Journal Title
XX(X):1–16

DOI: 10.1177/ToBeAssigned
www.sagepub.com/
SAGE

USTC FLICAR: A Multisensor 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 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. Therefore, the dataset is named "FLICAR" to denote flying cars. We believe this dataset can also represent the flying car scenarios, specifically the takeoff and landing of VTOL (Vertical Takeoff and Landing) flying cars. The dataset is available for download at: https://ustc-flicar.github.io/.

Keywords

Dataset, aerial working robot, simultaneous localization and mapping, bucket truck, flying car

1 Introduction

Aerial work plays a crucial role in our daily lives and industrial or agricultural production, as illustrated by the various typical aerial work scenes in Figure 1. However, aerial work is often characterized by low efficiency and high risk, as workers are exposed to dangers such as falling from high places, electrocution from overhead power lines, and being trapped or squeezed. Thousands of workers suffer serious injuries or fatalities each year as a result of aerial work. If robots can be utilized to replace workers in dangerous aerial working environments, the efficiency and safety of aerial work will be greatly improved, potentially saving lives.

1.1 Challenges

The automation of aerial work is faced with several challenges at various stages. One such challenge is the ability to lift a heavy robot into the air. To perform aerial work, robots must be equipped with flexible and powerful robotic arms, complex sensors, and sufficient computing resources for data processing and decision-making. However, achieving lightweighting at the current technical level is a challenge due to the need for these various systems. As a result, UAVs are limited in their payload capacity and cannot serve as a platform for heavy aerial work. To address this issue, we have identified the bucket truck as a suitable platform. Bucket trucks are high-capacity construction vehicles equipped with an extendable, hydraulic boom that carries a large bucket for raising workers to elevated, inaccessible areas. These trucks offer both strength and flexibility, with a typical payload capacity of over 200kg, and are able to reach any target position within their three-dimensional work space through the extension of the arms and rotation of the joints. A combined prototype of the working robot and the bucket truck is shown in Figure 2 (left).

Another challenge is for the robot to effectively interact with the complex aerial work environment. In order to do so, the aerial robot or vehicle must have accurate and real-time localization, as well as all-around accurate 3D perception and densification reconstruction of the environment through its visual and inertial system. This is a prerequisite for performing tasks such as object recognition, trajectory planning and control, and scene understanding.

There are specific challenges to visual and multisensor fusion localization and mapping in aerial work environments. Many aerial objects are difficult to detect and reconstruct

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}@mail.ustc.edu.cn, {jianmin, ebdong*, yanyongz}@ustc.edu.cn

Figure 1. Typical aerial work scenes in our daily life: (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

due to their small size and lack of texture, such as power lines, trusses, and tree branches. In contrast to structured scenes like warehouses or traffic, aerial work environments are often cluttered and unstructured, making it difficult to use general structured features to improve the performance of algorithms. Additionally, some aerial environments are emptier than ground environments, making it more difficult to perform visual or LiDAR odometry through matching and loopback detection. Aerial robots have more degrees of freedom and experience sudden changes in motion, which poses challenges to the accuracy and robustness of algorithms for complex motion and attitude. The bucket truck's huge hydraulic arm on which the aerial work robot is loaded will shake at low frequency and vibrate at high frequency while working, and the constantly moving hydraulic arm will become a dynamic disturbance in the environment, both of which can affect the accuracy of localization and mapping. Aerial work robots also face difficulties similar to those encountered by other outdoor robots, such as a high variety of lighting conditions in different weather, which can cause vision sensors to fail due to direct sunlight or darkness at night. The motivation for creating this dataset is to enable aerial work robots to overcome these challenges and advance their practical applications.

1.2 Related Works

Public datasets are essential for the application of autonomous systems in new scenes. They provide unified and fair benchmarks for evaluating various algorithms and allow researchers to quickly investigate and verify existing algorithms and develop new ones for a new scene without the need for expensive hardware equipment and tedious calibration and data collection. In this section, we will review some outstanding datasets related to our aerial work dataset, which are divided into two categories: ground and aerial. Table 1 provides a summary of the details.

For ground scenes, the most representative one is autonomous driving, which has made significant progress

Figure 2. *Left*: 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. *Right*: An imaginary picture of the flying car taking off and landing, drawn by our team.

and achieved great success in the past decade thanks to diverse public datasets. One of the most famous among them is the KITTI dataset Geiger et al. [2013]. A feature of this type of dataset is the abundance of sensors. Due to the load-carrying capacity and sufficient space of ground vehicles, various sensors such as stereo or surround view cameras, 3D LiDARs, IMUs, and INS/GNSS can be incorporated into the same spatiotemporal system for data collection. For localization and mapping tasks, ground scenes are usually based on RTK-GPS or LiDAR SLAM to generate ground truth values with centimeter-level accuracy.

Aerial autonomous systems have also made progress in recent years. The EuRoC dataset Burri et al. [2016] was the first to equip UAVs (Unmanned Aerial Vehicles) with synchronously triggered high frame rate stereo cameras and IMUs, allowing for the application of a tightly coupled visual-inertial system to the localization and attitude estimation of aerial robots. For aerial datasets, micro or small drones have been the most common aerial platforms used for data collection. Therefore, due to payload constraints, these platforms are usually equipped with only a few cameras and inertial sensors. In indoor or small scenes, motion capture systems are used to record motion trajectories and poses to generate 6DoF millimeter-level ground truth. In outdoor or larger scenes, laser trackers are used to record motion trajectories and generate millimeter-level ground truth.

According to the aforementioned description, there is a gap between ground datasets and aerial datasets. In contrast, one is a large scene with more sensors for flat 2D motion, and the other is a small scene with fewer sensors for aggressive 3D motion. To fill this gap, there are two routes to choose from: adding sensors to a drone with a larger payload or allowing a car to "fly" in the air for 3D movement. The NTU VIRAL dataset Nguyen et al. [2021] also recognized this gap and chose the first option, equipping a larger UAV with two Ouster 16-line 3D LiDARs in addition to a pair of cameras and an IMU. While the second option may sound like a fantasy, that is exactly what we do. Despite being in its infancy, it is feasible to use a bucket truck to allow a multi-sensor perception platform for autonomous driving to perform flexible 3D motion within a certain range in the air. As an aerial dataset, the USTC FLICAR dataset uses a typical suite of sensors for autonomous driving, including seven cameras, four 3D LiDARs, and three IMUs. When designing the sensor suite, we tried to maintain as much similarity as possible with existing autonomous driving datasets. For example, our Bumblebee stereo camera is the

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

	Sensors					
Dataset	IMU	Camera	LiDAR	Gruond 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. [2021]	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

same as those used in the Oxford RobotCar Maddern et al. [2017] and EU Long-term Yan et al. [2020] datasets. Our horizontal 3D LiDAR, the Velodyne HDL-32E, 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. Therefore, it is more convincing to compare algorithms using the same hardware between these datasets and the USTC FLICAR dataset. The USTC FLICAR dataset also has similarities with existing aerial datasets, but it is geared towards more delicate tasks. We obtained millimeter-level outdoor ground truth using a laser tracker.

