Date of publication Month Day, YYYY, date of current version September 19, 2025.
Digital Object Identifier 10.1109/ACCESS.2025.0429000

Indoor Benchmark of 3D LiDAR SLAM at iilab – Industry and Innovation Laboratory

JORGE DIOGO RIBEIRO[1,2], RICARDO B. SOUSA[1,2], JOÃO G. MARTINS[1,2], ANDRÉ S. AGUIAR[2], FILIPE N. SANTOS[2], and HÉBER MIGUEL SOBREIRA[2]
[1]Faculty of Engineering, University of Porto (FEUP), 4200-465 Porto, Portugal (email: rbs@fe.up.pt)
[2]INESC TEC – Institute for Systems and Computer Engineering, Technology and Science, 4200-465 Porto, Portugal
(email: {jorge.d.ribeiro,joao.g.martins,andre.s.aguiar,filipe.n.santos,heber.m.sobreira}@inesctec.pt)

Corresponding author: Jorge Diogo Ribeiro (e-mail: jorge.d.ribeiro@inesctec.pt).

This work is co-financed by Component 5 – Capitalization and Business Innovation, integrated in the Resilience Dimension of the Recovery and Resilience Plan within the scope of the Recovery and Resilience Mechanism (MRR) of the European Union (EU), framed in the Next Generation EU, for the period 2021–2026, within project GreenAuto, with reference 54.

ABSTRACT This paper presents the IILABS 3D – iilab Indoor LiDAR-based SLAM 3D dataset, a novel and publicly available resource designed to address current limitations in indoor benchmarking of 3D Light Detection And Ranging (LiDAR)-based Simultaneous Localization and Mapping (SLAM) algorithms. Existing SLAM datasets often focus on outdoor environments, rely on a single type of LiDAR sensor, or lack ground-truth data suitable for evaluating diverse indoor conditions. IILABS 3D fills this gap by providing a sensor-rich, indoor-exclusive dataset recorded in a controlled laboratory environment using a wheeled mobile robot platform. It includes four heterogeneous 3D LiDAR sensors – Velodyne VLP-16, Ouster OS1-64, RoboSense RS-Helios-5515, and Livox Mid-360 – featuring both mechanical spinning and non-repetitive scanning patterns, as well as an IMU and wheel odometry. The dataset also features calibration sequences, challenging benchmark trajectories, and high-precision ground-truth poses captured with a motion capture system. By combining diverse sensor technologies, extensive calibration data, and carefully designed indoor scenarios, IILABS 3D enables more comprehensive and reproducible evaluation of LiDAR-based SLAM algorithms, fostering innovation in autonomous navigation within complex indoor environments. The dataset information and associated tools are available on the project webpage: https://jorgedfr.github.io/3d_lidar_slam_benchmark_at_iilab/.

INDEX TERMS dataset, ground mobile robot, indoor environment, Light Detection And Ranging (LiDAR), Simultaneous Localization and Mapping (SLAM).

I. INTRODUCTION

DUE to the advantageous properties of Light Detection And Ranging (LiDAR) sensors, they are frequently employed in several robotic platforms such as mobile robots, autonomous vehicles, and unmanned aerial vehicles [1], [2]. LiDAR provides accurate ranging data across a wide Field of View (FoV), maintains performance under varying lighting conditions, and requires relatively low computational resources to generate 3D dense point clouds [3]. While vision and radar are also relevant technologies and may become essential for many tasks, LiDAR is expected to remain a core sensor for years to come [4]. Moreover, integrating LiDAR with complementary technologies such as Inertial Measurement Unit (IMU) sensors and wheel odometry can further enhance overall system robustness and reliability [5], [6].

Advances in LiDAR-based Simultaneous Localization and Mapping (SLAM) have driven significant progress in robotics and autonomous navigation [7]–[9]. Nevertheless, many real-world indoor applications still face challenges in achieving robust, accurate localization and mapping [10]. Existing datasets and benchmarking frameworks are often limited by their narrow focus on outdoor scenarios or by their reliance on a single LiDAR sensor, which restricts the exploration of sensor performance under diverse indoor conditions [11]. This gap is especially critical in industrial and service robotics, where the complex geometries and dynamic environments of indoor spaces demand high-precision solutions [10], [12].

In response, we introduce the IILABS 3D – iilab Indoor LiDAR-based SLAM 3D dataset. This dataset is a novel, publicly available resource designed to address the need for datasets specially focused on the benchmark of LiDAR-based SLAM algorithms in indoor environments. This dataset was captured using a mobile robot at the iilab – industry and innovation laboratory[1], and features multiple sensor sources, including four distinct 3D LiDAR sensors (Velodyne VLP-16, Ouster OS1-64, RoboSense RS-Helios-5515, and Livox Mid-360) with varying FoVs and scanning patterns, an IMU (Xsens MTi-630 AHRS), and wheel odometry. Furthermore, it comprises both calibration sequences and challenging benchmark trajectories, which makes it a good testbed for evaluating and comparing state-of-the-art SLAM algorithms in indoor environments.

In parallel, we present a detailed benchmark analysis of nine 3D LiDAR-based SLAM algorithms (LOAM [13], LeGO-LOAM [14], LIO-SAM [5], DLIO [15], VineS-LAM [16], KISS-ICP [17], GLIM [18], Kinematic-ICP [6], and MOLA-LO [4]) evaluated over six sequences and four distinct LiDAR configurations. In our analysis, we assess the accuracy of the odometry trajectories produced by the SLAM algorithms by comparing them against the ground-truth data from a Motion Capture (MoCap) system. We employ the Absolute Trajectory Error (ATE), Relative Translational Error (RTE), and Relative Rotational Error (RRE) metrics for a comprehensive evaluation. Our findings reveal that algorithms integrating IMU data consistently achieve superior accuracy on the IILABS 3D dataset[2], while those relying solely on wheel odometry tend to exhibit more significant errors. This result highlights a potential area for future research focused on improving wheel odometry integration in indoor SLAM systems.

Therefore, the principal contributions of this work are:

- **IILABS 3D Dataset:** We introduce a novel indoor LiDAR-based SLAM dataset featuring four diverse 3D LiDAR sensors, an IMU, and wheel odometry, complemented by calibration sequences and challenging benchmark trajectories, along with ground-truth data acquired using a MoCap system;
- **Comprehensive Benchmark Analysis:** We provide an evaluation of nine state-of-the-art 3D LiDAR-based SLAM algorithms over multiple sequences and sensor configurations, offering detailed insights into their accuracy under diverse indoor conditions;
- **Open-Source Resources:** We release all benchmark scripts[3], Dockerfiles, Robot Operating System (ROS) packages, and the iilabs3d-toolkit[4] for dataset download and evaluation metric computation, thereby promoting reproducibility and promoting further research in indoor LiDAR-based SLAM.

The rest of this article is organized as follows. Sec. II reviews previous works closely related to the presented contributions. Next, Sec. III introduces the proposed dataset, detailing its structure and key characteristics. A comprehensive benchmark analysis is presented in Sec. IV, evaluating nine different 3D LiDAR-based SLAM algorithms using the IILABS 3D dataset. Finally, Sec. V summarizes the findings and discusses potential directions for future research.

II. RELATED WORK

This paper presents an indoor LiDAR-based SLAM dataset. As a result, the related work discussion focuses on LiDAR-based SLAM approaches, benchmarks, and datasets. First, we review state-of-the-art 2D and 3D LiDAR-based SLAM algorithms. Next, we explore existing works and frameworks designed for benchmarking SLAM algorithms. Finally, we provide an overview of indoor SLAM datasets, highlighting their key characteristics and contributions. An exhaustive review of LiDAR-based SLAM algorithms is beyond the scope of this work. For a comprehensive analysis of the fundamentals of SLAM, the reader should refer to the works of Whyte and Bailey [8], [9] and Grisetti *et al.* [19]. For existing surveys on SLAM algorithms, readers may refer to Cadena *et al.* [11], Bresson *et al.* [1], and Xu *et al.* [12].

A. LIDAR-BASED SLAM ALGORITHMS

Typically, SLAM generates a static map, which can be used subsequently for localization. In contrast, long-term SLAM involves a dynamic update of the environment representation while accounting for changes over time [10]. Despite advancements in lifelong SLAM, many industrial applications continue to employ static maps because of their robustness and simplicity in environments where frequent updates are unnecessary [3], [11].

Early contributions for 2D LiDAR SLAM, such as GMapping [20] and HectorSLAM [21], generated static 2D occupancy grid maps. GMapping [20] introduced the Rao-Blackwellized particle filter, where each particle has its own map and robot pose. In contrast, HectorSLAM [21] relied singularly on scan-matching with scan-to-map alignment but without loop closure. Subsequent advancements in 2D SLAM shifted towards pose-graph implementations, improving scalability and facilitating loop closure detection. Cartographer [22] introduced submaps represented as probability grids within a pose graph framework, with scan matcher-based loop closure. Later, SLAM Toolbox [23] used sparse bundle adjustment to optimize scan-based pose graphs. An experimental feature implements lifelong mapping by removing extraneous or outdated information, bounding the computational resources over extended mapping periods.

While 2D SLAM remains popular in industrial applications for its simplicity and robustness, 3D LiDAR data provides richer spatial information than 2D lasers, improving dynamic object tracking, obstacle detection, and non-planar surface mapping. As a result, 3D LiDAR-based SLAM is being increasingly adopted for scenarios where 2D approaches are inadequate for dynamic and non-structured environments [3].

One of the earliest algorithms to achieve reliable performance with feature-based 3D LiDAR SLAM was

[1] https://www.inesctec.pt/en/laboratories/iilab-industry-and-innovation-lab
[2] https://doi.org/10.25747/VHNJ-WM80
[3] https://github.com/jorgedfr/3d_lidar_slam_benchmark_at_iilab
[4] https://github.com/jorgedfr/iilabs3d-toolkit

LOAM [13], which divided point clouds into two subsets – edge (corner) points and planar features –, enhancing both efficiency and accuracy in scan matching. Subsequent variations, such as LeGO-LOAM [14] and F-LOAM [24], built upon LOAM to optimize performance. LeGO-LOAM introduced ground segmentation, improving computational efficiency and reducing noise in real-time applications for ground-based mobile robots. F-LOAM enhanced robustness in dynamic environments through refined feature extraction and motion compensation. As for LOAM-Livox [25], the authors adapted the LOAM algorithm to accommodate non-uniform and non-repetitive scanning patterns (e.g., Livox sensors) that originate in less structured point clouds.

Significant advancements in 3D SLAM parametrization were introduced by KISS-ICP [17]. The algorithm automatically adjusted parameters to accommodate environment changes or motion profiles, reducing manual tuning. Kinematic-ICP [6] built on KISS-ICP for wheeled mobile robots on planar surfaces, using wheel odometry for scan de-skewing and integrating kinematic constraints into a point-to-point Iterative Closest Point (ICP) optimization scheme [26]. Additionally, MOLA-LO [4] introduced a modular system for localization and mapping, further demonstrating the versatility of adaptive parametrization in SLAM systems.

An alternative to wheeled odometry for correcting the skewed point clouds – due to the rotational mechanism of conventional LiDAR sensors and platform motion during data acquisition – is using IMU sensors. Loosely coupled sensor fusion methods, such as those employed in LOAM [13] and LeGO-LOAM [14], incorporate IMU data to provide motion priors for scan de-skewing but do not integrate the IMU measurements into the optimization process, limiting their accuracy. In contrast, recent approaches, such as LIO-SAM [5], DLIO [15] and FAST-LIO [27], implement tight coupling of LiDAR and IMU measurements into the factor graph optimization process, improving the accuracy of pose estimation and map consistency.

Overall, the SLAM field still continues to be actively researched with further contributions. Recent works by Aguiar *et al.* [16], Koide *et al.* [18], Ferrari *et al.* [28], and Pan *et al.* [29] introduced innovative techniques for SLAM. VineSLAM [16] employed a novel particle filter to integrate point and semi-plane features extracted from 3D LiDAR data, demonstrating its effectiveness in localizing robots in agricultural vineyards. GLIM [18] built upon the HDL Graph SLAM [30] framework to incorporate GPU acceleration for improving performance in large-scale environments. Furthermore, MAD-ICP [28] proposed a dynamic uncertainty-driven model update alongside efficient kd-tree data structures, enabling robust and adaptive LiDAR odometry. As for PIN-SLAM [29], the authors employed a point-based implicit neural representation to maintain global map consistency, efficiently processing large-scale 3D point cloud data.

B. BENCHMARKING FRAMEWORKS

Benchmarks have been widely used in SLAM research to ensure fair comparisons between algorithms by providing normalized testing conditions. Although benchmarks are commonly used in SLAM, some studies rely on proprietary datasets, selectively choosing sequences from well-known public datasets, such as KITTI [31], or comparing only to a limited subset of state-of-the-art algorithms. These evaluation practices may introduce bias in the results, either through fine-tuned parameters that favor the proposed algorithm or by selecting dataset sequences that highlight its strengths. Consequently, there is a growing need for benchmarks that promote more transparent and reproducible evaluations.

Tab. 1 presents open-source benchmarking frameworks. Frameworks that provide their own dataset, including TUM RGB-D [32], KITTI Vision [33], ETH3D [34], KITTI 360 [35], and VBR [36], offer multiple sequences to test algorithms under different conditions. The datasets are often divided into training and test sequences. Training sequences include accessible ground-truth data for algorithm tuning and preliminary evaluation, while testing sequences assess the algorithm with non-public ground-truth data, allowing independent and unbiased evaluation by the framework. Furthermore, benchmarking frameworks typically offer a Command Line Interface (CLI) toolkit for researchers to download dataset sequences and compute performance metrics. Two primary metrics are widely used in SLAM benchmarking: ATE in meters and Relative Pose Error (RPE) as a percentage. ATE quantifies the overall global alignment between the estimated and ground-truth trajectories, while RPE assesses local consistency between consecutive poses. Some frameworks further subdivide RPE into RTE and RRE, which measure translational and rotational discrepancies, respectively. Lastly, some frameworks maintain web-based ranking systems, allowing researchers to publicly submit results based on standardized metrics.

In contrast to only supporting a single dataset, frameworks such as SLAMBench [37]–[39] and SLAM HIVE [41], [42] support several publicly available datasets, enabling cross-domain comparisons. These frameworks are capable of evaluating various SLAM algorithms on multiple datasets, while also monitoring additional performance metrics such as CPU and RAM usage. SLAMBench supports more than 16 Visual SLAM algorithms and 9 datasets. In contrast, SLAM HIVE currently supports 8 SLAM algorithms and sequences from 3 datasets. Nevertheless, its modular design allows for the integration of additional algorithms and dataset sequences. Additionally, Liu *et al.* [42] introduced a Docker-based framework to enable graphical access to the benchmarking framework via cloud-based web pages.

Another widely used tool in SLAM benchmarking is EVO [40]. Inspired by TUM RGB-D [32], EVO provides a CLI toolkit with similar syntax and configuration, supports several data formats, and computes accuracy metrics such as ATE and RPE. The supported formats are TUM trajectory files [32], KITTI pose files [31], EuRoC MAV CSV files [43],

TABLE 1. State-of-the-art benchmarking frameworks for SLAM algorithms.

