

Article

Comprehensive Performance Evaluation between Visual SLAM and LiDAR SLAM for Mobile Robots: Theories and Experiments

Yu-Lin Zhao, Yi-Tian Hong  and Han-Pang Huang *

Department of Mechanical Engineering, National Taiwan University, Taipei 106, Taiwan; f07522844@ntu.edu.tw (Y.-L.Z.); r12522813@ntu.edu.tw (Y.-T.H.)

* Correspondence: hanpang@ntu.edu.tw; Tel.: +886-2-33662700; Fax: +886-2-23676064

Abstract: SLAM (Simultaneous Localization and Mapping), primarily relying on camera or LiDAR (Light Detection and Ranging) sensors, plays a crucial role in robotics for localization and environmental reconstruction. This paper assesses the performance of two leading methods, namely ORB-SLAM3 and SC-LeGO-LOAM, focusing on localization and mapping in both indoor and outdoor environments. The evaluation employs artificial and cost-effective datasets incorporating data from a 3D LiDAR and an RGB-D (color and depth) camera. A practical approach is introduced for calculating ground-truth trajectories and during benchmarking, reconstruction maps based on ground truth are established. To assess the performance, ATE and RPE are utilized to evaluate the accuracy of localization; standard deviation is employed to compare the stability during the localization process for different methods. While both algorithms exhibit satisfactory positioning accuracy, their performance is suboptimal in scenarios with inadequate textures. Furthermore, 3D reconstruction maps established by the two approaches are also provided for direct observation of their differences and the limitations encountered during map construction. Moreover, the research includes a comprehensive comparison of computational performance metrics, encompassing Central Processing Unit (CPU) utilization, memory usage, and an in-depth analysis. This evaluation revealed that Visual SLAM requires more CPU resources than LiDAR SLAM, primarily due to additional data storage requirements, emphasizing the impact of environmental factors on resource requirements. In conclusion, LiDAR SLAM is more suitable for the outdoors due to its comprehensive nature, while Visual SLAM excels indoors, compensating for sparse aspects in LiDAR SLAM. To facilitate further research, a technical guide was also provided for the researchers in related fields.

Keywords: 3D SLAM; Visual SLAM; LiDAR SLAM; 3D reconstruction; robotics



Citation: Zhao, Y.-L.; Hong, Y.-T.; Huang, H.-P. Comprehensive Performance Evaluation between Visual SLAM and LiDAR SLAM for Mobile Robots: Theories and Experiments. *Appl. Sci.* **2024**, *14*, 3945. <https://doi.org/10.3390/app14093945>

Academic Editors: José Miguel Molina Martínez and Andrea Prati

Received: 9 January 2024

Revised: 11 April 2024

Accepted: 2 May 2024

Published: 6 May 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

SLAM has emerged as a prominent subject within the robotics community in recent years, leading to the development of numerous associated open-source projects. These projects encompass a wide range of applications, including autonomous vehicles, drones, architectural surveying, and scene scanning. SLAM stands for Simultaneous Localization and Mapping, representing a vital technique with immense potential. It enables a robot to robustly ascertain its position (localization) while concurrently constructing a real-time representation of previously unexplored environments (mapping) [1]. Various sensors are employed to gauge distances in the real world and capture pertinent features. Among these sensors, cameras and LiDAR systems are the most prevalent. Consequently, the SLAM domain primarily encompasses visual-based SLAM and LiDAR-based SLAM, contingent upon the input from the specific sensors.

In our previous study, we discussed mainly mapping for mobile robots in outdoor environments involving LiDAR SLAM and Visual SLAM [2]. Thus, this study primarily revolves around the performance comparison between Visual SLAM and LiDAR SLAM in terms of localization and mapping for mobile robots operating in real-world environments.

With an emphasis on maintaining continuity and universality, we strive to opt for cost-effective solutions. Therefore, the chosen algorithm representatives for these two SLAM methods are ORB-SLAM3 [3] and SC-LeGO-LOAM [4], respectively.

Visual SLAM (VSLAM) mainly infers the motion of the camera and the surrounding environment based on the continuous color image stream. To ensure minimal distortion and reduce calculations related to the epipolar geometry-based positional relationships, depth image streams are often integrated into the camera's positioning. As exemplified by our research focus, ORB-SLAM3 [3] is a feature-based SLAM system that employs pinhole and fisheye lens models to facilitate visual, visual-inertial, and multi-map SLAM by monocular, stereo, and RGB-D cameras. Based mainly on ORB-SLAM2 [5] and ORB-SLAM visual-inertial [6], ORB-SLAM3 introduces several novel methods, including a multi-map system, bag-of-words for loop closure, and advanced mid-term and long-term data association techniques.

LiDAR SLAM can be roughly divided into 2D and 3D LiDAR SLAM, generating 2D and 3D maps separately. Early LiDAR SLAM systems, such as Gmapping [7], and Simultaneous Localization, Mapping, and Moving Object Tracking (SLAM-MOT) [8] primarily utilized 2D LiDAR sensors and were often based on the Extended Kalman Filter (EKF) algorithm [9]. EKF is the nonlinear version of the Kalman Filter (KF), which approximates the state of a nonlinear system through a linear system that employs a first-order Taylor expansion. As hardware advances and algorithms become increasingly refined, optimization-based techniques and the integration of multiple scanning channels with 3D LiDAR play increasingly significant roles in the field of LiDAR SLAM. SC-LeGO-LOAM [4] is an expansion of the LOAM (Lidar Odometry And Mapping) [10] approach and incorporates a scan context loop detection technique [11] that utilizes a global descriptor obtained through the scan context to detect loops. The main objective of SC-LeGO-LOAM is to perform real-time odometry and mapping by efficiently processing the point cloud data acquired from a 3D LiDAR sensor and to achieve accurate and robust localization and mapping.

An Inertial Measurement Unit (IMU) can also be used as a supplement to the SLAM position relationship, providing vision with fast-moving positioning. However, the external parameter matrix from the camera or LiDAR to IMU must be obtained through physical measurement and sensor calibration. If the external reference calibration is not accurate enough, the positioning effect of the experimental restoration may be seriously wrong. Therefore, the experiments involved in this paper did not use IMU data.

Three-dimensional reconstruction serves as both an application and a component of SLAM technology. It furnishes the essential three-dimensional geometric data necessary for constructing environment maps and plays a pivotal role in realizing the perception, positioning, and navigation functions of the SLAM algorithm within the environment. Typically, these 3D reconstruction representations take the form of point cloud models, mesh models, and geometric models. For example, Kinect Fusion [12] can use the RGB-D Kinect camera to achieve instant 3D reconstruction of the environment through dense point cloud sampling and real-time tracking algorithms. Furthermore, algorithms from the LOAM series can also find applications in the Architecture, Engineering, and Construction (AEC) field [13].

The dataset determines the upper limit of the algorithm, so another focus of this paper is to record datasets using 3D LiDAR scans and RGB-D camera streams. Current public datasets differ with target scenes, recorded sensor data, and ground truth collection methods. For example, the TUM (Technical University of Munich) dataset [14,15] is suitable for pure Visual SLAM and visual-IMU SLAM, where the motion capture system records the ground truth. Another example is the KITTI (Karlsruhe Institute of Technology and Toyota Technological Institute) dataset [16], which also provides high-quality data. KITTI employs Real-Time Kinematic GPS (RTK-GPS, [17]) in conjunction with Inertial Navigation Systems (INS) to establish ground truth data. But it uses a stereo camera sensor instead of an RGB-D camera sensor.

In previous SLAM research or common service robots in the market, SLAM technology is often used only in one specific scenario. As SLAM technology advances with modern technology, robots are required to continuously navigate between indoor and outdoor environments. However, different environments may be suitable for different SLAM algorithms. Our research aims to assist researchers in gaining a more comprehensive understanding of the performance of various SLAM approaches. Moreover, the literature [2,18], is also useful for us to choose suitable locations for conducting multiple experiments, thereby enhancing the variability and assessing the performance of various SLAM systems.

The summary of this research work can be succinctly encapsulated as follows:

(a) A systematic review of the SLAM algorithms including Gmapping, ORB-SLAM3, SC-LeGO-LOAM, and Cartographer is introduced for the comparison. (b) The robot platform used for implementation, the process to establish datasets including RGB-D and 3D LiDAR data, and the mapping approach utilized in this study are introduced. In addition, the benchmark calculated by the Cartographer algorithm for the trajectory error estimation is acquired. (c) The comparative analysis is divided into three main sections: localization, mapping, and performance evaluation. The localization focuses on evaluating the accuracy of pose tracking by ATE and RPE, which can be acquired by calculating the Euclidean distance between the ground-truth poses and the estimated poses [19]. The mapping section focuses on evaluating 3D reconstruction performance, considering factors such as the quality of reconstructed models and adaptability to varying environmental conditions. Additionally, the performance evaluation encompasses metrics such as CPU utilization and memory usage. Effective resource management is deemed crucial for real-time applications and overall system usability.

2. Theories

This section offers a concise overview of pertinent SLAM (Simultaneous Localization and Mapping) algorithms, specifically Gmapping, ORB-SLAM3, SC-LeGO-LOAM, and the Cartographer algorithm. Gmapping stands out as one of the most widely-used SLAM algorithms; however, it relies on odometry data as the input, making it suitable solely as a reference for positional effects in Section 4. The primary focus of evaluation lies on ORB-SLAM3 and SC-LeGO-LOAM. Additionally, the Cartographer algorithm comes into play for calculating ground-truth trajectories. Ultimately, 3D reconstruction is employed to assess the mapping outcomes.