Looking back at Table 1, it can be clearly seen that USTC FLICAR is the aerial dataset with the most sensors. This work extends the typical autonomous driving sensing suite to aerial scenes. We believe that our dataset is a significant contribution to provide benchmarks of testing existing algorithms for autonomous system and to develop new ones that are more suited to the particularities of aerial work scenes.

Going a step further, we named the dataset "FLICAR" because our vision is to advance the development of autonomous flying cars. Ahmed et al. [2020] has summerized challenges and strategies toward flying car future adoption. He argues that the ideal flying cars should have a high degree of autonomy and vertical take-off and landing (VTOL) capabilities. Advanced robotics and sensor fusion technology is an important factor in promoting ongoing development of flying car. Besides, the most essential part of the safe operation of a flying car will be the ground/air transition (takeoff/landing). As shown in Figure 2 (right), we draw an imaginary diagram of the flying car taking off and landing. How to correctly perceive, identify and safely avoid obstacles in the space adjacent to the ground during operation, such as power lines, trees, and buildings, is an important challenge to flying car. Therefore, aside from other parts of a flying car, let us focuses on its multi-sensor fusion perception, localization and mapping during its movement. USTC FLICAR dataset has the motion characteristics of VTOL like flying cars, i.e. ascent, descent and hover in the z direction, smooth rather than aggressive pitch and rotation in the air. If researchers want to test the performance of existing ground vehicle autonomous driving solutions in the air or develop and validate new algorithms, our dataset will provide real-world data in reference for flying cars during take-off and landing "parking" phases. The USTC FLICAR dataset will become a bridge for the application of autonomous

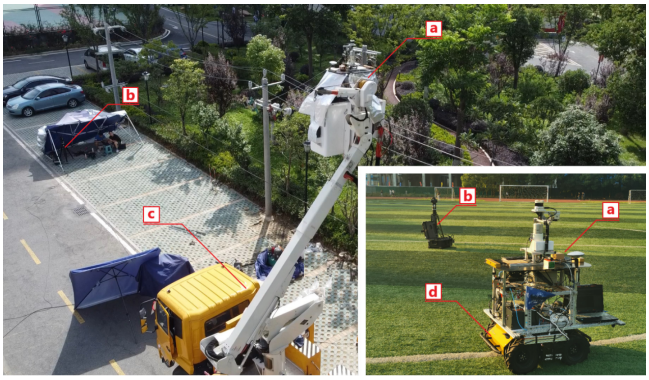

Figure 3. "Giraffe" and "Okapi" acquisition systems:
"Giraffe" aerial system: (a), (b) and (c).
"Okapi" ground system: (a), (b) and (d)
(a) multisensor data collection platform (Fig. 6), (b) laser tracker ground truth system, (c) bucket truck, (d) ground robot.

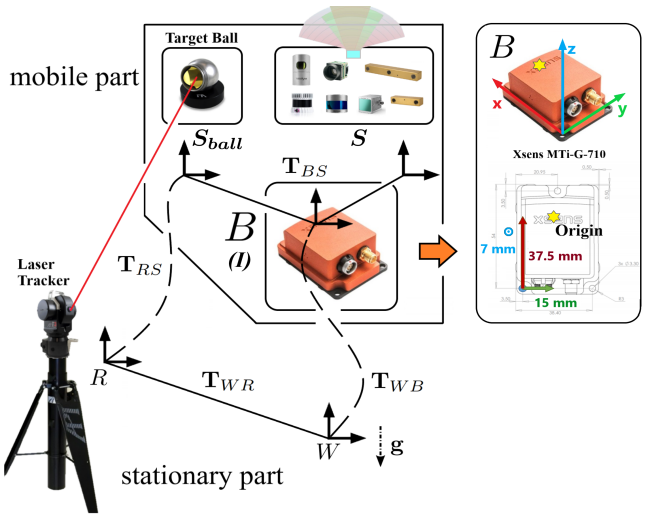

Figure 4. The sensor system used to capture the datasets consists of multiple sensors, each reporting measurements in its own reference frame $\boldsymbol{S}$. The datasets also include raw data from ground truth instruments, reported in the target ball frame $\boldsymbol{S_{ball}}$ and Laser tracker 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 4.7

driving fusion perception algorithms to the field of flying cars.

We organize the rest of the paper as follows: Section 2 describes in detail the various components of the data acquisition systems and the parameters, characteristics and function of each sensor. Section 3 describes the specific content of the dataset, the format of data storage and access methods. Section 4 presents the time synchronization, intrinsic and extrinsic calibration of sensors, and the generation of ground truth. Section 5 evaluated some state-of-the-art SLAM algorithms on the dataset as baselines and analysed the results. Finally, Section 6 summarizes the paper and discusses future work.

2 Acquisition Systems and Sensor Setup

The data was collected using the "Giaffe" acquisition system and "Okapi" acquisition system. As shown in Figure 3, the "Giraffe" system is an aerial system consisting of (a) multisensor data collection platform, (b) laser tracker ground truth system and (c) bucket truck; the "Okapi" system is a ground system similar to an autonomous vehicle, which equips the same sersors (a) and ground truth record system (b) on a ground robot (d) for data acquisition as a ground motion comparison with the data recorded by aerial system.

Both Systems are equipped with the following sensors:

2.1 Inertial Measurement Unit (IMU)

The main IMU of the acquisition system is an Xsens MTi-G-710 INS/GNSS module installed in the center of the system. The body frame is defined to be 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.

Xsens outputs the three-axis acceleration and three-axis angular velocity in its own coordinate system, and the quaternion attitude in the north-east-down (NED) coordinate system. Xsens is hardware synchronized to the same extrinsic GPS clock source with the cameras and LiDARs in the system, making up visual-inertial and LiDAR-inertial sensor units together. Two extra 6 axis IMUs are in OS0-128 LiDAR and LiVOX Avia LiDAR, as part of LiDAR-Inertial sensor unit.

2.2 3D LiDARs

3D LiDARs are important to 3D scenes accurate perception and understanding. This work includes three different kinds of mainstream LiDARs — Digital LiDAR, Mechanical LiDAR and MEMS LiDAR to provide aerial autonomous systems with no-blind-spot perception covering 360 degrees of horizontal and vertical views..