Benchmarking Framework	Supported Datasets	Sensor Data Types	Metrics	User Interface	Year
KITTI Vision [33]	KITTI dataset [31]	Monocular, Stereo, LiDAR	RTE (%), RRE (deg/m)	CLI toolkit; Public web server rankings	2012
TUM RGB-D [32]	TUM RGB-D dataset	RGB-D, IMU	ATE (m), RPE (%)	CLI toolkit	2012
SLAMBench [37]–[39]	Various public Visual SLAM datasets	Monocular, Stereo, RGB-D, IMU	ATE (m), Execution time (s), RAM usage (MB)	CLI framework	2015
ETH3D [34]	ETH3D dataset	Monocular, Stereo, RGB-D, IMU	ATE (m), RTE (%), RRE (deg/m)	CLI toolkit; Public web server rankings	2017
EVO [40]	–(1)	–(1)	ATE (m), RTE (%), RRE (deg/m)	CLI toolkit	2017
KITTI 360 [35]	KITTI 360 dataset	Monocular, Stereo, LiDAR	ATE (m), RPE (%)	CLI toolkit; Public web server rankings	2023
SLAM Hive [41], [42]	Various public SLAM datasets	Monocular, Stereo, RGB-D, LiDAR, IMU	ATE (m), RTE (%), CPU usage (%), RAM usage (MB)	Web-based graphical interface; Docker-based system	2023
VBR [36]	VBR dataset	Monocular, Stereo, LiDAR, IMU	ATE (m), RPE (%)	CLI toolkit; Public web server rankings	2024

(1) EVO [40] is a toolkit to evaluate the trajectory output by the SLAM algorithms. The supported trajectory formats are: TUM trajectory files [32], KITTI pose files [31], EuRoC MAV CSV files [43], and ROS 1 and ROS 2 bag files, including TF messages, pose, transform, and odometry topics.

Abbreviations: Absolute Trajectory Error (ATE), Command Line Interface (CLI), Inertial Measurement Unit (IMU), Light Detection And Ranging (LiDAR), Robot Operating System (ROS), Relative Pose Error (RPE), Relative Rotational Error (RRE), Relative Translational Error (RTE)

and ROS 1 and ROS 2 bag files, including pose, transform, and odometry topics. Furthermore, the toolkit generalizes trajectory plots, performing alignment, and converting between different trajectory representations. Subsequently, EVO's comprehensive functionality has led to widespread adoption. DLIO [15] used EVO to calculate the metrics in their results. Similarly, Hilti 2022 [44] dataset utilized EVO to compute the metrics in their SLAM challenge rankings. KITTI 360 [35] employed EVO to calculate metrics and generate the plots presented on its webpage. Also, MOLA [4] used EVO to produce quantitative metrics and visualizations when benchmarking its LiDAR Odometry (LO) pipeline on several public datasets.

C. DATASETS

While datasets have been developed for a wide range of SLAM applications, including autonomous driving [31], [45], long-term SLAM [10], [46], [47], semantic SLAM [35], and collaborative SLAM [48], there are still few datasets focusing on indoor sequences. Moreover, in datasets that include both indoor and outdoor sequences, the indoor portion is typically underrepresented — often accounting for less than 20% of the total data [36], [46], [48]–[52]. Narrowing the scope further to ground robots with 3D LiDAR sensors, the availability of suitable datasets becomes even more limited [46], [48]–[50], [52], [53], revealing a gap in resources to evaluate indoor 3D LiDAR-based SLAM. Tab. 2 provides an overview of state-of-the-art SLAM datasets that include indoor sequences and 3D LiDAR data.

A common characteristic among 3D LiDAR datasets with indoor sequences is the reliance on mechanical spinning LiDAR sensors. Most employ Velodyne or Ouster sensors with 16 to 128 beams. From Tab. 2, only six datasets incorporate more than one 3D LiDAR, and among those, only four datasets (Hilti 2021 [54], TIERS [51], A Multi-LiDAR Multi-UAV Dataset [56], and GEODE [52]) employ non-repetitive scanning sensors. However, considering different 3D LiDARs with various FoVs and scanning pattern configurations may provide valuable insights into which configurations best suit indoor environments. Our IILABS 3D dataset directly addresses this need by incorporating 4 heterogeneous LiDAR sensors enabling systematic comparisons across sensor configurations.

Regarding sensor fusion with 3D LiDAR, nearly all datasets include an external IMU. The only exceptions are TIERS [51] and A Multi-LiDAR Multi-UAV Dataset [56], where the internal IMU of the 3D LiDAR is used instead. In the case of ground-based platforms, only NCLT [46], FusionPortable [49], [55], and Ground-Challenge [53] provide wheel odometry data, which provides a cost-effective source of motion information in industrial and indoor applications from measuring the wheels' angular speed. In contrast, IILABS 3D provides both wheel odometry and an external IMU, being the only dataset that combines wheel odometry with heterogeneous 3D LiDARs beyond the common single-sensor setups based on Velodyne or Ouster. This setup enables the benchmarking of SLAM algorithms that fuse wheel odometry with diverse LiDAR configurations, extending evaluation possibilities beyond existing datasets.

Furthermore, ground-truth data varies among the datasets.

TABLE 2. State-of-the-art SLAM datasets with 3D LiDAR sensors and indoor sequences.

Dataset	Platforms	Sensors				Ground-Truth		Year
		Spinning LiDARs	Non-repetitive scanning LiDARs	IMU	Wheel Odometry	Trajectory	Map	
NCLT [46]	UGV	Velodyne HDL-32E	–	Yes	Yes	GNSS, SLAM	No	2015
Hilti 2021 [54]	Handheld, UAV	Ouster OS0-64	Livox Mid-70	Yes	–	MoCap	No	2021
FusionPortable [49], [55]	Handheld, Legged robot, UGV	Ouster OS1-128	–	Yes	Yes	Laser scan, GNSS	Yes	2022
Hilti 2022 [44]	Handheld	Hesai PandarXT-32	–	Yes	–	Laser scan	Yes	2022
M2DGR [50]	UGV	Velodyne VLP-32C	–	Yes	No	Laser scan, MoCap, GNSS	No	2022
NTU Viral [2]	UAV	2× Ouster OS1-16	–	Yes	–	Laser scan	No	2022
TIERS [51]	Mobile platform	Velodyne VLP-16, Ouster OS1-64, Ouster OS0-128	Livox Horizon, Livox Avia	No	No	MoCap, SLAM	No	2022
A Multi-LiDAR Multi-UAV Dataset [56]	UAV	Ouster OS1-64	Livox Avia, Livox Mid-360	No	–	MoCap	No	2023
Ground-Challenge [53]	UGV	Velodyne VLP-16	–	Yes	Yes	–	No	2023
GEODE [52]	Handheld, UGV, Car, Boat	Velodyne VLP-16, Ouster OS1-64	Livox Avia	Yes	No	Laser scan, MoCap, GNSS	Yes	2024
S3E [48]	UGVs	Velodyne VLP-16	–	Yes	No	MoCap, GNSS	No	2024
VBR [36]	Handheld	Ouster OS1-64, Ouster OS0-128	–	Yes	–	GNSS	No	2024
IILABS 3D (Ours)	UGV	Velodyne VLP-16, Ouster OS1-64, RoboSense RS-Helios-5515	Livox Mid-360	Yes	Yes	MoCap	No	2025

Abbreviations: Global Navigation Satellite System (GNSS), Inertial Measurement Unit (IMU), Motion Capture (MoCap), Unmanned Aerial Vehicle (UAV), Unmanned Ground Vehicle (UGV)

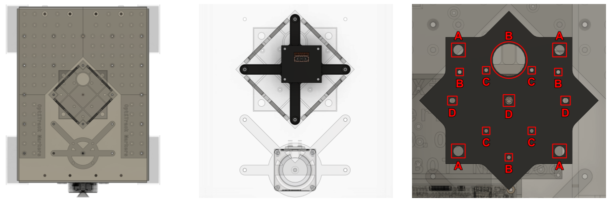
Several rely on Global Navigation Satellite System (GNSS), though this method is most effective in outdoor scenarios or indoors with ample window visibility. Many datasets employ laser scanning techniques to obtain ground-truth data, using devices such as Leica MS60 or adopting MoCap systems like OptiTrack or Vincon Vero cameras. Although laser scanners can be more easily moved and installed to cover larger areas, they provide ground-truth poses with lower accuracy and frequency compared to MoCap systems. As such, our IILABS 3D dataset adopts a MoCap system to generate its trajectory ground truth. Additionally, only three datasets (FusionPortable [49], [55], Hilti 2022 [44], and GEODE [52]) also provide ground-truth maps, typically generated from high-precision lasers (e.g., Leica RTC360).

In summary, existing datasets reveal significant gaps for indoor-oriented SLAM research. Many do not offer indoor-exclusive sequences, and most rely on a limited selection of mechanical spinning LiDARs, primarily Velodyne and Ouster models, with symmetric beam configurations and uniform FoV. This restricts the evaluation of algorithms across diverse sensor configurations. Moreover, non-repetitive scanning LiDARs remain scarcely represented despite their potential to generate denser point clouds [57], and wheel odometry data is rarely available despite its practical relevance [58]. Our IILABS 3D dataset is designed to address these shortcomings by combining heterogeneous LiDARs, an external IMU, wheel odometry, and a high-precision MoCap ground truth system.

III. IILABS 3D DATASET

In order to address the limitations of existing datasets and provide a more comprehensive resource for evaluating indoor LiDAR-based SLAM on ground-wheeled robots, we developed a new dataset named IILABS 3D – iilab Indoor LiDAR-based SLAM 3D [59]. This dataset is publicly available in the INESC TEC – Institute for Systems and Computer Engineering, Technology and Science research data repository[2]. Unlike existing datasets, which often lack components such as diverse 3D LiDAR configurations or wheel odometry data,

FIGURE 1. Revised Hangfa Discovery Q2 mobile platform with 3D sensor perception



(a) (b) (c)

FIGURE 2. 3D model views on AutoDesk Fusion 360 for the integration of the sensors into the Hangfa Discovery Q2 platform: (a) top view of the bend metal sheet; (b) support on the bottom of the bend metal sheet for the IMU; (c) support on the top of the bend metal sheet for 3D LiDARs (A: Ouster OS1-64; B: RoboSense RS-Helios-5515; C: Livox Mid-360; D: Velodyne VLP-16).

IILABS 3D is specifically designed to fill these gaps and support the benchmarking of LiDAR-based SLAM algorithms in indoor environments.

IILABS 3D features four different 3D LiDARs (Velodyne VLP-16, Ouster OS1-64, RoboSense RS-Helios-5515, and Livox Mid-360) with varying characteristics and design applications, offering a diverse range of sensor configurations for testing. The dataset also includes pre-processed wheel odometry data and IMU data to enhance sensor fusion capabilities. Additionally, the dataset contains calibration sequences to ensure transparency and reproducibility in benchmarking studies and challenge sequences to evaluate the robustness of algorithms under various conditions.

The following subsections provide detailed descriptions of the dataset's key components: the mobile robot and sensors used (Sec. III-A), the dataset sequences designed to present a range of challenges and benchmarking scenarios (Sec. III-B), and the ground-truth system employed (Sec. III-C).

A. MOBILE ROBOT AND SENSORS

The dataset was collected using a mobile robot platform equipped with 3D perception sensors, as shown in Fig. 1. This mobile robot, which builds on the Hangfa Discovery Q2 platform, was developed at CRIIS – Centre for Robotics in Industry and Intelligent Systems from INESC TEC. Detailed information about the robot is available in the work by Sousa *et al.* [60] and on its public webpage[5].

The Hangfa Discovery Q2 is a compact four-wheeled omnidirectional robot developed by Hangfa Robotics[6]. This robot has a coaxial pendulum suspension system on its rear wheels, ensuring ground contact of the four wheels while reducing vibrations on uneven surfaces. Also, the platform features QMA10 mecanum wheels with a 101.6 mm diameter, a load capacity of 30 kg, a 0.65 m/s maximum translational speed, and a 140°/s maximum rotation speed. However, the original platform lacks access to wheel odometry data and support for advanced sensing devices. As a result, the work of Sousa *et al.* [60] addressed those issues, while integrating multimodal perception into the platform.

Regarding sensor integration, the dataset acquisition considered three types of sensors equipped on the robot: four 3D LiDARs (Velodyne VLP-16, Ouster OS1-64, RoboSense RS-Helios-5515, and Livox Mid-360), a 2D laser scanner (Hokuyo UST-10LX), and an IMU (Xsens MTi-630 AHRS). These sensors are fixated on the platform using the 3D-printed supports illustrated in Fig. 2 and designed in AutoDesk Fusion 360. While the 3D LiDAR and IMU sensors are centered in the robot's geometric center, the 2D laser scanner is positioned forward without interfering with the data from any of the 3D LiDARs sensors. Notably, the generic support for 3D LiDAR from Sousa *et al.* [60] ensures the optical point alignment with the robot's geometric center and minimal occlusion by the chassis while accommodating a range of sensors – including Livox Mid-360, Ouster OS, RoboSense Helios, and Velodyne Puck series.

