electronics

MDPI

Article

Tightly Coupled 3D Lidar Inertial SLAM for Ground Robot

Daosheng Li [1], Bo Sun [2], Ruyu Liu [2] and Ruilei Xue [1,*]

[1] Mechanical Engineering, Xinjiang University, Urumqi 830002, China
[2] Haixi Institutes, Chinese Academy of Sciences Quanzhou Institute of Equipment Manufacturing, Jinjiang 362203, China
* Correspondence: 107552103801@stu.xju.edu.cn

Abstract: This paper proposes a robotic state estimation and map construction method. The traditional lidar SLAM methods are affected by sensor measurement noise, which causes the estimated trajectory to drift, especially along the altitude direction caused by lidar noise. In this paper, ground parameters in the environment are extracted to construct the ground factors to compress the trajectory estimation drifting along the altitude direction using the characteristics of constant robot pose relative to the ground. Our method uses tightly coupled lidar and inertial to obtain low-drift lidar odometry factors by factor graph optimization. The optimized lidar odometry factors are then added to a global factor graph, together with ground, loop closure, and GPS factors to obtain accurate robot state estimation and mapping after factor graph optimization. The experimental results show that our method has comparable results with advanced lidar SLAM methods, and even performs better in some complex and large-scale environments.

Keywords: tightly coupled; SLAM; factor graph

1. Introduction

Accurate self-motion estimation and mapping are prerequisites for mobile robots to achieve autonomous navigation [1,2]. Simultaneous localization and mapping (SLAM) technology usually uses cameras and lidar to sense the surrounding environment, including estimating the robot's trajectory and constructing a map describing the surrounding environment. Although visual SLAM [3,4] is easier to scene understanding and position recognition, cameras are limited by light variation and Field Of View (FOV). Lidar-based SLAM approaches avoid the disadvantages of cameras, while providing more reliable measurements. Specifically, as an active sensor, 3D lidar is virtually unaffected by lighting and can sense horizontal 360 degree FOV with a frequency of about 10 Hz. Therefore, this paper uses 3D lidar to realize real-time, six Degree-Of-Freedom (DOF) robotic state estimation and mapping.

Typical lidar SLAM uses the iterative closest point algorithm (ICP) [5,6] to align two frame point clouds with estimating lidar motion. Errors are accumulated during the recursive estimation of trajectory. Therefore, a pose graph [7] with nodes representing the robot states and edges representing the relative relationship between nodes is established to reduce cumulative errors. These edges are measurements from different sensors, such as lidar, Inertial Measurement Unit (IMU), and global positioning system (GPS), or construct loop closure edges [8] obtained by identifying the revisit position. Lastly, the nonlinear least square optimization algorithm is used to optimize the graph to reduce estimated trajectory errors.

For convenience, the noise of lidar measurement has long been modeled as a zero-mean Gaussian distribution. However, recent research [9] has found that the maximum error of lidar measurement at a high incidence angle is close to 20 cm. This measurement error in many ground robot working environments will cause the pose estimation to drift

check for updates

Citation: Li, D.; Sun, B.; Liu, R.; Xue, R. Tightly Coupled 3D Lidar Inertial SLAM for Ground Robot. *Electronics* **2023**, *12*, 1649. https://doi.org/10.3390/electronics12071649

Academic Editor: Janos Botzheim

Received: 24 February 2023
Revised: 22 March 2023
Accepted: 29 March 2023
Published: 31 March 2023

Copyright: © 2023 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/).

along the altitude direction, affecting localization accuracy and map quality. Some well-known methods use ground information to limit the robot's drift in the altitude direction. However, most of these are applied in an indoor planar or multi-floor environment, which is challenging to adapt to an outdoor environment. In this paper, we propose a robot state estimation method based on the factor graph [10,11]. Compared with a graph, the factor graph is very convenient for adding multi-modality measurements for smoothing and incremental trajectory estimation. The proposed method uses tightly coupled lidar and inertial to obtain low-drift lidar odometry, which is obtained by optimizing IMU preintegration factor and lidar odometry factor by the factor graph. Then, we compress the cumulative trajectory estimation error by ground, loop closure, and GPS constraint factors after factor graph optimization. In our method, the loop closure and GPS factors can effectively reduce the cumulative error as the scale of the environment increases. In particular, we use low-drift lidar odometry to construct a global ground and construct the ground factors based on the characteristic that the robot's pose is invariant to the ground. Since the ground factors are relative to the global ground rather than a specific plane, it can compress the robot's trajectory estimation drifting along the altitude direction caused by lidar noise, even in outdoor non-planar environments. After factor graph optimization, the points of each keyframe are saved to construct a point cloud map. Our contributions are summarized as follows:

- We present a lidar inertial tightly coupled SLAM method for robot 6DOF state estimation, which uses the factor graph to fuse multi-modality measurements from lidar, IMU, and GPS.
- We use ground factors to constrain the robot's trajectory estimation drifting along the altitude direction in outdoor non-planar environments.
- The proposed method has been tested extensively on public datasets and in real-world environments. Compared with popular lidar SLAM methods, our accuracy is improved.