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

• Mechanical LiDAR: 1 × Velodyne HDL-32E, 5/10 Hz, 32 beams, 1.33° angular resolution, ± 2 cm distance accuracy, collecting 1.39 million points/second, field of view: 360° HFoV, 41.3° VFoV (+10.67° 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, field of view: 360° HFoV, 40° VFoV (−25° to +15°), range: 200 m

• MEMS LiDAR: 1 × DJI LiVOX Avia, 10 Hz, 2 cm distance accuracy, collecting 0.24 million points/second, field of view: 70.4° HFoV, 77.2° VFoV (Non-repetitive Scanning), range: 450 m

The visual scope and installation position of each LiDAR are shown in Figure 5. Ouster OS0-128 and Velodyne HDL-32E LiDARs are installed in the center of the system as the main source of 3D environmental data for the system. In the actual work process, the point cloud accuracy of

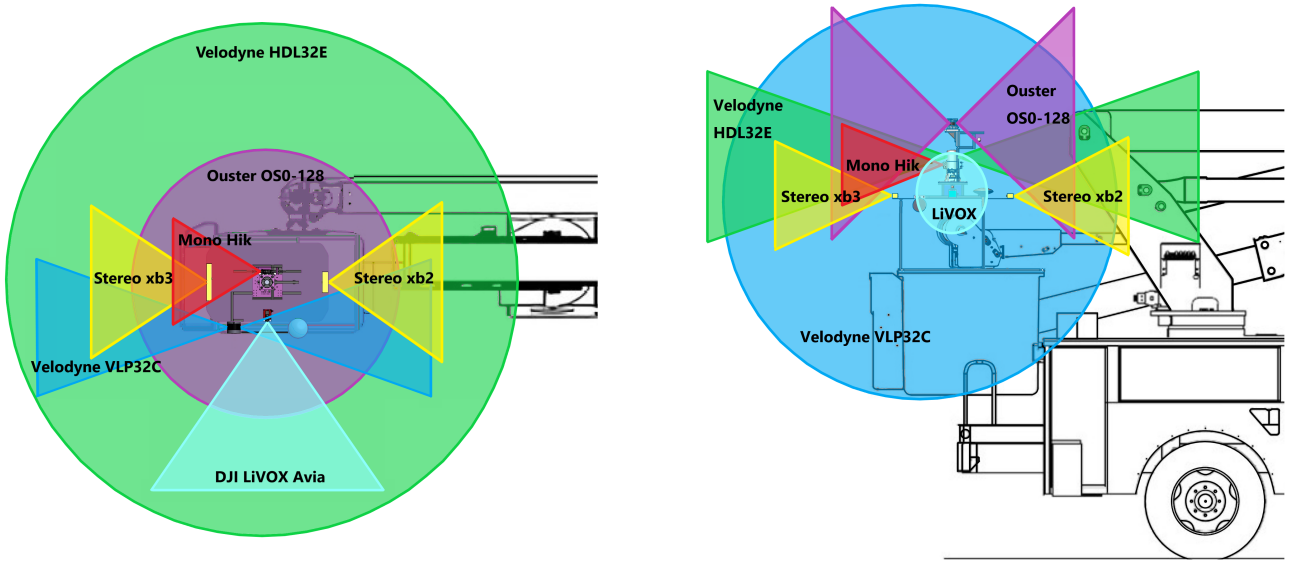

Figure 5. The visual scope of the acquisition system sensors, top view and side view.

Velodyne HDL-32E is the highest. And the Ouster OS0-128 LiDAR is a digital LiDAR. Digital LiDAR is based on custom system-on-a-chip ("SoC") with single photon avalanche diode ("SPAD") detectors. Therefore, it can not only output pointclouds, but also output depth images and signal-intensity images of LiDAR and visible light spectrum (Figure 11).

Another 360° 32 beams LiDAR Velodyne VLP-32C is installed vertically on the side of the acquisition system to supplement the blind area of vision. The aerial robot moves in three dimensions, and objects such as wires and branches may suddenly appear in all directions of the robot as obstacles. Therefore, the robot needs 360 degrees of perception in both horizontal and vertical directions. Besides, since the objects above the robot are few and small, there may only a few points in the point cloud within 180 degrees above the LiDAR.

A MEMS LiDAR LiVOX Avia is mounted horizontally on the sensor platform. A MEMS LiDAR LiVOX Avia is mounted horizontally on the sensor platform. The main characteristics of LiVOX Avia is that it has a view like camera and the mode of non-repetitive scanning. Pointclouds 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 m, 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 m, 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.

2.4 Laser Tracker

The laser tracker is fixed horizontally on the ground and is the only sensor that is independent of the overall sensor system, as shown in Figure 3 (b). During the motion of the sensor platform, the laser tracker tracks the active target ball that 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, angular accuracy: 3.5m/meter, system resolution: 0.1m, 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 power lines, trees, buildings, 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/.

The sensor platform complete various aerial motion through bucket truck , including movement in XYZ direction

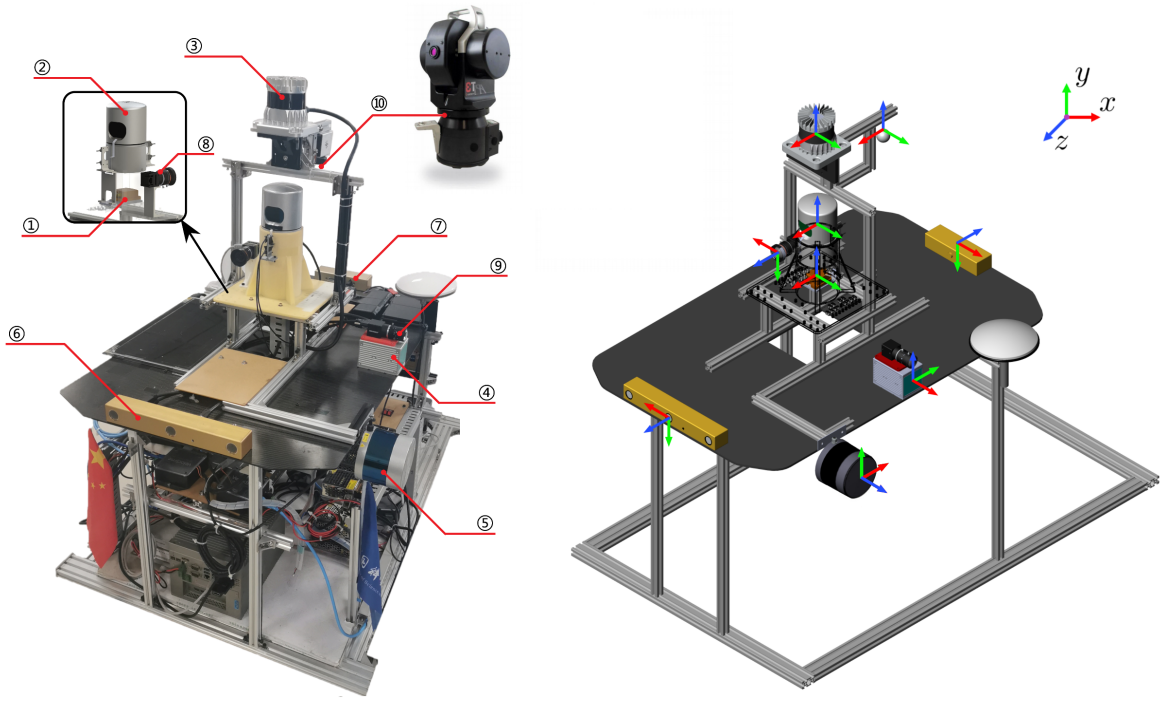

Figure 6. Sensor setup on a multi-sensor platform. The coordinate system shows the origin and orientation of each sensor mounted on the vehicle, with the following convention: X (red), Y (green), Z (blue). Note that velodyne LiDAR has two different coordinate system conventions. The coordinate system in the product manual is: Y: forward X: right Z: up; and the default coordinate system of Velodyne ROS package is: X: forward Y: left Z: up. Here shows the second coordinate system of ROS package. The number of the sensor in the figure corresponds to Table 2.