2.1. Gmapping

Gmapping combines the wheel odometry of the robot with current observation data so that the uncertainty can be reduced. However, it depends severely on the existence of odometers, which affect its robust performance [7]. To solve the error accumulation problem of learning grid maps with RBPFs (Rao-Blackwellized particle filters), Gmapping presents proposal distribution of statistics and adaptive resampling to improve the accuracy of particle filters. The adaptive resampling mechanism in Gmapping is based on a threshold N_{th} , which is usually set to half of the total particle number, and the currently effective number of particles N_{eff} , which can be calculated by Equation (1).

$$N_{eff} = \frac{1}{\sum_{i=1}^N (\tilde{w}^{(i)})^2} \quad (1)$$

where N is the total number of the particles and $\tilde{w}^{(i)}$ is the normalized weight of the particle i . Under the condition that N_{eff} is less than N_{th} , adaptive resampling will be performed.

This method combines robot encoder data with current observation data so that the uncertainty of robots can be reduced. However, it depends heavily on the existence of

odometers, which affected its robust performance. Furthermore, the number of particles became larger and influenced the calculation speed in a large scale scene.

2.2. ORB-SLAM3

ORB-SLAM3 [3] is a highly efficient system designed for Visual SLAM. It operates through three concurrent threads: tracking, local mapping, and loop and map merging, alongside an independent full bundle adjustment (BA) thread for map enhancement.

Bundle adjustment [20] is a non-linear optimization method utilized in the ORB-SLAM3 algorithm; it estimates accurate camera 3D locations and poses by minimizing the reprojection error with respect to the matched key points on the digital image. The SE(3), Special Euclidean Group, represents the special Euclidean group within Lie group algebra. It is a six-dimensional vector encompassing three rotational degrees of freedom and three translational degrees of freedom. The SE(3) manifold possesses a topological structure that renders it differentiable. Consequently, SE(3) can effectively determine an alignment. The error can be formulated as

$$\mathbf{e}_{i,j} = \mathbf{x}_{i,j} - \pi_i(\mathbf{T}_{iw}, \mathbf{X}_{w,j}) \quad (2)$$

and the minimized cost function is represented as Equation (3),

$$C = \sum_{i,j} \rho_h(\mathbf{e}_{i,j}^T \mathbf{\Omega}_{i,j}^{-1} \mathbf{e}_{i,j}) \quad (3)$$

where ρ_h is the Huber robust cost function and $\mathbf{\Omega}_{i,j} = \sigma_{i,j}^2 \mathbf{I}_{2 \times 2}$ is the covariance matrix related to the scale of the detected key point.

In ORB-SLAM3, the tracking thread processes image data, extracts key feature points, and matches them between consecutive frames to determine the current frame's position. To enhance pose estimation, this thread employs a motion-only BA, refining the accuracy of the calculated pose. Additionally, keyframes are discerned through predefined criteria. The local mapping thread implements the tracking thread's decisions, generating new keyframes and updating the local map. A culling mechanism is employed to systematically eliminate redundant keyframes and points, thereby bolstering the robustness of tracking. Furthermore, the local bundle adjustment optimizes all points within the local area for map accuracy. The loop and map merging thread operates within the atlas multi-map system, searching for similar scenes to achieve loop closure and integrate maps smoothly. Lastly, the full BA thread independently performs pose-graph optimization to refine the entire map without compromising real-time performance. By coordinating these threads, ORB-SLAM3 delivers real-time and optimal performance for diverse applications requiring accurate and up-to-date maps.

2.3. SC-LeGO-LOAM

SC-LeGO-LOAM [4], an extension of the LeGO-LOAM [21] method, incorporates a loop detection technique based on the scan context for enhanced loop identification. The process begins with the 3D LiDAR data points obtained from the scan, which are initially categorized into ground and segmented points using a ground segmentation method. Subsequently, an image segmentation method refines the segmented points into distinct clusters, each assigned a specific label for identification purposes.

By evaluating the roughness values c of both ground and segmented points and comparing them with a predefined threshold c_{th} , the classification of edge and planar features is then carried out. The roughness c is defined as follows:

$$c = \frac{1}{|S| \cdot \|r_i\|} \left\| \sum_{j \in S, j \neq i} (r_j - r_i) \right\| \quad (4)$$

where r_i and r_j are the Euclidean distances from points p_i and p_j in set S to LiDAR, respectively.

The estimation of the robot's motion follows, involving a comparison between the current feature set and that of the previous time step, with label matching enhancing the precision of matching results. The subsequent application of a two-step Levenberg–Marquardt (L–M) Optimization computes the pose transformation matrix. Loop detection is facilitated through the integration of a scan context method, transforming the 3D point cloud into polar coordinates and extracting global descriptors.

Utilizing a K-dimensional (KD) tree and conducting the nearest neighbor search, the frame with the highest similarity score is identified as the loopback frame, facilitating loop detection. The similarity score between queried and the candidate point clouds can be calculated by

$$d(I^q, I^c) = \frac{1}{N_s} \sum_{j=1}^{N_s} \left(1 - \frac{c_j^q \cdot c_j^c}{\|c_j^q\| \|c_j^c\|} \right) \quad (5)$$

where N_s represents axial sector number in scan context I and c_j^q, c_j^c represent column vectors at the same index in queried and the candidate frame, respectively.

The final stages encompass map optimization via the iterative closest point (ICP, [22]) method and the integration of positional data, culminating in the determination of an accurate final position and estimation.

2.4. Cartographer

The Cartographer [23] stands as an open-source real-time 2D and 3D mapping library and SLAM system, with its primary development carried out by Google. The original source code was introduced in 2016 [23] and has been updated subsequently. Consequently, we are presently utilizing the latest release version, which received its most recent update in 2021.

The Cartographer system employs a dual approach, incorporating both local and global optimization techniques to rectify errors and estimate poses. These two optimization methods are relatively independent and align with the distinct mapping processes: the front end, responsible for converting scans into submaps, and the back end, tasked with linking submaps into the global map. The latter can also be viewed as managing loop closures.

1. Scans-to-submap (Local map)

Initially, the LiDAR raw data were transformed into scans without any pose estimation. A sequence of consecutive scans was then employed to construct a submap, represented as a probability grid. To enhance the quality of the scans, a bicubic interpolation method was utilized for filtering and smoothing noisy data. This refined scan output is referred to as a grid scan. The scan matching process, utilizing a Ceres-based [24] scan matcher, was employed to perform this task, simplified as a nonlinear least squares problem.

$$\operatorname{argmin}_{\xi} \sum_{k=1}^K (1 - M_{\text{smooth}}(T_{\xi} h_k))^2 \quad (6)$$

where T_{ξ} transforms h_k from the scan frame to the submap frame according to the scan pose. The function $M_{\text{smooth}}: \mathbb{R}^2 \rightarrow \mathbb{R}$ is a smooth version of the probability values in the local submap. Consequently, this process enabled the estimation of the poses, with the grid scan subsequently incorporated into the submap.

2. Submaps-to-map (Global map)

As the front-end process gradually accumulates errors, the back end characterizes this optimization as a nonlinear least squares problem, akin to the local mapping step, termed sparse pose adjustment (SPA). This challenge involves handling scan poses and submap poses, incorporating constraints related to relative poses and their associated covariance matrices. To prevent an excessive computational burden when exploring the entire domain, the Cartographer employs a Branch-and-Bound (BAB) scan matching

approach. Its algorithm can be summarized as a method for refining the search process to obtain an optimal pose estimate.

2.5. Three-Dimensional Reconstruction

In this section, for the goal of 3D reconstruction, two maps generated separately by two SLAM methods are briefly introduced.

1. RGB-D Point Cloud

An RGB-D point cloud is a data representation that combines color information from a traditional RGB (Red, Green, and Blue) image with depth information to create a three-dimensional reconstruction of a scene or object. Figure 1 shows a typical RGB-D point cloud reconstruction process.

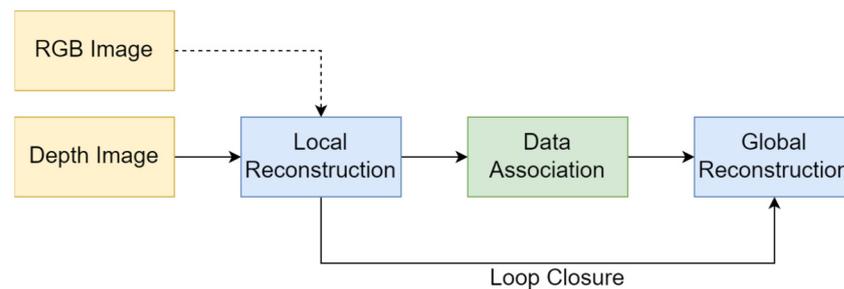


Figure 1. Overview of a typical RGB-D reconstruction process.

Initially, the preprocessed input data are aligned and used for local reconstruction. Next, we can determine the position and orientation of the cameras by establishing associations between visual features. This involves the process of matching observed features or landmarks in the environment with their corresponding features in a known or reference map. And loop closure detects and corrects errors in the estimated trajectory. Finally, based on the estimated camera pose, the input data are integrated into the 3D reconstruction mapping.