Furthermore, the four 3D LiDARs selected for the dataset offer various technologies and configurations, including mechanical spinning (Velodyne, Ouster, RoboSense) and non-repetitive scanning (Livox) data, a diverse FoV angles (see Fig. 3), and different beam configurations (16, 32, and 64 beams). This selection enriches the dataset's applicability across different SLAM scenarios (ground versus ceiling-based perception, symmetrical versus asymmetrical FoV of the sensor). Therefore, each 3D LiDAR used in the study contributes uniquely to the dataset, as follows:

- **Velodyne VLP-16**: a widely used sensor in recent datasets due to its robustness and compatibility with SLAM algorithms. It features 16 beams and a symmetrical vertical FoV from -15° to +15°, serving as a reliable baseline for benchmarking;
- **Ouster OS1-64**: common in autonomous driving and indoor datasets, this sensor provides high-resolution data

[5]https://sousarbarb.github.io/inesctec_mrdt_hangfa_discovery_q2
[6]https://www.hangfa-europe.com

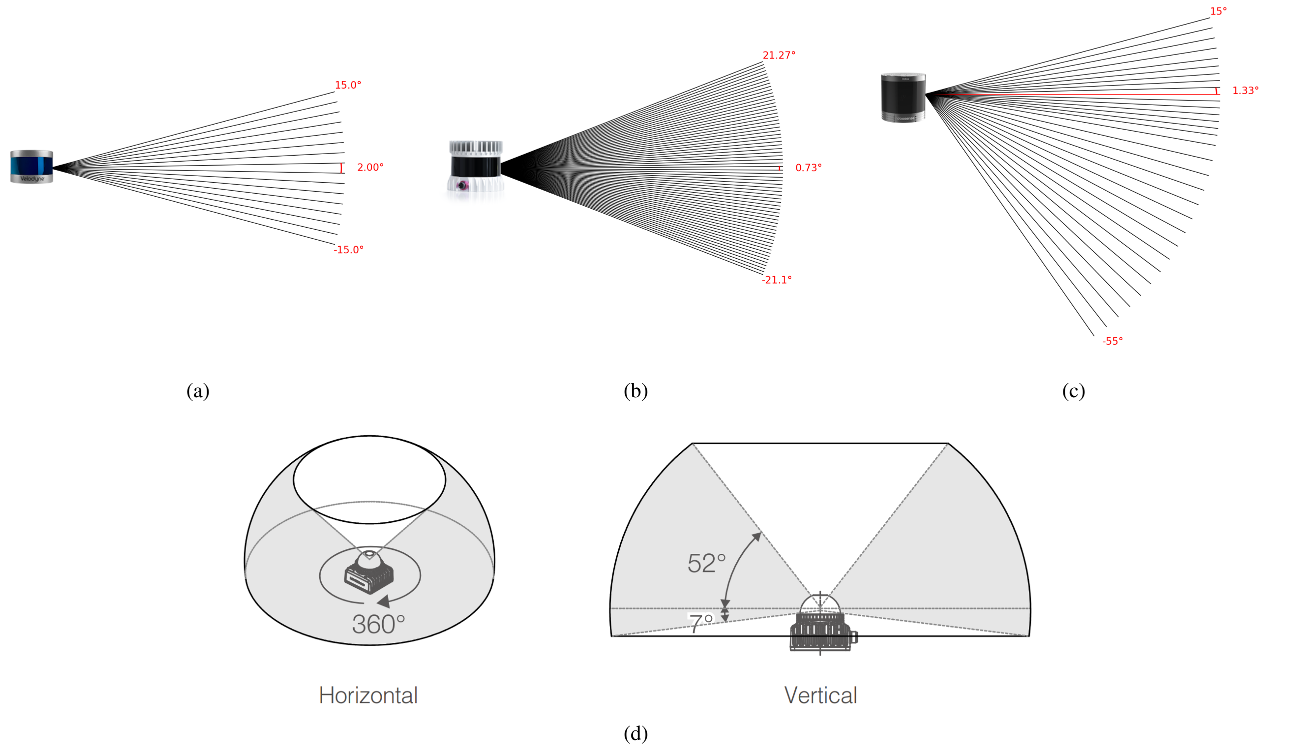


FIGURE 3. Field of View (FoV) of the different 3D LiDARs: (a) Velodyne VLP-16; (b) Ouster OS1-64; (c) RoboSense RS-Helios-5515; (d) Livox Mid-360.

with 64 beams and a vertical FoV from -21.1° to +21.27°. It supports configurable horizontal resolutions (512, 1024, or 2048) and operational frequencies (10 Hz or 20 Hz). An integrated IMU enhances SLAM compatibility and reduces reliance on external sensors;

- **RoboSense RS-Helios-5515**: a cost-effective sensor with a vertical FoV from -55° to +15°. Its design concentrates laser beams densely in the center while sparsely covering the edges, improving near-field accuracy and blind-spot detection. This configuration supports short-range and long-range perception, making it ideal for mobile robots in dynamic environments;
- **Livox Mid-360**: a non-repetitive scanning LiDAR with 360° horizontal coverage and a vertical FoV from -7° to +52°. With a range of approximately 40 meters, it is well-suited for indoor applications like warehouses, where it captures semi-static features such as ceilings and shelves, aiding SLAM algorithms in maintaining consistent localization.

In addition to 3D LiDARs, the robot is equipped with a Hokuyo UST-10LX 2D laser and an Xsens MTi-630 AHRS IMU. The 2D laser enables real-time 2D localization and parametric-based trajectory navigation [61] during the dataset acquisition. This approach ensures consistent trajectory reproduction regardless of which 3D LiDAR is present on the robot. Regarding the Xsens IMU, the dataset provides data from this sensor at a frequency of 400 Hz, enabling support for sensor fusion with inertial odometric data in sequences in which the LiDAR sensor does not have an internal IMU. In comparison, the Ouster OS1-64 and Livox Mid-360 sensors operate their internal IMU at 100 Hz and 200 Hz, respectively.

Notably, when using the UART-to-USB board included in the development kit to communicate with the external Xsens IMU, the communication driver based on FTDI chipsets experiences buffering when operating at frequencies higher than 100 Hz. Since IMU data messages are timestamped upon arrival in the ROS driver, the header timestamps are affected by this buffering, as illustrated in Fig. 4a. In order to address this issue, post-processing is applied using the internal clock timestamps available in the *time_ref* topic (see Fig. 4b) and an offset correction, considering the last elements of each buffer segment. The effect of this correction is illustrated in Fig. 4c.

Finally, the dataset also includes wheel odometry data in two forms: raw encoder readings and pre-processed odometry. The raw data consists of encoder counts, offering low-level measurements for researchers interested in custom odometry processing. Additionally, pre-processed odometry data is provided using a ROS package developed by the 5dpo Robotics Team[7] and published as open-source by Sousa *et al.* [62]. This package, part of a broader robotic framework designed for the Robot@Factory 4.0 competition,

[7]https://5dpo.github.io

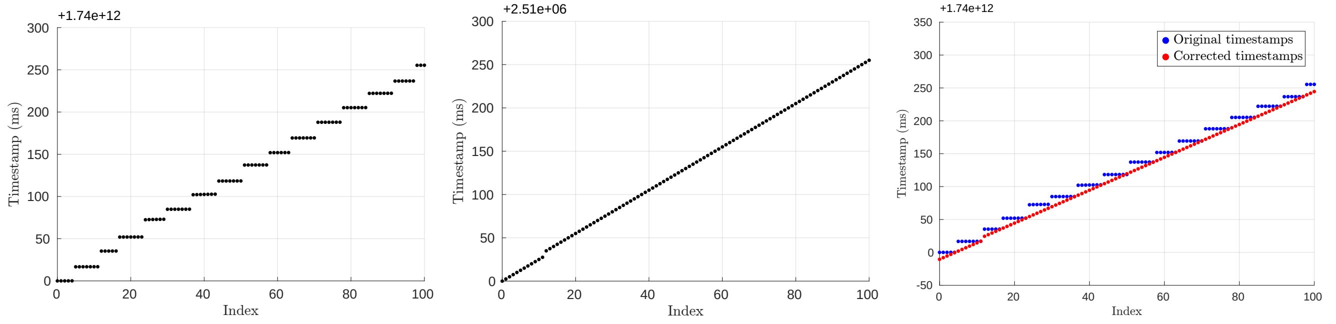


FIGURE 4. Timestamp buffering correction for the Xsens MTi-630 AHRS IMU: (a) original header timestamps; (b) timestamps from the *time_ref* topic; (c) corrected timestamps.

is available on GitHub[8].

B. DATASET SEQUENCES

The dataset is designed to include two distinct types of sequences: calibration and benchmark sequences. Calibration sequences are intended to provide the necessary data for replicating the results presented in this study and to support alternative calibration methodologies. These sequences ensure that intrinsic and extrinsic sensor calibration can be performed by users of the dataset. On the other hand, benchmark sequences are tailored to evaluate the performance of SLAM algorithms under varying conditions.

1) Calibration sequences

First, the extrinsic calibration sequences are designed to determine the transformation between the IMU and 3D LiDAR reference frames. While this transformation can be derived from the distances specified in the Computer Aided Design (CAD) 3D models of the platform [60], these sequences allow an alternative calibration method when the CAD data lacks sufficient precision. Two distinct extrinsic calibration sequences were performed. The first sequence involves lifting the robot in the air, followed by rotations around the z-axis and oscillations along the x-axis and y-axis. This setup ensures excitation in all three axes, making it compatible with various IMU–LiDAR calibration algorithms, such as LI-Init [63]. The second sequence follows a ground-based figure-eight trajectory, also known as the lemniscate of Bernoulli, as illustrated in Fig. 5. This trajectory is particularly suitable for extrinsic calibration algorithms intended for ground robots, such as GRIL-Calib [64]. The lemniscate of Bernoulli trajectory is mathematically defined as follows, where $PF_1 \cdot PF_2 = c^2$ and $a = c\sqrt{2}$:

$$\begin{cases} x = \dfrac{a \cdot \cos(t)}{1 + \sin^2(t)} \\ y = \dfrac{a \cdot \sin(t) \cdot \cos(t)}{1 + \sin^2(t)} \end{cases} \tag{1}$$

[8]https://github.com/5dpo/5dpo_ratf_2023

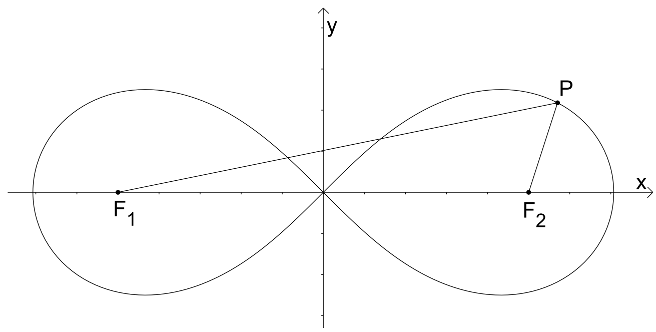


FIGURE 5. Extrinsic calibration using a ground-based figure-eight trajectory (lemniscate of Bernoulli).

Next, the intrinsic calibration sequence for the IMU sensors involves a three-hour stationary recording, during which the robot remains motionless. This procedure is intended to estimate the white noise and bias characteristics of IMU sensors. The dataset includes this calibration sequence for all three IMUs: the Xsens MTi-630 AHRS IMU and the internal IMUs of the Ouster OS1-64 and Livox Mid-360 3D LiDAR sensors.

As a result, the intrinsic calibration sequences can be used to generate Allan deviation plots to estimate the white noise and bias characteristics of both the gyroscope and accelerometer in the IMU. Preprocessed values for these parameters are provided in the dataset as YAML files, computed using the open-source package Allan Variance ROS [65]. For illustration, the calibration results of the Xsens MTi-630 AHRS IMU are presented in Fig. 6, while the same methodology applies to the internal IMUs of the other sensors. The Xsens IMU calibration yielded an accelerometer noise density of approximately $0.002\ \mathrm{m/s^2/\sqrt{Hz}}$ and a random walk of around $1.8 \times 10^{-5}\ \mathrm{m/s^3/\sqrt{Hz}}$. For the gyroscope, the noise density was approximately $1.4 \times 10^{-4}\ \mathrm{rad/s/\sqrt{Hz}}$, with a random walk of about $7.8 \times 10^{-6}\ \mathrm{rad/s^2/\sqrt{Hz}}$. To ensure a safe margin in the benchmark study, the YAML file values were amplified – bias values by a factor of 10 and white noise values by a factor of 5. The results, illustrated in Fig. 6, exemplify the noise characteristics and bias stability of the a IMU sensor.

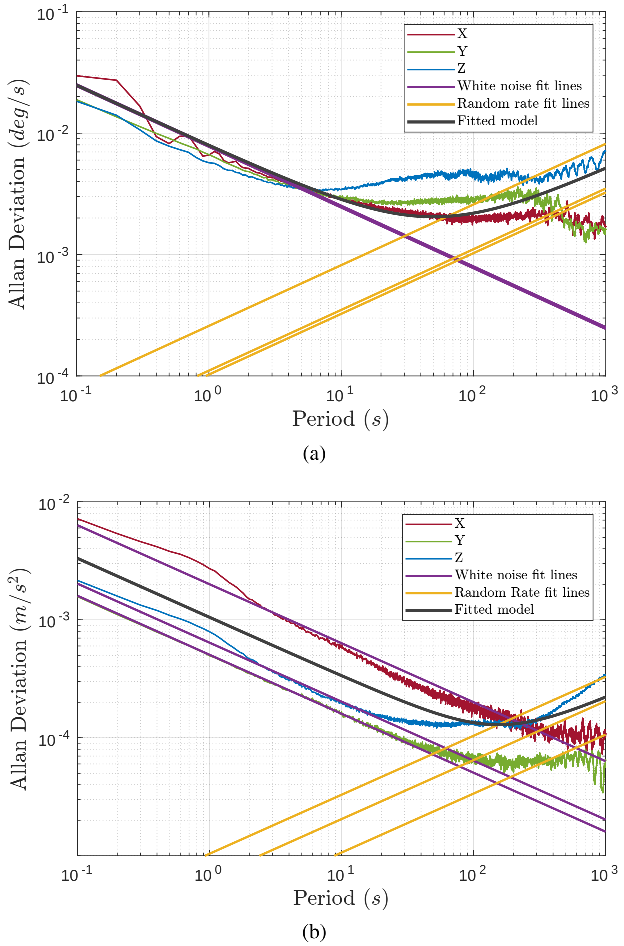


FIGURE 6. **Allan Deviation plots of the Xsens MTi-630 AHRS IMU generated by package Allan Variance ROS [65]: (a) gyroscope; (b) accelerometer.**

The wheel odometry calibration sequences consist of a set of square-shaped trajectories, a widely used approach for odometry calibration in mobile robotics [66]. Four distinct trajectories were executed, including both clockwise (CW) and counterclockwise (CCW) versions (see Fig. 7), with and without rotational maneuvers at each corner. As a result, the dataset includes sixteen odometry calibration sequences – four for each 3D LiDAR –, where the robot completes two full laps of each 2 m $\times$ 2 m square-shape trajectory. Although odometry calibration parameters are independent of other sensors, including the 3D LiDARs, providing a larger number of sequences enhances the precision and reliability of the calibration process. Furthermore, these sequences can also serve as alternative data for extrinsic calibration algorithms, complementing or substituting the figure-eight trajectory sequence when needed.

