Data Paper

The International Journal of Robotics Research
2024, Vol. 0(0) 1–24

DOI: 10.1177/02783649241303525
journals.sagepub.com/home/ijr

FusionPortableV2: A unified multi-sensor dataset for generalized SLAM across diverse platforms and scalable environments

Hexiang Wei[1,*], Jianhao Jiao[2,*], Xiangcheng Hu[1], Jingwen Yu[1,3], Xupeng Xie[1], Jin Wu[1], Yilong Zhu[1], Yuxuan Liu[1], Lujia Wang[4] and Ming Liu[4]

Abstract

Simultaneous Localization and Mapping (SLAM) has been widely applied in various robotic missions, from rescue operations to autonomous driving. However, the generalization of SLAM algorithms remains a significant challenge, as current datasets often lack scalability in terms of platforms and environments. To address this limitation, we present FusionPortableV2, a multi-sensor SLAM dataset featuring sensor diversity, varied motion patterns, and a wide range of environmental scenarios. Our dataset comprises 27 sequences, spanning over 2.5 hours and collected from four distinct platforms: a handheld suite, a legged robot, an unmanned ground vehicle (UGV), and a vehicle. These sequences cover diverse settings, including buildings, campuses, and urban areas, with a total length of 38.7 km. Additionally, the dataset includes ground truth (GT) trajectories and RGB point cloud maps covering approximately 0.3 km^2. To validate the utility of our dataset in advancing SLAM research, we assess several state-of-the-art (SOTA) SLAM algorithms. Furthermore, we demonstrate the dataset's broad application beyond traditional SLAM tasks by investigating its potential for monocular depth estimation. The complete dataset, including sensor data, GT, and calibration details, is accessible at https://fusionportable.github.io/dataset/fusionportable_v2.

Keywords

SLAM Dataset, sensor fusion, mapping, mobile robots, navigation

Received 6 April 2024; Revised 20 September 2024; Accepted 27 September 2024

1. Introduction

1.1. Motivation

Real-world robotic datasets are vital for algorithm development. They provide diverse environments and challenging real-world sequences for training and evaluation of systems. They reduce costs and workforce requirements, such as system integration, calibration, and field operations (Nguyen et al., 2022), promoting broader participation in robotic research and fostering novel algorithm development. As robotics research transitions from traditional handcrafted methods to data-driven and hybrid approaches (Brohan et al., 2022; Shah et al., 2023b), the importance of these datasets continues to grow. Building upon this trend, this paper contributes to the exploration of SLAM dataset's potential by introducing a diverse multi-sensor dataset. Our goal aims to develop a dataset to address the generalization challenges in robotic perception and navigation across various environments and operational conditions.

As outlined in Table 1, recent SLAM datasets exhibit two key trends: (1) they encompass a variety of large-scale real-world environments (e.g., urban roads (Burnett et al., 2023), subterranean (Reinke et al., 2022), and forests (Knights et al., 2023)), and (2) they incorporate heterogeneous

[1]Department of Electronic and Computer Engineering, The Hong Kong University of Science and Technology, Hong Kong SAR, China
[2]Robot Perception and Learning Lab, Department of Computer Science, University College London, UK
[3]Shenzhen Key Laboratory of Robotics and Computer Vision, Southern University of Science and Technology, China
[4]The Hong Kong University of Science and Technology (Guangzhou), Guangzhou, China

*Hexiang Wei and Jianhao Jiao Contributed Equally

Corresponding author:
Jianhao Jiao, Department of Computer Science, University College London, Gower Street, London WC1E 6BT, UK.
Emails: ucacjji@ucl.ac.uk; jiaojh1994@gmail.com

Table 1. Comparing SLAM Datasets: Highlights differences in platforms, sensors, and ground truth methods, focusing on FusionPortableV2's attributes. It classifies environment scales into small ($<100m^2$), medium ($<500m^2$), and large, acknowledging this as a broad estimation. Abbreviations include: UGV (Unmanned ground vehicle), MoCap (Motion capture), LT (Laser tracker), O (Out-of-the-box kinematic-inertial odometry), S (Subterranean), and V (Vegetated) areas. The symbol ● and ○ indicate whether dataset satisfies the option or not. ◑ indicates that the dataset misses robot kinematic data.

		Sensors Modality					Mobile Platform				Environment Scale				
Dataset		IMU	Frame Camera	Event Camera	LiDAR	GPS	Handhold	UGV	Legged	Vehicle	Small	Medium	Large	GT Pose	GT Map
Portable Suite	UZH-Event (Mueggler et al., 2017)	●	○	●	○	○	●	○	○	○	●	●	○	MoCap	/
	PennCOSYVIO (Pfrommer et al., 2017)	●	●	○	○	○	●	○	○	○	●	●	○	Apriltag	/
	TUM VI (Schubert et al., 2018)	●	●	○	○	○	●	○	○	○	●	●	●	MoCap	/
	UMA-VI (Zuñiga-Noël et al., 2020)	●	●	○	○	○	●	○	○	○	●	●	●	SfM	/
	Newer College (Ramezani et al., 2020)	●	●	○	●	○	●	○	○	○	●	●	○	ICP	Scanner
	Hilti-Oxford (Zhang et al., 2022)	●	●	○	●	○	●	○	○	○	●	○	○	ICP	Scanner
	VECTor (Gao et al., 2022)	●	●	●	●	○	●	○	○	○	●	○	○	MoCap/ICP	Scanner
Autonomous Driving	MIT DARPA (Huang et al., 2010)	●	●	○	●	●	○	○	○	●	○	○	●	D-GNSS	/
	KITTI (Geiger et al., 2013)	●	●	○	●	●	○	○	○	●	○	○	●	RTK-GNSS	/
	Oxford Robot Car (Maddern et al., 2017)	●	●	○	●	●	○	○	○	●	○	○	●	D-GNSS	/
	KAIST-Complex Urban (Jeong et al., 2019)	●	●	○	●	●	○	○	○	●	○	○	●	RTK-GNSS	SLAM
	Ford Multi-AV (Agarwal et al., 2020)	●	●	○	●	●	○	○	○	●	○	○	●	D-GNSS	SLAM
	DSEC (Gehrig et al., 2021)	○	●	●	●	●	○	○	○	●	○	○	●	RTK-GNSS	SLAM
	Boreas (Burnett et al., 2023)	●	●	○	●	●	○	○	○	●	○	○	●	RTX-GNSS	/
More Diversity	NCLT (Carlevaris-Bianco et al., 2016)	●	●	○	●	●	○	●	○	○	●	●	○	RTK-GNSS/ SLAM	/
	MVSEC (Zhu et al., 2018)	●	●	●	●	●	●	○	○	●	●	○	●	GNSS/MoCap/ SLAM	SLAM
	Rosario (Pire et al., 2019)	●	●	○	○	●	○	●	○	○	○	●(V)	○	RTK-GNSS	/
	M2DGR (Yin et al., 2021)	●	●	●	●	●	○	◑	○	○	●	●	○	RTK-GNSS/ MoCap/LT	/
	Nebula (Reinke et al., 2022)	●	○	○	●	●	○	●	◑(O)	○	○	●(S)	○	SLAM	Scanner
	FusionPortable (Jiao et al., 2022)	●	●	●	●	●	●	◑	◑	○	●	●	○	MoCap/RTK-GNSS/NDT	Scanner
	M3ED (Chaney et al., 2023)	●	●	●	●	●	○	○	◑	●	●	●(V)	●	RTK-GNSS/ SLAM	SLAM
	Ours (FusionPortableV2)	●	●	●	●	●	●	●	●	●	●	●	●	RTK-GNSS/LT	Scanner

sensors to enhance perceptual awareness. For example, cameras capture dense and high-resolution 2D images containing texture and pattern information of surroundings. However, cameras are vulnerable to adverse illumination conditions (e.g., darkness and glare) due to their passive nature in measuring. In contrast, range sensors such as LiDARs and Radars provide sparse but accurate structural information by exploiting their respective light sources. Integrating cameras with range sensors often yields more reliable results across a variety of perception tasks compared to relying on a single sensor type. Therefore, the complementary strengthen offered by various sensors drives the exploration of novel sensor fusion algorithms (Lin and Zhang, 2022), enabling robots to autonomously navigate through diverse environments with increased accuracy and robustness.

Despite the advancements in SLAM, a gap persists between the diversity of real-world scenarios and available datasets, especially concerning variations in environments, sensor modalities, and platforms executing diverse motions. This disparity affects the further development and evaluation of SLAM algorithms, potentially limiting their generalization capability and robustness in diverse real-world environments. Drawing on the recent success in the generalized manipulation and navigation models such as RT-2 (Brohan et al., 2023) and GNM (Shah et al., 2023a), it is our conviction that datasets featuring high motion and environmental diversity are crucial for the development of a versatile and generalized SLAM system.

1.2. Contributions

This paper aims to provide diverse, high-quality data using a consistent hardware setup, facilitating the development and benchmarking of robust and generalized SLAM systems. To achieve this objective, we present a comprehensive multi-sensor dataset, along with a detailed description of our data collection methodologies and complete benchmarking tools for evaluation. Building upon our previous FusionPortable dataset (Jiao et al., 2022), we introduce FusionPortableV2, a significant upgrade that expands the dataset in terms of data modalities, scenarios, and motion capabilities. This paper presents two main contributions:

1. **Diversity**: We have improved the FusionPortable dataset by expanding the range of platforms to include high-speed vehicles and to cover over 12 types of environments (e.g., campuses, underground areas, parking lots, and highways). These enhancements not only increase the dataset's diversity and complexity but also substantially improve the accuracy of ground truth data and the integration of raw kinematic data collected from ground robots. Our collection includes 27 sequences, spanning 2.5 hours and covering a total distance of 38.7 km. We also provide valuable ground truth trajectories and maps encompassing a campus area of approximately 0.3 km^2 to serve as benchmarks for evaluating various navigation algorithms beyond SLAM.
2. **Versatile**: We demonstrate the dataset's value for training models and evaluating algorithms through experiments on SLAM and monocular depth estimation. These experiments highlight the challenges and the need for cross-platform adaptability in diverse environments. With its rich variety of platforms and settings, the dataset can be extended to benchmarks for tasks such as cross-view and cross-modality localization (Shi et al., 2023), environmental mapping (Hu et al., 2024a; Kerbl et al., 2023), and navigation tasks with the high-quality map as simulation.

During the platform development and data collection process, we addressed numerous technical challenges and meticulously documented the issues encountered and their solutions. This guidance should be a valuable resource for future researchers in the field. To promote collaborative advancements, we have publicly released all data and implementation details. We believe this will open up a wide range of research opportunities in field robotics and aid in the development of versatile, resilient robotic systems.

1.3. Organization

The remainder of this paper is structured in the following manner: Section 2 discusses related works most of on SLAM datasets and summarizes key contributions of this paper. Section 3 outlines the hardware setup and sensor details. Section 4 covers sensor calibration procedures. Section 5 describes the dataset, including platform characteristics and scenarios. Section 6 introduces details post-processing steps on raw sensor measurements and GT data. Section 7 presents the methodologies used for evaluating localization, mapping, and monocular depth estimation. Known issues of this dataset are also discussed. Finally, Section 8 concludes the paper and suggests directions for future research.

2. Related works

In the last decade, the availability of high-quality datasets has significantly accelerated the development of SOTA SLAM algorithms by reducing the time and cost associated with data acquisition and algorithm evaluation. The rapid progress in sensor and robotics technology has led to the widespread adoption of multiple sensors across various robotic platforms. This evolution has set new benchmarks and hastened the enhancement of SOTA algorithms, spanning both handcrafted and data-driven methods such as VINS-Mono (Qin et al., 2018), FAST-LIO2 (Xu et al., 2022), VILENS (Wisth et al., 2022), DROID-SLAM (Teed and Deng, 2021), and Gaussian Splatting SLAM (Matsuki et al., 2024).

Recent advancements in the SLAM field have also extended to areas such as place recognition and collaborative

SLAM. Although these areas are not the primary focus of our work, they contribute significantly to the broader SLAM research community. Datasets such as Wild-Places (Knights et al., 2023) and HeLiPR (Jung et al., 2023) are specifically designed for LiDAR-based place recognition, emphasizing large-scale, long-term localization under varying appearance conditions (Yin et al., 2024). Collaborative SLAM, which enables information sharing and cooperative mapping among multiple robots, has gained attention with datasets such as Kimera-Multi (Tian et al., 2023), GrAco (Zhu et al., 2023), and S3E (Feng et al., 2022). These datasets provide multi-sensor data (e.g., cameras and LiDARs) and benchmarking tools, which are useful for both collaborative and single-robot SLAM evaluation. However, as they focus primarily on multi-robot scenarios with identical platforms, their utility for exploring cross-platform variability and single-robot SLAM challenges is limited.

In contrast, our FusionPortableV2 dataset is primarily designed for short-term odometry and SLAM, with a focus on accurately tracking the robot's pose and constructing consistent maps within a single session or over a shorter time period. The dataset features diverse platforms, sensor configurations, and environments, making it better suited for studying SLAM generalization across different conditions compared to most related works, as detailed in Table 1.

2.1. Specific-platform datasets

Early SLAM datasets predominantly focused on visual-inertial fusion, targeting specific platforms and environments. This focus was largely due to the ubiquity and convenience of visual and inertial sensors which are cheap and lightweight. They cover sequences which were captured by different platforms ranging from handheld devices (Pfrommer et al., 2017; Schubert et al., 2018; Zuñiga-Noël et al., 2020), drones (Burri et al., 2016; Delmerico et al., 2019; Li et al., 2024; Majdik et al., 2017), unmanned ground vehicles (Pire et al., 2019), and aquatic vehicles (Miller et al., 2018), respectively. Notably, the UZH-FPV dataset (Delmerico et al., 2019) stands out for its integration of event cameras and the inclusion of rapid trajectories from aggressive drone flights.

Concurrently, in the automotive industry, urban environment datasets introduce specific challenges including adverse lighting, weather conditions, and larger scales. Long-range sensors such as LiDARs and Radars are preferred for their capabilities, even though they were initially bulky and costly. The KITTI dataset (Geiger et al., 2013) sets a benchmark in autonomous driving with its rich urban sensor data collection. Further developments in driving-related datasets have expanded across dimensions of duration (Maddern et al., 2017), urban complexity (Jeong et al., 2019), and weather adversity (Agarwal et al., 2020). The DSEC dataset (Gehrig et al., 2021), akin to UZH-FPV, leverages stereo event cameras for extensive driving scenes. Moreover, Radars are essential for outdoor perception, offering advantages in range, velocity measurement via the Doppler effect and weather resilience. Related datasets such as Boreas (Burnett et al., 2023), Oxford Radar RoboCar (Barnes et al., 2020), and OORD (Gadd et al., 2024) collected data under conditions such as fog, rain, and snow.