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	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	10Hz
7	Stereo Camera back	PointGrey Bumblebee xb2	/cam_xb2/left/image_raw /cam_xb2/right/image_raw	sensor_msgs/Image	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_raw	sensor_msgs/Image	20Hz

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 3. 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 4.7. 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>`.

Figure 7. Images of several typical data acquisition sites.

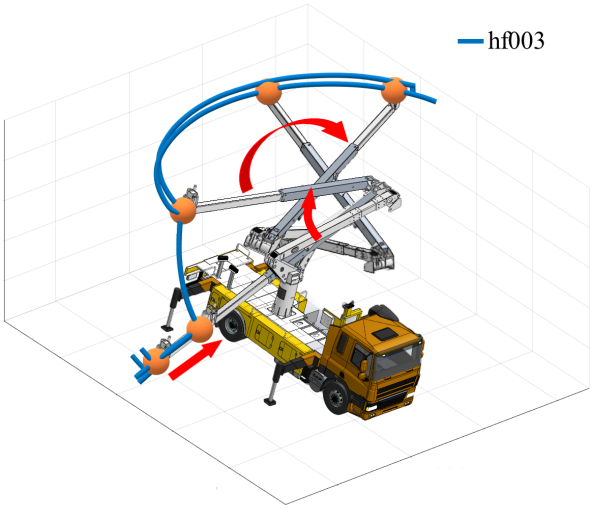

Figure 8. The joint action of the hydraulic arm of the bucket truck corresponds to the movement trajectory of the end sensor platform, taking the hf003 sequence as an example.

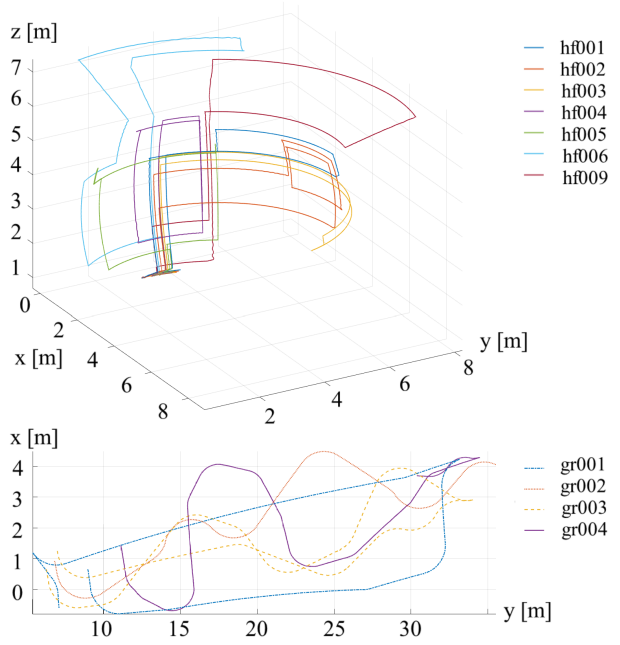

Figure 9. Representative aerial and ground trajectories in the dataset.

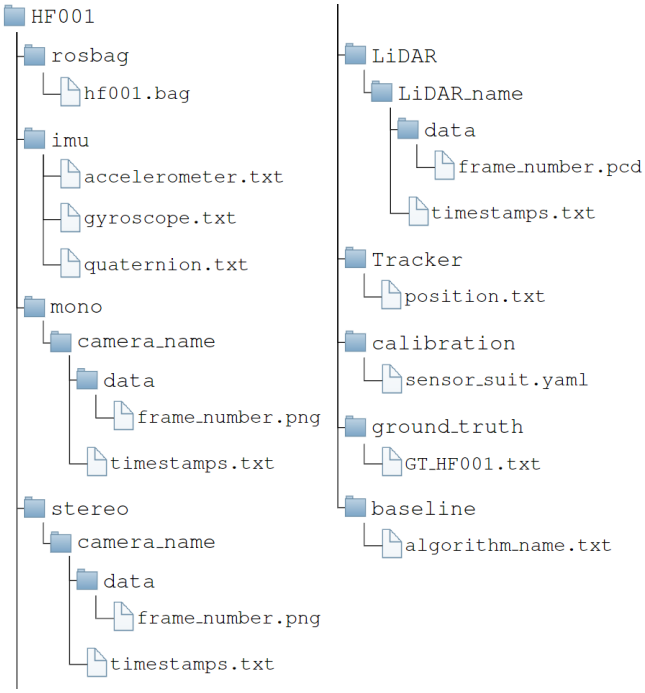

Figure 10. Dataset file structure.

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, 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 file which save the *(x,y,z)* coordinate of the pointcloud. If you want to use additional information of LiDAR scans such as reflection intensity *intensity*, scanning lines *ring* corresponding to pointcloud. 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,