2. Related Work

Over the past two decades, much research has been conducted on improving the accuracy of 3D lidar SLAM. IMLS-SLAM [12] only uses one 3D lidar sensor to complete a robot's trajectory estimation by registering the lidar point clouds. Since it is difficult for a single lidar method to correct the skew points caused by self-motion on lidar measurement and accurately estimate self-motion in some lidar degradation environments, lidar is often combined with IMU for state estimation and mapping. Depending on the degree of coupling, it can be divided into two types: loose coupling and tight coupling. In LOAM [13], IMU measurement is used to remove lidar point cloud distortion and as a priori of lidar odometry. LOAM is loosely coupled because IMU measurement is not used to further optimize odometry. More popular loose coupling methods [14–16] use an extended Kalman filter to fuse IMU, lidar, and extra GPS for robot state estimation. The loosely coupled method is simpler but less accurate than the tightly coupled method. In LIO-MAPPING [17], the lidar and IMU are tightly coupled to achieve accurate state estimation by minimizing the common costs. The tightly coupled framework allows it to function stably in the case of lidar degradation or rapid rotation. The LIO-SAM [18] uses the factor graph to add the measurements from lidar, IMU, and GPS to complete the smooth and incremental robot trajectory estimation through factor graph optimization, which is an efficient and tightly coupled method.

Lidar inertial tightly coupled methods provide accurate state estimation in a short time but still produce unacceptable errors over time, especially in the altitude direction. GPS can provide absolute measurements to limit position estimation error. However, the error of GPS measurements in the altitude direction is much larger than that of the other two directions, and it is difficult to eliminate the cumulative error of the trajectory when the GPS signal is weak. Some lidar SLAM methods use ground information to constrain the error of the robot state estimation in the altitude direction. For example, Zheng et al. [19,20]

propose fusing ground information with the pose estimation process so that the ground altitude is constrained by a two-dimensional plane. It does not consider constraining the robot pose in the space of six-DOF. Koide et al. [21] assume that the robot always moves along the same plane, limiting altitude changes in the altitude direction through ground detection. However, this is suitable indoors but not in outdoor environments. Geneva et al. [22] extract the plane in the three-axis direction of the environment instead of just the ground, use the closest point (CP) parameter to represent planes, and add the plane constraints into the graph optimization framework to limit the robot pose in 3D space. Wei et al. [23] and Zhang et al. [24] judge that the current ground of the robot has changed significantly from the last moment; therefore, they construct a non-planar global ground that uses the ground to limit the pose drift of the robot in a multi-floor environment.

Inspired by the above works, this paper proposes a lidar inertial tightly coupled robot state estimation method. In addition to using loop closure and GPS factors to constrain cumulative trajectory estimation errors, the method also uses ground factors to compress trajectory estimation drifting along the altitude direction.

3. Tightly Coupled Lidar Inertial Odometry

3.1. System Overview

To first define the symbolic descriptions that appear throughout this paper, we use W for the world coordinate frame, I for the IMU coordinate frame, and the robot state X in the world coordinate system as the following:

$$X = [R^T, p^T, v^T, b^T]^T \tag{1}$$

where $R \in SO(3)$ is the rotation matrix, $p \in R^3$ is the position vector, v is the velocity, and b is the IMU bias.

As a maximum a posteriori (MAP) problem, the state estimation problem can be conveniently solved using the factor graph. Our method flow is shown in Figure 1, which indicates that the inputs are three different types of sensor and the output is a point cloud map. The lidar odometry factors and the IMU preintegration factors from lidar and IMU are optimized by the factor graph to obtain smoother lidar odometry factors. Then, the optimized lidar odometry factors are added to a global factor graph with the ground, GPS, and loop closure factors. Finally, the pose of keyframes is adjusted through factor graph optimization to obtain an accurate trajectory estimation, and the point cloud of each keyframe is saved as the point cloud map. The generation of each constraint factor is described in detail below.