2) Benchmark sequences

On the other hand, the benchmark sequences were designed to evaluate the performance of SLAM algorithms under diverse conditions, introducing challenges that range from holonomic

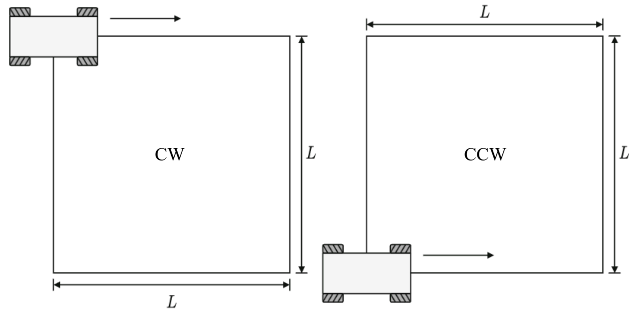
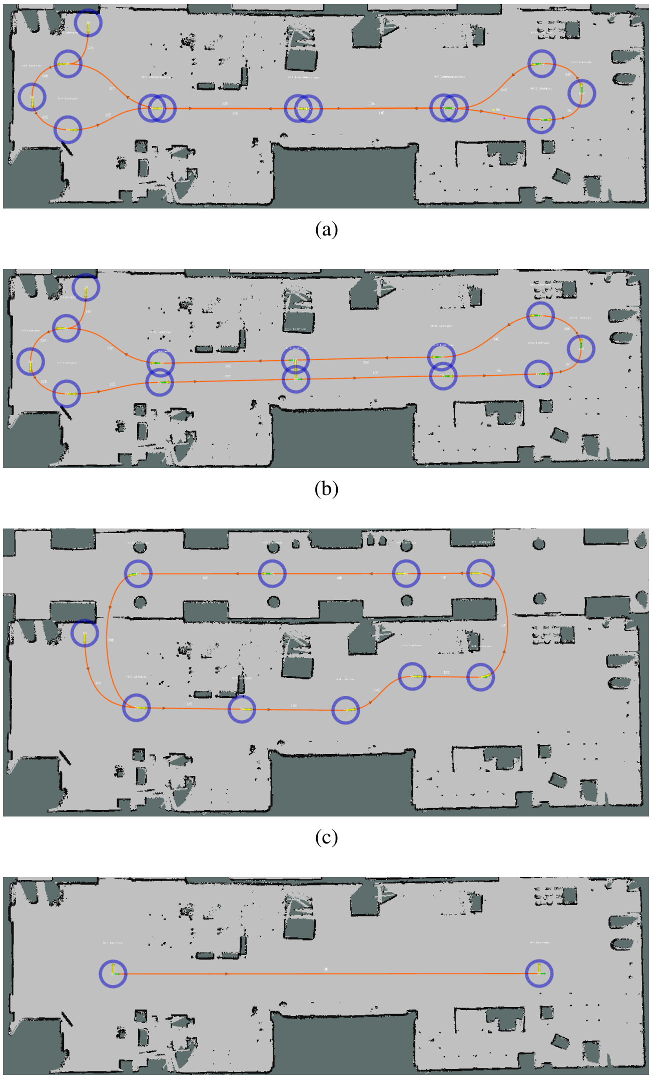


FIGURE 7. **Square-shaped trajectories used for odometry calibration, performed in both CW and CCW directions.**

motions to non-planar trajectories. All sequences were collected at iilab, with the majority conducted within the Nav A space, where the ground-truth system is installed. The Nav A space is characterized by its feature-rich environment, a fully planar floor, and the presence of large glass doors and windows, which can introduce challenges for LiDAR-based SLAM due to reflections and refractions.

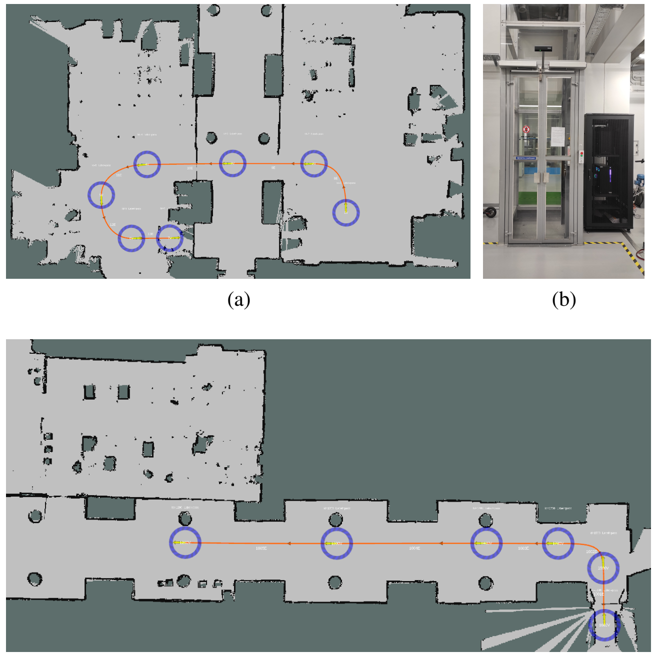
The trajectories for each sequence are depicted in Fig. 8 and Fig. 9, while illustrative photographs of the elevator and ramp are provided in Fig. 9b and Fig. 10. Furthermore, an explanation of each sequence is presented below:

- **Nav A Diff**: A 275-meter, 755-second sequence completing 5 cycles within the Nav A space. It serves as a baseline, capturing typical environmental conditions without additional challenges;
- **Nav A Omni**: A 112-meter, 387-second sequence with 2 cycles, featuring holonomic motion. The robot executes decoupled translational and rotational movements using omnidirectional steering;
- **Loop**: A 232-meter, 620-second sequence with 5 cycles. The robot exits and re-enters the Nav A space via different doors, briefly traversing an external corridor. This sequence was designed to assess the loop closure capabilities of the SLAM algorithms;
- **Slippage**: A 39-meter, 93-second sequence with only one cycle. The robot follows a straight-line path while performing lateral motions at high speeds that intentionally induce wheel slippage. This setup challenges SLAM algorithms that rely on wheel odometry by introducing significant drift;
- **Ramp**: A 29-meter, 168-second sequence with 3 cycles, manually controlled via teleoperation. The primary challenge of this sequence is the non-planar motion while ascending and descending a ramp, requiring SLAM to capture vertical displacements and pitch rotations;
- **Elevator**: An 85-meter, 381-second sequence with only one cycle. The robot moves between floors via an elevator with glass windows. This sequence tests whether vertical motion can be reliably detected through IMU sensor fusion or directly via 3D LiDAR point cloud analysis.



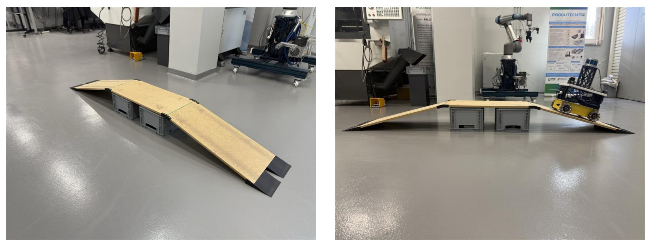
(a) (b) (c) (d)

FIGURE 8. Benchmark trajectories analyzed in this study: (a) Nav A Diff; (b) Nav A Omni; (c) Loop; (d) Slippage.



(a) (b) (c)

FIGURE 9. Elevator benchmark sequence: (a) trajectory on the first floor; (b) photograph of the elevator; (c) trajectory on the second floor.



(a) (b)

FIGURE 10. Ramp benchmark sequence: (a) ramp used in the experiment; (b) mobile robot navigating the ramp during the sequence.

In addition to the natural characteristics of the environment, specific challenges were deliberately introduced in some sequences to test SLAM robustness. One challenge consist in holonomic trajectories, were translation and rotation are decoupled by using the robot's omnidirectional steering. Another challenge is induced slippage, which causes significant wheel odometry drift. In addition, non-planar trajectories involve a ramp and an elevator, and high-speed motions introduce vibrations into the mobile robot. Additionally, some benchmark sequences extend beyond the Nav A area, briefly exploring locations such as the first and second floor corridors or the Nav B space. Although ground-truth data cannot be directly provided in these regions, all sequences begin and end within Nav A, ensuring that accurate ground-truth information is available at those timestamps.

Moreover, each benchmark sequence is repeated for all four 3D LiDAR sensors. In order to ensure consistency across all sequences, the 2D LiDAR and the robot's 2D localization and navigation stack [61] were employed in every sequence, except for the Ramp sequence, where teleoperation was used instead to capture non-planar motion dynamics.

Finally, in order to validate the induced odometry drift observed in the *slippage* sequence, we conducted a series of tests that evaluated both longitudinal and lateral motions at different linear velocities along the line-shaped trajectory. The Relative Translational Error (RTE) was computed on the wheel odometry data in 1-meter segments using the EVO tool [40], as illustrated in Fig. 11. The results confirm that odometry drift is more pronounced during longitudinal motions, likely due to the inherent limitations of the omnidirectional mobile robot's odometry model along that axis. Furthermore, when the linear velocity is increased, the mean RTE for longitudinal motions rises from 0.021 m to 0.035 m (a 66% increase), whereas for lateral motions, it increases

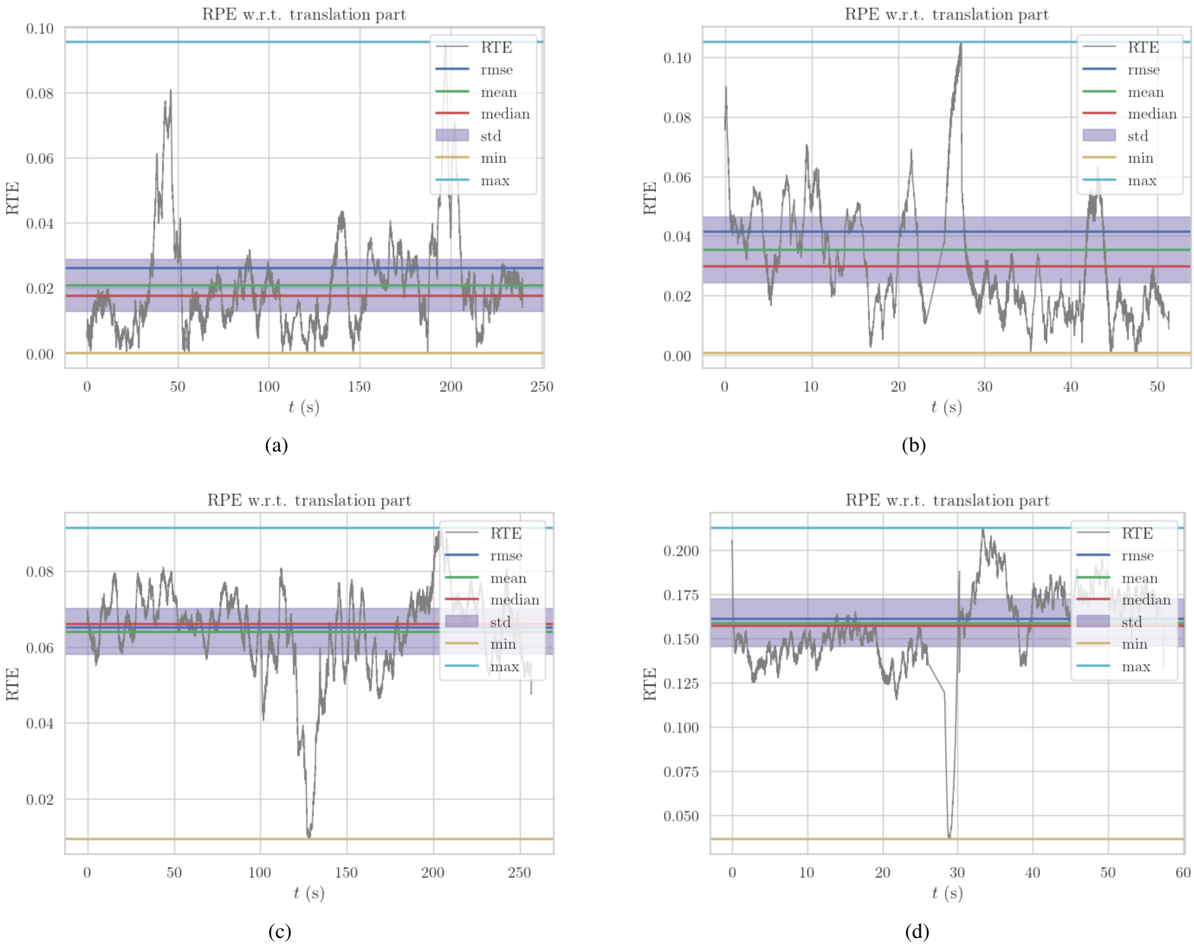


FIGURE 11. Odometry drift validation the Slippage trajectory: (a) with longitudinal motion at 0.1 m/s; (b) with longitudinal motion at 0.5 m/s; (c) with lateral motion at 0.1 m/s; (d) with lateral motion at 0.5 m/s.

from 0.064 m to 0.159 m (a 148% increase). These findings indicate that lateral motions are particularly susceptible to slippage-induced drift, thereby corroborating the use of this motion style in the *slippage* sequence.

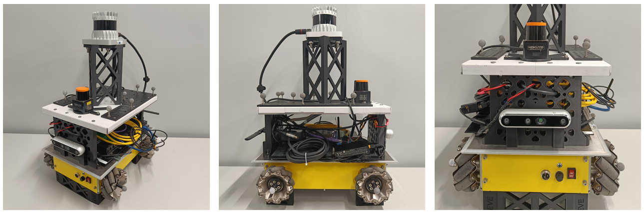
C. GROUND-TRUTH SYSTEM

We employed the OptiTrack MoCap system[9] to provide the ground-truth trajectory data for the dataset. MoCap systems like OptiTrack are widely used in robotics for providing high-precision ground-truth data, particularly in indoor environments where Global Navigation Satellite System (GNSS) is unavailable [48], [50]. OptiTrack leverages a network of cameras to track reflective markers with sub-millimeter accuracy, offering a reliable reference for evaluating robot localization and mapping algorithms. This study utilized the OptiTrack system installed in the iilab at INESC TEC, as shown in Fig. 12. The system operates at a frequency of 240 Hz and comprises 24 high-resolution cameras – model Prime^x 22 – strategically positioned throughout the lab to ensure full coverage of the robot's operating area.

[9]https://www.optitrack.com

FIGURE 12. OptiTrack system installed in the iilab at INESC TEC.

The ground-truth setup on the robot involves the integration of reflective spheres mounted in specific configurations to facilitate precise tracking and alignment. As illustrated in Fig. 13, four reflective spheres are attached to the robot's wheels to define the base link, aligning the MoCap rigid body with the robot's reference frame. Additionally, nine reflective spheres are placed on the robot's top in a non-symmetric structure. This asymmetrical arrangement ensures that the rigid body is uniquely identifiable by the OptiTrack system,



(a) (b) (c)

FIGURE 13. **Reflective sphere setup on the robot: (a) perspective view; (b) side view; (c) front view.**

even under occlusion or partial visibility. The setup process is configured using the Motive software[10], specifically designed to work with the OptiTrack system. The steps for setting up the ground-truth measurement are as follows:

1) Define the rigid body in the Motive software, aligning it with the robot's reference frame by incorporating the four reflective spheres mounted on the wheels;
2) Add the additional spheres on the top of the robot to the rigid body as constraints, enhancing its stability and accuracy during tracking;
3) Detach the wheel-mounted spheres from the rigid body definition. This ensures that their motion may occur during robot operation and does not introduce errors in the rigid body's pose estimation.

In order to record the ground-truth data, the NatNet 4 ROS driver[11] was employed to capture and store the OptiTrack data in rosbags. The OptiTrack system captures the positions of reflective spheres using cameras, which transmit raw positional data via Ethernet to a main computer running the Motive software in a Windows environment. The Motive software processes this data to compute the poses of rigid bodies and individual markers, transmitting the information via User Datagram Protocol (UDP) to a secondary system running the NatNet 4 ROS driver on Ubuntu.