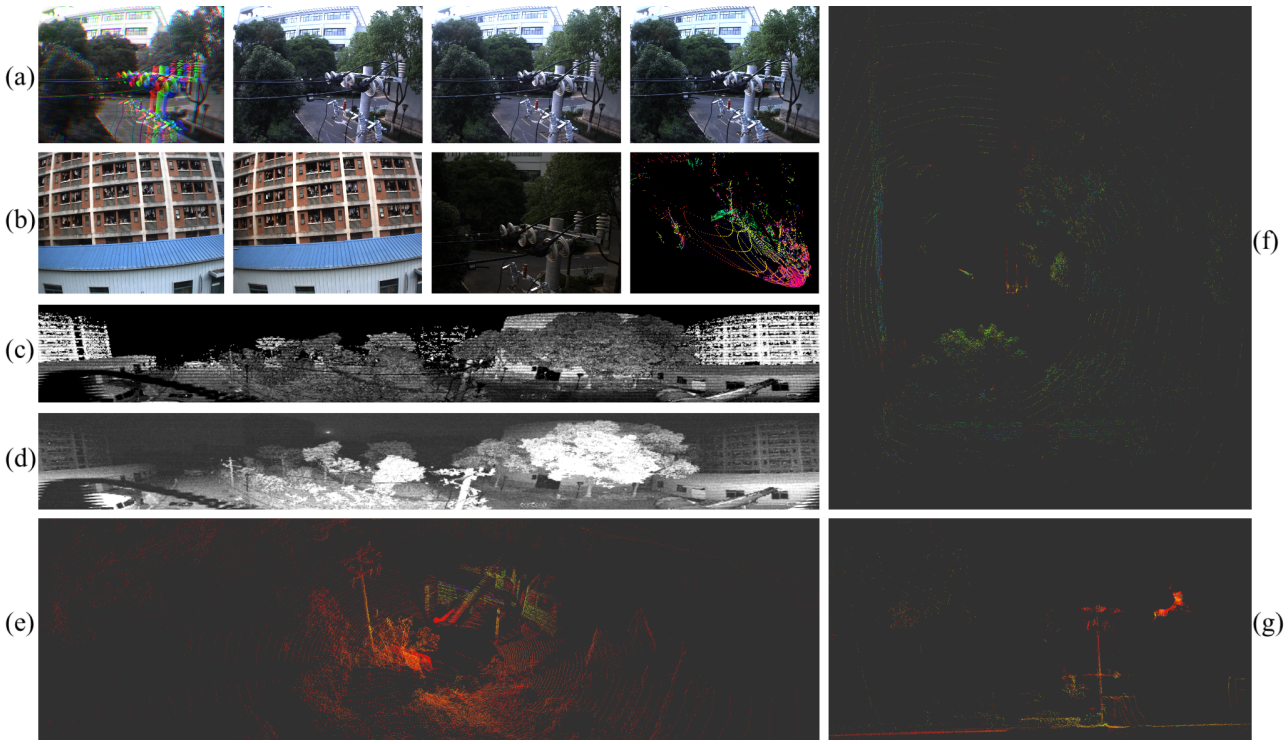

Figure 11. Visualization of multi-sensor data, at the same time. (a) Bumblebee-XB3 raw bayer image, left, center, right views. (b) Bumblebee-XB2 left,right views; Hikvision camera; LiVOX Avia pointclouds. (c) reflect image of Ouster OS0-128. (d) signal image of Ouster OS0-128. (e) Ouster OS0-128 pointclouds.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

topic type, frame rate, etc. of the data corresponding to the sensor is organized in Table 2.

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 spatio-temporal fusion. In the system, sensors are securely mounted using aluminium profile brackets, 3D prints, and carbon fibre 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 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

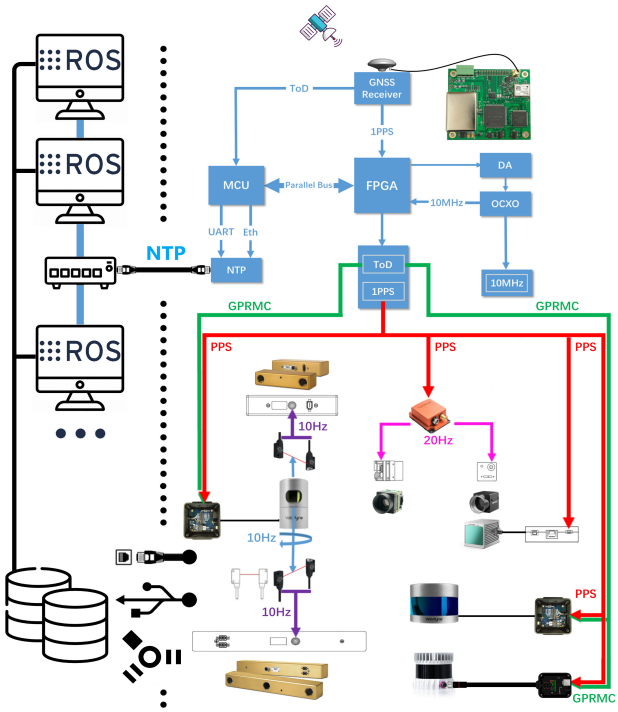

Figure 12. The multi-sensor time synchronization system structure, which is explained in detail in section 4.1.

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.

4.2 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 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 identity matrix for monocular cameras.

The camera projection matrix is used to project objects in the 3D world to the camera 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.

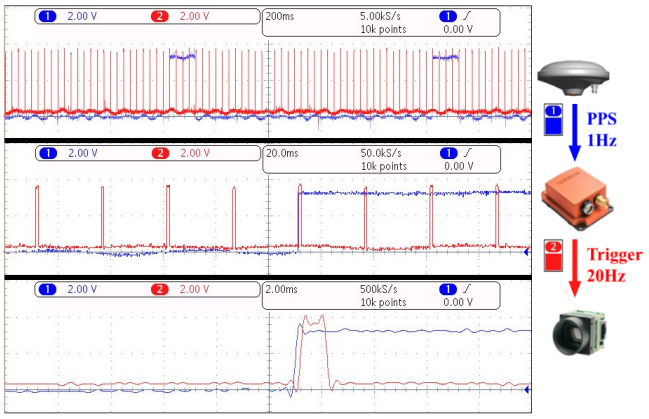

Figure 13. Time synchronization signals in visual-inertial system. The time resolution of the abscissa, from top to bottom, is 200ms, 20ms, and 2ms, from a Tektronix MDO3024 oscilloscope. Yellow: 1Hz PPS signal; Blue: 20Hz camera trigger signal.

4.3 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 IMU noise model. We fixed the IMU still on the anti-shake optical table for 4 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 4.2. The time synchronization accuracy between the IMU and the monocular camera is shown in Figure 13. 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 a 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:

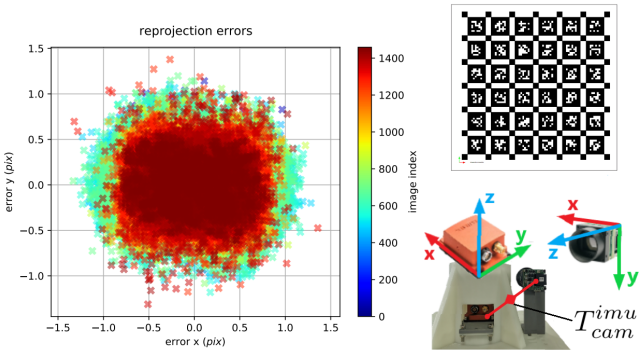

Figure 14. Reprojection error of camera-IMU extrinsics calibration using Kalibr.

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

$$T_{cam}^{imu} = \begin{bmatrix} R_{cam}^{imu} & t_{cam}^{imu} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{2}$$

The reprojection error of the camera-IMU extrinsic parameter calibration is shown in Figure 14, for most images the reprojection error is within 1.0 pixel. The mean, median, and standard deviation of the reprojection error are 0.352 pixels, 0.321 pixels, and 0.197 pixels, respectively.

4.4 LiDAR Inertial Calibration

We use the online method to calibrate the extrinsic parameters of the rotation between the LiDAR and the IMU, which can be seen as 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 ICP scan-to-scan registration. $\boldsymbol{q}_{\boldsymbol{L}}^{\boldsymbol{i}}$ is the rotation transformation from LiDAR to IMU.

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

$$\boldsymbol{q}_{i_k}^{i_{k+1}} \boldsymbol{q}_{\boldsymbol{L}}^{\boldsymbol{i}} = \boldsymbol{q}_{\boldsymbol{L}}^{\boldsymbol{i}} \boldsymbol{q}_{L_k}^{L_{k+1}} \tag{3}$$

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

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

$$(\boldsymbol{Q}_{i_k}^{i_{k+1}}{}^{+} - \boldsymbol{Q}_{L_k}^{L_{k+1}}{}^{-}) \boldsymbol{q}_{\boldsymbol{L}}^{\boldsymbol{i}} = 0 \tag{5}$$

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

$$\begin{bmatrix} \boldsymbol{Q}_{i_0}^{i_1}{}^{+} - \boldsymbol{Q}_{L_0}^{L_1}{}^{-} \\ \cdots \\ \boldsymbol{Q}_{i_{N-1}}^{i_N}{}^{+} - \boldsymbol{Q}_{L_{N-1}}^{L_N}{}^{-} \end{bmatrix} \boldsymbol{q}_{\boldsymbol{L}}^{\boldsymbol{i}} = \boldsymbol{A}_{\boldsymbol{4N\times4}} \boldsymbol{q}_{\boldsymbol{L}}^{\boldsymbol{i}} = 0 \tag{6}$$