2. LiDAR Point Cloud

LiDAR Point cloud maps, generated by LiDAR SLAM, provide information regarding the positions of objects. This is achieved through the emission of laser beams by the LiDAR sensor, which is then reflected back from the objects to calculate their respective positions. Due to the properties of 3D LiDAR beams being emitted in a specific angular range and covering multiple directions at the same time, the point cloud maps possess the advantage of fast and wide-ranging simultaneous generation. As a result, it is well-suited for applications involving large-scale environments. To alleviate computational load and down-sampling, some mapping algorithms will preserve features like points, lines, and surfaces as data points.

3. Dataset and Benchmark

3.1. Robot Platform

This section describes the robot platform setups used in the experiments, which are modified from [25]. The differential wheeled mobile robot deployed several sensors, computer, and power. The sensors contain Stereolab Depth Camera Zed2, LSLIDAR 16-Line mechanical LiDAR, and wheel odometry. Thus, we collect wheel odometry, RGB and depth images, and 3D LiDAR scans. The transform matrixes of the different sensor frames are also calculated as Figure 2. The mobile robot (Figure 2) was constructed by our laboratory. The positions of the depth camera and 3D LiDAR mounted on the robot have been accurately measured by the 3D model file and the extrinsic calibration parameters have been computed using the D–H (Denavit–Hartenberg) parameter table in the field of robotics. It is worth mentioning that the robot also has 2D LiDAR; however, it is not

used because of poor accuracy. The computer deployed an Intel Core i7-9700TE (8 cores @ 3.8 GHz), which was used for all the dataset records and SLAM experiments. And 64-bit Ubuntu 20.04 was used for all of the experiments.

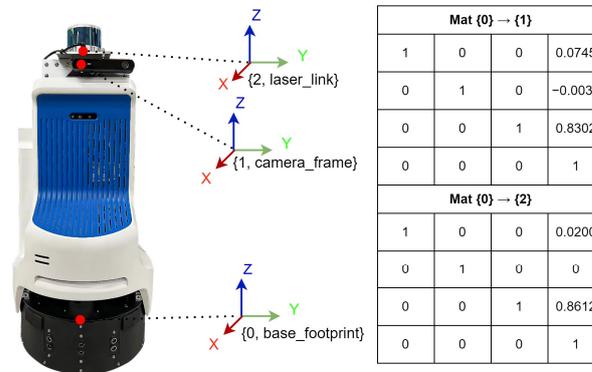


Figure 2. Hardware and coordinate system.

3.2. Benchmark

This section primarily focuses on the acquisition of cost-effective datasets with the ground truth serving as the benchmark. In essence, our task is to devise a methodology for obtaining ground truth data and crafting scenarios that align with our specific evaluation requirements for assessing the performance of various algorithms. In order to manage costs, it becomes crucial to obtain datasets that are economically viable while still serving as a valid benchmark for algorithmic effectiveness. As a result, we need to derive ground truth values based on the available sensors and data sources. Fortunately, the wheeled robots are typically designed to travel on flat ground and often have fixed sensors. Thus, the 2D SLAM algorithm aids in deriving the ground truth, encompassing both the robot's poses and their corresponding LiDAR timestamps. Additionally, we utilize interpolation to enhance the data content and synchronize LiDAR and image timestamps, contributing to the accurate calculation of error between camera poses obtained from ORB-SLAM3 and ground truth. Finally, leveraging the Cartographer as our foundation, we have proposed a technique for calculating ground truth. At the same time, we select scenes with engineering drawings or tiles as references while recording dataset scenes.

In this study, we recorded real-world datasets from two scenes on the campus of National Taiwan University (NTU) in Taipei, Taiwan. These scenes include (1) the corridor on the sixth floor of the College of Engineering Building as an indoor environment and (2) the second floor outdoor venue of the College of Engineering Building as an outdoor environment. Combined with comparing the engineering drawings of the building, we believe that the accuracy of this dataset can be guaranteed. In addition, the paper [23] shows that the relative error of the Cartographer is less than 1%, which is within our error margin. At this point, we have obtained benchmarks for comparing localization effects.

3.3. Mapping (3D Reconstruction)

Our subsequent goal is to obtain the mapping benchmark by expanding upon our established localization. As illustrated in Figure 3, this paper introduces two distinct RGB-D reconstruction processes based on different data association methods. Figure 3a shows a RGB-D reconstruction with loop closure, relying on ORB-SLAM3. In contrast, Figure 3b illustrates RGB-D reconstruction based on ground truth, preventing additional loop closure as acquiring ground truth incorporates loopback correction. To optimize the effect of mapping and reduce the amount of calculation, we use three filters when constructing the map. (i) Pass-through filters are employed to filter data along a specific dimension efficiently. It can remove errors caused by the inherent characteristics of the system and the imaging principle. (ii) Voxel filters are utilized to reduce point sampling and mitigate bias. (iii) Statistical outlier removal filters serve the purpose of eliminating outliers.

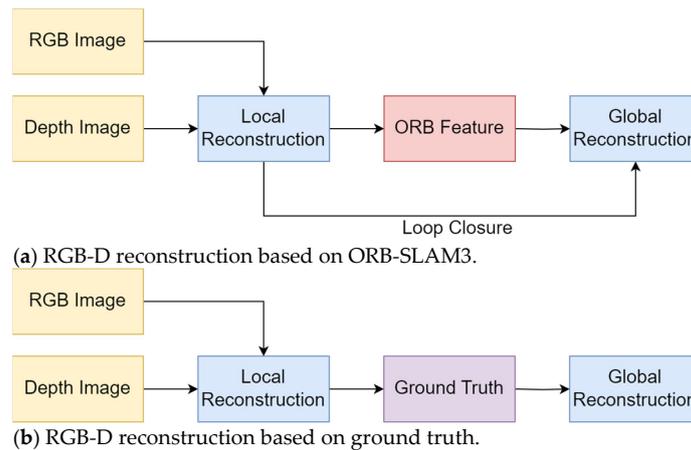


Figure 3. Two types of RGB-D reconstruction processing methods are referred to in this paper. (a) Based on ORB-SLAM3, RGB-D reconstruction with a loop closure. (b) Based on ground truth, RGB-D reconstruction.

4. Comprehensive Performance Evaluation

The evaluation and comparison of the proposed SLAM algorithms can be divided into two categories: localization and mapping in an indoor environment and localization and mapping in an outdoor environment. The organization aims to compare the overall performance of SLAM algorithms in two environments. There are five outdoor datasets and five indoor datasets, including 3D LiDAR data, RGB-D images, and the corresponding benchmarks. In this paper, the trajectory paths for the self-recorded indoor dataset are about 90 m, while the outdoor dataset has a trajectory path of about 40 m. Additionally, when running SLAM results (both trajectory and mapping), programs (Ros-noetic-rviz 1.14.20 on ROS Netics and debugging messages) are employed to ensure that both SLAM methods in every running achieve loop detection and result in the best SLAM outcomes. During the recording of datasets in this paper, the mobile robot moved along the same route five times with the same starting and ending positions each time. Therefore, in Table 1, for example, I1 represents the first recording in an indoor environment, while O5 represents the fifth recording in an outdoor environment. The numbers in the descriptors only indicate the order of dataset recording in each environment. In the indoor environment, we used the third dataset (I3) for our implementation. In the outdoor environment, we used the fourth dataset (O4) for our implementation. It has been modified in the manuscript.

Table 1. ATE with SE(3) Umeyama alignment in an indoor environment (Unit: m).

		Gmapping	ORB-SLAM3	SC-LeGO-LOAM			Gmapping	ORB-SLAM3	SC-LeGO-LOAM
Dataset I1	MAX	4.91518	0.78351	2.38522	Dataset I4	MAX	1.88854	0.46741	0.56568
	MEAN	2.68788	0.48961	0.5693		MEAN	0.90588	0.27761	0.27293
	RMSE	3.05934	0.50410	0.69376		RMSE	1.09383	0.29589	0.29494
	STD	1.4611	0.11998	0.39648		STD	0.61307	0.10238	0.1118
Dataset I2	MAX	2.39814	0.65616	0.73112	Dataset I5	MAX	1.31456	0.79217	0.92859
	MEAN	1.23568	0.39849	0.35394		MEAN	0.50987	0.3197	0.4894
	RMSE	1.48806	0.40884	0.38915		RMSE	0.65464	0.36851	0.53221
	STD	0.82911	0.09143	0.16174		STD	0.41060	0.18328	0.20911
Dataset I3	MAX	4.80831	0.38631	1.07315	Average	MAX	3.06494	0.61711	1.13675
	MEAN	2.45052	0.16167	0.39211		MEAN	1.55796	0.32941	0.41554
	RMSE	2.87831	0.17358	0.44974		RMSE	1.83484	0.35018	0.47196
	STD	1.50985	0.0632	0.22025		STD	0.96475	0.11205	0.21988

4.1. Indoor Environment

The indoor datasets describe a rectangular corridor (low-texture, low-structure) with a total length of over 90 m. Due to different robot speeds, the durations of the datasets are approximately 350 to 450 s.

4.1.1. Localization

Figure 4 displays the ground truth of the robot poses obtained from one of the indoor datasets (I3). Other datasets have similar trajectories. The absolute trajectory error (ATE) and relative pose error (RPE) are utilized to evaluate SLAM algorithms [26]. ATE measures the difference between the actual and estimated values of camera poses, directly showing the advantages and disadvantages of the algorithm. Relative pose error is a measure of the difference between the estimated pose and the ground truth pose at each point in time along the trajectory. In addition, SE(3) transformation parameters are calculated using the Umeyama algorithm [27] to align the two values.