Additionally, to determine whether a secondary computer was necessary for handling ground-truth data, a comparative study was conducted on the performance of Wi-Fi versus Ethernet connections for transmitting OptiTrack data. While the NatNet 4 ROS driver could theoretically run directly on the robot using a Wi-Fi connection, the robot's inability to connect to the main computer via Ethernet during dataset recording posed a challenge. This limitation made it essential to evaluate whether the use of a secondary computer connected via Ethernet could provide a significant advantage. A fast and stable connection was critical since the timestamps of the ROS messages are based on the ROS time when the NatNet 4 ROS driver processes the message, rather than the time when the Motive software generates it. Tests revealed that an Ethernet connection offered significantly reduced jitter compared to Wi-Fi, ensuring higher precision and reliability in data transmission. As illustrated in Fig. 14, the jitter values

[10]https://optitrack.com/software/motive
[11]https://github.com/L2S-lab/natnet_ros_cpp

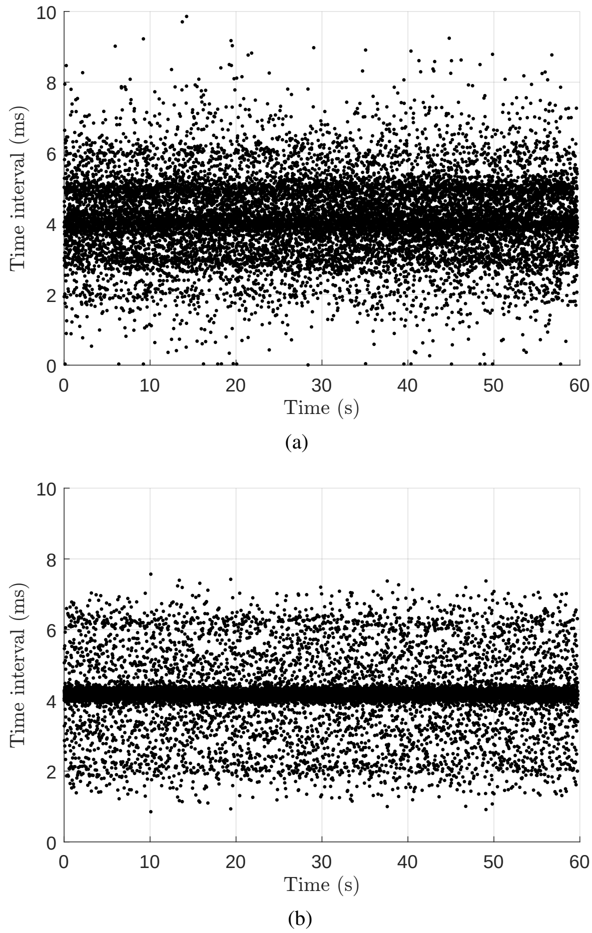


FIGURE 14. **Jitter comparison for OptiTrack data transmission: (a) Wi-Fi connection; (b) Ethernet connection.**

were 1.1607 ms for the Wi-Fi setup and 0.8472 ms for the Ethernet configuration. This notable reduction highlights the importance of using a secondary computer with an Ethernet connection to ensure accurate ground-truth data recording.

Given the use of a secondary computer for handling ground-truth data, clock synchronization between this computer and the robot was essential for accurate benchmarking. Initial experiments utilized the Network Time Protocol (NTP) for direct time synchronization, with the secondary computer serving as the NTP server and the robot as the client. However, this configuration resulted in offsets and jitters of approximately 100ms. The issue stemmed from variable delays in the Wi-Fi connection, where the time taken to transmit messages could fluctuate significantly. Such high latency was unacceptable for this study, as the benchmark requires precise synchronization between the odometry provided by the SLAM algorithms and ground-truth data. Since the dataset includes LiDAR data at 10 Hz, even small timing discrepancies can lead to significant errors in evaluating the SLAM algorithm's performance. In order to address this issue, an alternative synchronization strategy was implemented, utilizing a public NTP server in both devices, specifically the

Lisbon Astronomical Observatory (`ntp02.oal.ul.pt`). This configuration achieved significantly smaller offsets and jitter, approximately 1 ms for the Ethernet-connected secondary computer and 10 ms for the robot connected via Wi-Fi.

Consequently, the offset and jitter between the robot and the secondary computer were reduced to around 10 ms, ensuring adequate synchronization for the dataset. The unexpected improvement with the external server is likely related to external conditions, such as fluctuating Wi-Fi transmission delays, affecting the local NTP setup. So, we recommend that readers, replicating this benchmark setup, conduct similar synchronization tests in their own environments to determine the most effective configuration for their specific conditions.

Finally, the accuracy of our ground-truth system was assessed through an OptiTrack calibration procedure. The calibration yielded a mean three-dimensional residual error of 0.646 mm for a single reflective sphere, as reported in the calibration file. It is important to note that the OptiTrack system specifies this error for individual markers rather than for the localization of a complete rigid body. Since the rigid body of the mobile robot is defined by nine spheres of 15.9 mm diameter (the same size as those in the calibration tool), the pose estimation benefits from marker redundancy and geometric constraints. Therefore, the expected trajectory error of the rigid body is bounded above by the single-marker error, and in practice is considerably smaller [67]. This sub-millimeter precision confirms the reliability of our ground-truth trajectories, establishing the dataset as a robust benchmark for evaluating SLAM algorithms.

IV. BENCHMARK

This section presents a comprehensive benchmark study of nine state-of-the-art 3D LiDAR-based SLAM algorithms. The evaluation uses the dataset described in Sec. III. By benchmarking multiple SLAM approaches on this dataset, we provide a detailed comparison that highlights their performance in terms of accuracy and robustness.

The section is structured as follows. First, in Sec. IV-A, the methodology is introduced, describing the evaluation metrics, the selected algorithms, and the overall benchmarking process. Next, in Sec. IV-B, the results are presented in a structured manner, summarizing the performance of each approach. Finally, in Sec. IV-C, the results are analyzed and discussed, considering both the differences between the algorithms and the impact of the 3D LiDAR sensors used.

A. METHODOLOGY

This study focuses on a selection of open-source, 3D LiDAR-based SLAM algorithms that collectively represent a diverse set of strategies, sensor integrations, and loop closure techniques. As summarized in Tab. 3, the algorithms include foundational methods such as LOAM [13] and LeGO-LOAM [14], as well as more recent contributions like GLIM [18], Kinematic-ICP [6], and MOLA-LO [4]. Some rely solely on LiDAR data, as exemplified by KISS-ICP [17], while others fuse data from additional sensors, such as IMUs (LIO-SAM [5], DLIO [15]) or wheel odometry (Kinematic-ICP [6], VineSLAM [16]). They also differ in loop closure strategies: certain methods employ basic distance/time thresholds (LeGO-LOAM [14], GLIM [18]), whereas others leverage advanced descriptors using Scan Context [69], [70]. Furthermore, although most of these approaches are based on LO, VineSLAM [16] adopts a particle-filter framework.

Additionally, not all algorithms officially support all 3D LiDAR sensors. In particular, the oldest algorithms contain feature extraction modules that limit their support for non-repetitive scanning 3D LiDAR sensors, such as the Livox Mid-360. Moreover, LOAM [13] and LeGO-LOAM [14] were initially optimized for mechanical spinning LiDARs that have a symmetric, uniform vertical FoV. Consequently, the RoboSense RS-Helios-5515 does not fully meet these criteria. However, we still evaluated the performance of these algorithms with this sensor for comparative analysis. Furthermore, although MOLA-LO [4] officially supports both IMU and wheel-odometry fusion, we were unable to evaluate these configurations in our experiments, as the sensor fusion modules are, at the moment, still under development.

Given that some original implementations were built on legacy code bases, this study employs refined forks or variants of the older algorithms. Specifically, A-LOAM [68] replaces LOAM to enhance code clarity and numerical stability. Likewise, LeGO-LOAM-BOR[12] refines LeGO-LOAM by improving its software structure. However, both of these implementations have removed the loosely coupled IMU-based deskewing step present in the original algorithms – a drawback not critical for our low-speed indoor dataset. In addition, LIORF[13] extends LIO-SAM by supporting a broader range of sensors (e.g., RoboSense LiDARs and 6-axis IMUs) and incorporating a Scan Context-based loop closure module. These updated forks retain the original algorithms' core principles while ensuring compatibility with modern hardware and software frameworks.

Moreover, this benchmark study employs the open-source tool EVO [40] to compute and analyze the following key metrics for evaluating SLAM algorithm performance:

- **Absolute Trajectory Error (m):** This metric evaluates the overall trajectory accuracy by computing the Root Mean Square Error (RMSE) between time-aligned ground truth and estimated trajectories. Prior to error calculation, trajectories are spatially aligned using Umeyama's method [71];
- **Relative Translational Error (%) and Relative Rotational Error (°/m):** Initially introduced in the KITTI Vision Benchmark Suite [31], these metrics quantify the drift in odometry by comparing relative pose estimates over successive trajectory segments. RTE expresses translational drift as a percentage of segment length, while RRE measures rotational drift in degrees

[12] https://github.com/facontidavide/LeGO-LOAM-BOR
[13] https://github.com/YJZLuckyBoy/liorf

TABLE 3. Overview of the 3D LiDAR-based SLAM algorithms evaluated in this benchmark study.

SLAM Algorithm	Open-source Code	ROS Version	3D LiDAR Support				Additional Sensor Support		Loop Closure	Year
			VLP-16	OS1-64	RS-5515	Mid-360	IMU	Wheel Odometry		
LOAM [13]	A-LOAM [68]	Noetic (ROS 1)	X	X	X(1)	–	–	–	–	2014
LeGO-LOAM [14]	LeGO-LOAM-BOR[12]	Noetic (ROS 1)	X	X	X(1)	–	–	–	X	2018
LIO-SAM [5]	LIORF[13]	Noetic (ROS 1)	X	X	X	–	X	–	X	2020
DLIO [15]	Official	Noetic (ROS 1)	X	X	X	X	X	–	–	2022
VineSLAM [16]	Official	Foxy (ROS 2)	X	X	X	X	X	X	–	2022
KISS-ICP [17]	Official	Humble (ROS 2)	X	X	X	X	–	–	–	2023
GLIM [18]	Official	Humble (ROS 2)	X	X	X	X	X	–	X	2024
Kinematic-ICP [6]	Official	Humble (ROS 2)	X	X	X	X	–	X	–	2024
MOLA-LO [4]	Official	Humble (ROS 2)	X	X	X	X	–(2)	–(2)	–	2024

(1) LOAM and LeGO-LOAM are optimized for mechanical spinning LiDARs that provide a symmetric and uniform vertical field of view. The RoboSense RS-Helios-5515 does not meet these criteria; however, data from sequences recorded with this sensor were still included for comparative analysis.

(2) Although MOLA-LO officially supports sensor fusion with both an IMU and wheel odometry, we were unable to evaluate these configurations in our experiments because the sensor fusion models were still insufficiently documented at the time of this article.

Abbreviations: Inertial Measurement Unit (IMU), Robot Operating System (ROS), Livox Mid-360 (Mid-360), Ouster OS1-64 (OS1-64), RoboSense RS-Helios-5515 (RS-5515), Velodyne VLP-16 (VLP-16)

per meter. These metrics provide a systematic assessment of how errors accumulate over time.

It is important to note that the present benchmark focuses exclusively on accuracy-related metrics. Other performance indicators, such as computational resource usage or real-time capabilities, are not considered, as they depend strongly on the hardware platform and runtime environment. Moreover, since the dataset was designed for offline benchmarking, the analysis is not intended to reflect deployment on embedded or resource-constrained systems. For studies centered on runtime efficiency and hardware-specific performance, we refer the reader to complementary benchmarking frameworks such as SLAMBench [37]–[39] and SLAM-HIVE [41], [42], which are discussed in the related work section.

Finally, the benchmark study results were obtained by evaluating the selected SLAM algorithms across all benchmark sequences from the IILABS 3D dataset. In total, the selected SLAM algorithms span a broad range of release years, reflecting both long-standing foundational approaches and the latest research advances. In order to address compatibility challenges (e.g., some algorithms require ROS 1 on Ubuntu 20.04, whereas others need ROS 2 on Ubuntu 22.04), we used Docker containers[14]. This containerized setup ensured that each SLAM system operated under consistent computational conditions, allowing for fair and controlled evaluations across the full set of benchmark sequences. In order to promote reproducibility, the Dockerfiles and ROS packages used for benchmarking the SLAM algorithms are available as open-source at GitHub[3]. Furthermore, a dedicated toolkit for downloading the dataset and computing the evaluation metrics is also available on GitHub[4].

Additionally, it is important to note that the odometry estimates produced by the SLAM algorithms are not always provided in the same reference frame as the ground-truth data (i.e., the *base_link* frame). For example, the output odometry may be provided in different reference frames: some algorithms report it in the LiDAR frame (e.g., A-LOAM and LIORF), others in the IMU frame (e.g., GLIM), some in the robot footprint frame (e.g., Kinematic-ICP), and yet others directly in the robot frame (e.g., DLIO, KISS-ICP, and MOLA-LO). In order to ensure a fair comparison against the ground-truth data, the odometry trajectories were transformed to the robot frame by applying the appropriate CAD-specified transformations. These are the same transformations provided as static transforms in the ROS tf topic of the dataset and as configuration parameters for the algorithms when required, thereby reinforcing consistency in the analysis.

[14]https://www.docker.com

B. RESULTS

The benchmark results of this study are presented in Tab. 4, Tab. 5, and Fig. 15, considering all six sequences, four 3D LiDAR sensors, and nine SLAM algorithms. The benchmark analysis is approached differently in the *elevator* sequence, since ground-truth data is unavailable for the entire trajectory. A quantitative analysis is conducted based on accuracy metrics for five of the sequences. In contrast, for the *elevator* sequence, a qualitative analysis is performed by examining the odometry trajectories produced by the SLAM algorithms.

The quantitative analysis is organized into two tables that report the ATE, RTE, and RRE for all four 3D LiDAR sensors. Tab. 4 presents the performance metrics obtained from sequences recorded with the RoboSense RS-Helios-5515 and Livox Mid-360 sensors, whereas Tab. 5 details similar metrics for data collected with the Velodyne VLP-16 and Ouster OS1-64 sensors. The RTE and RRE values are computed in both tables over 10-meter segments. Due to constraints inherent to some algorithms, not all combinations of sequences, sensors, and methods are available. For example, results for sequences acquired with the Livox Mid-360 are provided only for those algorithms that support this sensor, and quantitative outcomes for the *ramp* sequence are omitted for Kinematic-ICP, which is tailored for planar motion estimation. Nevertheless, qualitative results for this algorithm in the *elevator* sequence are included for illustrative purposes. Complementing the numerical data, Fig. 15 displays the odometry trajectories for the *elevator* sequence across all four sensor types. Due to the limited availability of ground-truth data provided only at the beginning and end of the trajectory, a quantitative evaluation is not feasible for this sequence, and a qualitative assessment is instead based on a visual analysis of the trajectories.

The results indicate that most algorithms maintain robust performance without significant divergences. A closer examination reveals that methods incorporating IMU sensor fusion generally outperform those that do not. However, algorithms relying on wheel odometry tend to exhibit more significant errors, with some exceptions. Among the evaluated approaches, LIORF consistently demonstrates the best overall performance, GLIM exhibits notably lower rotational drift, and KISS-ICP registers reduced translational drift. In the *elevator* sequence, only a few algorithms (LIORF, DLIO, GLIM, and MOLA-LO) could accurately estimate the trajectory. Moreover, the choice of sensor has an influence on performance, with the Livox Mid-360 and Ouster OS1-64 sensors enabling more accurate trajectory estimation compared to the RoboSense RS-Helios-5515 and Velodyne VLP-16. Finally, the integration of loop closure modules appears to have a marginal impact, as the performance differences between the original implementations and their loop closure–enhanced counterparts are minimal. These observations provide a solid foundation for the subsequent discussion of the underlying trends and algorithmic differences.