The trend toward multi-sensor fusion spurred the creation of diverse and complex datasets. Datasets such as NCLT (Carlevaris-Bianco et al., 2016), M2DGR (Yin et al., 2021), NTU-VIRAL (Nguyen et al., 2022), ALITA (Yin et al., 2022), and FusionPortable (Jiao et al., 2022) also pose challenges for SLAM, given their diverse environmental appearance and structure. These datasets feature a variety of environments, including dense vegetation, open spaces, and complex buildings with multiple levels and detailed layouts. The changing lighting, seasonal foliage variations, and movement of pedestrians and vehicles add complexity to campus environments. As highlighted in NCLT dataset, these factors are crucial for life-long SLAM challenges. However, these datasets, collected via specific platforms such as unmanned ground vehicles (UGVs) and drones, fall short in showcasing diverse motion patterns, especially aggressive maneuvers.

Recent advancements in sensor technology have enabled the development of portable multi-sensor suites capable of collecting high-quality, multi-modal data in diverse environments (e.g., multi-floor buildings). Notable examples include the Newer College (Ramezani et al., 2020) and Hilti-Oxford (Zhang et al., 2022) datasets. These datasets also offer dense, high-quality 3D global maps, enabling the generation of high-rate 6-DoF reference poses and the evaluation of mapping algorithms. The RELLIS-3D dataset (Jiang et al., 2021), though it primarily focuses on semantic scene understanding in off-road environments, also provides diverse sensor data that can be valuable for SLAM research.

2.2. Cross-platform datasets

As SLAM research progressed from single-platform or collaborative applications, there has been a growing interest in cross-platform generalization. Building upon the specific-platform datasets discussed in Section 2.1, several datasets explore the generalization of algorithms across different platforms and scales, aiming to integrate motion characteristics from varied platforms with minimal parameter tuning for diverse scenarios. The MVSEC dataset (Zhu et al., 2018) collected multi-sensor data with diverse platforms, excluding UGV sequences. Conversely, the Nebula dataset (Reinke et al., 2022), developed during the DARPA Subterranean Challenge, includes field environments with both wheeled and legged robots, providing precise maps and trajectories. However, it lacks urban data and primarily focuses on LiDAR-based perception. The M3ED dataset (Chaney et al., 2023), although closely aligned with our objectives, lacks indoor data and platform-specific kinematic measurements, underscoring the unique contribution of our dataset.

Table 2. The sensors used in this dataset and their corresponding specifications. The detailed definition of ROS message type and naming of coordinate frames of each sensor are provided in the dataset website.

Sensor	Characteristics	ROS Topic	ROS Message Type	Rate (Hz)
3D LiDAR	Ouster OS1-128, 45°vert. × 360°horiz. FOV	/os_cloud_node/points	sensor_msgs/PointCloud2	10
	IMU: ICM20948, 9-axis MEMS	/os_cloud_node/imu	sensor_msgs/Imu	100
	Range, near-ir, reflectivity, signal images	/os_image_node/(range,nearir,…)_image	sensor_msgs/Image	10
Frame Camera	Stereo FILR BFS-U3-31S4C, global shutter 66.5°vert. × 82.9°horiz. FOV 1024 × 768 resolution	/stereo/frame_(left,right)/image_raw	sensor_msgs/CompressedImage	20
Event Camera	Stereo DAVIS346, 67°vert., × 83°horiz. FOV 346 × 240 resolution	/stereo/davis_(left,right)/events	dvs_msgs/EventArray	30
	Images that capture color data	/stereo/davis_(left,right)/image_raw	sensor_msgs/CompressedImage	20
	IMU: MPU6150, 6-axis MEMS	/stereo/davis_(left,right)/imu	sensor_msgs/Imu	1000
IMU	STIM300, 6-axis MEMS	/stim300/imu	sensor_msgs/Imu	200
INS	3DM-GQ7-GNSS/INS Dual-antenna, RTK-enabled INS	/3dm_ins/nav/odom	nav_msgs/Odometry	10
		/3dm_ins/gnss_(left,right)/fix	sensor_msgs/NavStatFix	10
		/3dm_ins/imu	sensor_msgs/Imu	200
Wheel Encoder	Omron E6B2-CWZ6C, 1000P/R	/mini_hercules/encoder	sensor_msgs/JointState	100
Legged Sensor	Built-in joint encoders and contact sensors	/unitree/joint_state	sensor_msgs/JointState	50
	Built-in IMU	/unitree/imu	sensor_msgs/Imu	50
	Out-of-the-box kinematic-inertial odometry	/unitree/body_odom	nav_msgs/Odometry	50

Despite these advancements, there remains an absence of datasets that comprehensively cover a wide range of platforms, environments, and sensor modalities within a single, unified framework (Table 2).

3. System overview

This section presents our developed multi-sensor suite, designed for integration with various mobile platforms through plug-and-play functionality. All sensors are securely mounted on an aluminum alloy frame, facilitating a unified installation. Additionally, we detail the devices employed for collecting GT trajectories and maps.

3.1. Suite setup and synchronization

The Multi-Sensor Suite (MSS) integrates exteroceptive and proprioceptive sensors, including a 3D Ouster LiDAR, stereo frame and event cameras, and IMUs, as depicted in its CAD model in Figure 1. We use two PCs that are synchronized via a Network Time Protocol (NTP) server for data collection. The primary PC processes data from the frame cameras and IMU, while the auxiliary PC handles additional data types. Both PCs are equipped with a 1 TB SSD, 64 GB of DDR4 memory, and an Intel i7 processor, running Ubuntu with a real-time kernel patch and employing the Robot Operating System (ROS) for data collection. This distributed architecture reduces the number of ROS nodes running on each individual PC, thereby alleviating the risk of queuing problems during data collection. After the mission, separate data collected by these two PCs are merged and post-processed offline. The subsequent sections will elaborate on the synchronization approach and the features of each sensor.

3.1.1. Synchronization. The synchronization process is illustrated in Figure 2. Generally, the field-programmable gate array (FPGA) board synchronizes with the pulse-per-second (PPS) signal from the external GNSS receiver, producing higher frequency trigger signals for the IMU, stereo frame cameras, and LiDAR clock alignment. In GPS-denied environments, it utilizes its internal low drift oscillator for synchronization, achieving a time accuracy below 1 ms between multiple trigger signals. To synchronize LiDAR and camera data, we phase-lock[1] the LiDAR's rotation so its forward-facing direction aligns with the camera's capture timing, accounting for the continuous nature of LiDAR's spinning data. We use the internal (Master–Slave mode) mechanism to synchronize stereo event cameras, where the left event camera is assigned as the master to send trigger signals to the right camera.

Figure 2(b) illustrates our synchronization scheme. The FPGA sends trigger signals to connected sensors, which starts capturing data after a small delay (commonly $t_{delay} <$ 1 ms). $t_{capture}$ is defined as the time for capturing data such as camera exposure or LiDAR rotation. It is followed by a transmission process that sends data to the host with time $t_{transmit}$. Non-triggerable sensors such as wheeled encoders use their internal clocks for timestamping, resulting in a time offset, t_{offset}, relative to the FPGA's trigger signal. Our scheme cannot recover t_{delay} and $t_{capture}$.

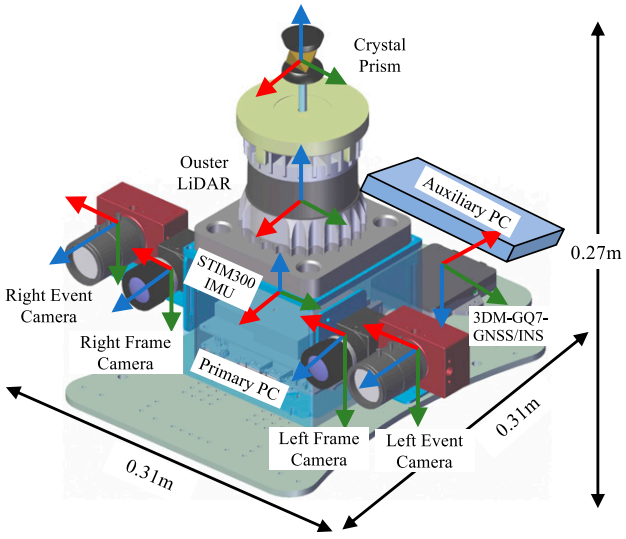


Figure 1. CAD model of the sensor rig where axes are marked: red: *X*, green: *Y*, blue: *Z*. It visualizes the position of each component of the handheld multi-sensor suite.

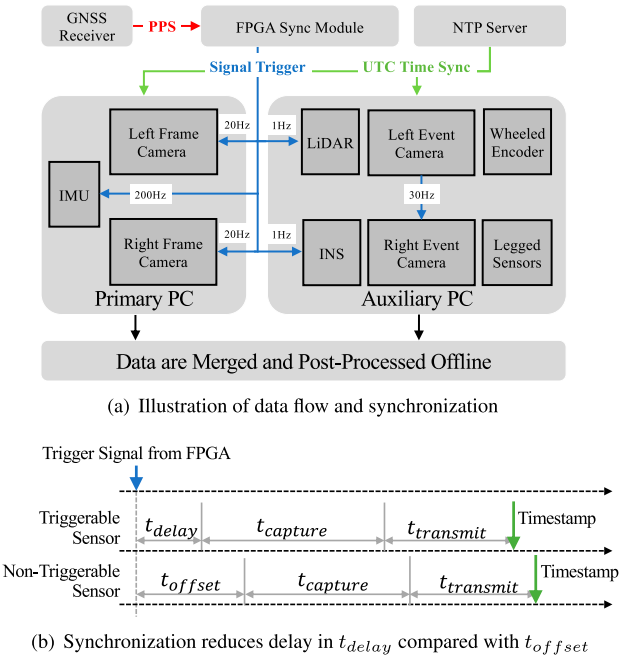


Figure 2. (a) Illustration of data collection which shows the data flow and synchronization processes. The red arrow indicate PPS signals for synchronization, green arrows show UTC time synchronization, and blue arrows represent sensor triggering signals, and black arrows depict the flow of raw data. (b) The timing diagram for triggerable (our case) and non-triggerable sensors, illustrating the unknown time offset caused by the delay in starting data capture, the duration of data capture, and the time required for data transmission from the sensor to the PC. Our synchronization solution can reduce the time delay (i.e., $t_{offset} - t_{delay}$) but cannot address other factors, which require online time calibration algorithms.

3.1.2. 3D LiDAR. Our LiDAR choose the OS1-128 Gen5 LiDAR that operates at 10 Hz. It features a built-in IMU capturing gyroscope, acceleration, and magnetometer data and generates four types of images to facilitate the usage of image-based algorithms: **range**, **near-ir**, **reflectivity**, and **signal** image. Each image measures different properties of the surroundings: (1) range images display the distance of the point from the sensor origin, calculated using the time of flight of the laser pulse; (2) near-ir images capture the strength of sunlight at the 865 nm light wavelength collected, also expressed in the number of photons detected that were not produced by the sensor's laser pulse; (3) reflectivity images display the reflectivity of the surface or object that was detected by the sensor; and (4) signal images show the strength of the light returned from the given point, which is influenced by various factors including distance, atmospheric conditions, and objects' reflectivity. The timestamp of the LiDAR data represents the end of the scanning period.

3.1.3. Stereo frame cameras. Our setup includes two FLIR BFS-U3-31S4C global-shutter color cameras for stereo imaging, synchronizing to output images at 1024 × 768 pixels and 20 Hz. The exposure time τ for both cameras is manually set as a fixed value. Image timestamps are adjusted by subtracting 0.5τ to properly align image features with IMU measurements since an image is the result of integrating light over the exposure interval (Furgale et al., 2013; Rehder et al., 2016b). To prevent abrupt changes in color space, the white balance settings are also fixed. Additionally, metadata like exposure time and gain are included for image analysis.

3.1.4. Stereo event cameras. Two event cameras, which are known for their high temporal resolution, extensive dynamic range, and energy efficiency, are used for data collection. These cameras output events data, 346 × 260 frame images, and high-rate IMU measurements. Frame images cannot be synchronized, resulting in a 10–20 ms delay. Infrared filters are used to lessen LiDAR light interference. Exposure times are set fixedly, whereas outdoor settings use auto-exposure to maintain image quality under varying light conditions.

3.1.5. Inertial measurement unit. The STIM300 IMU, a tactical-grade sensor[2], serves as the primary inertial sensor of our system, mounted beneath the LiDAR. It has a bias instability of 0.3°/h for the gyroscope and 0.04 mg for the accelerometer. The sensor outputs angular velocity and acceleration measurements at 200 Hz. Other components, including the LiDAR, event cameras, and the 3DM-GQ7 Inertial Navigation System (INS), are also integrated with IMUs. Further details are provided in subsequent sections.