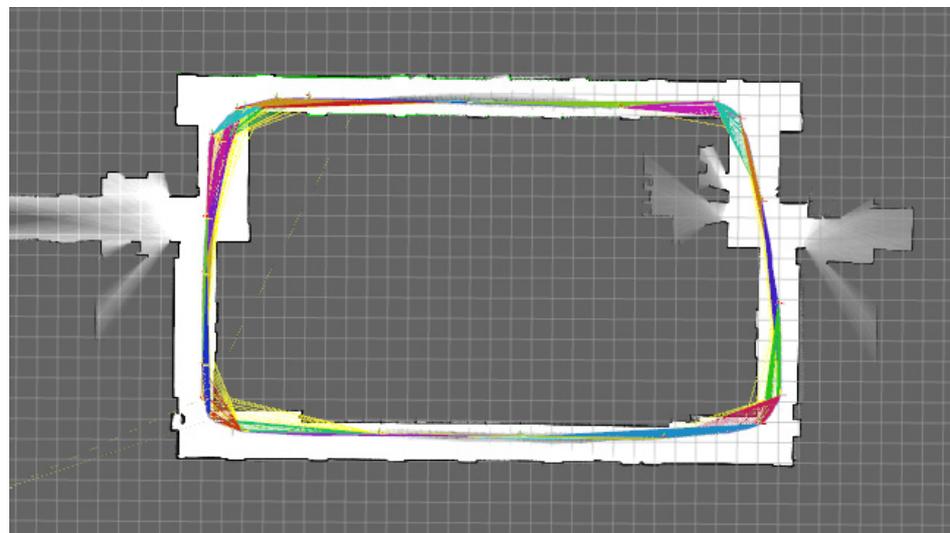


Figure 4. The ground truth of the robot poses obtained from one of the indoor datasets (I3).

Four metrics are chosen to assess the algorithm, comprising the maximum value (MAX), mean value (MEAN), root mean square error (RMSE), and standard deviation (STD).

The RMSE can quantify the accuracy of the constructed trajectory through the following mathematical expression

$$RMSE = \sqrt{\frac{\sum_{t=1}^n (X_t - \hat{X}_t)^2}{n}}, \quad (7)$$

where \hat{X}_t represents the ground truth of the localization of the mobile robot, X_t is the measured value, n is the total number of positions, and t stands for the serial number.

The STD reflects the degree of stability of the build trajectory error and the mathematical expression is

$$\sigma = \sqrt{\frac{\sum_{t=1}^n (X_t - \bar{X}_t)^2}{n-1}}, \quad (8)$$

where \bar{X}_t is the mean value.

Additionally, Gmapping serves as an extra reference. However, given that Gmapping is a 2D LiDAR SLAM algorithm utilizing odometry as an additional input, its involvement is limited to the evaluation of localization.

The results of the metrics ATE (with SE(3) Umeyama alignment) are shown in Table 1. It is obvious that in indoor environments, ORB-SLAM3 has smaller errors, highlighted in bold font, and outperforms other methods. Moreover, ORB-SLAM3 also demonstrates smaller standard deviation, indicating higher stability for localization. Due to the smooth surface of the indoor floor, the robot is susceptible to slipping, leading to inaccurate mileage counting values. Consequently, this issue adversely impacts the precision of the Gmapping algorithm.

As illustrated in Figure 5, we conducted a comparison of RPE in an indoor environment (I3), unveiling the discernible impact of loop closure on ORB-SLAM3 and SC-LeGO-LOAM, respectively. Analysis of the four sub-figures allows us to draw the following conclusions. (1) In indoor environments, loop closure detection effectively corrects the results of the two SLAM algorithms, enhancing their accuracy and robustness. (2) The RPE of Visual SLAM is positively correlated with the robot's speed and changes in the field of view, with the most pronounced performance observed in Figure 5a. The local maximum values occur at corners, while the RPE approaches zero in the initial and final static states. (3) Conversely, the correlation of LiDAR SLAM's RPE is relatively inconspicuous. We attribute this to the LiDAR's 360-degree horizontal range, which scans the operator behind it, introducing a different dynamic.

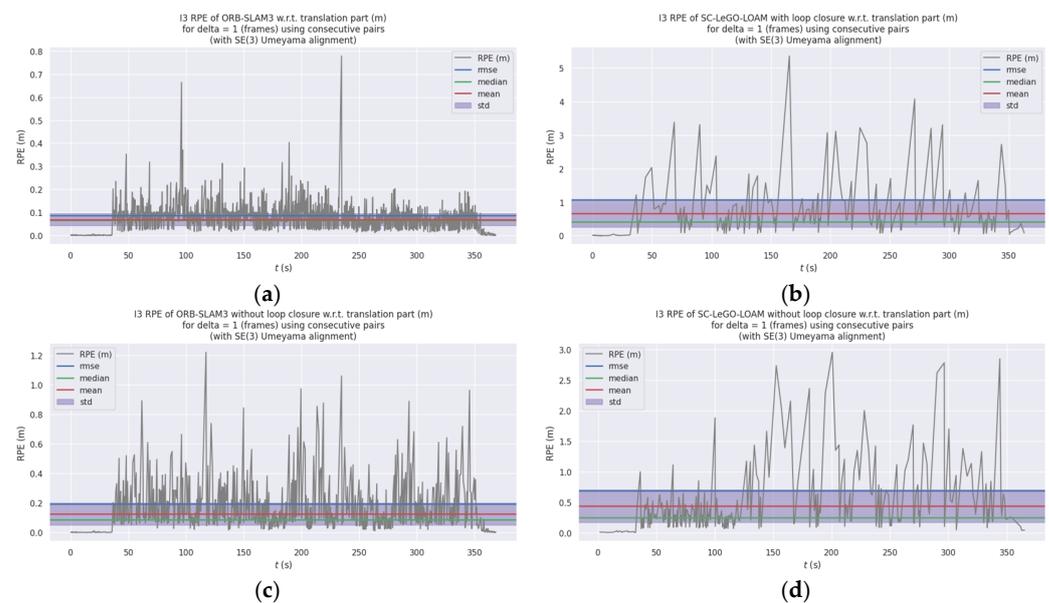


Figure 5. I3 RPE with regard to the translation part (m) for $\delta = 1$ (frames) using consecutive pairs (with SE(3) Umeyama alignment, while processing panning rotation and scale): (a) ORB-SLAM3 with loop closure; (b) SC-LeGO-LOAM with loop closure; (c) ORB-SLAM3 without loop closure; and (d) SC-LeGO-LOAM without loop closure.

4.1.2. Mapping

1. LiDAR SLAM mapping analysis

The robot trajectory is illustrated by the colored line is shown in Figure 6, which presented from a top-view perspective of the LiDAR SLAM mapping result, the yellow point cloud denotes the lowest elevation corresponding to the floor, while the red point cloud represents the highest elevation corresponding to the ceiling. The central point cloud in the map captures reflections from exterior high-rise objects encountered when traversing corridor windows. After completing a circuit around the building, it is observed that the 3D reconstructed map of LiDAR SLAM presents the architectural structure of the indoor environment in a comprehensive geometrically precise manner. However, when viewed locally, the map exhibits sparser details and there is a lack of discernibility on planar features.

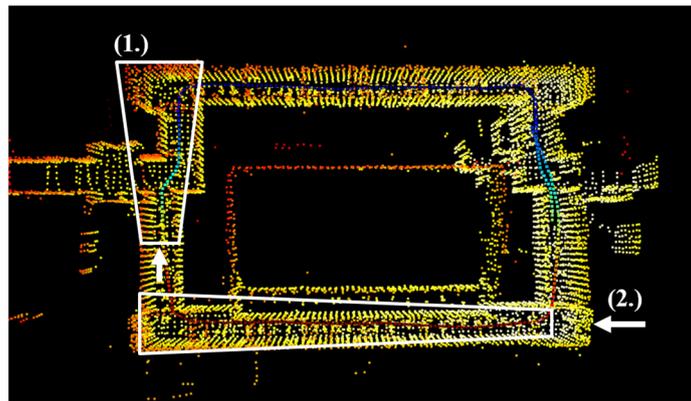


Figure 6. A top-view perspective of the LiDAR SLAM mapping result. The colored point clouds depict the robot's poses. The white boxes represent specific scenarios: (1.) Multiple corners. (2.) Extended corridor.

2. Visual SLAM based on two mapping methods

Figure 7 depicts local reconstructed maps based on the ground truth or ORB-SLAM3 within a depth range of 7 m, with the reconstruction areas and perspectives corresponding to the white field of view (FOV) trapezoids and arrows labeled (1) and (2) in Figure 6.