Due to the extensive scope of this benchmark study, which includes six sequences, four 3D LiDAR sensors, and nine SLAM algorithms, it is not practical to include illustrations of all resulting trajectories within this paper. The figures presented here, such as Fig. 15, Fig. 16, and Fig. 17, represent just a selected subset that highlights key trends and specific cases. For a complete set of trajectory visualizations, please refer to the project webpage[15].

C. DISCUSSION

As outlined in the previous subsection, all algorithms generally achieved robust performance across the benchmark sequences, yielding an ATE between 0.02 m and 0.05 m, an RTE around 1%, and an RRE below 0.2°/m, which is consistent with the expectations for a feature-rich environment such as the iilab. However, a few notable exceptions merit closer inspection. For example, the GLIM algorithm exhibited significant difficulties with the Velodyne VLP-16 sequences. The sensor's narrow vertical FoV, from -15° to 15°, appears to have induced drift along the z-axis, most likely since it could not provide enough points of the ground and/or ceiling. Moreover, in the *loop* sequence, where the robot transitions from the Nav A space to the corridor through a glass door, GLIM struggled with discrepancies between consecutive keyframe local maps in those transitions. These issues culminated in an ATE of 1.563 m and an RTE of 18.69% for that sequence, as illustrated in Fig. 17.

Moreover, algorithms incorporating IMU sensor fusion generally delivered superior accuracy. Among these, LIORF consistently demonstrated the best overall performance by achieving the lowest ATE, RTE, and RRE in several sequences. However, when the metrics are considered individually, the rankings vary: while LIORF excels in ATE, KISS-ICP and GLIM register the best performance in terms of RTE and RRE, respectively. One contributing factor to the enhanced performance of IMU-based methods is the use of the high-precision Xsens MTi-630 AHRS IMU, which operates at 400 Hz and provides accurate motion estimates that improve the initialization step of the ICP-based modules within these SLAM systems.

Conversely, algorithms that rely on wheel odometry data, such as Kinematic-ICP and VineSLAM, produced less accurate results. For instance, Kinematic-ICP consistently achieved an ATE exceeding 0.1 m in most sequences and even surpassed 0.5 m in the *slippage* sequence. This underperformance is likely related to its unicycle-based correction in the ICP-based pipeline, which is unsuitable for robots with omnidirectional steering. On the other hand, although the VineSLAM achieved satisfactory results in all sequences, with an ATE below 0.15 m, its performance was outstripped by several other methods. Despite being the only approach that fuses both IMU and wheel odometry data for sensor fusion, VineSLAM relies on a particle filter, whose computational complexity results in a lower update frequency for robot pose estimation. Furthermore, its overall trajectory exhibits more jitter than those generated by the LO-based algorithms, as illustrated in Fig. 16.

[15]https://jorgedfr.github.io/3d_lidar_slam_benchmark_at_iilab

TABLE 4. Benchmark results of 3D LiDAR-based SLAM algorithms using sequences from Livox Mid-360 and RoboSense RS-Helios-5515 sensors. Metrics include Absolute Trajectory Error (ATE) in meters (m), Relative Translational Error (RTE) in percentage (%), and Relative Rotational Error (RRE) in degrees per meter (°/m).

SLAM Algorithm	RoboSense RS-Helios-5515 Sequences					Livox Mid-360 Sequences				
	Nav A Diff	Nav A Omni	Loop	Slippage	Ramp	Nav A Diff	Nav A Omni	Loop	Slippage	Ramp
A-LOAM [68]	0.032 m 0.92% 0.097 °/m	0.032 m 1.05% 0.084 °/m	0.045 m 1.03% 0.100 °/m	0.042 m 1.32% 0.084 °/m	0.057 m 0.59% 0.073 °/m	–	–	–	–	–
LeGO-LOAM-BOR [12]	0.040 m 1.45% 0.114 °/m	0.038 m 1.29% 0.104 °/m	0.039 m 1.37% 0.132 °/m	0.043 m 1.71% 0.114 °/m	0.041 m 0.59% 0.168 °/m	–	–	–	–	–
LIORF [13]	0.029 m 1.23% 0.090 °/m	0.024 m 1.24% 0.109 °/m	0.023 m 1.28% 0.093 °/m	0.029 m 0.97% 0.077 °/m	0.034 m 0.47% 0.119 °/m	–	–	–	–	–
DLIO [15]	0.068 m 1.55% 0.107 °/m	0.050 m 1.64% 0.122 °/m	0.040 m 1.36% 0.102 °/m	0.074 m 1.31% 0.073 °/m	0.030 m 0.30% 0.125 °/m	0.027 m 1.01% 0.056 °/m	0.021 m 0.99% 0.069 °/m	0.017 m 1.14% 0.050 °/m	0.030 m 0.92% 0.042 °/m	0.012 m 0.17% 0.073 °/m
VineSLAM [16]	0.108 m 2.00% 0.148 °/m	0.078 m 1.97% 0.177 °/m	0.123 m 2.55% 0.199 °/m	0.095 m 1.67% 0.111 °/m	0.056 m 0.80% 0.315 °/m	0.188 m 2.70% 0.223 °/m	0.106 m 1.95% 0.188 °/m	0.124 m 2.07% 0.146 °/m	0.096 m 1.56% 0.072 °/m	0.108 m 1.12% 0.335 °/m
KISS-ICP [17]	0.046 m 1.10% 0.091 °/m	0.046 m 1.10% 0.080 °/m	0.047 m 1.00% 0.098 °/m	0.041 m 0.96% 0.081 °/m	0.039 m 0.58% 0.125 °/m	0.031 m 0.76% 0.064 °/m	0.030 m 0.75% 0.067 °/m	0.026 m 0.83% 0.059 °/m	0.033 m 0.95% 0.052 °/m	0.028 m 0.53% 0.099 °/m
GLIM [18]	0.137 m 1.60% 0.051 °/m	0.055 m 1.16% 0.038 °/m	0.560 m 7.36% 0.049 °/m	0.030 m 1.43% 0.038 °/m	0.057 m 0.65% 0.036 °/m	0.017 m 0.78% 0.044 °/m	0.016 m 0.81% 0.052 °/m	0.104 m 1.76% 0.036 °/m	0.025 m 0.82% 0.039 °/m	0.057 m 0.64% 0.055 °/m
Kinematic-ICP [6]	0.337 m 2.87% 0.184 °/m	0.178 m 2.27% 0.143 °/m	0.103 m 1.62% 0.118 °/m	0.676 m 8.86% 0.110 °/m	–	0.085 m 1.62% 0.066 °/m	0.181 m 2.31% 0.108 °/m	0.053 m 1.38% 0.084 °/m	0.584 m 7.45% 0.106 °/m	–
MOLA-LO [4]	0.044 m 1.26% 0.104 °/m	0.031 m 1.15% 0.076 °/m	0.032 m 1.25% 0.104 °/m	0.040 m 1.21% 0.099 °/m	0.134 m 0.94% 0.230 °/m	0.025 m 0.77% 0.057 °/m	0.022 m 0.77% 0.063 °/m	0.018 m 0.98% 0.058 °/m	0.027 m 0.93% 0.047 °/m	0.023 m 0.26% 0.056 °/m
LeGO-LOAM-BOR + LC [12]	0.045 m 1.44% 0.113 °/m	0.043 m 1.37% 0.099 °/m	0.050 m 1.81% 0.133 °/m	0.043 m 1.69% 0.112 °/m	0.041 m 0.56% 0.168 °/m	–	–	–	–	–
LIORF + LC [13]	0.029 m 1.22% 0.093 °/m	0.023 m 1.21% 0.104 °/m	0.022 m 1.27% 0.091 °/m	0.031 m 1.00% 0.078 °/m	0.022 m 0.33% 0.110 °/m	–	–	–	–	–
GLIM + LC [18]	0.148 m 1.76% 0.051 °/m	0.059 m 1.27% 0.038 °/m	0.487 m 6.66% 0.048 °/m	0.030 m 1.41% 0.037 °/m	0.057 m 0.64% 0.036 °/m	0.016 m 0.78% 0.044 °/m	0.016 m 0.81% 0.051 °/m	0.103 m 1.74% 0.036 °/m	0.025 m 0.82% 0.039 °/m	0.057 m 0.64% 0.056 °/m

Abbreviations: Loop Closure (LC)

When evaluating the results on a per-sensor basis, the Livox Mid-360 consistently enabled the best outcomes across sequences. For example, in the *nav a diff* sequence, the optimal metrics achieved by GLIM and KISS-ICP were an ATE of 0.017 m, an RTE of 0.76%, and an RRE of 0.044°/m, outperforming results obtained with the other sensors. This improvement can be attributed to the Livox Mid-360's distinctive vertical FoV, which extends from -7° to 52°, thereby capturing abundant ceiling features that enhance the performance of the ICP-based local optimization modules. Notably, the VineSLAM algorithm did not exhibit a similar improvement with this sensor, reinforcing the notion that the advantages of the Livox Mid-360 are closely tied to the presence of a practical local optimization component. Additionally, specific algorithms appear to perform better with particular sensors, likely due to the extensive parameter sets of these algorithms, which can be more fine-tuned for specific sensors. For instance, GLIM excelled with the Livox Mid-360, while MOLA-LO delivered better results with the Ouster OS1-64.

Despite these particular cases, the minimal differences observed in algorithm performance among the mechanical spinning LiDARs (Velodyne VLP-16, RoboSense RS-Helios-5515, and Ouster OS1-64) suggest that the vertical resolution of these sensors, which is determined by the number of beams and vertical FoV, has a limited impact in indoor environments. Similarly, the unique characteristics of the RoboSense

TABLE 5. Benchmark results of 3D LiDAR-based SLAM algorithms using sequences from Velodyne VLP-16 and Ouster OS1-64 sensors. Metrics include Absolute Trajectory Error (ATE) in meters (m), Relative Translational Error (RTE) in percentage (%), and Relative Rotational Error (RRE) in degrees per meter (°/m).

SLAM Algorithm	Velodyne VLP-16 Sequences					Ouster OS1-64 Sequences				
	Nav A Diff	Nav A Omni	Loop	Slippage	Ramp	Nav A Diff	Nav A Omni	Loop	Slippage	Ramp
A-LOAM [68]	0.031 m 1.28% 0.066 °/m	0.049 m 1.58% 0.084 °/m	0.039 m 1.68% 0.104 °/m	0.040 m 1.53% 0.083 °/m	0.031 m 0.38% 0.072 °/m	0.039 m 1.11% 0.072 °/m	0.043 m 1.34% 0.089 °/m	0.029 m 1.15% 0.080 °/m	0.041 m 1.18% 0.088 °/m	0.038 m 0.39% 0.052 °/m
LeGO-LOAM-BOR [12]	0.054 m 1.33% 0.114 °/m	0.047 m 1.43% 0.110 °/m	0.042 m 1.67% 0.119 °/m	0.053 m 1.87% 0.151 °/m	0.050 m 0.75% 0.189 °/m	0.042 m 1.14% 0.097 °/m	0.036 m 1.23% 0.106 °/m	0.033 m 1.43% 0.083 °/m	0.045 m 1.44% 0.104 °/m	0.036 m 0.52% 0.150 °/m
LIORF [13]	0.030 m 1.33% 0.057 °/m	0.027 m 1.43% 0.064 °/m	0.025 m 1.63% 0.086 °/m	0.035 m 1.40% 0.084 °/m	0.030 m 0.36% 0.078 °/m	0.032 m 1.06% 0.060 °/m	0.031 m 1.24% 0.070 °/m	0.021 m 1.31% 0.055 °/m	0.028 m 1.19% 0.057 °/m	0.017 m 0.21% 0.026 °/m
DLIO [15]	0.064 m 1.77% 0.095 °/m	0.049 m 1.81% 0.121 °/m	0.036 m 1.69% 0.093 °/m	0.065 m 1.42% 0.075 °/m	0.031 m 0.30% 0.112 °/m	0.042 m 1.40% 0.086 °/m	0.032 m 1.49% 0.110 °/m	0.029 m 1.41% 0.083 °/m	0.044 m 1.23% 0.062 °/m	0.020 m 0.19% 0.080 °/m
VineSLAM [16]	0.083 m 2.03% 0.147 °/m	0.089 m 2.00% 0.166 °/m	0.128 m 2.24% 0.182 °/m	0.077 m 2.18% 0.148 °/m	0.048 m 0.62% 0.283 °/m	0.144 m 2.61% 0.201 °/m	0.139 m 2.86% 0.242 °/m	0.143 m 2.83% 0.251 °/m	0.147 m 2.57% 0.153 °/m	0.086 m 1.08% 0.369 °/m
KISS-ICP [17]	0.052 m 1.41% 0.093 °/m	0.047 m 1.38% 0.091 °/m	0.045 m 1.50% 0.096 °/m	0.045 m 1.56% 0.088 °/m	0.041 m 0.91% 0.149 °/m	0.036 m 1.07% 0.073 °/m	0.038 m 1.10% 0.079 °/m	0.032 m 1.00% 0.063 °/m	0.035 m 1.12% 0.069 °/m	0.030 m 0.64% 0.087 °/m
GLIM [18]	0.525 m 2.99% 0.048 °/m	0.286 m 2.73% 0.046 °/m	1.563 m 18.69% 0.083 °/m	0.082 m 2.38% 0.036 °/m	0.068 m 0.82% 0.040 °/m	0.029 m 1.15% 0.066 °/m	0.021 m 1.18% 0.075 °/m	0.034 m 1.17% 0.058 °/m	0.036 m 1.23% 0.074 °/m	0.053 m 0.34% 0.050 °/m
Kinematic-ICP [6]	0.183 m 2.43% 0.140 °/m	0.180 m 2.45% 0.152 °/m	0.147 m 1.96% 0.137 °/m	0.676 m 8.63% 0.093 °/m	–	0.510 m 3.18% 0.224 °/m	0.191 m 2.59% 0.172 °/m	0.152 m 1.95% 0.137 °/m	0.788 m 9.94% 0.135 °/m	–
MOLA-LO [4]	0.045 m 1.41% 0.088 °/m	0.040 m 1.48% 0.088 °/m	0.037 m 1.69% 0.106 °/m	0.039 m 1.54% 0.092 °/m	0.338 m 3.53% 0.822 °/m	0.028 m 0.97% 0.070 °/m	0.026 m 1.07% 0.080 °/m	0.020 m 1.06% 0.057 °/m	0.028 m 1.12% 0.064 °/m	0.015 m 0.24% 0.054 °/m
LeGO-LOAM-BOR + LC [12]	0.059 m 1.33% 0.104 °/m	0.047 m 1.37% 0.108 °/m	0.051 m 1.67% 0.115 °/m	0.053 m 1.81% 0.134 °/m	0.046 m 0.73% 0.183 °/m	0.046 m 1.17% 0.098 °/m	0.042 m 1.24% 0.108 °/m	0.042 m 1.71% 0.088 °/m	0.045 m 1.43% 0.105 °/m	0.031 m 0.46% 0.148 °/m
LIORF + LC [13]	0.028 m 1.33% 0.055 °/m	0.027 m 1.45% 0.068 °/m	0.019 m 1.50% 0.076 °/m	0.040 m 1.42% 0.097 °/m	0.027 m 0.34% 0.073 °/m	0.031 m 1.07% 0.061 °/m	0.031 m 1.25% 0.069 °/m	0.020 m 1.33% 0.054 °/m	0.028 m 1.21% 0.056 °/m	0.016 m 0.20% 0.026 °/m
GLIM + LC [18]	0.461 m 2.91% 0.046 °/m	0.242 m 2.69% 0.048 °/m	1.262 m 15.76% 0.077 °/m	0.109 m 2.74% 0.035 °/m	0.069 m 0.86% 0.041 °/m	0.030 m 1.13% 0.066 °/m	0.021 m 1.20% 0.075 °/m	0.030 m 1.20% 0.058 °/m	0.035 m 1.23% 0.073 °/m	0.053 m 0.34% 0.050 °/m

Abbreviations: Loop Closure (LC)

RS-Helios-5515, such as its asymmetric and non-uniform FoV resolution, do not appear to affect performance in these settings significantly.