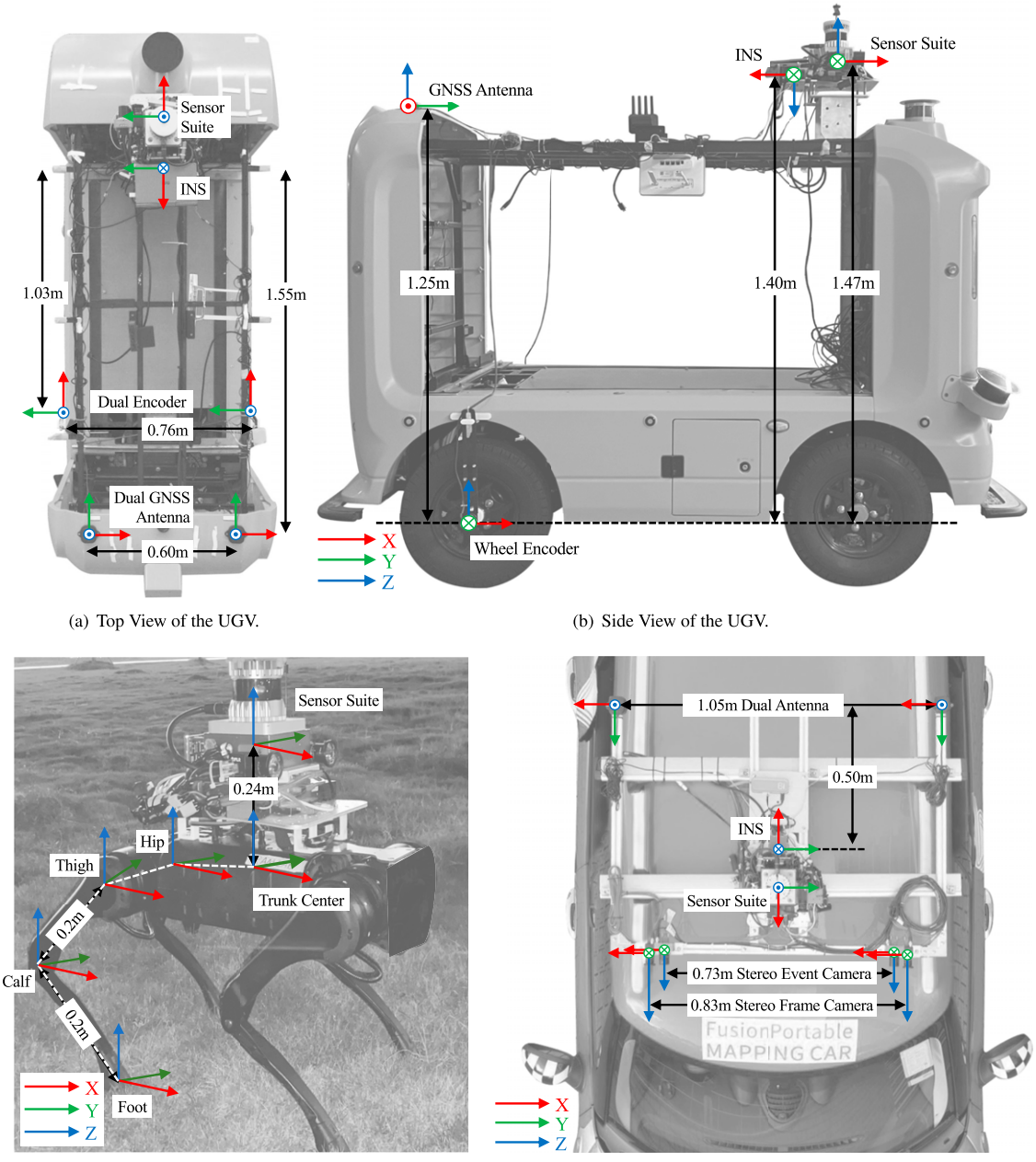
3.2. Platform-specific sensor setup

Our goal is to create a diverse dataset by capturing sequences with multiple mobile platforms, thereby increasing

the dataset's complexity and challenge compared to those relying on a single platform. Each platform is equipped with a handheld multi-sensor suite and platform-specific sensors, as shown in Figure 3. Figure 4 displays the platforms and exemplifies typical scenes from which data were gathered. Platform-specific sensor settings are introduced in the subsequent sections, while the description of their motion and scenario patterns are presented in Section 5.1.

3.2.1. Legged robot. We have selected the Unitree A1 quadruped robot as our legged platform, as shown in Figure 3(c). This robot is equipped with 12 joint motor encoders and four contact sensors per leg, located at the hip, thigh, calf, and foot. These sensors provide kinematic measurements at a rate of 50 Hz. The MSS is affixed to the robot's dorsal side and communicates with the kinematic sensors via Ethernet. In addition to the raw sensor measurements, we record metadata for each motor, which includes torque, velocity, position, and temperature, along with kinematic-inertial odometry data.

3.2.2. Unmanned ground vehicle. The MSS is integrated into a four-wheeled Ackerman UGV (see Figure 3(a), 3(b)),



(a) Top View of the UGV.

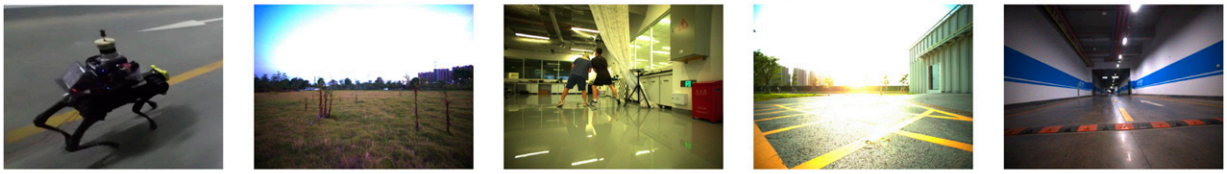
(b) Side View of the UGV.

(c) Legged Robot.

(d) High-Speed Vehicle.

Figure 3. Layouts of the platform-specific sensor setup, including different coordinate systems and their relative translation. More detailed and accurate dimensional data are provided in our calibration files.

(a) The handheld multi-sensor rig and scene images covering the grassland, lab, escalator, and underground tunnel.



(b) The legged robot and scene images covering the grassland, lab, campus, and underground tunnel.

(c) The UGV and scene images covering the outdoor parking lot, garage, and campus.

(d) The high-speed vehicle and scene images covering the mountain road, urban road, highway, and tunnel.

(e) Devices for generating GT trajectories and maps: 3DM-GQ7, Leica MS60, Leica BLK360, and Leica RTC360.

Figure 4. Platform-specific data samples: **(a)** the handheld multi-sensor rig, **(b)** the legged robot, **(c)** the low-speed UGV, **(d)** the high-speed vehicle, and **(e)** the diverse GT generation device. The depicted scenes highlight the dataset's comprehensiveness in covering a spectrum of platforms and environmental conditions.

originally designed for logistics transportation (Liu et al., 2021). To optimize signal reception, the dual GNSS antennas of the INS are positioned at UGV's rear side. Kinematic data for the UGV is acquired through two incremental rotary encoders, strategically positioned at the center of the rear wheel. These encoders, featuring 1000 pulses per revolution, produce measurement data at a rate of approximately 100 Hz, which is then recorded.

3.2.3. Vehicle. As depicted in Figure 3(d), we follow the KITTI setup (Geiger et al., 2013) by extending the baseline of both the stereo cameras and the dual antenna, with the stereo frame camera having a baseline of 83 cm and the event camera having a baseline of 73 cm. This extended baseline enhances the accuracy of depth estimation for distant objects, as compared with that in the UGV. The MSS is mounted on the vehicle's luggage rack using a custom-designed aluminum frame.

3.3. Ground truth provision setup

High-precision, dense RGB point cloud maps and GT trajectories are essential for evaluating SLAM and perception algorithms. This section describes three types of GT devices featured in our dataset, selected to meet the varied needs of the sequences. Through the integration of data from these GT devices, our dataset provides comprehensive support for algorithm benchmarking, not only in localization and mapping but also across diverse applications and requirements.

3.3.1. Dense RGB point cloud map. For creating dense point cloud maps of outdoor scenarios, the Leica RTC360 laser scanner was selected, because of its high scanning rate of up to 2 million points per second and accuracy under 5.3 mm within a 40 m radius. Some indoor areas were

scanned with the Leica BLK360, which operates at a rate of 0.68 million points per second and achieves an accuracy of 4 mm within a 10 m range. All scans are registered and merged by the Leica Cyclone software[3], resulting in a dense and precise RGB point cloud map. This map with the resolution as 8 cm, covering all data collection areas, can be used to evaluate the mapping results of algorithms[4] ranging from model-based (Lin and Zhang, 2022) and learning-based methods (Pan et al., 2024).

3.3.2. 3-DoF GT trajectory. For indoor and small-scale outdoor environments, the Leica MS60 total station[5] was utilized to measure the GT trajectory of the robot at the 3-DoF position. As shown in Figure 4(a), the tracking prism is placed atop the LiDAR. The GT trajectory was captured at a frequency between 5 and 8 Hz, achieving an accuracy of 1 mm. However, due to occasional instability in the measurement rate, the GT trajectory is resampled at 20 Hz using cubic spline interpolation for a more consistent evaluation. For further details on this process, please refer to Section 6.2.1.

3.3.3. 6-DoF GT trajectory. While the stationary Leica MS60 provides accurate measurements, it cannot track the prism when it is occluded or outside the visible range. Consequently, we employ the INS to capture 6-DoF GT trajectories in large-scale and outdoor environments with available GNSS satellites. This sensor integrates data from its internal dual-antenna RTK-GNSS, which provides raw data at a frequency of 2 Hz, and an IMU, to deliver estimated poses with an output rate of up to 30 Hz. Before commencing data collection, we ensure that the GNSS has initialized with a sufficient satellite lock and the RTK is in a fixed status, typically achieving a positioning accuracy of up to 1.4 cm. This initialization process usually takes 1–3 min in outdoor. To maintain the reliability of our ground truth data, we adhere to strict criteria. First, the INS filter must be in a stable navigation status. Second, both dual-antenna GNSS must be in RTK-fixed status, receiving signals from at least 20 satellites. Lastly, the positional error covariance must have converged to an optimal state.

4. Sensor calibration

We meticulously calibrate the *intrinsics* of each sensor, their *extrinsics*, and the *time offsets* between certain sensors beforehand using off-the-shelf softwares.[6,7,8,9,10] and [11], The STIM300 IMU's coordinate system is designated as the *body frame*, serving as the primary reference for most extrinsic calibrations. For indirectly calibrated sensors, conversion is achieved through matrix multiplication: $\mathbf{T}_C^A = \mathbf{T}_B^A \mathbf{T}_C^B$. The positioning and orientation of sensors across devices and platforms are illustrated in Figures 1 and 3. A summary of the calibration process is provided in Table 3. We show detailed results and guidelines for good calibration on the dataset's website due to page constraints.

4.1. Intrinsic calibration

We calibrate IMUs and cameras using the off-the-shelf Kalibr toolbox (Furgale et al., 2013; Rehder et al., 2016a). For wheel encoder intrinsics, such as wheel radius and axle track, we implement the motion-based calibration algorithm outlined in (Jeong et al., 2019). This involves manually maneuver the UGV through significant transformations, as depicted in Figure 5. We calculate the UGV's planar motion for each interval $\tau \in [t_k, t_{k+1}]$ using encoder data to determine linear $v = (\omega_l r_l + \omega_r r_r)/2$ and angular $\omega = (\omega_l r_l - \omega_r r_r)/b$ velocities. Concurrently, the INS captures more accurate motion estimates, and intrinsics are then optimized by minimizing the trajectory alignment error between these two trajectories.

4.2. Extrinsic calibration

Extrinsic calibrations, encompassing 6-DoF transformations and time offsets for IMU–IMU, IMU–camera, IMU-prism, camera–camera, and camera–LiDAR pairs, are typically obtained with off-the-shelf toolboxes. We specifically describe the calibration between the IMU and the prism that defines the reference frame of GT measurements relative to the total station (Leica MS60). We design the indirect calibration method since the total station provides only 3-DoF and low-rate trajectories. We observed that the prism is visible to infrared cameras in the motion capture room. We place and adjust three tracking markers around the prism to approximate its center, as shown in Figure 6. We move the handheld device to perform the "8"-shape trajectory. Both the motion capture system and LiDAR-inertial odometry (Xu et al., 2022) can estimate trajectories of the prism and STIM300 IMU (*body frame*), respectively. With these trajectories, the IMU–Prism extrinsics are estimated using the hand-eye calibration algorithm (Furrer et al., 2018).

5. Dataset description

This section begins by outlining the general motion patterns and potential applications for each platform. Following this, we consider the challenges posed by the dataset (as detailed in Section 5.2) and proceed to generate **27** sequences designed for algorithm development and evaluation (refer to Section 5.3). A summary of the essential characteristics of these sequences is provided in Table 4. In our prior publication, FusionPortable (Jiao et al., 2022), we presented more handheld and legged robot sequences captured with a similar hardware configuration but without kinematic data. This dataset encompasses **10** sequences featuring a variety of environments including garden, canteen, and escalator, captured via handheld devices, and **6** sequences obtained from a legged robot within a motion capture room.

Table 3. Description of intrinsic and extrinsic parameter calibration.

Type	Sensor	Calibrated Parameter	Approach
Intrinsics	IMU	Noisy Density, Random Walk	Allen variance analysis Toolbox
	Wheel Encoder	Wheel Radius, Axle Track	Minimize alignment error between $\mathcal{T}_{gt}$ and $\mathcal{T}_{est}$ (Jeong et al., 2019)[#]
	Camera	Focal Length, Center Point, Distortion	Minimize reprojection error (Zhang, 2000)
Extrinsics	IMU–IMU	Rotation, Translation	Optimization (Rehder et al., 2016a)
	IMU–Camera	Rotation, Translation, Cons. Time Offset	Optimization (Furgale et al., 2013)
	IMU–Prism	Translation	Hand-eye calibration (Furrer et al., 2018)
	IMU–Legged Sensors	Rotation, Translation	Obtained from the CAD Model
	Camera–Camera	Rotation, Translation	Minimize reprojection errors (Zhang, 2000)
	Camera–LiDAR	Rotation, Translation	Minimize point-to-line and point-to-plane errors (Jiao et al., 2023)

detailed in Section 4.2 of the paper.

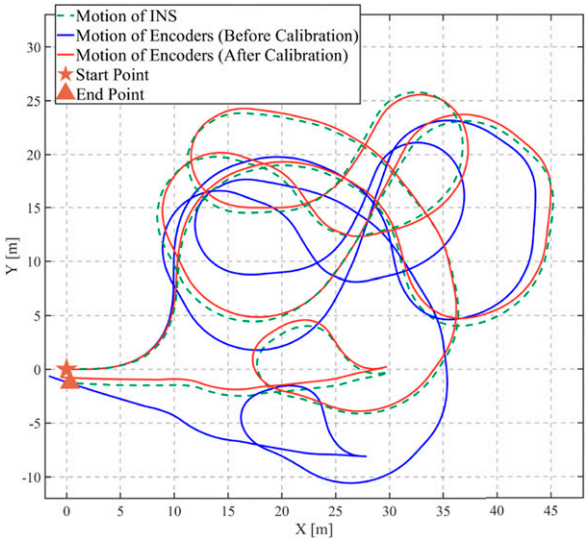


Figure 5. Comparison of trajectories: estimated motion by the INS (3DM-GQ7) (red), integration of encoders' measurements before calibration (green), and after calibration (blue) using the sequence `Ugv_parking00` for calibration. The ATE of the trajectory alignment is 0.98 *m*.

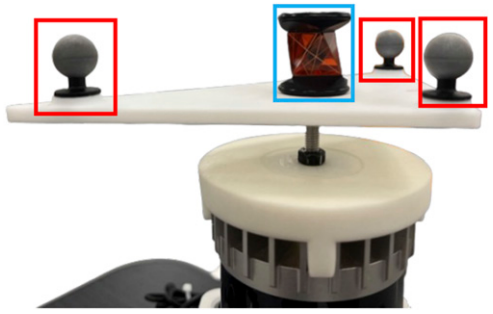


Figure 6. Sensor placement for the IMU–Prism calibration. Reflective balls for motion capture cameras (MCC) and the prism are marked in red and blue, respectively. We use MCC's measurements to infer high-rate motion of the prism.

5.1. Analysis of platforms characteristics

Each platform has its motion patterns (e.g., speed, angular velocity, dynamic frequency) and working ranges. Figure 7 visualize typical motion patterns of different platforms on some example sequences. We can clearly observe that the legged robot's motion is highly dynamic and the vehicle's motion is fast but smooth. Drawing from this observation, we meticulously design sequences to highlight the unique features of each platform.