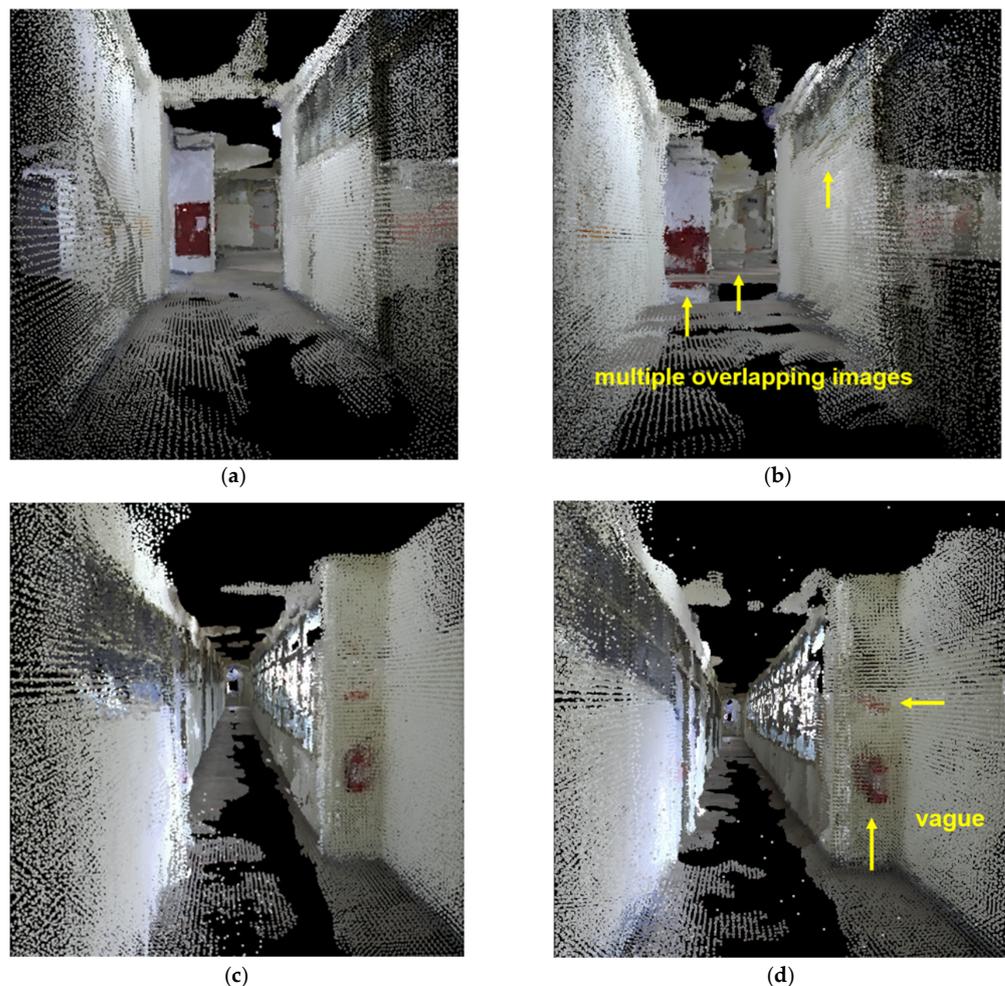


Figure 7. Local reconstructed indoor maps within a depth range of 7 m. (a) The first perspective (1) based on ground truth; (b) The first perspective (1) based on ORB-SLAM3; (c) The second perspective (2) based on ground truth; (d) The second perspective (2) based on ORB-SLAM3.

Firstly, a comparison of the 3D reconstruction effects between the two Visual SLAM mapping approaches is performed. In the first perspective, Figure 7a,b, the reconstruction from ORB-SLAM3, exhibits multiple image overlap phenomena on the fire hydrant on the wall, the door in the background, and the window frame on the right wall. This results in distorted and shifted object shapes. In contrast, the reconstruction based on the ground truth appears more detailed, continuous, and visually superior, closely aligning with the actual scene.

Moving to the second perspective within the indoor environment, Figure 7c,d, the fire extinguisher on the wall and its signage similarly appear more blurred in the reconstruction from ORB-SLAM3, consistent with the issues observed in the previous location. This problem arises from the differences in data sources and algorithms between the ORB-SLAM3 and Cartographer models. ORB-SLAM3 calculates camera poses based on feature point matching for each image, leading to cumulative errors in the z-axis direction due to robot vibrations during movement. This results in discontinuous height differences in the overlay of each frame during mapping. In contrast, the Cartographer model utilizes 3D LiDAR data with wheel odometry for calculation, imposing fixed constraints on the pose in the z-axis direction. This ensures consistency in height along the z-axis in the ground truth, resulting in a smoother overlay of each frame during mapping.

3. Comparison between LiDAR SLAM and Visual SLAM

In this section, we compare Visual SLAM and LiDAR SLAM mapping based on three aspects.

The first aspect to be discussed is the ability to identify items. In Figure 7, the map constructed by Visual SLAM displays more detailed and rich surface features. For example, the fire hydrant on the wall surface and the colors of surrounding objects are reconstructed, enhancing map identifiability and closely resembling the real scene. In contrast, the LiDAR SLAM reconstruction only outlines the structural aspects of the environmental architecture. The scanning characteristics of LiDAR result in uniform point clouds on the same plane of corridor walls, making it challenging to distinguish features and positions of individual objects in finer detail. As a result, the visual discernibility is comparatively lower in LiDAR SLAM.

The second aspect is deficiencies caused by indoor lighting sources. Defects, such as a lack of point cloud, are observable in the reconstruction based on Visual SLAM on both the floor and ceiling. This is attributed to the installation of fluorescent lights on the ceiling, causing reflections on the floor that introduce errors in depth measurements based on the binocular disparity of Zed2. These errors are subsequently mitigated through filtering during the mapping process. Conversely, the scanning nature of 3D LiDAR in LiDAR SLAM is unaffected by indoor lighting sources and it does not experience the same issue.

The third aspect is the blind spot of local maps. Due to the limited FOV inherent in the camera's imaging principle, certain parts of the scene may not be captured, such as corners during turns at intersections. This issue results in the absence of objects around corners, as depicted in Figure 8. However, in LiDAR SLAM, the 3D LiDAR used in this experiment provides a 360-degree scan of the environment, eliminating blind spots in the LiDAR reconstruction results.

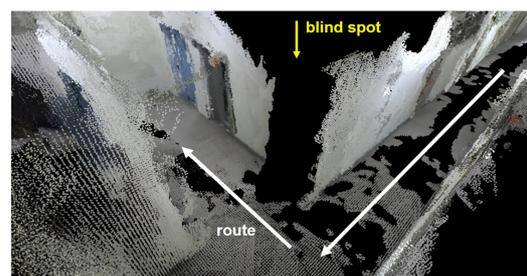


Figure 8. Blind spots in local mapping around corners.

4.2. Outdoor Environment

The outdoor datasets describe a rectangular pathway over the yard with a total length of over 40 m. The durations of the datasets are approximately 200 s with a stable robot speed.

4.2.1. Localization

Figure 9 displays the ground truth of the robot poses obtained from one of the outdoor datasets (O4). Other datasets start at slightly different locations but have similar journeys. The results of the metrics absolute trajectory error (with SE(3) Umeyama alignment) are shown in Table 2. In outdoor environments, SC-LeGO-LOAM demonstrates better positioning accuracy, namely lower errors highlighted in bold font, particularly in scenarios with richer textures compared to indoor environments. Moreover, SC-LeGO-LOAM also demonstrates smaller standard deviation, indicating higher stability for localization.

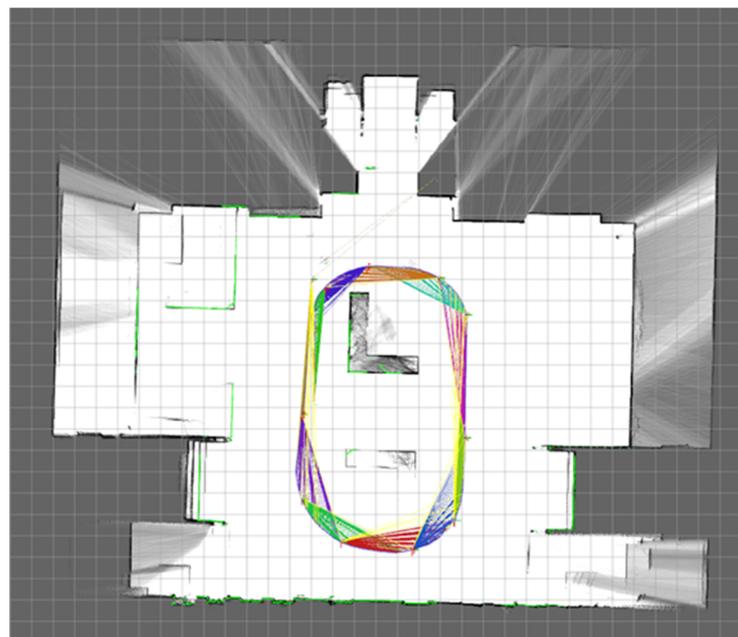


Figure 9. The ground truth of the robot poses was obtained from one of the outdoor datasets (O4).

Table 2. ATE with SE(3) Umeyama alignment in an outdoor environment (Unit: m).

		Gmapping	ORB-SLAM3	SC-LeGO-LOAM			Gmapping	ORB-SLAM3	SC-LeGO-LOAM
Dataset O1	MAX	0.18999	0.62861	0.16018	Dataset O4	MAX	0.27816	1.87328	0.16234
	MEAN	0.13034	0.36844	0.08666		MEAN	0.131768	0.34227	0.05432
	RMSE	0.13735	0.39105	0.09413		RMSE	0.14738	0.38218	0.06009
	STD	0.04333	0.13104	0.03674		STD	0.06602	0.17004	0.02568
Dataset O2	MAX	0.19983	0.62096	0.16714	Dataset O5	MAX	0.23663	0.66725	0.17237
	MEAN	0.10030	0.30119	0.06705		MEAN	0.10678	0.34068	0.06945
	RMSE	0.11021	0.32724	0.07239		RMSE	0.11844	0.36635	0.07693
	STD	0.04567	0.12794	0.02728		STD	0.05125	0.13473	0.03308
Dataset O3	MAX	0.20131	0.78556	0.19174	Average	MAX	0.22118	0.91513	0.17075
	MEAN	0.09545	0.31753	0.07116		MEAN	0.11292	0.33402	0.06972
	RMSE	0.10189	0.34955	0.0776		RMSE	0.12306	0.36327	0.07622
	STD	0.03564	0.14614	0.03095		STD	0.04839	0.14198	0.03074

As shown in Figure 10, we conduct a comparison between RPE in an outdoor environment (O4), revealing the discernible impact of loopback on ORB-SLAM3 and SC-LeGO-LOAM, respectively. This further underscores the beneficial impact of loop closure detection in error elimination. However, unlike the indoor environment discussed in the preceding section, the outdoor environment possesses richer texture information. Consequently, the role of loop closure in LiDAR SLAM is mitigated and the operator’s influence on the SLAM results is not as substantial.