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

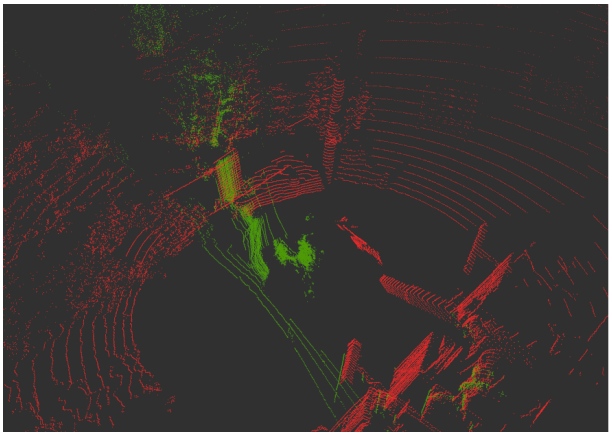

Figure 15. The point clouds obtained from the calibration process. The red point clouds are obtained from the horizontal LiDAR Velodyne HDL-32E, and the green point clouds are obtained from the vertical LiDAR Velodyne VLP-32C.

The translation vector t_{Lidar}^{imu} between the LiDAR and the IMU is determined according to the size of the 3D model drawing, because the connecting parts are 3D printed, so the accuracy can be trusted.

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*: $\mathrm{t}_{Lidar}^{imu} \in \mathbb{R}^{1\times3}$

$$T_{Lidar}^{imu} = \begin{bmatrix} R_{Lidar}^{imu} & t_{Lidar}^{imu} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{7}$$

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.

The principle of our multi-LiDAR extrinsic parameter calibration method is based on the NDT (Normal Distributions Transform) algorithm. The basic idea of the NDT algorithm for extrinsic parameter calibration is to construct a probability distribution map of the environment by analyzing and clustering LiDAR data, then match the probability distributions to obtain the pose transform with the highest fitting degree between the point clouds of the two LiDARs.

As shown in Figure 15, the pointclouds of the vertical LiDAR Velodyne VLP-32C are projected into the coordinate system of horizontal LiDAR Velodyne HDL-32E and displayed together.

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*: $\mathrm{t}_{Lidar1}^{Lidar2} \in \mathbb{R}^{1\times3}$

$$T_{velo}^{cam} = \begin{bmatrix} R_{Lidar1}^{Lidar2} & t_{Lidar1}^{Lidar2} \\ 0_{1\times3} & 1 \end{bmatrix} \tag{8}$$

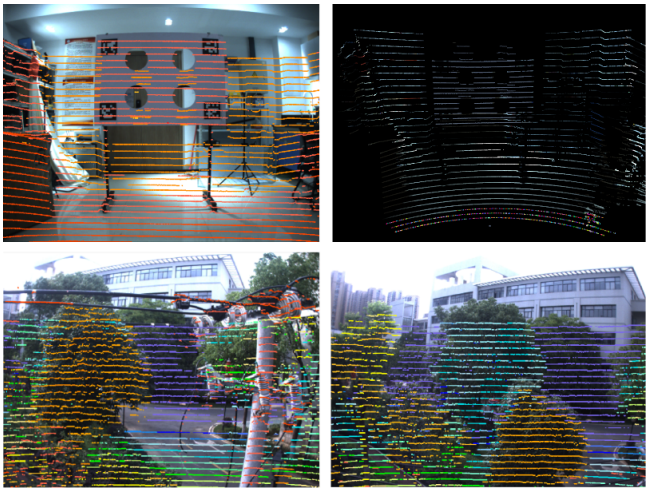

Figure 16. Velo2Cam Beltrán et al. camera-LiDAR calibration. *Up-Left*: special calibration board, project LiDAR point cloud to image. *Up-Right*: colorize LiDAR point cloud with image. *Down*: Aerial scenes LiDAR points fusion with images. (Velodyne-HDL-32E and Bumblebee-xb3-center)

4.6 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 pointcloud 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 16 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, autonomic 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 autonomic targetless camera-LiDAR calibration.

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 are to 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 4.2 denotes the camera intrinsic parameter.And h represents the scaling factor due to down-sampling. Examples of data fusion are shown in Figure 16. Besides, considering the loss of fusion data caused by the resolution mismatch between LiDAR point cloud and RGB image, our team proposed VPFNet Zhu et al. [2021], which bridges the resolution gap between the two sensors by adding virtual points, thereby retaining more information for processing.

4.7 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. The definition of the coordinates below can be reviewed in Figure 4 :

• 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 $\boldsymbol{p}_{\boldsymbol{S_{ball}}}^{\boldsymbol{R}} \in \mathbb{R}^3$, 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 $\boldsymbol{a}_{\boldsymbol{I(B)}} \in \mathbb{R}^3$ and angular velocity $\boldsymbol{\omega}_{\boldsymbol{I(B)}} \in \mathbb{R}^3$ 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 $\boldsymbol{q}_{\boldsymbol{B}}^{\boldsymbol{W}} \in \mathbb{H}$ of the sensor platform in world frame $\boldsymbol{W}$ (NED coordinates). $\boldsymbol{q}_{\boldsymbol{W}}$ is a a statistical optimal 3D orientation estimate computed by 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 LiDAR-Inertial system together with Xsens IMU. Here its data is not used directly, but used as a 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 17. 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 IMU noise model, the measurements of angular velocity and acceleration from IMU are defined as:

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

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

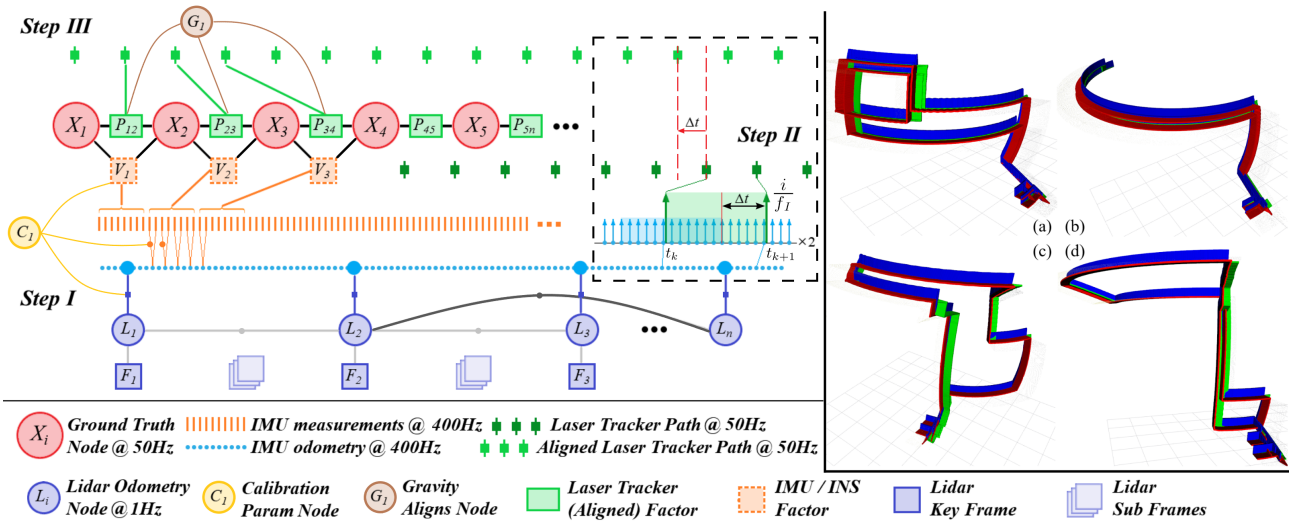

Figure 17. Example of a factor graph created by our system. The states to be estimated are represented by circles, and the measured values are represented by squares. For visualization, we group all calibration parameters into one node C_1, G_1 is the rotation from gravity alignment to laser tracking 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.