With respect to loop-closure modules, the results remain inconclusive. In most sequences, the algorithms equipped with a loop-closure module (LeGO-LOAM-BOR, LIORF, and GLIM) did not require loop-closure optimization to correct large trajectory drifts. The only exception was the GLIM algorithm, which relied on its loop-closure module to mitigate z-axis drift, mainly when using the Velodyne VLP-16 sensor, as illustrated in Fig. 16. As such, a direct comparison between the different loop-closure modules is not possible.

Finally, regarding the *elevator* sequence, only a small subset of the algorithms could accurately estimate vertical motion during both ascent and descent. Among the nine SLAM methods, three are explicitly designed for ground applications (LeGO-LOAM-BOR, VineSLAM, and Kinematic-ICP), and, as expected, these approaches failed to capture vertical movements. In contrast, three algorithms that incorporate IMU data (LIORF, DLIO, and GLIM) managed to estimate vertical motion accurately with certain sensors, notably the Ouster OS1-64 and Livox Mid 360. At the same time, performance with the Velodyne VLP-16 and RoboSense RS-Helios-5515 was less reliable, likely due to the lower point-cloud density provided by these sensors. Among the IMU-based methods, LIORF successfully tracked vertical motion during ascent

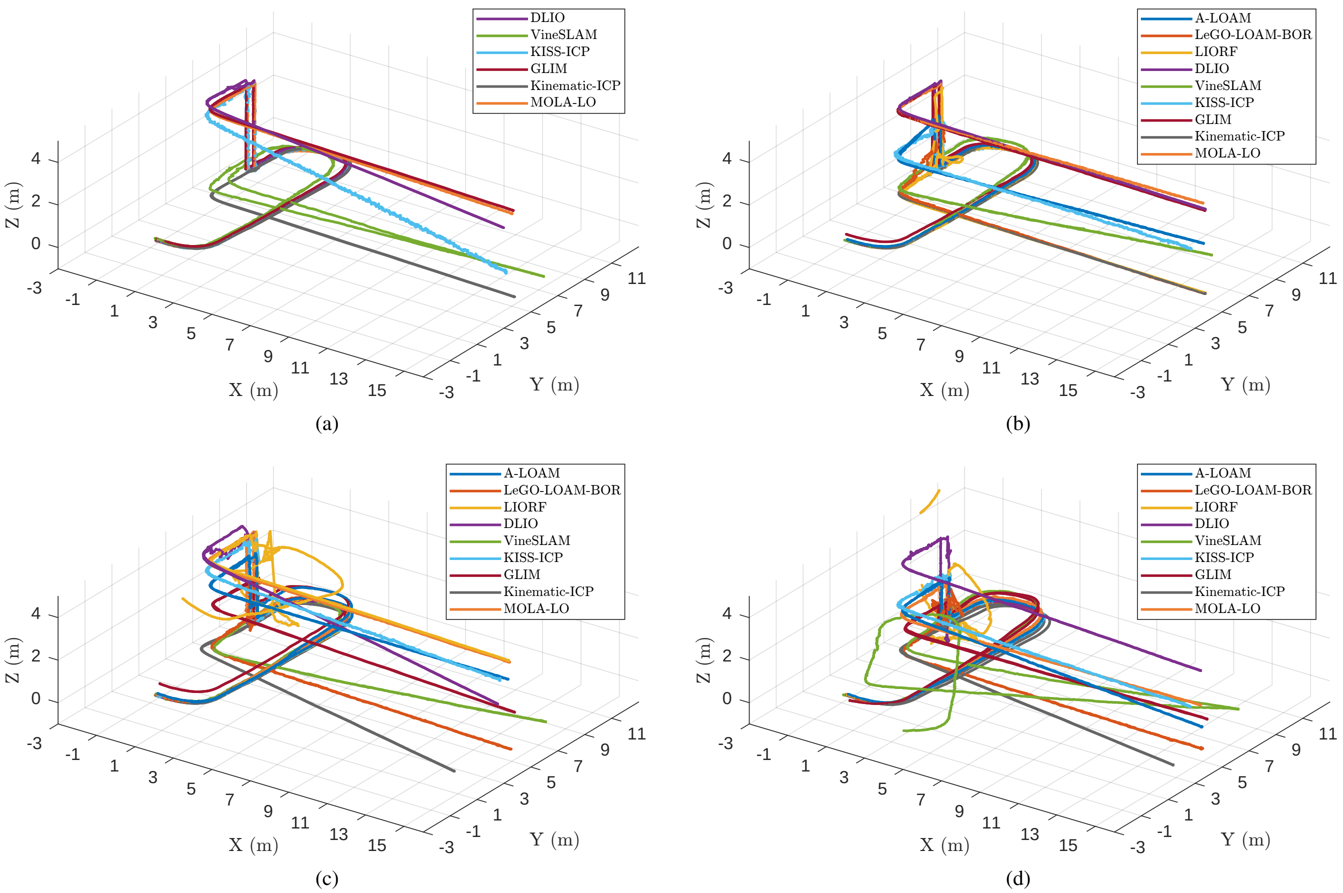


FIGURE 15. Odometry trajectories of 3D LiDAR-based SLAM algorithms in the *elevator* sequence for different 3D LiDAR sensors: (a) Livox Mid-360; (b) Ouster OS1-64; (c) RoboSense RS-Helios-5515; (d) Velodyne VLP-16.

for three sensors (Ouster OS1-64, Livox Mid-360, and RoboSense RS-Helios-5515) but produced inaccurate estimates during descent for two of them (Ouster OS1-64 and RoboSense RS-Helios-5515), and it completely diverged for the Velodyne VLP-16. Additionally, of the three algorithms that rely solely on LiDAR data (A-LOAM, KISS-ICP, and MOLA-LO), both A-LOAM and KISS-ICP failed to capture the full vertical displacement by exhibiting an offset in the vertical distance to the second floor. In contrast, MOLA-LO surprisingly managed to estimate vertical motion accurately for three sensors (Ouster OS1-64, Livox Mid-360, and RoboSense RS-Helios-5515), albeit with an offset error for the Velodyne VLP-16, likely due to insufficient point-cloud information from that sensor.

V. CONCLUSION

In this study, we presented a comprehensive benchmark of state-of-the-art 3D LiDAR-based SLAM algorithms evaluated on the IILABS 3D dataset. This dataset was developed to address key limitations of existing resources by providing multiple sensor sources along with calibration sequences and challenging benchmark trajectories, all acquired using a mobile robot at the iilab. The sensor sources include four 3D LiDARs (Velodyne VLP-16, Ouster OS1-64, RoboSense RS-Helios-5515) with different FoVs and scanning patterns, an IMU (Xsens MTi-630 AHRS), and wheel odometry.

By testing nine algorithms over six sequences and four distinct LiDAR sensors, our study offers a comparison of their performance in terms of accuracy and robustness under diverse indoor conditions. Overall, the results indicate that most SLAM methods perform robustly, with algorithms incorporating IMU sensor fusion consistently delivering superior accuracy, whereas approaches that rely on wheel odometry tend to yield larger errors. As such, given that wheel odometry is more accessible than IMU sensors in many industrial environments, there exists a clear research gap in developing robust SLAM methods that effectively leverage wheel odometry. Moreover, the analysis revealed that sensor characteristics play a critical role: the Livox Mid-360 and Ouster OS1-64 sensors, with their dense point clouds and favorable FoV, enabled more precise trajectory estimation compared to the RoboSense RS-Helios-5515 and Velodyne VLP-16. Additionally, the challenging task of accurately capturing vertical motion in the *elevator* sequence further highlighted the limitations of current SLAM approaches.

Additionally, it is important to note that among the nine

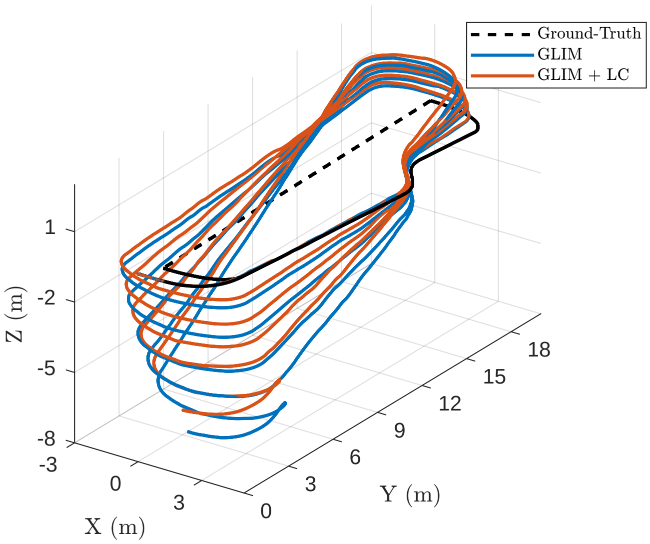


FIGURE 16. Odometry trajectories of the GLIM algorithm in the *loop* sequence using the Velodyne VLP-16 LiDAR sensor, with and without loop-closure module

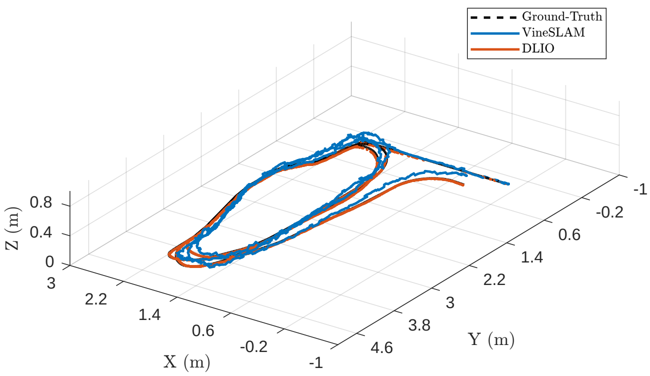


FIGURE 17. Odometry trajectories of the VineSLAM and DLIO algorithms in the *ramp* sequence using the Velodyne VLP-16 LiDAR sensor

evaluated algorithms, only VineSLAM and MOLA-LO provide both mapping and dedicated localization modules. While all the algorithms are capable of generating maps, only these two offer integrated localization functionalities that use the previously built map to accurately determine the robot's pose. This gap in the state-of-the-art highlights a promising direction for future research aimed at developing SLAM solutions that seamlessly combine robust mapping with efficient localization in challenging indoor environments.

Finally, we provide open-source benchmark scripts used to evaluate all nine SLAM algorithms presented in this work, along with a dedicated toolkit for downloading the IILABS 3D dataset and computing evaluation metrics. These resources not only facilitate the replication of this benchmark study but also support further research in indoor LiDAR-based SLAM.

REFERENCES

[1] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser. Simultaneous localization and mapping: a survey of current trends in autonomous driving. *IEEE Transactions on Intelligent Vehicles*, 2(3):194–220, 2017.

[2] T.-M. Nguyen, S. Yuan, M. Cao, Y. Lyu, T.H. Nguyen, and L. Xie. NTU VIRAL: a visual-inertial-ranging-LiDAR dataset, from an aerial vehicle viewpoint. *The International Journal of Robotics Research*, 41(3):270–280, 2022.

[3] R. Roriz, J. Cabral, and T. Gomes. Automotive LiDAR technology: a survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(7):6282–6297, 2022.

[4] J.L. Blanco-Claraco. A flexible framework for accurate LiDAR odometry, map manipulation, and localization. *The International Journal of Robotics Research*, 2025.

[5] T. Shan, B. Englot, D. Meyers, W. Wang, C. Ratti, and D. Rus. LIO-SAM: tightly-coupled LiDAR inertial odometry via smoothing and mapping. In *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 5135–5142, 2020.

[6] T. Guadagnino, B. Mersch, I. Vizzo, S. Gupta, M.V.R. Malladi, L. Lobefaro, G. Doisy, and C. Stachniss. Kinematic-ICP: enhancing LiDAR odometry with kinematic constraints for wheeled mobile robots moving on planar surfaces. *arXiv Preprint*, arXiv:2410.10277, 2024.

[7] R. Siegwart, I.R. Nourbakhsh, and D. Scaramuzza. *Introduction to autonomous mobile robots*, volume 23. MIT press, 2011.

[8] H. Durrant-Whyte and T. Bailey. Simultaneous localization and mapping: part I. *IEEE Robotics & Automation Magazine*, 13(2):99–110, 2006.

[9] T. Bailey and H. Durrant-Whyte. Simultaneous localization and mapping (SLAM): part II. *IEEE Robotics & Automation Magazine*, 13(3):108–117, 2006.

[10] R.B. Sousa, H.M. Sobreira, and A.P. Moreira. A systematic literature review on long-term localization and mapping for mobile robots. *Journal of Field Robotics*, 40(5):1245–1322, 2023.

[11] C. Cadena, L. Carlone, H. Carrillo, Y. Latif, D. Scaramuzza, J. Neira, I. Reid, and J.J. Leonard. Past, present, and future of simultaneous localization and mapping: toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[12] X. Xu, L. Zhang, J. Yang, C. Cao, W. Wang, Y. Ran, Z. Tan, and M. Luo. A review of multi-sensor fusion SLAM systems based on 3D LiDAR. *Remote Sensing*, 14(12), 2022.

[13] J. Zhang and S. Singh. LOAM: LiDAR odometry and mapping in real-time. In *Robotics: Science and systems*, pages 1–9, 2014.

[14] T. Shan and B. Englot. LeGO-LOAM: lightweight and ground-optimized Lidar odometry and mapping on variable terrain. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4758–4765, 2018.

[15] K. Chen, R. Nemiroff, and B.T. Lopez. Direct LiDAR-inertial odometry: lightweight LIO with continuous-time motion correction. *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3983–3989, 2023.

[16] A.S. Aguiar, F.N. Santos, H. Sobreira, J. Boaventura-Cunha, and A.J. Sousa. Localization and mapping on agriculture based on point-feature extraction and semiplanes segmentation from 3D LiDAR data. *Frontiers in Robotics and AI*, 9, 2022.