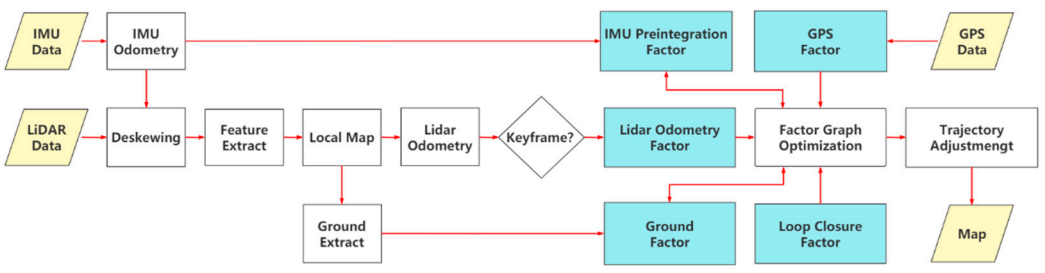


Figure 1. System overview of our framework. The system consists of three inputs and one output. The multi-modality measurements from lidar, IMU, and GPS are fused by the factor graph and adjust the trajectory estimation to output point cloud maps after factor graph optimization.

3.2. IMU Preintegration Factor

The initial angular velocity and acceleration measured by IMU in I at time t are defined as Equations (2) and (3):

$$\hat{\omega}_t = \omega_t + b_t^{\omega} + n_t^{\omega} \tag{2}$$

$$\hat{a}_t = R_{Wt}^{I}(a_t - g) + b_t^a + n_t^a \tag{3}$$

where b_t is varying bias and n_t is white noise. R_W^I is the rotation matrix from frame W to frame I. g is the gravity vector. The velocity, position, and rotation of the robot measured using the IMU at time $t + \Delta t$ can be given by Equations (4)–(6):

$$v_{t+\Delta t} = v_t + g\Delta t + R_I^W{}_t(\hat{a}_t - b_t^a - n_t^a)\Delta t \tag{4}$$

$$p_{t+\Delta t} = p_t + v_t\Delta t + \frac{1}{2}g\Delta t^2 + \frac{1}{2}R_I^W{}_t(\hat{a}_t - b_t^a - n_t^a)\Delta t^2 \tag{5}$$

$$R_{t+\Delta t} = R_I^W{}_t \exp((\hat{\omega}_t - b_t^\omega - n_t^\omega)\Delta t) \tag{6}$$

The IMU preintegration factor represents the relative motion estimation between two timesteps. The IMU preintegration measurements, Δv_{ij}, Δp_{ij}, and ΔR_{ij} between time I and time j, can be calculated using the following equations:

$$\Delta v_{ij} = R_t{}^T(v_j - v_i - g\Delta t_{ij}) \tag{7}$$

$$\Delta p_{ij} = R_i{}^T(p_j - p_i - v_i\Delta t_{ij} - \frac{1}{2}g\Delta {t_{ij}}^2) \tag{8}$$

$$\Delta R_{ij} = R_i{}^T R_j \tag{9}$$

3.3. Lidar Odometry Factor

Unnecessary pressure is placed on the real-time performance of the system by optimizing the robot's pose with the intrinsic frequency of the lidar. This paper selects a lidar frame with pose changes of more than 0.8 m or 0.15 rad as a keyframe and outputs the optimized keyframe poses as the robot's trajectory. The robot's state X of a keyframe is added to the factor graph as variable nodes, while the normal frames only participate in the lidar odometry calculation and do not join optimization. Because the points in a single lidar frame are not dense enough, we maintain a sliding window with a fixed number of 20 keyframes. Then, we splice keyframes to construct a local map M_t according to the pose relationship between keyframes in the sliding window to improve the accuracy of the lidar odometry by scan-to-map. Our method used the motion estimated by IMU as the initial value of lidar odometry. Then, we use optimized IMU preintegration factors and lidar odometry factors on a factor graph to obtain accurate odometry and estimate the bias of IMU to achieve the lidar inertial tight coupling. When a point cloud F_{t+1} with corrected skew points arrives, it is divided into two types of feature points: edge points F_{t+1}^e and plane points F_{t+1}^p, $F_{t+1} = \{F_{t+1}^e, F_{t+1}^p\}$, according to the curvature of points. Our method first converts F_{t+1} to the vicinity of M_t according to the motion estimated by IMU, as shown in Equation (10). Then, it calculates the distance between the corresponding feature points in F_{t+1} and M_t to obtain the lidar odometry.