5.1.1. Handheld. Since the handheld MSS is commonly held by a user, it offers flexibility for data collection scenarios. The handheld multi-sensor device provides adaptable data collection across diverse settings, akin to market counterparts like the Leica BLK2GO mobile scanning device, which excels in precision scanning and motion estimates. Therefore, we collect data in scenarios including a *laboratory* with furniture and dynamic elements, uneven *grasslands*, an *escalator* for vertical transitions, and an *underground parking lot* resembling long tunnels. The device performs motion influenced by the user's walking or running, sometimes leading to camera shake and rapid directional shifts. Each sequence contains at least one loop. The average movement speed is around 2 m/s.

5.1.2. Legged robot. The quadruped robot carries a sensor suite and commonly operates in *indoors*, *outdoors*, and *underground* for missions such as rescue, inspection, and document transportation. It exhibits complex motion patterns that involve a combination of walking, trotting, and running gaits. Deformable and rugged terrain can also affect motion's stability. Our experiments reveal that high-frequency jitters and sudden bumps are challenging to SOTA LiDAR-inertial odometry methods (Xu et al., 2022). Therefore, we believe that the integration sensor measurements from the joint motor and contact for a better

Table 4. Statistics and key challenges of each sequence are reported. Abbreviations: T: Total time. D: Total distance traveled. L: Large. M: Medium. S: Small. $\|\bar{\boldsymbol{v}}\|$: Mean linear velocity. $\|\boldsymbol{v}\|_{\max}$: Max linear velocity ($3\sigma$). 3-DoF (GNSS): Refer to Section 6.2.2. 6-DoF (SLAM): Use FAST-LIO2 (Xu et al., 2022) to generate the reference trajectory.

Platform	Sequence	T [s]	D [m]	$\|\bar{\boldsymbol{v}}\|/\|\boldsymbol{v}\|_{\max}[m/s]$	Scale	Motion	Challenges	GT pose	GT map
Handheld	handheld_grass00	140	80	0.55/1.76	S	6-DoF Walk	Textureless	3-DoF (Tracker)	Yes
	handheld_room00	140	63	0.41/1.40	S	6-DoF Walk	Dynamic	3-DoF (Tracker)	Yes
	handheld_room01	113	46	0.36/1.52	S	6-DoF Walk	Dynamic	3-DoF (Tracker)	Yes
	handheld_escalator00	247	95	0.45/1.45	S	6-DoF Walk	Non-inertial	3-DoF (Tracker)	Yes
	handheld_escalator01	254	88	0.47/1.54	S	6-DoF Walk	Non-inertial	3-DoF (Tracker)	Yes
	handheld_underground00	380	403	1.07/3.38	M	6-DoF Walk	Structureless	3-DoF (Tracker)	Yes
Legged Robot	legged_grass00	301	112	0.35/1.51	S	Jerky	Deformable	3-DoF (Tracker)	Yes
	legged_grass01	355	97	0.32/1.58	S	Jerky	Deformable	3-DoF (Tracker)	Yes
	legged_room00	173	57	0.28/1.30	S	Jerky	Dynamic	3-DoF (Tracker)	Yes
	legged_transition00	233	98	0.41/1.60	S	Jerky	Illumination	3-DoF (Tracker)	Yes
	legged_underground00	274	167	0.58/2.46	M	Jerky	Structureless	3-DoF (Tracker)	Yes
UGV	ugv_parking00	178	319	1.77/2.80	M	Smooth	Structureless	6-DoF (INS)	Yes
	ugv_parking01	292	434	1.48/4.07	M	Smooth	Structureless	6-DoF (INS)	Yes
	ugv_parking02	80	242	3.04/4.59	M	Jerky	Structureless	6-DoF (INS)	Yes
	ugv_parking03	79	218	2.75/4.92	M	Jerky	Structureless	6-DoF (INS)	Yes
	ugv_campus00	333	898	2.69/4.90	M	Smooth	Scale	6-DoF (INS)	Yes
	ugv_campus01	183	343	1.86/4.40	M	Jerky	Fast Motion	6-DoF (INS)	Yes
	ugv_transition00	491	445	1.04/3.25	S	Smooth	GNSS-Denied	3-DoF (Tracker)	Yes
	ugv_transition01	375	356	0.92/3.51	S	Smooth	GNSS-Denied	3-DoF (Tracker)	Yes
Vehicle	vehicle_campus00	610	2708	4.43/9.31	M	Height-Change	Scale	6-DoF (INS)	No
	vehicle_campus01	420	2086	4.96/8.59	M	Height-Change	Scale	6-DoF (INS)	No
	vehicle_street00	578	8042	13.90/19.52	L	High-Speed	Dynamic	3-DoF (GNSS)	No
	vehicle_tunnel00	668	3500	5.24/15.00	L	High-Speed	LiDAR	3-DoF (GNSS)	No
	vehicle_downhill00	512	3738	7.29/15.59	L	Height-Change	Illumination	6-DoF (INS)	No
	vehicle_highway00	694	9349	13.46/30.87	L	High-Speed	Structureless	6-DoF (INS)	No
	vehicle_highway01	377	3641	9.64/24.36	L	High-Speed	Structureless	6-DoF (INS)	No
	vehicle_multilayer00	607	1021	1.68/4.53	M	Spiral	Perceptual Aliasing	6-DoF (SLAM)	No

motion estimation deserve further study and thus provide these data (Yang et al., 2023). The operational speed of the robot is approximately 1.5 m/s.

5.1.3. Unmanned ground vehicle. The UGV is typically designed for last-mile delivery and navigates middle-scale areas like campuses and factories. Constrained by Ackermann steering geometry, the UGV executes planar and smooth movements in response to the operator's inputs. Data collection is conducted in various environments, including an *outdoor parking lot* (open space), a *campus*, and the challenging *transition zones* between indoor and outdoor environments (where GNSS signals are unstable). To mimic real-world complexities, commands for sudden stops and 45° turns are occasionally issued. The UGV can move at speeds of approximately 5 m/s.

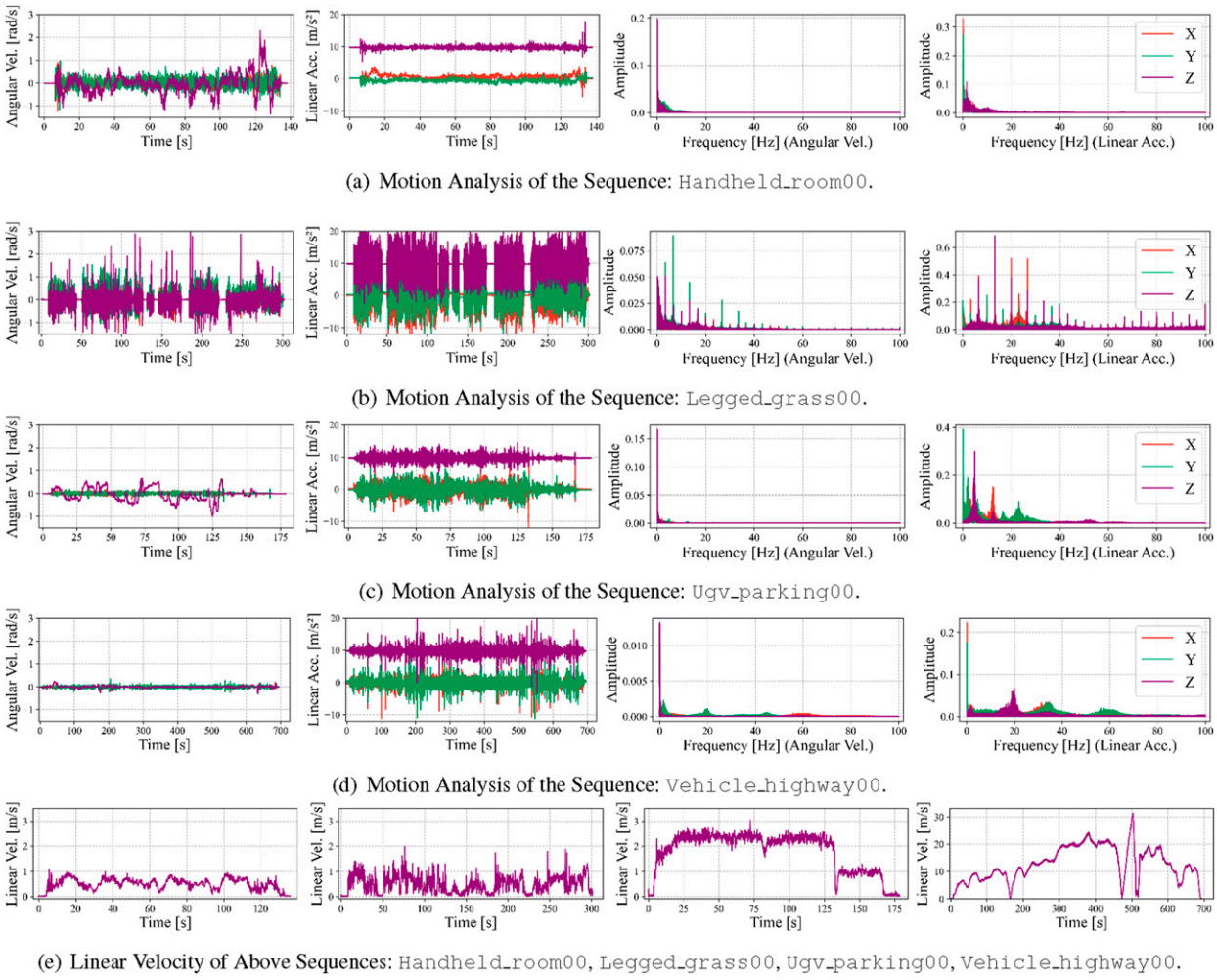


Figure 7. Motion analysis with four mobile platforms in terms of linear acceleration [m/s^2], angular velocity [rad/s], and velocity [m/s]. We use measurements from STIM-300 to get linear acceleration (including gravity) and angular velocity as well as SLAM results to get rough velocity. (a)–(d): The left two columns of each row illustrate time-domain data, revealing immediate dynamic behaviors, while the right two columns display frequency-domain data, highlighting motion features of different platforms. Each line in the legend represents linear acceleration and angular velocity of each axis: X (Red), Y (Green), and Z (Purple). (e) The linear velocity of these sequences. The acceleration of the vehicle is low since the vehicle is driving on a highway, where acceleration and deceleration occus less frequently.

5.1.4. Vehicle. The vehicle collects data across diverse urban environments in Hong Kong, navigating through *mountain roads* with elevation shifts, *multi-story parking lots* with varying heights and orientations, dynamic *downtown areas* with buildings, *highways*, and GNSS-denied *underground tunnels*, which are structureless and lack distinctive textures. It operates at a range of speeds from 10 km/h to 100 km/h, sometimes with abrupt speed and directional changes influenced by traffic and road conditions.

5.2. Challenging factors

Prior to data collection, we acknowledge that practical factors contribute to sensor degradation and potential algorithmic failure. Our data sequences, integrated with the platforms described, aim to comprehensively evaluate algorithm performance in terms of accuracy, efficiency, and robustness. Additionally, we anticipate these sequences will draw the development of novel algorithms.

5.2.1. Illumination conditions. Different illumination conditions, such as bright sunlight, shadows, and low light, affect the quality of visual sensors and pose challenges for visual perception algorithms. For example, in bright sunlight, cameras are sometimes overexposed, resulting in a loss of appearance information. On the contrary, cameras are sometimes underexposed in low light conditions, leading to image noise and poor visibility.

5.2.2. Richness of texture and structure. Structured environments (e.g., offices or buildings) can mainly be explained using geometric primitives, while semi-structured environments have both geometric and complex elements like trees and sundries. Scenarios like narrow corridors are structured but may challenge state estimators. Additionally, texture-rich scenes facilitate visual algorithms to extract stable features (e.g., points and lines), while texture-less may negatively affect the performance. Also, in texture-less environments, only a small amount of events is triggered.

5.2.3. Dynamic objects. In dynamic environments, several elements (e.g., pedestrians or cars) are moving when the data are captured. This is in contrast to static environments. For instance, moving cars cause noisy reflections and occlusions to LiDAR data, while pedestrians cause motion blur to images. Overall, dynamic objects induce negative effects from several aspects such as incorrect data association, occlusion, and "ghost" points remaining on the map.

5.2.4. Intermittent GNSS. The intermittent GNSS signal issue typically arises in environments like places where dense and towering urban clusters are presented, overpasses, and indoor–outdoor transition areas. A special example is the city center of Hong Kong. In such scenarios, GNSS signals are often obstructed, leading to sporadic reception and significant uncertainty.

5.2.5. Scale variability. Developing SLAM and perception algorithms for large-scale environments may encounter challenges such as an increased computational load and a heightened risk of perceptual aliasing. The former necessitates stricter demands on algorithm latency and memory usage, whereas the latter requires more accurate long-term associations for place recognition (Yin et al., 2024), given the potential for environments to include geographically distant yet visually similar locations.

5.2.6. Viewpoint change. We consider the viewpoint change from two perspectives: "yaw-change" and "roll-pitch-change." The former often occurs in sequences with loops. Several sequences in our dataset feature at least one loop, posing challenges for place recognition and image matching algorithms. For instance: `ugv_transition00` and `ugv_transition01` contain loops in both indoor and outdoor environments; `vehicle_multilayer00` captures multiple loops across different floors of a multi-story parking lot; and `vehicle_campus00` and `vehicle_campus01` collectively cover the entire HKUST campus, where loops are also present. We also provide a similar sequence (`campus_road_day`) in FusionPortable (Jiao et al., 2022), collected 15 months ago, during which several buildings and roads were updated.

The latter aspect is particularly relevant to the cross-view localization (CVL) problem (Shi et al., 2023), which involves viewpoint changes between down-facing satellite images and ground-level images. Open-source satellite images can be obtained from tools such as Google Earth. Since several sequences include accurate GT geo-localization information and covers diverse scenarios (e.g., campus, urban roads, downhill mountain roads), it can serve as a challenging benchmark for CVL.

5.3. Sequence description. Table 4 summarizes the characteristics of our proposed sequences, detailing aspects such as temporal and spatial dimensions, motion patterns, locations, textural and structural richness, and whether GT poses and maps cover. Figures 8 and 9 illustrate the coverage areas of the sequences from a satellite view perspective.