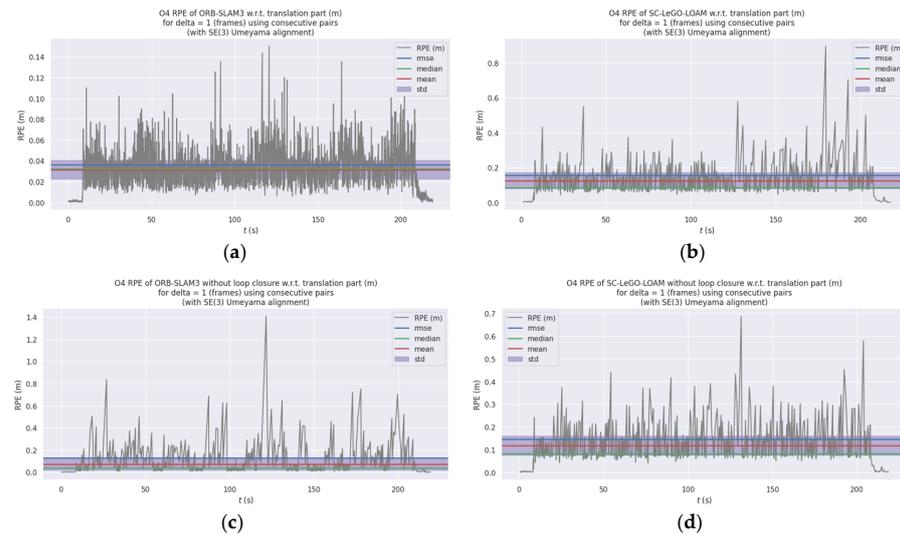


Figure 10. O4 RPE with regard to the translation part (m) for $\delta = 1$ (frames) using consecutive pairs (with SE(3) Umeyama alignment, while processing panning rotation and scale): (a) ORB-SLAM3 with loop closure; (b) SC-LeGO-LOAM with loop closure; (c) ORB-SLAM3 without loop closure; and (d) SC-LeGO-LOAM without loop closure.

4.2.2. Mapping

1. LiDAR SLAM mapping analysis

Figure 11 shows LiDAR SLAM mapping in different views pairs and the real scene and the robot trajectory is represented as colored lines. Figure 11a,b are presented as top and side views of the reconstructed map, respectively. Figure 11c provides the real scene of the outdoor environment. By circling around the central platform during data collection, it is evident from Figure 11a,b that the 3D LiDAR reconstruction faithfully captures the complex architectural structures surrounding the platform, including every beam, column, and the stairs on either side.

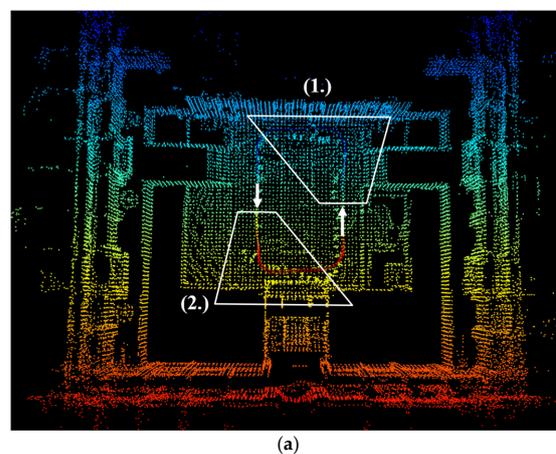


Figure 11. Cont.

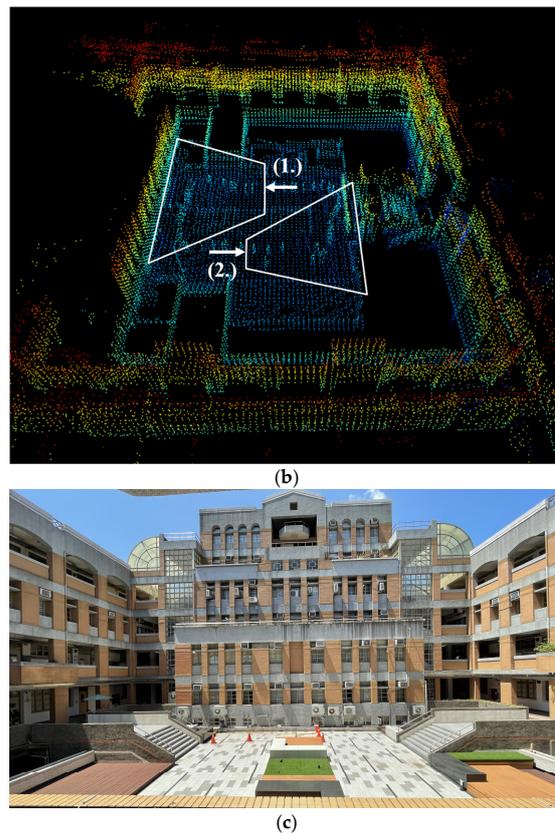


Figure 11. LiDAR SLAM outdoor mapping and the real scene. (a) Top view of the reconstructed map; (b) Side view of the reconstructed map; (c) The real scene.

2. Visual SLAM based on two mapping methods

Figure 12 depicts local reconstructed maps based on the ground truth or ORB-SLAM3 within a depth range of 7 m, with the reconstruction areas and perspectives corresponding to the white FOV trapezoids and arrows labeled (1) and (2) in Figure 11.