$$\widetilde{p}_i = R_{t+1}^t p_i + t_{t+1}^t \tag{10}$$

where p_i is a feature point in F_{t+1}. R_{t+1}^t and t_{t+1}^t are the estimated motion of the IMU after the transformation from I to W coordinate frame. When p_i is an edge point, the distance between the corresponding relations of feature points in F_{t+1} and M_t is calculated, as shown in Equation (11):

$$d_\varepsilon = \frac{|(\widetilde{p}_i - p_b) \times (\widetilde{p}_i - p_a)|}{|p_a - p_b|} \tag{11}$$

where d_ε is the distance from the edge point in F_{t+1} to the edge point in M_t. p_i is the edge point in F_{t+1}, p_a and p_b are the edge feature points corresponding to p_i in M_t. The correspondence between feature points is detailed in reference [13]. When p_i is a plane point, the distance between the corresponding relations of plane points in F_{t+1} and M_t are calculated, as shown in Equation (12):

$$d_{\mathrm{H}} = (\widetilde{p}_i - p_j) \cdot \frac{(p_l - p_j) \times (p_m - p_j)}{\left|(p_l - p_j) \times (p_m - p_j)\right|} \tag{12}$$

where d_{H} is the distance from the plane point in F_{t+1} to the plane in M_t. p_l, p_j, and p_m are the plane points corresponding to p_i in M_t. Our method uses the Gaussian–Newton method to minimize the distance between the feature points in F_{t+1} and the corresponding feature points in M_t, as shown in Equation (13):

$$\min_{T_{t+1}} = \left\{ \sum_{\widetilde{p}_i \in F^e_{t+1}} d_\varepsilon + \sum_{\widetilde{p}_i \in F^p_{t+1}} d_{\mathrm{H}} \right\} \tag{13}$$

where T_{t+1} is the robot's state X of F_{t+1}. Finally, the lidar odometry factor between two keyframes is calculated, as shown in Equation (14):

$$\Delta T_t^{t+1} = {T_t}^T T_{t+1} \tag{14}$$

3.4. Ground Factor

It is reasonable to assume that the ground points near the robot can be modeled as an infinite plane in the form of $HF(\vec{n}, d)$ (Hesse form, *HF*), where $\vec{n}$ is the normal vector and d is the distance between the plane and the origin of the specified coordinate frame. *HF* is a hyperparametric planar description with the influence of a singular information matrix during optimization. In this paper, we use the *HF* parameter to facilitate the conversion between different grounds. $\Pi = d\vec{n}$ is a minimal representation [22], while the infinite plane of *HF* is represented by Π during the process of factor graph optimization. Our method uses the RANSAC [25] algorithm to extract the ground of the *HF* parameter in the local map to which each keyframe belongs. If we assume ground with parameter $HF(\vec{n}^a, d^a)$ corresponding to keyframe F_a, then for each point p_i^a belonging to the ground of F_a satisfies Equation (15):

$$\left(\vec{n}^a\right)^T p_i^a - d^a = 0 \tag{15}$$

if the transformation T_b^a between keyframes F_a and F_b is known, that is $p_i^a = R_b^a p_i^b + t_b^a$, substituted into Equation (15), then the ground parameter of keyframe F_b is naturally represented under keyframe F_a as:

$$\begin{cases} \vec{n}^b = (R_b^a)^T \vec{n}^a \\ d^b = d^a - (\vec{n}^a)^T t_b^a \end{cases} \tag{16}$$

Equation (16) converts the ground of the current keyframe into the previous keyframe coordinate frame. In this paper, the conversion relationship between two keyframes is determined by the optimized lidar odometry factor, which is accurate in a short time. Then we compare the HF parameters between two keyframes. If the changes of normal vector angle Δ_θ and distance Δ_d of HF between two keyframes are less than the threshold, the two keyframes are considered to belong to the same ground. Otherwise, they belong to different grounds. We use Equation (16) to splice different grounds into a global ground. Finally, we add the difference between the ground parameter observed at each keyframe with the estimated global ground as a ground factor to the factor graph.

Equations (17) and (18) use a ground factor corresponding to the measured likelihood of a model with Gaussian noise:

$$L(\Pi^i, \Pi^W) = \exp\left\{-\frac{1}{2}\|h(\Pi^i) - \Pi^W\|_{\Sigma}^2\right\} \tag{17}$$

$$h(\Pi^i) = R_i^W \vec{n}^i d^i + (R_i^W \vec{n}^i)^T t_i^W (R_i^W \vec{n}^i) \tag{18}$$

where Σ is the noise covariance, Π^W is the global ground, and $h(\Pi^i)$ is the ground parameter measured from the keyframe F_i in the W coordinate frame.