5.4. Dataset organization

Figure 10 outlines our dataset's organization. Sensor data were captured using the ROS bag tool[12]. ROS bags are used due to their numerous advantages, such as mature tools for debugging, visualization of the tf tree, especially with multiple sensors present, a broadcasting mechanism for batching output messages of different types, and the ability to convert them into individual files. To facilitate download, ROS bags were compressed with 7-Zip. Each bag follows the naming convention <**platform_environment**>. Sensor calibration parameters are saved in *yaml* files naming as <**frame_id**> (e.g., `frame_cam00.yaml` for the left RGB camera and `frame_cam01.yaml` for the right.) During the 6-month dataset construction period, calibration was performed and documented multiple times, with parameters organized by calibration date (e.g., `20230426_calib`). Sequences must utilize the appropriate calibration files and

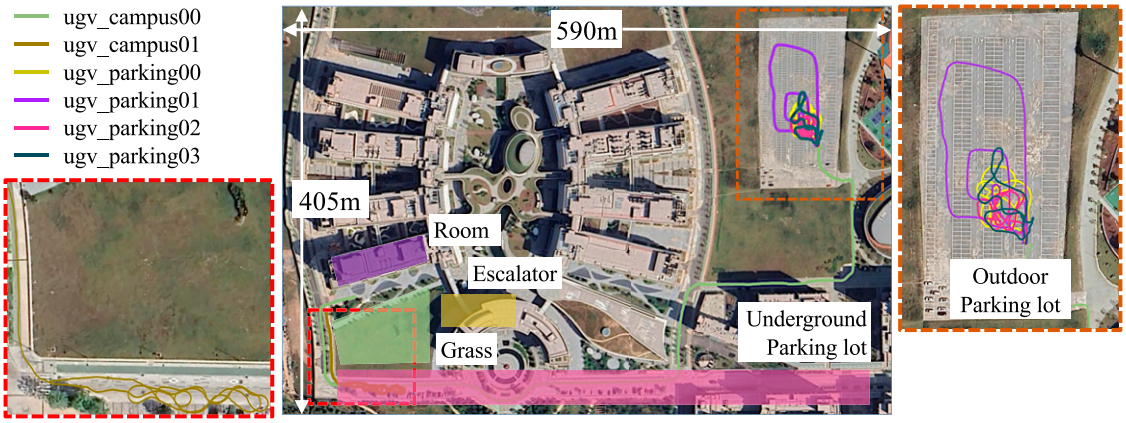


Figure 8. Trajectories of several sequences collected using the low-speed UGV in the campus, where environments with different structures and texture including room, escalator, grassland, parking lots are presented.

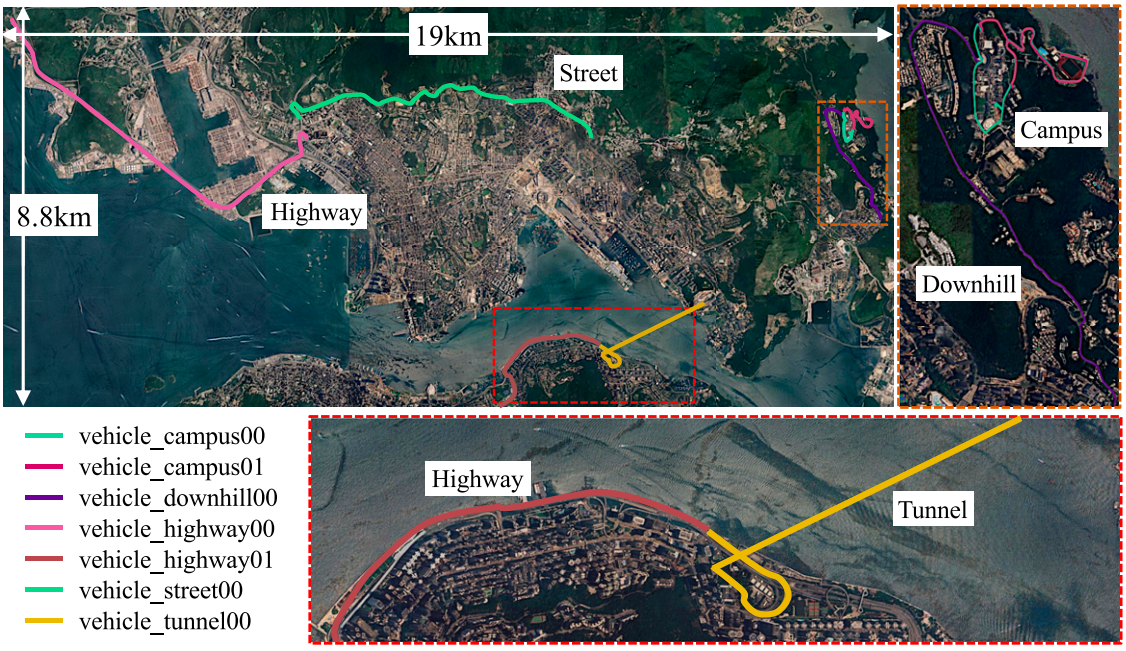


Figure 9. Trajectories of several sequences collected using the high-speed vehicle in Hong Kong.

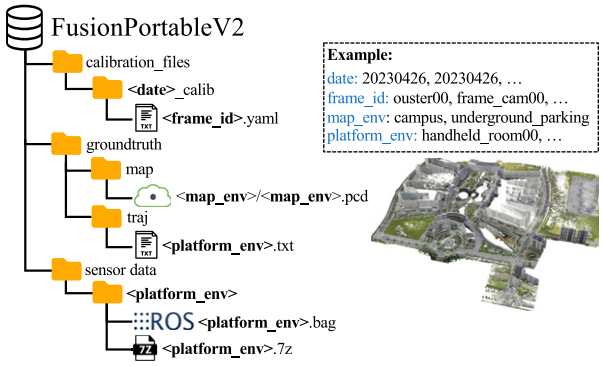


Figure 10. The dataset organization.

such correspondences are provided in the development package. GT poses at the TUM format are recorded in files matching the sequence names, detailing timestamp, orientation (as Hamilton quaternion), and translation vector per line. GT map is provided as the `pcd` format, naming as <**location_environment**>. Please note that all the data provided have undergone additional post-processing steps, following the procedures detailed in Section 6.

5.5. Development tools

We release a set of tools that enable users to tailor our dataset to their specific application needs. Components are introduced as follows:

5.5.1. Software development kit (SDK). We present a Python-only SDK that is both extensible and user-friendly. The kit includes foundational functions such as loading calibration parameters and visualizing them using a TF tree, parsing ROS messages into discrete files, data post-processing, and basic data manipulation. Figure 11 shows the point cloud projection function provided by the package.

5.5.2. Evaluation. We provide a set of scripts and tools for algorithm evaluation including localization and mapping.

5.5.3. Application. We provide open-source repositories for users to try different applications with our dataset covering localization, mapping, monocular depth estimation, and anonymization of specific objects. All can be found on the dataset website.

6. Data post-processing

The raw data captured by sensors and GT devices undergo post-processing before public release. The specifics are outlined as follows.

6.1. Privacy management

Data collection in public spaces such as the campus and urban roads was conducted with strict adherence to privacy regulations. We employed the anonymization technique[13] introduced in (Burnett et al., 2023) to obscure all human faces and license plates in images from our stereo frame cameras. Building upon the original implementation, we enhanced the algorithm's efficiency using the ONNX Runtime Library[14]. This upgraded version is now ROS-compatible, offering a valuable resource to the community, and is included in our development tools.

(a) Handheld_escalator00 (b) Vehicle_campus00

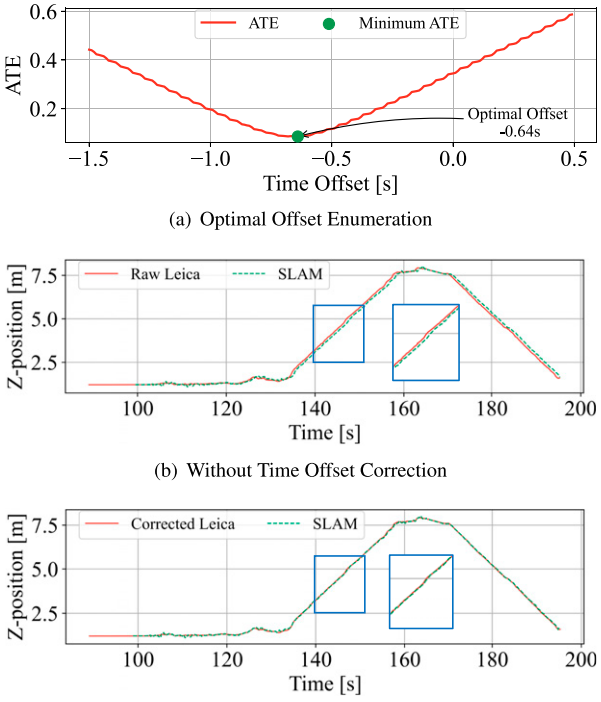
Figure 11. The projected point cloud onto the left frame image using our SDK shows points' colors indicating relative distances. This involves a basic implementation, including a data loader, calibration loader, point cloud manipulation, and camera model.

6.2. GT data processing

Due to diverse sources of GT trajectories and maps, it is necessary to standardize the GT data through processing and conversion and then verify them. These steps should be executed sequentially.

6.2.1. 3-DoF GT poses of total station. The preprocessing initiates with temporal alignment, crucial for synchronizing Leica MS60 total station measurements with sensor data, following the approach proposed in (Nguyen et al., 2022). This synchronization finds the optimal time offset that minimizes the Absolute Trajectory Error (ATE) between the MS60's recorded poses $\mathcal{T}_{ms60}$ and the SLAM-generated poses $\mathcal{T}_{alg}$. This is done by enumerating offsets at intervals of 0.01 s to generate various versions of time-shifted poses from the total station poses, as shown in Figure 12. Upon adjusting their timestamps with the optimal δ, all MS60 poses (recorded at 5–8 Hz) are resampled to a denser 20 Hz sequence via cubic spline interpolation. This method not only yields a smooth, continuous trajectory but also temporally synchronizes the data with SLAM algorithm outputs. To maintain data accuracy, intervals longer than 1 s indicating obstructions are omitted from the sequence.

6.2.2. 6-DoF GT poses of INS. Each 6-DoF pose provided by the INS is accompanied by a variance value, indicating the measurement's uncertainty. This uncertainty increases when the GNSS signal is obstructed. For a fair comparison, we manually removed data points with excessive uncertainty to serve as ground truth. We also provide the original data along with relevant tools for users to process the data according to their needs. For the sequences `vehicle_street00` and `vehicle_tunnel00`, where intermittent GNSS signals disrupt the INS odometry filter convergence, we directly used the original GNSS data sampled at 2 Hz and provide 3-DoF GT poses instead. For the sequence `vehicle_multilayer00` which was collected in a multi-layer parking garage, most of the trajectory is indoors without access to GNSS signals. Therefore, we used the results from the FAST-LIO2 algorithm as the baseline, given the rich structured information available. The start and end points of the sequence are located on the rooftop, where GNSS data can be used as loop closure references.



(a) Optimal Offset Enumeration

(b) Without Time Offset Correction

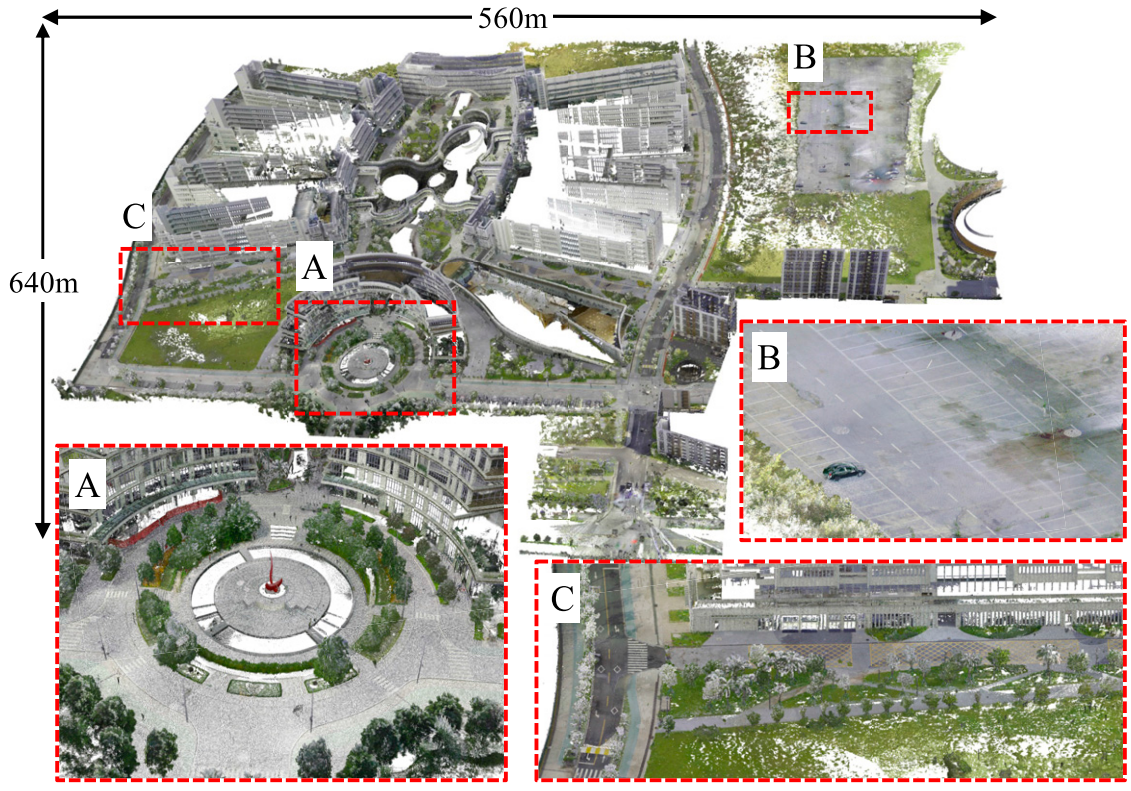
(c) With Time Offset Correction

Figure 12. Alignment process for `handheld_escalator00` sequence. (a) depicts the ATE versus time offset, pinpointing the optimal offset at −0.64 s. (b) and (c) illustrate the Z-axis trajectories for the first 200 seconds, before and after alignment, respectively, highlighting the applied time offset correction.