where $\boldsymbol{g_W}$ is the constant gravity vector in the world frame $\boldsymbol{W}$, and $\boldsymbol{R_W^B}$ 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], 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 persecond, making the results more accurate. The transformation $T_{lidar}^{imu} \in SE(3)$ from $\boldsymbol{S_{lidar}}$ to $\boldsymbol{B}$ has been obtained by the method mentioned in Section 4.4.

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

$$\boldsymbol{x_{B_0}} = \left[\boldsymbol{t_B}\ \boldsymbol{p_{B_0}}\ \boldsymbol{q_{B_0}}\right]^T \tag{13}$$

• Step II. Motion Correlation Analysis

The purpose of step II is to spatio-temporally align the laser tracker measurements with the body frame based on motion correlation analysis. Our methor is similar to Qiu et al. [2020]. In this step we have to analysis two signals, the first is IMU odometry $\boldsymbol{x_{B_0}}$ gotten in step I, the second is the measurement of laser tracking system $\boldsymbol{x_R} = \left[\boldsymbol{t_R}\ \boldsymbol{p}_{S_{ball}}^{R}\ \boldsymbol{p0}_{S_{ball}}^{B}\right]^T$, $\boldsymbol{p0}_{S_{ball}}^{B}$ 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 tripod and hydraulic legs respectively 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, 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 finish this step state $\boldsymbol{x_R}$ is aligned to $\boldsymbol{t_B}$. Here we get:

$$\boldsymbol{x_{B_1}} = \left[\boldsymbol{t_B}\ \boldsymbol{p}_{S_{ball}}^{R}\ \boldsymbol{p0}_{S_{ball}}^{B}\right]^T \tag{15}$$

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

$$\boldsymbol{x_{raw}} = \left[\boldsymbol{t_B}\ \boldsymbol{p}_{S_b}^{R}\ \boldsymbol{p0}_{S_b}^{B}\ \boldsymbol{q}_{B}^{W}\ \boldsymbol{g_W}\ \boldsymbol{a_B}\ \boldsymbol{\omega_B}\ \boldsymbol{b_{a,\omega}}\ \boldsymbol{n_{a,\omega}}\right]^T \tag{16}$$

• 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:

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

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

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

here we are estimating N inertial states at laser tracker frequency 50Hz, along with a calibration state $\boldsymbol{x_C}$ containing the spacial-temporal parameters between the traget ball frame $\boldsymbol{S_b}$ and IMU frame $\boldsymbol{B}$, and $\bar{q}_{\boldsymbol{W}}^{\boldsymbol{R}}$ the rotation between the global laser tracking frame $\boldsymbol{R}$ and global inertial frame $\boldsymbol{W}$. $\bar{q}_{\boldsymbol{R}}^{\boldsymbol{B_i}}$ 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. $\boldsymbol{p}_{\boldsymbol{B_i}}^{\boldsymbol{R}}$ and $\boldsymbol{v}_{\boldsymbol{B_i}}^{\boldsymbol{R}}$ 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_B^R 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

Table 3. ATE 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.

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 (Visual)	Hikcam	0.097	0.028	0.093	0.081	0.163	0.144*	0.084	×	×
	Xb3-C	0.144	0.182	0.106	0.113	0.417*	0.178	0.126	×	×
ORB-SLAM3 (V-Mono-Inertial)	Hikcam+Xsens	0.017	0.181	0.090	0.082	0.086*	0.118	0.078	×	×
	Xb3-C+Xsens	0.150	0.276	0.156	0.116	0.159*	0.184*	0.407	×	×
VINS-Mono (V-Mono-Inertial)	Hikcam+Xsens	0.146	0.291	0.332	0.092	0.204	0.232	0.131	1.381	1.794
	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

inertial state $\boldsymbol{x}_{\boldsymbol{B}_i}$ 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 17.

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 3 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 18, 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

Figure 18. 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.

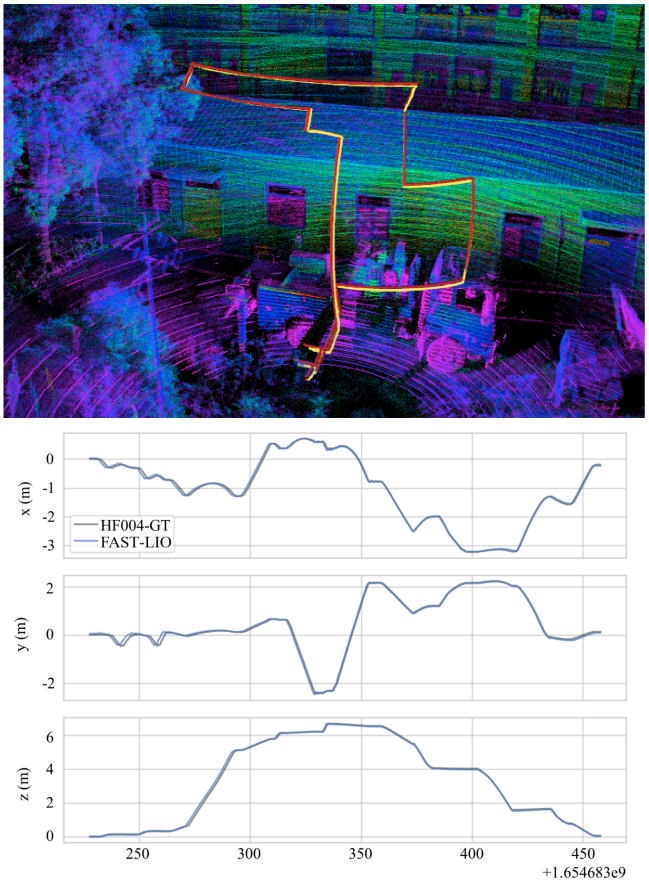

Figure 19. The ground truth is marked in red and the SLAM trajectory is marked in yellow. The figure shows the running results of Fast-LIO on the hf004 sequence.

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 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 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 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.

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 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. 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 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 Summary and future work

The USTC FLICAR is a unique dataset that focus on the task of 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 covering 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 autonomous driving sensing kit and bucket truck is a general autonomous aerial platform with high potential for various aerial work tasks.