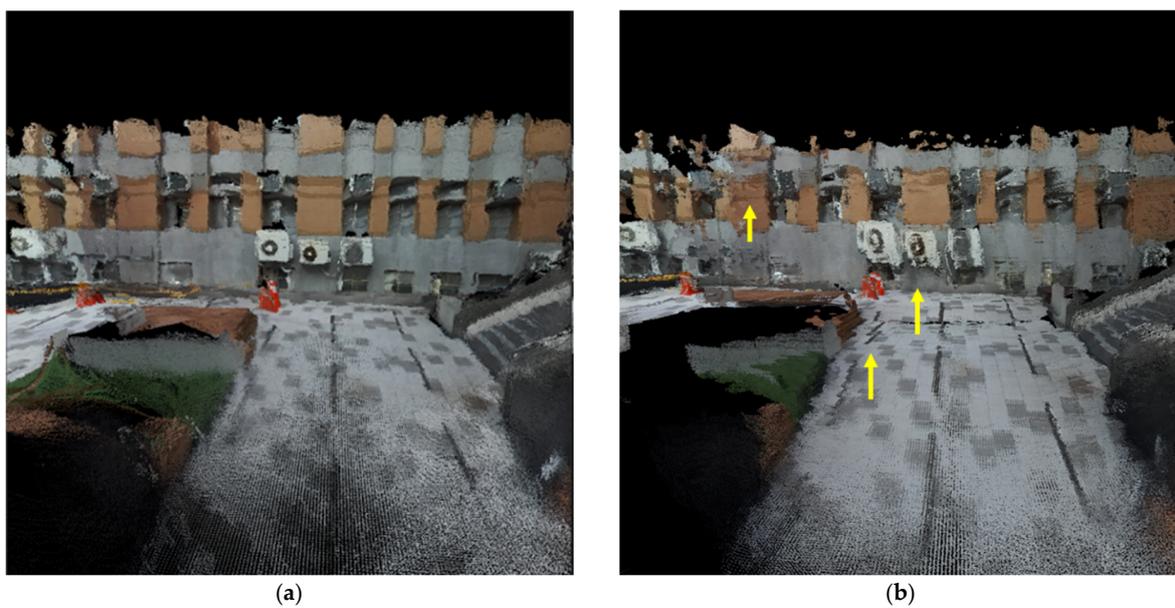


Figure 12. *Cont.*

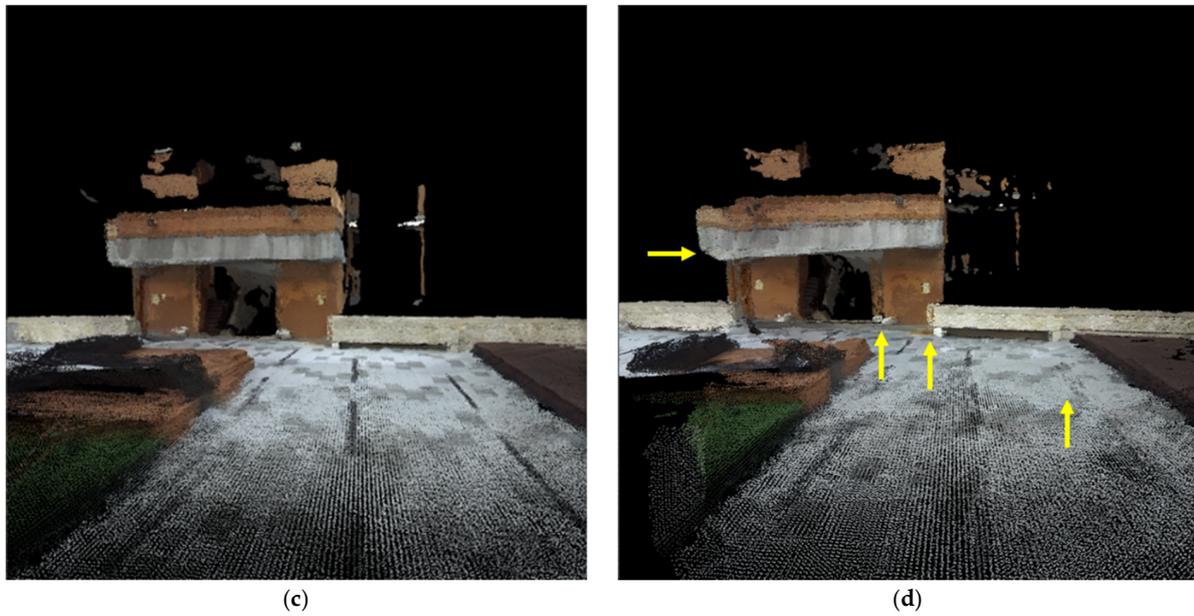


Figure 12. Local reconstructed outdoor maps within a depth range of 7 m. (a) The first perspective (1) based on ground truth; (b) The first perspective (1) based on ORB-SLAM3; (c) The second perspective (2) based on ground truth; (d) The second perspective (2) based on ORB-SLAM3.

In the first perspective, the reconstruction from ORB-SLAM3, Figure 12b, exhibits phenomena of multiple image overlaps on the beams of the building, outdoor air conditioning units, and patterns on the floor. This results in a blurry appearance of objects and the floor patterns appear distorted as a consequence. However, the reconstruction based on ground truth, Figure 12a, while showing slight distortion in some subtle areas, maintains an overall finer and more continuous effect.

Moving to the second perspective, Figure 12c,d, similar issues are observed in the reconstruction from ORB-SLAM3, including blurred floor patterns and double-image displacement in the architectural structure. The reason for these problems is consistent with the indoor conditions discussed in the previous section and will not be reiterated here.

3. Comparison between LiDAR SLAM and Visual SLAM

In this section, a comparison between Visual SLAM and LiDAR SLAM is conducted. By examining the Visual SLAM-based reconstruction map, Figure 13a, and the LiDAR SLAM-based reconstruction map, Figure 13b, we discuss the following three points sequentially.

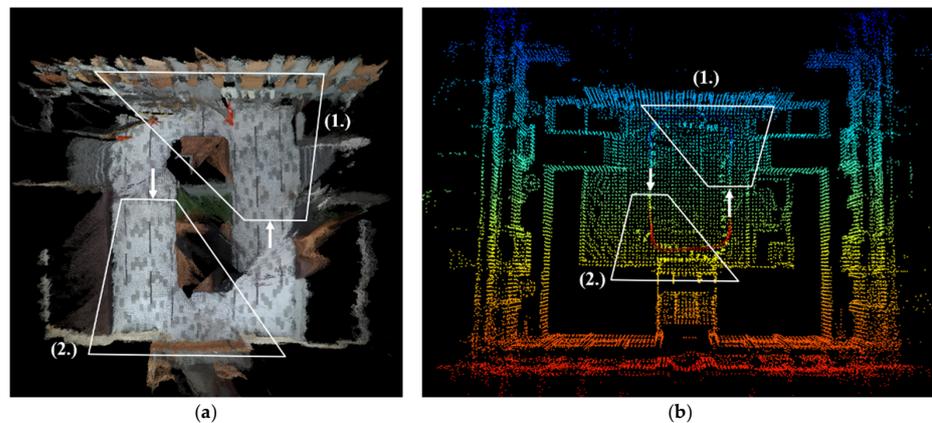


Figure 13. (a) Visual SLAM based on ground truth and (b) LiDAR SLAM based on 3D LiDAR. The colored point clouds depict the robot’s poses. The white boxes represent specific scenarios: (1.) Reconstruction with loop. (2.) Reconstruction without loop.

The first aspect is the difference in the size of the reconstructed area. From the results of both mapping approaches, under the same robot trajectory, the global map reconstructed by Visual SLAM covers only a portion of the LiDAR SLAM global map. This difference arises from the distinct maximum measurement distances provided by the two sensors. In hardware specifications, the optimal measurement depth for the camera is approximately 20 m and exceeding this range introduces larger errors. Therefore, in Visual SLAM reconstruction, the mapping range is set to pixels within a depth of 7 m to ensure more accurate mapping. In contrast, the 3D LiDAR used in this experiment has a maximum scanning range of up to 150 m, resulting in a broader map coverage by LiDAR SLAM while maintaining precision in distant areas.

The second aspect is object recognizability. In outdoor scenes, while LiDAR SLAM provides a more comprehensive environmental structure and outline, it lacks the capability to recognize local objects and features. This limitation, similar to that in an indoor environment, arises from the scanning nature of LiDAR, where point clouds on the same plane exhibit no height differences, making it challenging to distinguish features between each point cloud. Consequently, LiDAR SLAM cannot recognize objects as effectively as Visual SLAM, which utilizes complete color and feature shapes for object identification.

The third aspect is about defects and blind spots in local maps. In the mapping analysis of Visual SLAM, we observed the issue of multiple image overlaps caused by vibrations. Additionally, as in indoor mapping, the limited field of view (FOV) results in some local environments that cannot be captured, creating blind spots in the mapping results. However, in LiDAR SLAM, the blind spot issues are also eliminated by the usage of a 360-degree 3D LiDAR for the reconstruction. Moreover, the SC-LeGO-LOAM algorithm used in LiDAR SLAM leverages feature set matching between timestamps for pose estimation and its detection range is broader. Therefore, the impact of vibrations on mapping effectiveness is milder in LiDAR SLAM, with no apparent ghosting phenomena.

4.3. Computational Resource Analysis

To quantitatively demonstrate the computational resource requirements of the two SLAM methods, we simultaneously record the CPU resource allocation during the map creation process. In this study, we run 10 datasets for the nondeterministic nature of the system and show average results for the CPU utilization under various types of tasks and memory usage increasing ratio in Table 3. "User" is responsible for time spent by normal processes executing in user mod [28]. "System" shows time spent by executing processes in kernel mode. And "Idle" is the portion of CPU time during which the processor is inactive. In addition to these three main states, there are other types of CPU usage; therefore, these three types will not add up to 100%. Moreover, "Memory" is the temporary storage for active processes. Finally, the increasing ratio is computed by subtracting the normal memory usage occupancy rate from the active occupancy rate and dividing the result by the normal memory usage occupancy rate.

According to Table 3, we conducted comparisons from three different perspectives and the results are presented in Table 4. Firstly, we compare the CPU utilization statistics of the two SLAM methods indoors and outdoors (Scenario 1). In indoor environments, Visual SLAM occupies 51.25% more CPU compared to LiDAR SLAM, while outdoors, it occupies 35.02% more than LiDAR SLAM. This result confirms that Visual SLAM has higher computational requirements than LiDAR SLAM. The reason behind this lies in the fact that the map constructed by Visual SLAM stores more information.

Next, we compare each CPU utilization of the two SLAM methods indoors and outdoors with the original CPU utilization without running any programs (Scenario 2). The CPU utilization of Visual SLAM during the execution of indoor and outdoor datasets is 77.80% and 70.67% higher compared to the original usage, respectively. On the other hand, for LiDAR SLAM, it is 17.55% and 26.41% higher than the original usage, respectively. The relatively minor increase in indoor CPU utilization may be due to the simpler structure of indoor walls, resulting in a lower point cloud volume and lower density for mapping.

Table 3. Average results for the CPU utilization under various types of task and memory usage increasing ratio * (Unit: %).

	CPU Utilization			Memory Usage	
	User	Sys	Idle	Occupancy Rate	Increasing Ratio
Org	14.93	3.03	73.08	27.20	NA
ORB_I	26.54	3.13	60.83	34.20	22.18
ORB_O	25.48	3.20	61.80	33.69	12.12
SC_I	17.55	3.08	70.25	31.49	2.40
SC_O	18.87	3.32	68.45	30.42	2.68

* Note that the notation ‘Org’ denotes the average CPU usage in User without SLAM algorithm running. ‘ORB_I’, ‘ORB_O’, ‘SC_I’, and ‘SC_O’ mean the average amount of CPU time spent in user mode indoors using the ORB algorithm, outdoors using the ORB algorithm, indoors using SC-the LeGO-LOAM algorithm, and outdoors using SC-LeGO-LOAM algorithm, respectively.

Table 4. Three comparisons among different perspectives * (Unit: %).

Scenario 1		Scenario 2		Scenario 3	
In Terms of CPU Usage in User		In Terms of CPU Usage in User		In Terms of Memory Increasing Ratio	
ORB_I vs. SC_I	51.25	ORB_I vs. Org	77.80	ORB_I	22.18
ORB_O vs. SC_O	35.02	ORB_O vs. Org	70.67	ORB_O	12.12
		SC_I vs. Org	17.55	SC_I	2.40
		SC_O vs. Org	26.41	SC_O	2.68

* The comparison of A to B refers to the latter, B. The notations carry the same meaning as those in Table 3.