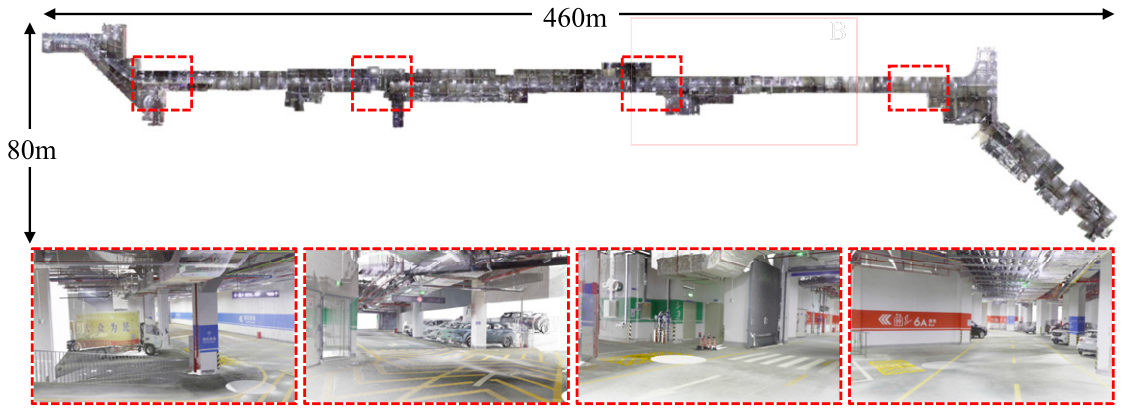
6.2.3. Accuracy of GT maps. In the construction of our ground truth (GT) maps, we employed a high-precision Leica scanner to capture detailed environments both indoors and outdoors. Figure 13 displays the complete RGB point cloud map of the entire campus and a section of the underground parking garage. The fine details observable in the depicted areas highlight the high quality of both the point cloud and its color fidelity. According to the Leica Cyclone software report, the GT map exhibits an average error of less than 15 mm across all pairwise scans, the average error is 3.4 mm and with 90% of these scans maintaining an error margin of 10 mm or less. The precision of our GT map exceeds that of point cloud maps constructed via current LiDAR SLAM technologies by nearly two orders of magnitude, making it suitable for algorithm evaluation.

7. Experiment

We select eight representative sequences (two sequences from each platform) from the dataset to conduct a series



(a) Campus



(b) Underground Parking Lot

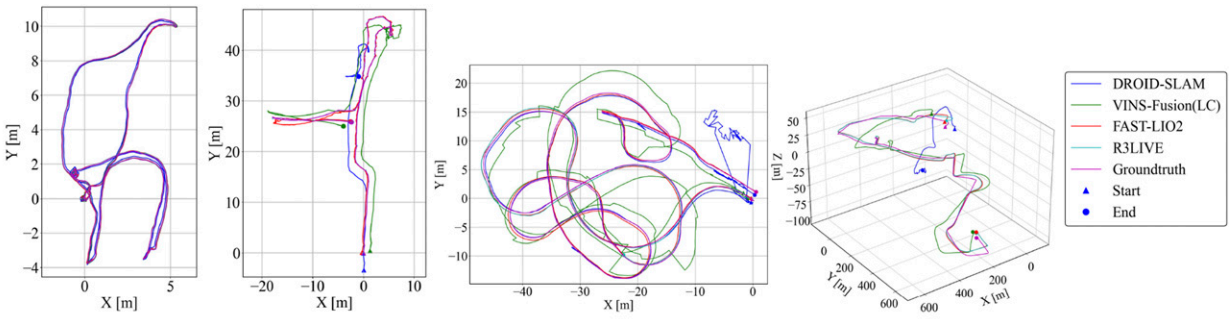
Figure 13. GT RGB point cloud map of the (a) campus ($\approx 0.36 km^2$) with 8 cm-resolution and (b) the underground parking lot ($\approx 0.037 km^2$) with 4 mm-resolution. It almost encompasses the range of most sequences except for those related to vehicles. For detailed map information, please refer to our video presentation.

algorithm evaluation and verification. Experiments include localization, mapping, and monocular depth estimation.

7.1. Evaluation of localization

7.1.1. Experiment setting. As one of the main applications, this dataset can be used to benchmark SOTA SLAM algorithms. Here in, for evaluation of localization systems with different input modalities, we select four SOTA SLAM algorithms (including a learning-based method): DROID-SLAM (left frame camera) (Teed and Deng, 2021), VINS-Fusion (LC) (IMU + stereo frame cameras, with loop closure enabled) (Qin et al., 2018), FAST-LIO2 (IMU+LiDAR) (Xu et al., 2022), and R3LIVE (IMU+LiDAR+left frame camera) (Lin and Zhang, 2022). The customized data loaders of each method are publicly released to foster research.

Marching from traditional to deep learning-based SLAM methods, we evaluate DROID-SLAM, an end-to-end deep visual SLAM algorithm. We employ DROID-SLAM on the monocular image stream with a pre-trained model[15] without fine-tuning to present a fair comparison to model-based



(a) handheld_room00 (b) legged_grass00 (c) ugv_parking00 (d) vehicle_campus00

Figure 14. This figure illustrates the comparative performance of leading SLAM algorithms operating across four distinct platforms and environmental contexts, from indoor spaces to a university campus. It is evident that LiDAR-based methods such as FAST-LIO2 and R3LIVE consistently outperform their vision-based counterparts across all scenarios, maintaining a higher trajectory accuracy. On the other hand, the performance of vision-based algorithms, particularly DROID-SLAM, deteriorates as the environment scale increases, with significant scale recovery issues observed in the expansive `vehicle_campus00` sequence. This trend underscores the superior robustness of LiDAR-based SLAM in varied and large-scale environments.

Table 5. Localization accuracy: We calculate translation ATE [m] for each sequence. The best result among all methods is shown in **bold**, and the best result among vision-based methods (VINS-Fusion and DROID-SLAM) is underlined.

Sequence	R3LIVE	FAST-LIO2	VINS-Fusion (LC)	DROID-SLAM
handheld_room00	***0.057***	0.058	0.063	0.118
handheld_escalator00	0.093	***0.085***	0.258	4.427
legged_grass00	***0.069***	0.327	1.801	7.011
legged_room00	***0.068***	0.093	0.149	0.135
ugv_campus00	***1.486***	1.617	1.866	43.869
ugv_parking00	0.424	***0.271***	2.400	2.019
vehicle_campus00	10.070	***8.584***	66.428	×
vehicle_highway00	×	686.940	×	×

methods. Meanwhile, testing the limits of its generalization ability is crucial for SLAM. The pre-trained model is trained by supervision from optical flow and poses on the synthetic dataset TartanAir (Wang et al., 2020), covering various conditions (e.g., appearance and viewpoint changes) and environments (e.g., from small-scale indoor to large-scale suburban). All the experiments are conducted on NVIDIA GPU GeForce RTX 3090 with a downsampled image resolution of 320 × 240. The average runtime is 16 FPS with a global bundle adjustment (BA) layer. The average GPU memory consumption is below 11 GB.

7.1.2. Evaluation. We choose the typical evaluation metric: mean ATE to evaluate the accuracy of estimated trajectories against the GT using the EVO package.[16] Table 5 reports the quantitative localization results and Figure 14 presents quantificative results.

Our evaluation of SOTA SLAM systems, as summarized in Table 5, demonstrates that each system's performance varies across different environments, depending on its sensor configuration and algorithmic approach. Due to the precise geometric information inherent in LiDAR raw data, methods incorporating LiDAR generally exhibit higher accuracy. However, as scene scale increases and becomes more complex (like the highway), segments lacking visual texture or structural features become challenging. FAST-LIO2, which utilizes IMU and LiDAR data, showcased robust performance across a diverse array of environments. This highlights the inherent strength of LiDAR-based systems in tackling various and complex scenarios. In contrast, R3LIVE, which integrates IMU, LiDAR, and visual data, consistently demonstrated superior accuracy in different settings, particularly outperforming FAST-LIO2 in scenarios where LiDAR degradation and jerky motion pattern are present (e.g., `ugv_campus00`, `legged_grass00`). However, in environments featuring intricate visual features such as water surfaces or reflective glass on the `ugv_parking00`, the presence of visual mechanisms in R3LIVE may lead to a performance decrease.

For vision-based methods, VINS-Fusion outperforms DROID-SLAM on average, demonstrating robustness and generalization ability over learning-based methods. However, it is important to note that DROID-SLAM, using only monocular input, surpasses VINS-Fusion in three specific sequences: `legged_room00`, and `ugv_parking00`. These sequences are characterized by smaller-scale environments with constrained boundaries, such as closed rooms or parking areas. This result highlights the promising potential of employing deep learning techniques in SLAM algorithms, particularly in scenarios where the environment is more confined and structured. The superior performance of DROID-SLAM in these specific cases suggests that learning-based methods can excel in certain conditions, despite their overall lower average performance compared to traditional approaches like VINS-Fusion.

7.2. Evaluation of mapping

Localization and mapping represent the foundational tasks for robotic navigation, and evaluating trajectory accuracy alone

does not suffice to encapsulate the efficacy of such processes comprehensively. Within the framework of SLAM algorithms predicated on Gaussian models, the map serves as a crucial output, and its accuracy assessment indirectly mirrors the precision of localization. For the broader spectrum of mapping tasks, whether conducted online or offline, sparse or dense, direct evaluation of map accuracy remains crucial. Hence, a module dedicated to assessing map accuracy has been developed to address this need, ensuring a holistic appraisal of navigational competencies.

7.2.1. Experiment setting. For evaluating point cloud maps estimated by SOTA SLAM algorithms, we first downsampled the estimated maps using a 0.1 m grid. Initial alignment with the GT map was performed using CloudCompare software. We set the maximum threshold distance for corresponding points at 0.2 m. Thus, after evaluation, point pairs with a distance less than 0.2 m were considered as the same point for distance calculation.

7.2.2. Evaluation. We use the mapping evaluation metrics in PALoc (Hu et al., 2024b) to complement our localization evaluation. After initial alignment, the error metrics were then calculated using our map evaluation library as introduced in Section 3.3.1. In the presence of high-precision RGB point cloud map ground truth, the accuracy of the maps reconstructed by the algorithm can be evaluated. We register the estimated point cloud map $\mathcal{M}$, reconstructed by the algorithm, to the ground truth point cloud map $\mathcal{G}$, subsequently obtaining the corresponding set of associated points. By computation of precision metrics on this set of associated points, we can also mitigate minor map variations due to temporal discrepancies in ground truth point cloud collection and slight interference from dynamic obstacles. We compare these two maps based on four metrics: *Reconstruction Error (RE)* in terms of the Root Mean Squared Error (RMSE), *Completeness (COM)*, and *Chamfer Distance (CD)*. They are defined as below:

- *Reconstruction Error* computes the average point-to-point distance between $\mathcal{M}$ and $\mathcal{G}$ (Pan et al., 2024):

$$RE = \sqrt{\frac{1}{|\mathcal{M}|}\sum_{\boldsymbol{p}\in\mathcal{M}} \underbrace{\min(\tau,\ \|\boldsymbol{p}-\boldsymbol{q}\|)}_{d(\boldsymbol{p},\mathcal{G})}^{2}}. \tag{1}$$

where τ is the inlier distance and $\mathbf{q} \in \mathcal{G}$ is the nearest point to $\mathbf{q}$. We empirically set $\tau = 0.2$ m in experiments.

- *Completeness* describes how does $\mathcal{M}$ cover extent of $\mathcal{G}$. $\mathcal{G}'$ is the subset of $\mathcal{G}$. Each element of $\mathcal{G}'$ has a nearby point from $\mathcal{M}$ such as

$$\begin{aligned} COM &= \frac{|\mathcal{G}'|}{|\mathcal{G}|}, \\ \mathcal{G}' &= \{\mathbf{q}\in\mathcal{G} \mid \quad \exists \mathbf{p}\in\mathcal{M}, \|\mathbf{p}-\mathbf{q}\|\leq\tau\}. \end{aligned} \tag{2}$$

- *Chamfer Distance* computes the Chamfer-L1 Distance (Mescheder et al., 2019) as:

$$CD = \frac{1}{2|M|}\sum_{\boldsymbol{p}\in\mathcal{M}} d(\boldsymbol{p},\mathcal{G}) + \frac{1}{2|\mathcal{G}|}\sum_{\boldsymbol{q}\in\mathcal{G}} d(\boldsymbol{q},\mathcal{M}). \tag{3}$$

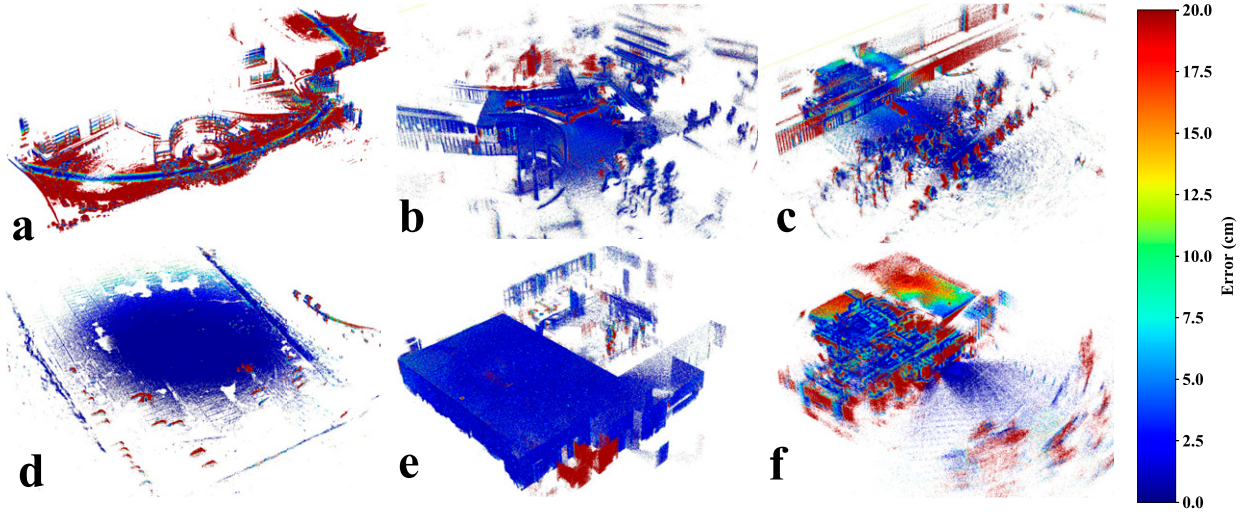


Figure 15. This figure presents the mapping performance of FAST-LIO2 in various environments: (a) `ugv_campus00`, (b) `handheld_escalator00`, (c) `legged_grass00`, (d) `ugv_parking00`, (e) `handheld_room00`, and (f) `legged_room00`. The color gradient, from red to blue, illustrates the range of errors in the map points generated by FAST-LIO2, with red indicating higher errors (up to 20 cm) and blue denoting lower errors (down to 0), where deeper blue signifies higher mapping precision. Notably, (a) shows significant z-axis drift in an outdoor large-scale scenario, resulting in predominantly high-error red areas in the map evaluation. Conversely, (f) Illustrates the algorithm's application on a quadruped platform in an indoor office environment characterized by intense ground movement, glass-induced noise, and numerous dynamic obstacles, which are depicted by the red areas signifying higher error.

Table 6. Mapping accuracy: We calculate four metrics to evaluate *FAST-LIO2* ***(FL2)*** and *R3LIVE* ***(R3L)***.

