. and Zhang, F. [2022], Robust Real-time LiDAR-inertial Initialization, *in* '2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)', IEEE, pp. 3948–3955.