Finally, due to the indoor dataset recording time being approximately 350 to 450 s and the outdoor dataset recording time being around 200 s, we compare the memory usage increasing ratios of the two SLAM methods indoors relative to outdoors (Scenario 3). It can be observed that the memory usage increasing ratio of Visual SLAM indoors is approximately twice that of outdoors, consistent with the doubled dataset recording time. However, in LiDAR SLAM, the memory usage increasing ratio outdoors is slightly higher than indoors, possibly due to the larger scanning range and point cloud volume in outdoor environments.

5. Conclusions

This study undertook a thorough assessment of SLAM algorithms in both indoor and outdoor environments, leveraging state-of-the-art techniques. Two pivotal SLAM algorithms, namely ORB-SLAM3 and SC-LeGO-LOAM, were utilized, along with the ground truth creation method Cartographer. Additionally, two 3D reconstruction methods employing RGB-D and LiDAR point clouds were employed. The experiments entailed the collection of real-world datasets across varied settings, with a focus on analyzing the algorithms’ efficacy in terms of localization, mapping, and computational resource demands. The ensuing findings are derived from the outcomes:

- In terms of localization performance, ORB-SLAM3 exhibits superior positioning accuracy in indoor environments, while SC-LeGO-LOAM excels in outdoor settings. This distinction arises from the nature of SLAM algorithms, which benefit from environmental intricacies, making it easier for the robot to calculate and match features;
- In terms of mapping capability, the effectiveness of mapping is closely tied to the sensor’s characteristics and each has its unique drawbacks. For instance, SLAM mapping using depth cameras may suffer from imperfections and blind spots due to indoor lighting sources and limitations on FOV, respectively. Three-dimensional LiDAR-based

mapping may lack the level of detail and textural information required for precise object recognition. Nevertheless, in accordance with the distinct mapping spatial scopes between indoor and outdoor environments, LiDAR SLAM is more suitable in an outdoor environment due to its comprehensive characteristics. In contrast, Visual SLAM is more suitable in an indoor environment due to its recognizability characteristic and ability to compensate for the sparse aspects inherent in LiDAR SLAM;

- Regarding performance evaluation, quantitative analysis of computational resource requirements indicates that Visual SLAM demands more CPU resources compared to LiDAR SLAM, attributing this to the storage of additional depth and color data. The study also explores CPU utilization and memory usage increase ratios, emphasizing the impact of environmental factors on resource demands.

As part of future work, expanding the datasets to encompass indoor and outdoor environments, more complex robotic experiment scenarios, the ability to identify objects, and longer recording times will allow for a comprehensive comparison of SLAM performance in these diverse situations. At the same time, the use of more sensors should also be considered to test the performance of more sensor fusion algorithms.

Author Contributions: Conceptualization, Y.-L.Z. and H.-P.H.; Funding acquisition, H.-P.H.; Methodology, Y.-L.Z. and Y.-T.H.; Project administration, H.-P.H.; Resources, H.-P.H.; Software, Y.-L.Z.; Supervision, H.-P.H.; Validation, Y.-L.Z. and Y.-T.H.; Visualization, Y.-L.Z. and Y.-T.H.; Writing—original draft, Y.-L.Z. and Y.-T.H.; Writing—review and editing, Y.-L.Z., Y.-T.H. and H.-P.H. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded in part by the National Science and Technology Council, Taiwan, under Grant MOST 112-2221-E-002-149-MY3.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Zang, Q.; Zhang, K.; Wang, L.; Wu, L. An Adaptive ORB-SLAM3 System for Outdoor Dynamic Environments. *Sensors* **2023**, *23*, 1359. [[CrossRef](#)] [[PubMed](#)]
2. Hong, Y.-T.; Huang, H.-P. A Comparison of Outdoor 3D Reconstruction between Visual SLAM and LiDAR SLAM. In Proceedings of the 2023 International Automatic Control Conference (CACs), Penghu, Taiwan, 26–29 October 2023; pp. 1–6. [[CrossRef](#)]
3. Campos, C.; Elvira, R.; Rodríguez, J.J.G.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM3: An Accurate Open-Source Library for Visual, Visual-Inertial, and Multimap SLAM. *IEEE Trans. Robot.* **2021**, *37*, 1874–1890. [[CrossRef](#)]
4. Xue, G.; Wei, J.; Li, R.; Cheng, J. LeGO-LOAM-SC: An Improved Simultaneous Localization and Mapping Method Fusing LeGO-LOAM and Scan Context for Underground Coalmine. *Sensors* **2022**, *22*, 520. [[CrossRef](#)] [[PubMed](#)]
5. Mur-Artal, R.; Tardós, J.D. ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras. *IEEE Trans. Robot.* **2017**, *33*, 1255–1262. [[CrossRef](#)]
6. Mur-Artal, R.; Tardós, J.D. Visual-Inertial Monocular SLAM With Map Reuse. *IEEE Robot. Autom. Lett.* **2017**, *2*, 796–803. [[CrossRef](#)]
7. Grisetti, G.; Stachniss, C.; Burgard, W. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Trans. Robot.* **2007**, *23*, 34. [[CrossRef](#)]
8. Wang, C.-C.; Thorpe, C.; Thrun, S.; Hebert, M.; Durrant-Whyte, H. Simultaneous Localization, Mapping and Moving Object Tracking. *Int. J. Robot. Res.* **2007**, *26*, 889–916. [[CrossRef](#)]
9. Bailey, T.; Nieto, J.; Guivant, J.; Stevens, M.; Nebot, E. Consistency of the EKF-SLAM Algorithm. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 3562–3568. [[CrossRef](#)]
10. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-time. *Robot. Sci. Syst.* **2014**, *2*, 9. [[CrossRef](#)]
11. Kim, G.; Kim, A. Scan Context: Egocentric Spatial Descriptor for Place Recognition Within 3D Point Cloud Map. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4802–4809. [[CrossRef](#)]

12. Izadi, S.; Kim, D.; Hilliges, O.; Molyneaux, D.; Newcombe, R.; Kohli, P.; Shotton, J.; Hodges, S.; Freeman, D.; Davison, A.; et al. KinectFusion: Real-time 3D reconstruction and interaction using a moving depth camera. In Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology, Santa Barbara, CA, USA, 16–19 October 2011; pp. 559–568. [[CrossRef](#)]
13. Yarovoi, A.; Cho, Y.K. Review of simultaneous localization and mapping (SLAM) for construction robotics applications. *Autom. Constr.* **2024**, *162*, 105344. [[CrossRef](#)]
14. Schubert, D.; Goll, T.; Demmel, N.; Usenko, V.; Stückler, J.; Cremers, D. The TUM VI Benchmark for Evaluating Visual-Inertial Odometry. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 1680–1687. [[CrossRef](#)]
15. Sturm, J.; Engelhard, N.; Endres, F.; Burgard, W.; Cremers, D. A benchmark for the evaluation of RGB-D SLAM systems. In Proceedings of the 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems, Vilamoura-Algarve, Portugal, 7–12 October 2012; pp. 573–580. [[CrossRef](#)]
16. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The KITTI dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [[CrossRef](#)]
17. Langley, R.B. RTK GPS. *GPS World* **1998**, *9*, 70–76.
18. Ye, W.; Zhao, Y.; Vela, P. Characterizing SLAM Benchmarks and Methods for the Robust Perception Age. *arXiv* **2019**, arXiv:1905.07808.
19. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. SVO: Semidirect Visual Odometry for Monocular and Multicamera Systems. *IEEE Trans. Robot.* **2017**, *33*, 249–265. [[CrossRef](#)]
20. Mur-Artal, R.; Montiel, J.M.M.; Tardós, J.D. ORB-SLAM: A Versatile and Accurate Monocular SLAM System. *IEEE Trans. Robot.* **2015**, *31*, 1147–1163. [[CrossRef](#)]
21. Shan, T.; Englot, B. LeGO-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In Proceedings of the 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018; pp. 4758–4765. [[CrossRef](#)]
22. Segal, A.; Haehnel, D.; Thrun, S. Generalized-ICP. *Robot. Sci. Syst.* **2009**, *2*, 435. [[CrossRef](#)]
23. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-Time Loop Closure in 2D LIDAR SLAM. In Proceedings of the 2016 IEEE International Conference on Robotics and Automation (ICRA), Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278. [[CrossRef](#)]
24. Agarwal, S.; Mierle, K. Ceres Solver: Tutorial & Reference. Available online: <http://ceres-solver.org/> (accessed on 1 December 2023).
25. Zhao, Y.L.; Huang, H.P.; Chen, T.L.; Chiang, P.C.; Chen, Y.H.; Yeh, J.H.; Huang, C.H.; Lin, J.F.; Weng, W.T. A Smart Sterilization Robot System With Chlorine Dioxide for Spray Disinfection. *IEEE Sens. J.* **2021**, *21*, 22047–22057. [[CrossRef](#)]
26. Grupp, M. EVO: Python Package for the Evaluation of Odometry and SLAM. Available online: <https://github.com/MichaelGrupp/evo> (accessed on 1 December 2023).
27. Umeyama, S. Least-squares estimation of transformation parameters between two point patterns. *IEEE Trans. Pattern Anal. Mach. Intell.* **1991**, *13*, 376–380. [[CrossRef](#)]
28. Rodola, G. psutil on pypi.org. Available online: <https://pypi.org/project/psutil> (accessed on 1 December 2023).

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.