	RE [m, ↓]		COM [%, ↑]		CD [m, ↓]	
Sequence	FL2	R3L	FL2	R3L	FL2	R3L
handheld_room00	***0.144***	0.269	***0.949***	0.802	***0.109***	0.131
handheld_escalator00	***0.273***	0.544	***0.846***	0.340	0.128	***0.126***
legged_grass00	***0.092***	0.199	***0.818***	0.406	***0.158***	0.161
legged_room00	0.442	***0.196***	***0.445***	0.316	***0.132***	0.163
ugv_campus00	***0.765***	0.767	***0.232***	0.217	0.112	***0.107***
ugv_parking00	***0.105***	0.122	***0.956***	0.567	***0.110***	0.166

Figure 15 employs the map evaluation module to present the assessment outcomes of the FAST-LIO2 algorithm across six indoor and outdoor data sequences, with varying colors representing the accuracy levels across different map regions. The map accuracy estimated by FAST-LIO2 notably decreases in outdoor large-scale scenes (Figure 15(a)) or areas with dense vegetation (Figure 15(c) and Figure 15(f)), attributable to significant measurement noise from trees or overall z-axis drift in outdoor LiDAR odometry and mapping applications. Conversely, indoor settings, barring the effects introduced by dynamic obstacles (Figure 15(f)), predominantly exhibit high map quality (Figure 15(e)).

Table 6 further delineates the map evaluation results for FAST-LIO2 and R3LIVE across six indoor and outdoor sequences, with R3LIVE retaining only points with RGB colors, hence showing inferior performance on the COM metric compared to FAST-LIO2. However, FAST-LIO2 outperforms R3LIVE in most scenes in terms of RE and CD metrics, particularly in handheld and quadruped robot sequences. In expansive campus environments, both algorithms exhibit comparable CD metrics, as seen in scenarios like `handheld_escalator` and `ugv_campus`. Figure 16 illustrates the mapping results of R3LIVE on the `handheld_room00` sequence, employing a color scheme consistent with that of Figure 15. The presence of glass within the room introduces noise, resulting in some blurred regions within the map. This depiction underscores the impact of environmental features on mapping clarity.

7.3. Evaluation of depth estimation

The diversity of sensors, mobile platforms, and scenarios make our dataset appealing for algorithm verification not limited to localization and mapping. In this section, we demonstrate that our dataset can serve for the evaluation of advanced perception algorithms. Due to the easily accessible GT, we set the benchmark for measuring the generalization ability of unsupervised monocular depth prediction. The benchmark measures how unsupervised monocular depth prediction networks could perform on scenes collected from different data collection platforms.

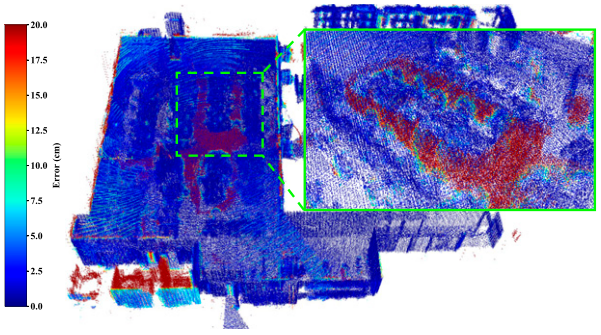


Figure 16. This figure demonstrates the map evaluation results within the `handheld_room00` dataset using R3LIVE. The estimated point cloud map is compared against the ground truth map, with the color gradient from blue to red indicating accuracy discrepancies ranging from 0 to 20 cm. The inset highlights significant errors in the seating area within the room.

7.3.1. Data preparation. Each frame image is accompanied by a GT depth image of identical size for evaluation. Depth images are produced by projecting point clouds, generated by FAST-LIO2 (Xu et al., 2022) through IMU interpolation, onto these frames:

$$D_{gt}(\mathbf{x}) = Z, \quad \mathbf{x} = \lfloor \pi(\mathbf{p}^c) \rceil, \quad \mathbf{p}^c = \mathbf{R}_l^c \mathbf{p}^l + \mathbf{t}_l^c. \tag{4}$$

where Z is the z-axis value of $\mathbf{p}^c$, $\pi(\cdot)$ is the camera projection function, $\lfloor \cdot \rceil$ is the rounding operation, and $(\mathbf{R}_l^c, \mathbf{t}_l^c)$ represents the extrinsics from the left frame camera to the LiDAR.

7.3.2. Experiment setting. Monocular depth estimation tests are essential for evaluating a system's ability to perceive relative object distances from a single camera view, a key aspect of understanding spatial relationships. Based on the FSNet that is a self-supervised depth estimation model (Liu et al., 2023), we fine-tune this model with our dataset using GT poses. We organize the train-validation data into two groups. The first group allocates 70% of handheld indoor sequence data (i.e., `handheld_room00` and `handheld_escalator00`) for training, and combines 30% of this data with all vehicle-related outdoor sequences for validation. The second group uses 70% of vehicle sequence data (i.e., `vehicle_campus00` and `vehicle_parking00`) for training, and blends 30% of this data with all handheld sequences for validation. We train the FSNet with these groups of data, respectively, and obtain two models: **FSNet-Handheld** and **FSNet-Vehicle**.

7.3.3. Evaluation. We assess models' performance of unsupervised monocular depth prediction models use the proposed scale-invariant metrics in (Eigen et al., 2014): *Absolute Relative Difference* (ARD), *Squared Relative Difference* (SRD), *Root*

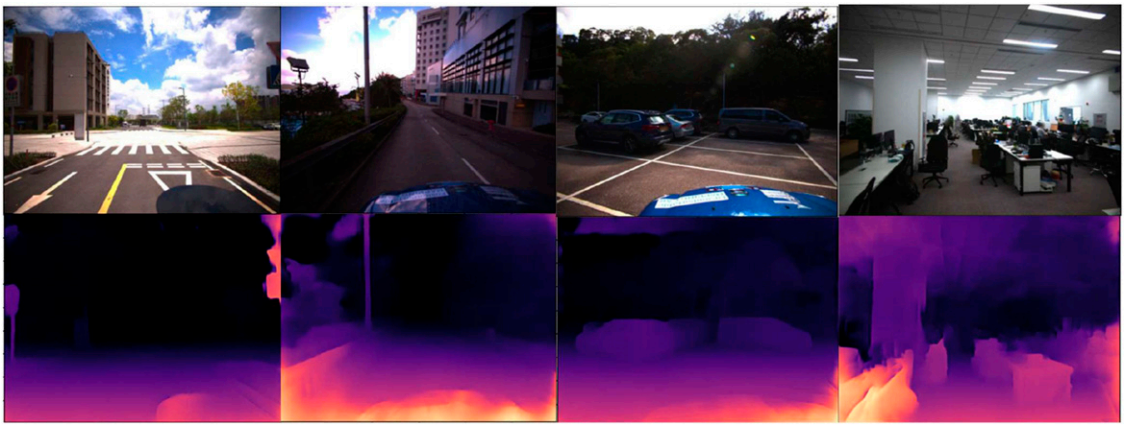


Figure 17. Results of the unsupervised depth prediction method Liu et al. (2023) which can generalize over different environments.

Table 7. Performance of FSNet MonoDepth (Liu et al., 2023) on FusionPortableV2. Results are categorized by the training domain and testing domain. The left four metrics: ARD, SRD, RMSE-linear, and RMSE-log are error metrics (the lower the better). The right three metric: $\delta < \delta_{thr}$ are accuracy metrics (the higher the better).

Model	Test sequence	ARD	SRD	RMSE-linear [m, ↓],	RMSE-log	$\delta < 1.25$ [%, ↑]	$\delta < 1.25^2$ [%, ↑]	$\delta < 1.25^3$ [%, ↑]
FSNet-Handheld	Handheld	0.592	7.885	5.750	0.552	0.440	0.697	0.825
	Vehicle	0.235	1.724	5.210	0.287	0.670	0.887	0.961
FSNet-Vehicle	Handheld	1.031	15.786	6.970	0.742	0.300	0.531	0.679
	Vehicle	***0.125***	***1.522***	***4.561***	***0.199***	***0.882***	***0.955***	***0.978***

Mean Squared Error (RMSE)-linear, RMSE-log, and *Threshold*. They are defined as follows:

$$\begin{aligned}
\text{ARD} &= \frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}\in\mathcal{X}} |d_{est} - d_{gt}|/d_{gt}, \\
\text{SRD} &= \frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}\in\mathcal{X}} \|d_{est} - d_{gt}\|^2/d_{gt}, \\
\text{RMSE}-\text{linear} &= \sqrt{\frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}\in\mathcal{X}} \|d_{est} - d_{gt}\|^2}, \\
\text{RMSE}-\text{log} &= \sqrt{\frac{1}{|\mathcal{X}|}\sum_{\mathbf{x}\in\mathcal{X}} \|\log d_{est} - \log d_{gt}\|^2}, \\
\text{Threshold} &= \%\ \text{of}\ d_{est} \\
&\text{s.t. } \max\left(\frac{d_{est}}{d_{gt}}, \frac{d_{gt}}{d_{est}}\right) = \delta \leq \delta_{thr}.
\end{aligned} \tag{5}$$

where d_{est} is the estimated depth value at the pixel $\mathbf{x}$ with the corresponding GT depth d_{gt} and where $\delta_{thr} \in \{1.25, 1.25^2, 1.25^3\}$. Quantitative and some qualitative results are presented in Figure 17 and Table 7, respectively. These results not only validate our dataset for depth estimation but also highlight the limitations of current unsupervised depth prediction techniques. These limitations are revealed as the FSNet-Handheld model struggles with generalization to handheld sequences, despite training on data with similar appearance. The challenge for monocular depth estimation is further amplified by the significant scale variation in indoor sequences. Advancements in depth formulation and learning strategies are expected to markedly improve the performance in future benchmarks. For a more detailed and comprehensive analysis, we encourage viewing the dataset videos available on our website.

7.4. Known issues and limitations

Creating a comprehensive dataset spanning multiple platforms, sensors, and scenes is labor-intensive. Despite our efforts to resolve many issues, we acknowledge the presence of several imperfections within the dataset. We detail these common challenges in the subsequent sections and present our technical solutions. We hope this discussion will provide valuable insights and lessons for future researchers.

7.4.1. Calibration. Achieving the life-long sensor calibration poses significant challenges (Maddern et al., 2017). We try our best to provide the best estimate of calibration parameters. Calibration was performed each time when the data collection platform changed, employing SOTA methods for parameter adjustments, which were also manually verified and fine-tuned. For extrinsic parameters difficult to estimate, such as the relative transformation between specific components, we refer to the CAD model. Efforts were made to reinforce the mechanical structure and

minimize external disturbances during the data collection process. Nevertheless, it is acknowledged that high accuracy for specific traversals cannot be assured. We encourage users to use our calibration estimates as initial values and to explore innovative approaches for long-term extrinsic calibration, as suggested in these studies (Luo et al., 2024; Ulrich and Hillemann, 2023; Wu et al., 2021). To aid in these endeavors, we provide raw calibration data and reports, allowing users to develop their methodologies and consider our estimates as a foundational benchmark.

7.4.2. Synchronization. Section 3.1.1 presents our hardware synchronization solution that guarantees the IMU, frame cameras, and LiDAR are triggered by the same clock source. However, the timestamp of the ROS message of each sensor data has minor differences since the time of data transmission and decode varies. For vehicle-related sequences, the average relative time latency (ARTL) among stereo frame images is smaller than 20 ms. This is mainly caused by the long connection between the camera and the signal trigger. For other sequences, the ARTL is smaller than 5 ms. Due to the special design of the event cameras, the ARTL between the left event camera and the LiDAR is unstable and sometimes smaller than 15 ms.

7.4.3. Partial loss of information. In the construction of the dataset, real-world challenges have led to partial loss of sensor information in certain sequences, reflecting practical issues encountered during robotic deployment. Specifically, in `ugv_transition00` and `ugv_transition01`, the wheel encoder driver was not activated correctly, resulting in the absence of wheel encoder data. Additionally, the `legged_grass00` sequence experienced a brief interruption in data transmission, amounting to several seconds of lost data, due to a loose RJ45 network port connection. These instances underscore the importance of robustness in algorithm development to handle incomplete or missing sensor data in realistic operational conditions.

7.4.4. Camera exposure setting. To ensure image consistency during the whole sequence, we fixed the camera exposure time with a specific value before collecting each sequence, mitigating color varies from illumination changes. This scheme is also important to stereo matching since consistent brightness is commonly desirable. However, this scheme can darken images in significantly different lighting conditions, such as entering a tunnel. The darker appearance can be a challenge for most visual perception algorithms.

7.4.5. Limited diversity and volume. Drawing inspiration from foundational models in computer vision and natural language processing, developing a robot-agnostic general model for robotics could be an exciting avenue for future research (Khazatsky et al., 2024; Padalkar et al., 2023; Shah et al., 2023a). However, acquiring a large, diverse real-world dataset poses significant challenges. We address key problems in data collection for field robots, including system integration and data post-processing. But we acknowledge the limitations in our dataset's scale and diversity: (1) **Platforms**: Our dataset does not include drones, underwater robots, or multi-robot systems. (2) **Scenarios**: Critical environments such as factories, forests, and underground mines, which are also important for SLAM but not presented. (3) **Time Periods**: The dataset was collected over a short time frame, lacking day–night cycles, seasonal variations, and structural changes, which are crucial for place recognition and long-term SLAM (Barnes et al., 2020; Carlevaris-Bianco et al., 2016). (4) **Volume**: The dataset is relatively small, with a limited number of sequences and short durations. FusionPortableV2 is just the beginning. As more researchers and institutions contribute to this field, we anticipate the development of larger, more diverse datasets that will support the creation of a truly generalizable robotic model.

8. Conclusion and future work

This paper presents the FusionPortableV2 dataset, a comprehensive multi-sensor collection designed to advance research in SLAM and mobile robot navigation. The dataset is built around a compact, multi-sensor device that integrates IMUs, stereo cameras (both frame-based and event-based), LiDAR, and INS, all carefully calibrated and synchronized. This primary device is deployed on various platforms, including a legged robot, a low-speed UGV, and a high-speed vehicle, each equipped with additional platform-specific sensors such as wheel encoders and legged sensors. The FusionPortableV2 dataset features a diverse range of environments, spanning indoor spaces, grasslands, campuses, parking lots, tunnels, downhill roads, and highways. This environmental diversity challenges existing SLAM and navigation technologies with realistic scenarios involving dynamic objects and variable lighting conditions. To ensure the dataset's utility for the research community, we have meticulously designed 27 sequences, totaling 2.5 hours of data, and provided ground truth data for the objective evaluation of SOTA methods in localization, mapping, and monocular depth estimation.

As we explore future directions, we aim to enhance this dataset's applicability beyond SLAM by developing novel navigation methods based on the proposed dataset. We will continue to improve the quality of the data and the integration of the system to facilitate easier use by non-expert users. Alongside this dataset, we also release our implementation details and tools to encourage further research advancements. Future update will be provided on the dataset's website.