3.5. Loop Closure Factor

When the robot revisits the vicinity of the historical keyframe, a closed loop is formed between the current and the historical keyframe. This provides the relative pose relationship of two keyframes, which can compress the cumulative error of continuous keyframe pose estimation between them. This paper detects loop closure using a Euclidean space-based method [18]. In this paper, the search for a loop closure occurs at a distance set to 10 m. When a new keyframe is added as a variable node to the factor graph, the variable node that meets the distance limit is searched from the factor graph. Then, the point cloud registration is performed in the same way as the lidar odometry to obtain the relative transformation ΔT between the two nodes, and the loop closure factor ΔT was added to the factor graph. In our experiments, loop closure had a positive effect. In an environment with loop closure, loop closure pulls the estimated trajectory back to the correct position.

3.6. GPS Factor

While lidar and inertial can accurately estimate the robotic state in a short time, they still produce large pose drift in large-scale and complex environments. One solution is the absolute measurement provided by GPS. When a GPS signal is available, our method converts the GPS measurements into the local Cartesian coordinate frame [18,26] to build GPS factors. Whenever a new keyframe is generated, we linearly interpolate the GPS measurement based on the time stamp of the lidar and associate the new node with the GPS factor. This process is selective because position estimation errors increase slowly, and the frequent insertion of noisy GPS measurements has a negative effect. This paper adds a GPS factor if the estimated location covariance is larger than the received GPS location covariance. Our method can work entirely without GPS, and when GPS is used, it effectively limits the cumulative error caused by position estimation.

4. Experiments and Analysis

4.1. Experimental Equipment

To validate our method, we conducted extensive experiments on the public datasets KITTI [27] and M2DGR [28], and in real-world environments. Our experimental platform is shown in Figure 2. It is a ground robot weighing about 160 kg on an Ackerman chassis, with a 65 cm wheelbase and 38 cm wheel diameter, equipped with an RS-LiDAR-16 lidar, an Xsens Mti-g710 9-axis IMU, a CHCNAV CGI-610 integrated navigation, and an AMD R7 4800 H central processing unit (CPU) laptop with 16G memory. All experiments were completed on this experimental platform. In our experimental environments, the frequencies of lidar, Xsens IMU, and CHCNAV GPS were set to 10 Hz, 400 Hz, and 4 Hz, respectively.



Figure 2. Experimental platform. The primary sensors include a 16-line 3D lidar, a 9-axis IMU, and an integrated navigation system.

4.2. Experiments on Public Datasets

In this section, our method compares absolute trajectory errors (ATE) with several well-known lidar SLAM methods on the KITTI 07 sequence and the M2DGR street_04 sequence, including only a lidar using method A-LOAM [13] and two multi-sensor fusion methods, LIO-MAPPING [17] and LIO-SAM [18]. The two public dataset sequences are typical vehicle driving environments, including flat ground, buildings, and trees. The robot's driving speed in sequences 07 and 04 was 6 m/s and 1 m/s, respectively. Tables 1 and 2 show the ATE of our method and the other three controlled methods. The experimental results show that the translation and rotation errors of A-LOAM were greater than those of the other three methods because the empty environment and some sharp curves of these two sequences were not friendly to the single-sensor method. Our method had a similar translation error in the horizontal direction to the two multi-sensor fusion methods, but a remarkable difference was reflected in the altitude direction. Due to the ground factor, the drift of our method along the altitude direction was not apparent, the altitude error was compressed below 0.4m, and our rotation error was lower than that of the other three methods in both sequences. Thanks to the lidar inertial tightly coupled framework, the errors of our method grew slowly over time, and accurate robot trajectory estimation was obtained by correcting the position with GPS and loop closure constraints. In addition, we obtained similar results to those shown in Tables 1 and 2 in several other outdoor sequences of these two public datasets. All four methods were run in real time and the translation error of our method was controlled within 1 m, which is lower than the other three methods.

Table 1. Root mean square error (RMSE) for translation and rotation in sequence 07 of the KITTI dataset.

APPROACH	x[m]	y[m]	z[m]	trans[m]	yaw[deg]	pitch[deg]	roll[deg]	rot[deg]
A-LOAM	1.183	1.216	1.752	2.412	0.725	0.548	0.740	1.172
LIO-MAPPING	0.588	0.692	0.845	1.247	0.433	0.601	0.516	0.903
LIO-SAM	0.435	0.646	0.817	1.129	0.497	0.528	0.563	0.918
ours	0.487	0.459	0.390	0.775	0.514	0.392	0.267	0.701

Table 2. RMSE for translation and rotation in sequence street_04 of the M2DGR dataset.