[17] I. Vizzo, T. Guadagnino, B. Mersch, L. Wiesmann, J. Behley, and C. Stachniss. KISS-ICP: in defense of point-to-point ICP – simple, accurate, and robust registration if done the right way. *IEEE Robotics and Automation Letters*, 8(2):1029–1036, 2023.

[18] K. Koide, M. Yokozuka, S. Oishi, and A. Banno. GLIM: 3D range-inertial localization and mapping with GPU-accelerated scan matching factors. *Robotics and Autonomous Systems*, 179, 2024.

[19] G. Grisetti, R. Kümmerle, C. Stachniss, and W. Burgard. A tutorial on graph-based SLAM. *IEEE Intelligent Transportation Systems Magazine*, 2(4):31–43, 2010.

[20] G. Grisetti, C. Stachniss, and W. Burgard. Improved techniques for grid mapping with Rao-Blackwellized particle filters. *IEEE Transactions on Robotics*, 23(1):34–46, 2007.

[21] S. Kohlbrecher, O. von Stryk, J. Meyer, and U. Klingauf. A flexible and scalable SLAM system with full 3D motion estimation. In *2011 IEEE International Symposium on Safety, Security, and Rescue Robotics*, pages 155–160, 2011.

[22] W. Hess, D. Kohler, H. Rapp, and D. Andor. Real-time loop closure in 2D LiDAR SLAM. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278, 2016.

[23] S. Macenski and I. Jambrecic. SLAM toolbox: SLAM for the dynamic world. *Journal of Open Source Software*, 6(61), 2021.

[24] H. Wang, C. Wang, C.-L. Chen, and L. Xie. F-LOAM: fast LiDAR odometry and mapping. In *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4390–4396, 2021.

[25] J. Lin and F. Zhang. LOAM Livox: a fast, robust, high-precision LiDAR odometry and mapping package for LiDARs of small FoV. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3126–3131, 2020.

[26] P.J. Besl and N.D. McKay. A method for registration of 3-D shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[27] W. Xu and F. Zhang. FAST-LIO: a fast, robust LiDAR-inertial odometry package by tightly-coupled iterated Kalman filter. *IEEE Robotics and Automation Letters*, 6(2):3317–3324, 2021.

[28] S. Ferrari, L.D. Giammarino, L. Brizi, and G. Grisetti. MAD-ICP: it is all about matching data – robust and informed LiDAR odometry. *IEEE Robotics and Automation Letters*, 9(11):9175–9182, 2024.

[29] Y. Pan, X. Zhong, L. Wiesmann, T. Posewsky, J. Behley, and C. Stachniss. PIN-SLAM: LiDAR SLAM using a point-based implicit neural representation for achieving global map consistency. *IEEE Transactions on Robotics*, 40:4045–4064, 2024.

[30] K. Koide, J. Miura, and E. Menegatti. A portable three-dimensional LiDAR-based system for long-term and wide-area people behavior measurement. *International Journal of Advanced Robotic Systems*, 16(2), 2019.

[31] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: the KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.

[32] J. Sturm, N. Engelhard, F. Endres, W. Burgard, and D. Cremers. A benchmark for the evaluation of RGB-D SLAM systems. In *2012 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 573–580, 2012.

[33] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *2012 IEEE Conference on Computer Vision and Pattern Recognition*, pages 3354–3361, 2012.

[34] T. Schöps, J.L. Schönberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger. A multi-view stereo benchmark with high-resolution images and multi-camera videos. In *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2538–2547, 2017.

[35] Y. Liao, J. Xie, and A. Geiger. KITTI-360: a novel dataset and benchmarks for urban scene understanding in 2D and 3D. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(3):3292–3310, 2023.

[36] L. Brizi, E. Giacomini, L.D. Giammarino, S. Ferrari, O. Salem, L.D. Rebotti, and G. Grisetti. VBR: a vision benchmark in Rome. In *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 15868–15874, 2024.

[37] L. Nardi, B. Bodin, M.Z. Zia, J. Mawer, A. Nisbet, P.H.J. Kelly, A.J. Davison, M. Luján, M.F.P. O'Boyle, G. Riley, N. Topham, and S. Furber. Introducing SLAMBench, a performance and accuracy benchmarking methodology for SLAM. In *2015 IEEE International Conference on Robotics and Automation (ICRA)*, pages 5783–5790, 2015.

[38] B. Bodin, H. Wagstaff, S. Saecdi, L. Nardi, E. Vespa, J. Mawer, A. Nisbet, M. Lujan, S. Furber, A.J. Davison, P.H.J. Kelly, and M.F.P. O'Boyle. SLAMBench2: multi-objective head-to-head benchmarking for visual SLAM. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3637–3644, 2018.

[39] M. Bujanca, P. Gafton, S. Saeedi, A. Nisbet, B. Bodin, M.F.P. O'Boyle, A.J. Davison, P.H.J. Kelly, G. Riley, B. Lennox, M. Luján, and S. Furber. SLAMBench 3.0: systematic automated reproducible evaluation of SLAM systems for robot vision challenges and scene understanding. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 6351–6358, 2019.

[40] M. Grupp. evo: Python package for the evaluation of odometry and SLAM., 2017.

[41] Y. Yang, B. Xu, Y. Li, and S. Schwertfeger. The SLAM Hive benchmarking suite. In *2023 IEEE International Conference on Robotics and Automation (ICRA)*, pages 11257–11263, 2023.

[42] X. Liu, Y. Yang, B. Xu, D. Feng, and S. Schwertfeger. Benchmarking SLAM algorithms in the cloud: the SLAM Hive benchmarking suite. *arXiv Preprint*, arXiv:2406.17586, 2024.

[43] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M.W. Achtelik, and R. Siegwart. The EuRoC micro aerial vehicle datasets. *The International Journal of Robotics Research*, 35(10):1157–1163, 2016.

[44] L. Zhang, M. Helmberger, L.F.T. Fu, D. Wisth, M. Camurri, D. Scaramuzza, and M. Fallon. Hilti-Oxford dataset: a millimeter-accurate benchmark for simultaneous localization and mapping. *IEEE Robotics and Automation Letters*, 8(1):408–415, 2023.

[45] G. Kim, Y.S. Park, Y. Cho, J. Jeong, and A. Kim. MulRan: multimodal range dataset for urban place recognition. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6246–6253, 2020.

[46] N. Carlevaris-Bianco, A.K. Ushani, and R.M. Eustice. University of Michigan North Campus long-term vision and LiDAR dataset. *International Journal of Robotics Research*, 35(9):1023–1035, 2015.

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

[48] D. Feng, Y. Qi, S. Zhong, Z. Chen, Q. Chen, H. Chen, J. Wu, and J. Ma. S3E: a multi-robot multimodal dataset for collaborative SLAM. *IEEE Robotics and Automation Letters*, pages 1–8, 2024.

[49] J. Jiao, H. Wei, T. Hu, X. Hu, Y. Zhu, Z. He, J. Wu, J. Yu, X. Xie, H. Huang, et al. 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)*, pages 3851–3856, 2022.

[50] J. Yin, A. Li, T. Li, W. Yu, and D. Zou. M2DGR: a multi-sensor and multi-scenario SLAM dataset for ground robots. *IEEE Robotics and Automation Letters*, 7(2):2266–2273, 2022.

[51] L. Qingqing, Y. Xianjia, J.P. Queralta, and T. Westerlund. Multi-modal LiDAR dataset for benchmarking general-purpose localization and mapping algorithms. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3837–3844, 2022.

[52] Z. Chen, Y. Qi, D. Feng, X. Zhuang, H. Chen, X. Hu, J. Wu, K. Peng, and P. Lu. Heterogeneous LiDAR dataset for benchmarking robust localization in diverse degenerate scenarios. *arXiv Preprint*, arXiv:2409.04961, 2024.

[53] J. Yin, H. Yin, C. Liang, H. Jiang, and Z. Zhang. Ground-challenge: a multi-sensor SLAM dataset focusing on corner cases for ground robots. In *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1–5, 2023.

[54] M. Helmberger, K. Morin, B. Berner, N. Kumar, G. Cioffi, and D. Scaramuzza. The Hilti SLAM challenge dataset. *IEEE Robotics and Automation Letters*, 7(3):7518–7525, 2022.

[55] H. Wei, J. Jiao, X. Hu, J. Yu, X. Xie, J. Wu, Y. Zhu, Y. Liu, L. Wang, and M. Liu. FusionPortableV2: a unified multi-sensor dataset for generalized SLAM across diverse platforms and scalable environments. *The International Journal of Robotics Research*, 2024.

[56] I. Catalano, X. Yu, and J.P. Queralta. Towards robust UAV tracking in GNSS-denied environments: a multi-LiDAR multi-UAV dataset. In *2023 IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pages 1–7, 2023.

[57] A. Xie, Y. Qian, W. Yan, C. Wang, and M. Yang. Non-repetitive: a promising LiDAR scanning pattern. In *2024 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2653–2659, 2024.

[58] M.H. Korayem, A. Nakhai, and T.B. Rostam. Design, modelling and errors measurement of wheeled mobile robots. *The International Journal of Advanced Manufacturing Technology*, 28:403–416, 2006.

[59] J.D. Ribeiro, R.B. Sousa, J.G. Martins, A.S. Aguiar, F.N. Santos, and H.M. Sobreira. IILABS 3D: iilab indoor LiDAR-based SLAM dataset, 2025. [Dataset].

[60] R.B. Sousa, H.M. Sobreira, J.G. Martins, P.G. Costa, M.F. Silva, and A.P. Moreira. Integrating multimodal perception into ground mobile robots. In *2025 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 104–111, 2025.

[61] H.M. Sobreira, M. Pinto, A.P. Moreira, P.G. Costa, and J. Lima. Robust robot localization based on the perfect match algorithm. In *CONTROLO'2014 – Proceedings of the 11th Portuguese Conference on Automatic Control*, pages 607–616, 2015.

[62] R.B. Sousa, C.D. Rocha, J.G. Martins, J.P. Costa, J.T. Padrão, J.M. Sarmento, J.P. Carvalho, M.S. Lopes, P.G. Costa, and A.P. Moreira. A robotic framework for the Robot@Factory 4.0 competition. In *2024 IEEE International Conference on Autonomous Robot Systems and Competitions (ICARSC)*, pages 66–73, 2024.

[63] F. Zhu, Y. Ren, and F. Zhang. Robust real-time LiDAR-inertial initialization. In *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3948–3955, 2022.

[64] T.Y. Kim, G. Pak, and E. Kim. GRIL-Calib: targetless ground robot IMU-LiDAR extrinsic calibration method using ground plane motion constraints. *IEEE Robotics and Automation Letters*, 9(6):5409–5416, 2024.

[65] R. Buchanan. Allan Variance ROS, 2021.

[66] R.B. Sousa, M.R. Petry, P.G. Costa, and A.P. Moreira. OptiOdom: a generic approach for odometry calibration of wheeled mobile robots. *Journal of Intelligent & Robotic Systems*, 105(2), 2022.
[67] J.S. Furtado, H.H.T. Liu, G. Lai, H. Lacheray, and J. Desouza-Coelho. Comparative analysis of OptiTrack motion capture systems. In *Advances in Motion Sensing and Control for Robotic Applications*, pages 15–31, 2019.
[68] T. Qin and S. Cao. A-LOAM: advanced implementation of LOAM, 2019.
[69] G. Kim and A. Kim. Scan context: egocentric spatial descriptor for place recognition within 3D point cloud map. In *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 4802–4809, 2018.
[70] G. Kim, S. Choi, and A. Kim. Scan context++: structural place recognition robust to rotation and lateral variations in urban environments. *IEEE Transactions on Robotics*, 38(3):1856–1874, 2022.
[71] S. Umeyama. Least-squares estimation of transformation parameters between two point patterns. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 13(4):376–380, 1991.

JORGE DIOGO RIBEIRO received the B.Sc. degree in Electrical and Computer Engineering from the Faculty of Engineering of the University of Porto (FEUP) in 2022, and the M.Sc. degree in Electrical and Computer Engineering at FEUP in 2024. He is currently pursuing a Ph.D. degree in Electrical and Computer Engineering at FEUP and works as an assistant researcher at the Centre for Robotics in Industry and Intelligent Systems (CRIIS) of INESC TEC – Institute for Systems and Computer Engineering, Technology and Science. His research interests include robotics, mobile robots, control, trajectory planning, and localization and mapping.

RICARDO B. SOUSA received the M.Sc. degree in Electrical and Computer Engineering (integrated course) from the Faculty of Engineering of the University of Porto (FEUP) in 2020. He is currently pursuing a Ph.D. degree in Electrical and Computer Engineering at FEUP and works as a researcher at the Centre for Robotics in Industry and Intelligent Systems (CRIIS) of INESC TEC – Institute for Systems and Computer Engineering, Technology and Science. His research interests focus on automation, calibration, control, localization and mapping, mobile robots, and sensor fusion.

JOÃO G. MARTINS received the M.Sc. degree in Electrical and Computer Engineering, with a specialization in automation, from the Faculty of Engineering of the University of Porto (FEUP) in 2023. He is currently pursuing a Ph.D. degree in Electrical and Computer Engineering at FEUP and works as an assistant researcher at the Centre for Robotics in Industry and Intelligent Systems (CRIIS) of INESC TEC – Institute for Systems and Computer Engineering, Technology and Science. His research interests include robotics, computer vision, and extended reality, with applications in industrial robotics ranging from autonomous systems to human–machine interaction.

ANDRÉ S. AGUIAR received the M.Sc. degree in Electrical Engineering, with a specialization in robotics, from the Faculty of Engineering of the University of Porto (FEUP) in 2019, and the Ph.D. degree in Electrical Engineering from the University of Trás-os-Montes and Alto Douro (UTAD) in 2023. He is currently a researcher at the Centre for Robotics in Industry and Intelligent Systems (CRIIS) of INESC TEC – Institute for Systems and Computer Engineering, Technology and Science. His research interests include robotics navigation, simultaneous localization and mapping, computer vision, point cloud processing, and deep learning.

FILIPE N. SANTOS received the B.Sc degree in Electrical and Computer Engineering from the Instituto Superior de Engenharia do Porto (ISEP) in 2003, the M.Sc. degree in Electrical and Computer Engineering from Instituto Superior Técnico (IST) of the Universidade Técnica de Lisboa in 2007, and the Ph.D. degree in Electrical and Computer Engineering from the Faculty of Engineering of the University of Porto (FEUP) in 2014. He is currently a researcher at the Centre for Robotics in Industry and Intelligent Systems (CRIIS) of INESC TEC – Institute for Systems and Computer Engineering, Technology and Science. His work is particularly focused on developing robotic solutions for agricultural and forestry applications, addressing challenges in real-world operational environments. His research interests include visual perception and navigation, localization, mapping, and path planning.

HÉBER MIGUEL SOBREIRA received the M.Sc. degree in Electrical Engineering from the Faculty of Engineering of the University of Porto (FEUP) in 2009, and the Ph.D. degree in Electrical and Computer Engineering from the FEUP in 2017. He is currently a researcher at the Centre for Robotics in Industry and Intelligent Systems (CRIIS) of INESC TEC – Institute for Systems and Computer Engineering, Technology and Science. His research interests primarily include the navigation and control of indoor autonomous vehicles.