Looking towards the future, there are several areas that we hope to expand upon in order to further advance the practical applications of aerial work robots. These include the addition of radar and camera sensor data with joint semantic annotations and semantic segmentation. This will allow for more accurate and detailed understanding of the environment and objects within it. Additionally, we plan on incorporating additional sensors such as infrared night vision cameras to further enhance the capabilities of the dataset. We also plan to continue collecting new data in order to capture a greater range of scenarios and environments. Overall, we believe that the USTC FLICAR dataset represents an important step towards making aerial work safer, more efficient, and more accessible.

Acknowledgements

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

Sheikh Shahriar Ahmed, Kevin F Hulme, Grigorios Fountas, Ugur Eker, Irina V Benedyk, Stephen E Still, and Panagiotis Ch Anastasopoulos. The flying car—challenges and strategies toward future adoption. *Frontiers in Built Environment*, 6:106, 2020.

Dan Barnes, Matthew Gadd, Paul Murcutt, Paul Newman, and Ingmar Posner. The oxford radar robotcar dataset: A radar extension to the oxford robotcar dataset. *arXiv preprint arXiv: 1909.01300*, 2019. URL https://arxiv.org/pdf/1909.01300.

Jorge Beltrán, Carlos Guindel, Arturo de la Escalera, and Fernando García. Automatic extrinsic calibration method for lidar and camera sensor setups. *IEEE Transactions on Intelligent Transportation Systems*, 2022. doi: 10.1109/TITS.2022.3155228.

Michael Burri, Janosch Nikolic, Pascal Gohl, Thomas Schneider, Joern Rehder, Sammy Omari, Markus W Achtelik, and Roland Siegwart. The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

Holger Caesar, Varun Bankiti, Alex H Lang, Sourabh Vora, Venice Erin Liong, Qiang Xu, Anush Krishnan, Yu Pan, Giancarlo Baldan, and Oscar Beijbom. nuscenes: A multimodal dataset for autonomous driving. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 11621–11631, 2020.

Carlos Campos, Richard Elvira, Juan J Gómez Rodríguez, José MM Montiel, and Juan D Tardós. Orb-slam3: An accurate open-source library for visual, visual–inertial, and multimap slam. *IEEE Transactions on Robotics*, 37(6):1874–1890, 2021.

Nicholas Carlevaris, Arash K Ushani, and Ryan M Eustice. University of michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research*, 35(9):1023–1035, 2016.

Jeffrey Delmerico, Titus Cieslewski, Henri Rebecq, Matthias Faessler, and Davide Scaramuzza. Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6713–6719. IEEE, 2019.

Andreas Geiger, Philip Lenz, Christoph Stiller, and Raquel Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

Patrick Geneva and Guoquan Huang. vicon2gt: Derivations and analysis. Technical Report RPNG-2020-VICON2GT, University of Delaware, 2020. Available: http://udel.edu/~ghuang/papers/tr_vicon2gt.pdf.

Albert S Huang, Matthew Antone, Edwin Olson, Luke Fletcher, David Moore, Seth Teller, and John Leonard. A high-rate, heterogeneous data set from the darpa urban challenge. *The International Journal of Robotics Research*, 29(13):1595–1601, 2010.

Jinyong Jeong, Younggun Cho, Young-Sik Shin, Hyunchul Roh, and Ayoung Kim. Complex urban dataset with multi-level sensors from highly diverse urban environments. *The International Journal of Robotics Research*, 38(6):642–657, 2019.

Xingchen Li, Yuxuan Xiao, Beibei Wang, Haojie Ren, Yanyong Zhang, and Jianmin Ji. Automatic targetless lidar–camera calibration: a survey. *Artificial Intelligence Review*, pages 1–39, 2022.

Will Maddern, Geoffrey Pascoe, Chris Linegar, and Paul Newman. 1 year, 1000 km: The oxford robotcar dataset. *The International Journal of Robotics Research*, 36(1):3–15, 2017.

András L Majdik, Charles Till, and Davide Scaramuzza. The zurich urban micro aerial vehicle dataset. *The International Journal of Robotics Research*, 36(3):269–273, 2017.

Thien-Minh Nguyen, Shenghai Yuan, Muqing Cao, Yang Lyu, Thien Hoang Nguyen, and Lihua Xie. Ntu viral: A visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint. *Internationl Journal of Robotics Research*, 2021.

Edwin Olson. Apriltag: A robust and flexible visual fiducial system. In *2011 IEEE international conference on robotics and automation*, pages 3400–3407. IEEE, 2011.

Gaurav Pandey, James R McBride, and Ryan M Eustice. Ford campus vision and lidar data set. *The International Journal of Robotics Research*, 30(13):1543–1552, 2011.

Taihú Pire, Martín Mujica, Javier Civera, and Ernesto Kofman. The rosario dataset: Multisensor data for localization and mapping in agricultural environments. *The International Journal of Robotics Research*, 38(6):633–641, 2019.

Tong Qin, Peiliang Li, and Shaojie Shen. Vins-mono: A robust and versatile monocular visual-inertial state estimator. *IEEE Transactions on Robotics*, 34(4):1004–1020, 2018.

Kejie Qiu, Tong Qin, Jie Pan, Siqi Liu, and Shaojie Shen. Real-time temporal and rotational calibration of heterogeneous sensors using motion correlation analysis. *IEEE Transactions on Robotics*, 37(2):587–602, 2020.

Joern Rehder, Janosch Nikolic, Thomas Schneider, Timo Hinzmann, and Roland Siegwart. Extending kalibr: Calibrating the extrinsics of multiple imus and of individual axes. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4304–4311. IEEE, 2016.

Tixiao Shan and Brendan Englot. 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)*, pages 4758–4765. IEEE, 2018.

Tixiao Shan, Brendan Englot, Drew Meyers, Wei Wang, Carlo Ratti, and Daniela Rus. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ international conference on intelligent robots and systems (IROS)*, pages 5135–5142. IEEE, 2020.

Tixiao Shan, Brendan Englot, Carlo Ratti, and Daniela Rus. Lvi-sam: Tightly-coupled lidar-visual-inertial odometry via smoothing and mapping. In *2021 IEEE international conference on robotics and automation (ICRA)*, pages 5692–5698. IEEE, 2021.

Joan Sola. 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*, 2012.

Jürgen Sturm, Nikolas Engelhard, Felix Endres, Wolfram Burgard, and Daniel Cremers. A benchmark for the evaluation of rgb-d slam systems. In *2012 IEEE/RSJ international conference on intelligent robots and systems*, pages 573–580. IEEE, 2012.

Wei Xu and Fu Zhang. Fast-lio: A fast, robust lidar-inertial odometry package by tightly-coupled iterated kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.

Zhi Yan, Li Sun, Tomáš Krajník, and Yassine Ruichek. Eu long-term dataset with multiple sensors for autonomous driving. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 10697–10704. IEEE, 2020.

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

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

Hanqi Zhu, Jiajun Deng, Yu Zhang, Jianmin Ji, Qiuyu Mao, Houqiang Li, and Yanyong Zhang. Vpfnet: Improving 3d object detection with virtual point based lidar and stereo data fusion. *arXiv preprint arXiv:2111.14382*, 2021.