APPROACH	x[m]	y[m]	z[m]	trans[m]	yaw[deg]	pitch[deg]	roll[deg]	rot[deg]
A-LOAM	1.873	1.932	4.175	4.967	1.142	1.311	0.905	1.960
LIO-MAPPING	0.766	0.824	3.590	3.762	0.581	0.727	0.814	1.236
LIO-SAM	0.581	0.893	3.203	3.376	0.564	0.740	0.739	1.188
ours	0.617	0.649	0.257	0.932	0.692	0.392	0.271	0.840

4.3. Experiments on PARK and BLOCK

We used the experimental equipment in Figure 2 to acquire more datasets from challenging environments to test our method sufficiently. Figure 3a shows an experimental environment named PARK, where the robot goes around counterclockwise from the red start point back to the origin, with a length of 1328 m. This paper uses integrated navigation CHCNAV CGI-610 to provide the ground-truth trajectory. Table 3 shows the ATE between different methods and the ground-truth trajectory in PARK, and their trajectories are shown in Figure 3b,c. The lidar measurement noise makes the trajectory drift along the altitude direction. As shown in Table 3 and Figure 3c, the three comparison methods have more than twice the translation error in the vertical direction than the horizontal direction. Excessive error accumulation caused the loop closure failure of LIO-MAPPING and LIO-SAM, the same as A-LOAM without back-end optimization. As shown in Figure 3b, they did not return to the starting point. However, due to the ground factor constraining the drifting in the altitude direction, our translation error in the altitude direction was only 0.48 m, which was always within the range of loop closure that occurred safely. However, with the help of different constraint factors, only our method returned to the starting point. Our method always approached the ground-truth trajectory. The translation error of our method was 0.952 m, while the rotation error was 0.950 degrees in PARK, which is more accurate than the other three methods.