Acknowledgments

This research greatly benefited from the guidance and expertise of many contributors. We extend our profound gratitude to colleagues: Xiaoyang Yan, Ruoyu Geng, Lu Gan, Bowen Yang,

Tianshuai Hu, Mingkai Jia, Mingkai Tang, Yuanhang Li, Shuyang Zhang, Bonan Liu, Jinhao He, Sheng Wang, Ren Xin, Yingbing Chen, etc. from HKUST and HKUSTGZ for their suggestions in improving our dataset's quality. Special acknowledgment goes to the BIM Lab at HKUST, particularly Prof. Jack Chin Pang Cheng and his students, for their crucial BLK360 scanning expertise and insights that were essential in shaping our dataset. The authors also gratefully acknowledge Dr. Thien-Minh Nguyen (NTU) and Ms. Qingwen Zhang (KTH) for their insightful feedback and technical support; Prof. Dimitrios Kanoulas (UCL), Dr. Martin Magnusson (Örebro University), Prof. Hong Zhang (Southern University of Science and Technology), and Dr. Peng Yin (CityU Hong Kong) for their invaluable dataset writing advice; Seth G. Isaacson (University of Michigan) for his suggestions on dataset structure; and OpenAI's GPT for enhancing the paper's linguistic quality. The author(s) received no financial support for the research, authorship, and/or publication of this article.

Declaration of conflicting interests

The author(s) declared no potential conflicts of interest with respect to the research, authorship, and/or publication of this article.

Funding

The author(s) received no financial support for the research, authorship, and/or publication of this article.

ORCID iDs

Hexiang Wei https://orcid.org/0000-0002-8025-278X
Jianhao Jiao https://orcid.org/0000-0001-7372-266X
Xiangcheng Hu https://orcid.org/0000-0003-3535-6886
Jingwen Yu https://orcid.org/0000-0003-3336-3935
Yilong Zhu https://orcid.org/0000-0001-5332-0794

Supplemental Material

Supplemental material for this article is available online.

Notes

1. https://static.ouster.dev/sensor-docs/image_route1/image_route2/time_sync/time-sync.html
2. https://www.sensonor.com/media/1132/ts1524r9-datasheet-stim300
3. https://leica-geosystems.com/products/laser-scanners/software/leica-cyclone/leica-cyclone-register-360
4. https://github.com/JokerJohn/Cloud_Map_Evaluation
5. https://leica-geosystems.com/en-us/products/total-stations/multistation/leica-nova-ms60
6. https://github.com/ori-drs/allan_variance_ros
7. https://www.mathworks.com/help/vision/camera-calibration.html
8. https://github.com/ethz-asl/kalibr/wiki/Multi-IMU-and-IMU-intrinsic-calibration
9. https://github.com/ethz-asl/kalibr/wiki/camera-imu-calibration
10. https://github.com/unitreerobotics/unitree_ros/blob/master/robots/a1_description/urdf/a1.urdf
11. https://github.com/HKUSTGZ-IADC/LCECalib
12. https://wiki.ros.org/rosbag
13. https://github.com/understand-ai/anonymizer
14. https://onnxruntime.ai
15. https://github.com/princeton-vl/DROID-SLAM
16. https://github.com/MichaelGrupp/evo

References

Agarwal S, Vora A, Pandey G, et al. (2020) Ford multi-av seasonal dataset. *The International Journal of Robotics Research* 39(12): 1367–1376.

Barnes D, Gadd M, Murcutt P, et al. (2020) The Oxford Radar RobotCar dataset: a radar extension to the Oxford RobotCar dataset. In: *2020 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 6433–6438.

Brohan A, Brown N, Carbajal J, et al. (2022) Rt-1: robotics transformer for real-world control at scale. arXiv preprint arXiv:2212.06817.

Brohan A, Brown N, Carbajal J, et al. (2023) Rt-2: vision-language-action models transfer web knowledge to robotic control. arXiv preprint arXiv:2307.15818.

Burnett K, Yoon DJ, Wu Y, et al. (2023) Boreas: a multi-season autonomous driving dataset. *The International Journal of Robotics Research* 42(1-2): 33–42.

Burri M, Nikolic J, Gohl P, et al. (2016) The euroc micro aerial vehicle datasets. *The International Journal of Robotics Research* 35(10): 1157–1163.

Carlevaris-Bianco N, Ushani AK and Eustice RM (2016) University of Michigan north campus long-term vision and lidar dataset. *The International Journal of Robotics Research* 35(9): 1023–1035.

Chaney K, Cladera F, Wang Z, et al. (2023) M3ed: multi-robot, multi-sensor, multi-environment event dataset. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 4015–4022.

Delmerico J, Cieslewski T, Rebecq H, et al. (2019) Are we ready for autonomous drone racing? the uzh-fpv drone racing dataset *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 6713–6719.

Eigen D, Puhrsch C and Fergus R (2014) Depth map prediction from a single image using a multi-scale deep network. *Advances in Neural Information Processing Systems* 27.

Feng D, Qi Y, Zhong S, et al. (2022) S3e: A large-scale multimodal dataset for collaborative slam. arXiv preprint arXiv: 2210.13723.

Furgale P, Rehder J and Siegwart R (2013) Unified temporal and spatial calibration for multi-sensor systems. In: *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 1280–1286.

Furrer F, Fehr M, Novkovic T, et al. (2018) Evaluation of combined time-offset estimation and hand-eye calibration on robotic datasets *Field and Service Robotics*. Springer, 145–159.

Gadd M, De Martini D, Bartlett O et al. (2024) Oord: the oxford offroad radar dataset. arXiv preprint arXiv:2403.02845.

Gao L, Liang Y, Yang J, et al. (2022) Vector: a versatile event-centric benchmark for multi-sensor slam. *IEEE Robotics and Automation Letters* 7(3): 8217–8224.

Gehrig M, Aarents W, Gehrig D, et al. (2021) Dsec: a stereo event camera dataset for driving scenarios. *IEEE Robotics and Automation Letters* 6(3): 4947–4954.

Geiger A, Lenz P, Stiller C, et al. (2013) Vision meets robotics: the kitti dataset. *The International Journal of Robotics Research* 32(11): 1231–1237.

Hu X, Wu J, Jiao J, et al. (2024a) Ms-mapping: an uncertainty-aware large-scale multi-session lidar mapping system. arXiv preprint arXiv:2408.03723.

Hu X, Zheng L, Wu J, et al. (2024b) Paloc: advancing slam benchmarking with prior-assisted 6-dof trajectory generation and uncertainty estimation. *IEEE/ASME Transactions on Mechatronics*.

Huang AS, Antone M, Olson E, et al. (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 YS, et al. (2019) Complex urban dataset with multi-level sensors from highly diverse urban environments. *The International Journal of Robotics Research* 38(6): 642–657.

Jiang P, Osteen P, Wigness M, et al. (2021) RELLIS-3D Dataset: data, benchmarks and analysis. In: *2021 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 1110–1116.

Jiao J, Wei H, Hu T, et al. (2022) Fusionportable: a multi-sensor campus-scene dataset for evaluation of localization and mapping accuracy on diverse platforms. In: *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 3851–3856.

Jiao J, Chen F, Wei H, et al. (2023) LCE-Calib: automatic lidar-frame/event camera extrinsic calibration with a globally optimal solution. *IEEE/ASME Transactions on Mechatronics* 28(5): 2988–2999.

Jung M, Yang W, Lee D, et al. (2023) Helipr: heterogeneous lidar dataset for inter-lidar place recognition under spatiotemporal variations. *The International Journal of Robotics Research*. Online.

Kerbl B, Kopanas G, Leimkühler T, et al. (2023) 3d Gaussian splatting for real-time radiance field rendering. *ACM Transactions on Graphics* 42(4): 1–14.

Khazatsky A, Pertsch K, Nair S et al. (2024) Droid: a large-scale in-the-wild robot manipulation dataset. arXiv preprint arXiv: 2403.12945.

Knights J, Vidanapathirana K, Ramezani M, et al. (2023) Wild-Places: a large-scale dataset for lidar place recognition in unstructured natural environments *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 11322–11328.

Li H, Zou Y, Chen N, et al. (2024) Mars-lvig dataset: a multi-sensor aerial robots slam dataset for lidar-visual-inertial-gnss fusion. *The International Journal of Robotics Research*. Online.

Lin J and Zhang F (2022) R3live: a robust, real-time, rgb-colored, lidar-inertial-visual tightly-coupled state estimation and mapping package. In: *2022 International Conference on Robotics and Automation (ICRA)*. IEEE, 10672–10678.

Liu T, hai Liao Q, Gan L, et al. (2021) The role of the hercules autonomous vehicle during the covid-19 pandemic: an autonomous logistic vehicle for contactless goods transportation. *IEEE Robotics & Automation Magazine* 28(1): 48–58.

Liu Y, Xu Z, Huang H, et al. (2023) Fsnet: redesign self-supervised monodepth for full-scale depth prediction for autonomous driving. *IEEE Transactions on Automation Science and Engineering*. 1–11. DOI: 10.1109/TASE.2023.3290348.

Luo Z, Yan G, Cai X, et al. (2024) Zero-training lidar-camera extrinsic calibration method using segment anything model. In: *2024 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 14472–14478.

Maddern W, Pascoe G, Linegar C, et al. (2017) 1 year, 1000 km: the oxford robotcar dataset. *The International Journal of Robotics Research* 36(1): 3–15.

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

Matsuki H, Murai R, Kelly PH, et al. (2024) Gaussian splatting slam. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 18039–18048.

Mescheder L, Oechsle M, Niemeyer M, et al. (2019) Occupancy networks: learning 3d reconstruction in function space. In: *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 4460–4470.

Miller M, Chung SJ and Hutchinson S (2018) The visual–inertial canoe dataset. *The International Journal of Robotics Research* 37(1): 13–20.

Mueggler E, Rebecq H, Gallego G, et al. (2017) The event-camera dataset and simulator: event-based data for pose estimation, visual odometry, and slam. *The International Journal of Robotics Research* 36(2): 142–149.

Nguyen TM, Yuan S, Cao M, et al. (2022) Ntu viral: a visual-inertial-ranging-lidar dataset, from an aerial vehicle viewpoint. *The International Journal of Robotics Research* 41(3): 270–280.

Padalkar A, Pooley A, Jain A, et al. (2023) Open x-embodiment: robotic learning datasets and rt-x models. arXiv preprint arXiv:2310.08864.

Pan Y, Zhong X, Wiesmann L, et al. (2024) Pin-slam: lidar slam using a point-based implicit neural representation for achieving global map consistency. arXiv preprint arXiv: 2401.09101.

Pfrommer B, Sanket N, Daniilidis K, et al. (2017) Penncosyvio: a challenging visual inertial odometry benchmark. In: *2017 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 3847–3854.

Pire T, Mujica M, Civera J, et al. (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.

Ramezani M, Wang Y, Camurri M, et al. (2020) The newer college dataset: handheld lidar, inertial and vision with ground truth. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4353–4360.

Rehder J, Nikolic J, Schneider T, et al. (2016a) Extending kalibr: calibrating the extrinsics of multiple imus and of individual axes. In: *2016 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4304–4311.

Rehder J, Siegwart R and Furgale P (2016b) A general approach to spatiotemporal calibration in multisensor systems. *IEEE Transactions on Robotics* 32(2): 383–398.

Reinke A, Palieri M, Morrell B, et al. (2022) Locus 2.0: robust and computationally efficient lidar odometry for real-time 3d mapping. *IEEE Robotics and Automation Letters* 7(4): 9043–9050.

Schubert D, Goll T, Demmel N, et al. (2018) The tum vi benchmark for evaluating visual-inertial odometry. In: *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 1680–1687.

Shah D, Sridhar A, Bhorkar A, et al. (2023a) Gnm: a general navigation model to drive any robot. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 7226–7233.

Shah D, Sridhar A, Dashora N, et al. (2023b) Vint: a foundation model for visual navigation. arXiv preprint arXiv:2306.14846.

Shi Y, Wu F, Perincherry A, et al. (2023) Boosting 3-dof ground-to-satellite camera localization accuracy via geometry-guided cross-view transformer. In: *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 21516–21526.

Teed Z and Deng J (2021) Droid-slam: deep visual slam for monocular, stereo, and rgb-d cameras. *Advances in Neural Information Processing Systems* 34: 16558–16569.

Tian Y, Chang Y, Quang L, et al. (2023) Resilient and distributed multi-robot visual slam: datasets, experiments, and lessons learned. In: *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 11027–11034.

Ulrich M and Hillemann M (2023) Uncertainty-aware hand–eye calibration. *IEEE Transactions on Robotics*.

Wang W, Zhu D, Wang X, et al. (2020) Tartanair: a dataset to push the limits of visual slam. In: *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 4909–4916.

Wisth D, Camurri M and Fallon M (2022) Vilens: visual, inertial, lidar, and leg odometry for all-terrain legged robots. *IEEE Transactions on Robotics* 39(1): 309–326.

Wu J, Wang M, Jiang Y, et al. (2021) Simultaneous hand–eye/robot–world/camera–imu calibration. *IEEE/ASME Transactions on Mechatronics* 27(4): 2278–2289.

Xu W, Cai Y, He D, et al. (2022) Fast-lio2: fast direct lidar-inertial odometry. *IEEE Transactions on Robotics* 38(4): 2053–2073.

Yang S, Zhang Z, Fu Z, et al. (2023) Cerberus: low-drift visual-inertial-leg odometry for agile locomotion. In: *2023 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 4193–4199.

Yin J, Li A, Li T, et al. (2021) M2dgr: a multi-sensor and multi-scenario slam dataset for ground robots. *IEEE Robotics and Automation Letters* 7(2): 2266–2273.

Yin P, Zhao S, Ge R, et al. (2022) ALITA: a large-scale incremental dataset for long-term autonomy. arXiv preprint arXiv:2205.10737.

Yin P, Jiao J, Zhao S, et al. (2024) General place recognition survey: towards real-world autonomy. arXiv preprint arXiv: 2405.04812.

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

Zhang L, Helmberger M, Fu LFT, et al. (2022) Hilti-Oxford Dataset: a millimeter-accurate benchmark for simultaneous localization and mapping. *IEEE Robotics and Automation Letters* 8(1): 408–415.

Zhu AZ, Thakur D, Özaslan T, et al. (2018) The multivehicle stereo event camera dataset: an event camera dataset for 3d perception. *IEEE Robotics and Automation Letters* 3(3): 2032–2039.

Zhu Y, Kong Y, Jie Y, et al. (2023) Graco: a multimodal dataset for ground and aerial cooperative localization and mapping. *IEEE Robotics and Automation Letters* 8(2): 966–973.

Zuñiga-Noël D, Jaenal A, Gomez-Ojeda R, et al. (2020) The uma-vi dataset: visual–inertial odometry in low-textured and dynamic illumination environments. *The International Journal of Robotics Research* 39(9): 1052–1060.