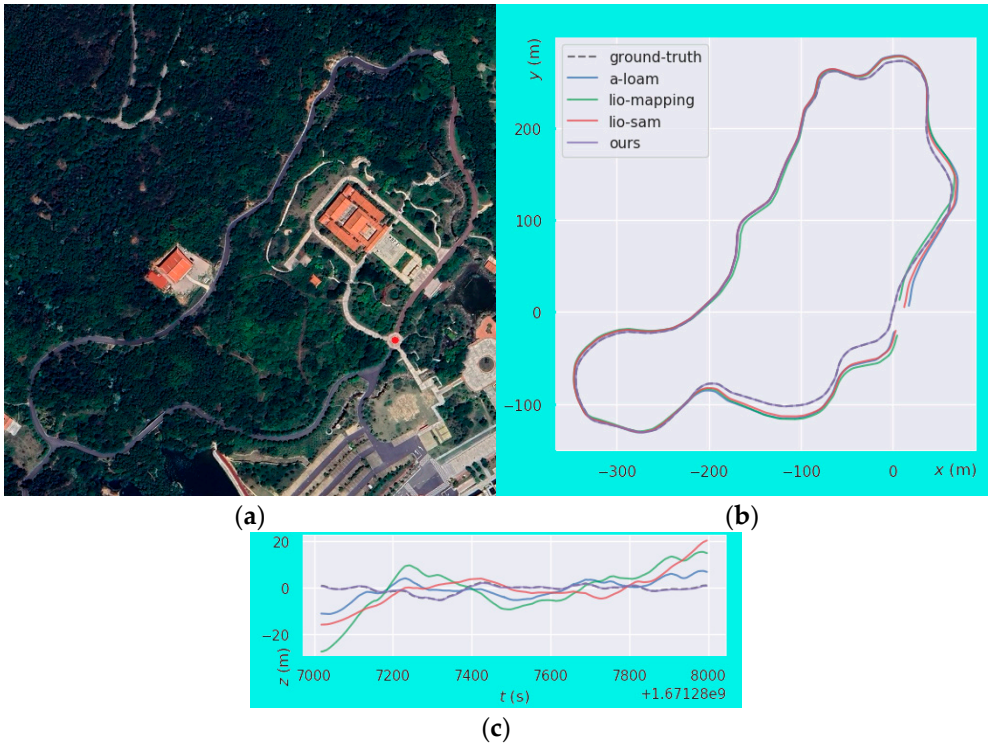
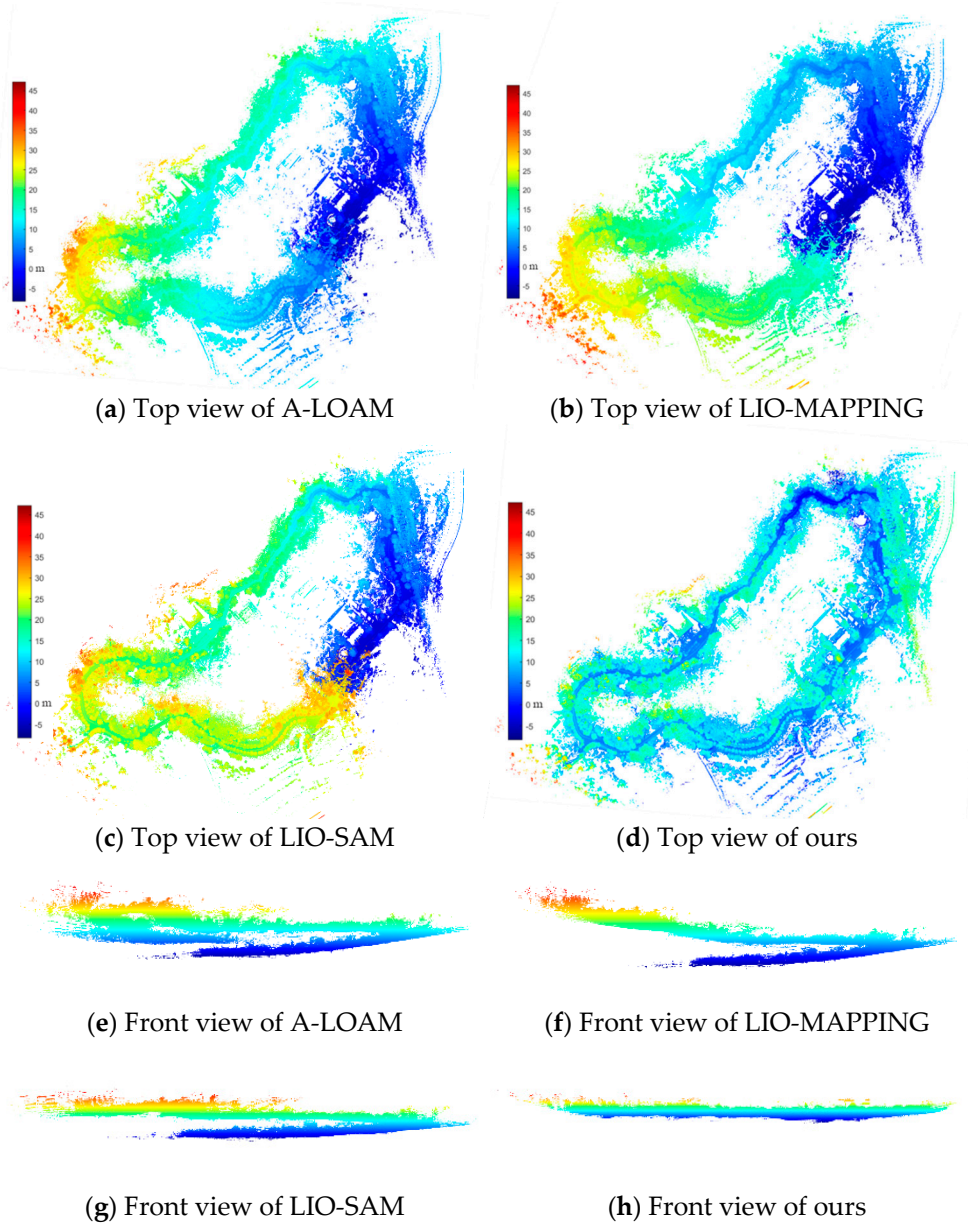


Figure 3. Satellite image of PARK and trajectories of different methods in PARK. (**a**) Satellite image of PARK; (**b**) Horizontal trajectory of different methods in PARK; (**c**) Vertical trajectory over time of different methods in PARK.

Table 3. Translation and rotation RMSE of different methods in PARK.

APPROACH	x[m]	y[m]	z[m]	trans[m]	yaw[deg]	pitch[deg]	roll[deg]	rot[deg]
A-LOAM	2.089	1.826	4.523	5.306	0.869	0.944	0.826	1.526
LIO-MAPPING	1.761	1.940	8.825	9.206	0.759	0.840	0.738	1.351
LIO-SAM	1.583	1.435	6.102	6.465	0.608	0.563	0.833	1.175
ours	0.537	0.621	0.482	0.952	0.763	0.461	0.328	0.950

Figure 4 shows the point cloud maps of A-LOAM, LIO-MAPPING, and LIO-SAM in PARK. The color of their point cloud gradually changed from blue at the starting point to red in a counterclockwise direction, indicating that their point cloud height gradually increased and the estimated trajectory deviated from the ground-truth trajectory. There was a downhill slope at 3/4 of the PARK, and the altitude of the point clouds of the three methods began to drop, but at last, they all formed an altitude distance from the starting point, as shown in Figure 4e–g. The point cloud map of our method in PARK is shown in Figure 4d,h. The point cloud altitude was always close to the ground-truth trajectory. Under the ground constraint factor, the altitude of our point cloud changed with the relief of the terrain rather than drifting toward the altitude with time.



(**a**) Top view of A-LOAM (**b**) Top view of LIO-MAPPING

(**c**) Top view of LIO-SAM (**d**) Top view of ours

(**e**) Front view of A-LOAM (**f**) Front view of LIO-MAPPING

(**g**) Front view of LIO-SAM (**h**) Front view of ours

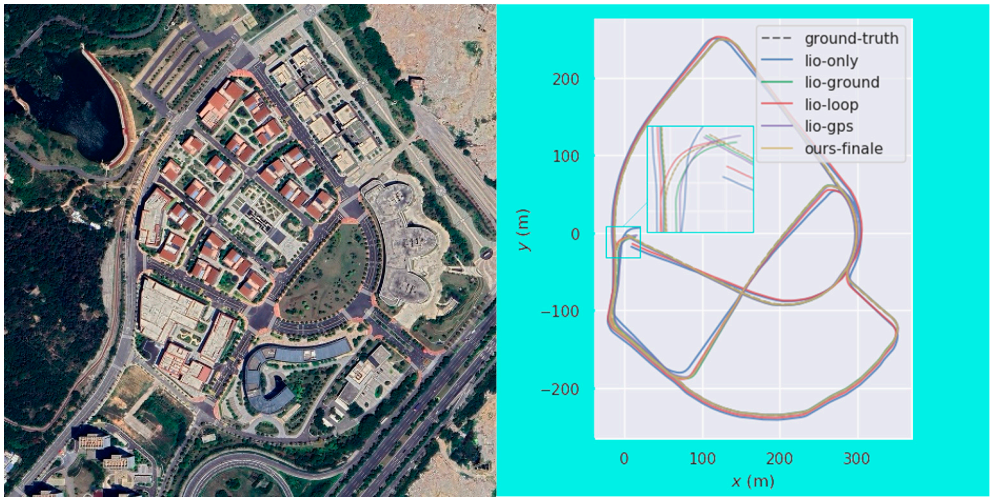
Figure 4. Point cloud maps were generated by different methods in PARK. The point clouds were rendered according to the altitude value in the vertical direction. Blue indicates low altitude and red indicates high altitude.

Furthermore, to illustrate the influence of different constraint factors in our method on the SLAM trajectory, the experiments based on tightly coupled lidar inertial odometry (LIO) were carried out with ground, loop closure, and GPS factors, respectively, in a large-

scale environment BLOCK with a length of 2334 m. In the different methods, lio-only (LIO) was the minimization method of our system, meaning that only lidar odometry and IMU factors were used for the robot's odometry estimation, and there was no constraint factor to correct the cumulative error of trajectory estimation. The ground factor was added to LIO; we named this method lio-ground. Similarly, we added loop closure and GPS factor separately to LIO and named the two methods lio-loop and lio-gps, respectively. The ATE between the different methods and the ground-truth trajectory is given in Table 4, while Figure 5 shows the corresponding trajectory and the satellite image of BLOCK.

Table 4. Translation and rotation RMSE of different methods in BLOCK.

APPROACH	x[m]	y[m]	z[m]	trans[m]	yaw[deg]	pitch[deg]	roll[deg]	rot[deg]
lio-only	1.809	1.946	5.651	6.244	0.836	1.120	0.964	1.698
lio-ground	1.913	1.857	1.214	2.929	0.734	0.491	0.578	1.055
lio-loop	1.370	1.238	3.644	4.085	0.705	0.514	0.810	1.191
lio-gps	1.215	1.008	4.736	5.087	0.923	0.706	0.625	1.319
ours-finale	0.571	0.612	0.429	0.941	0.529	0.388	0.326	0.733

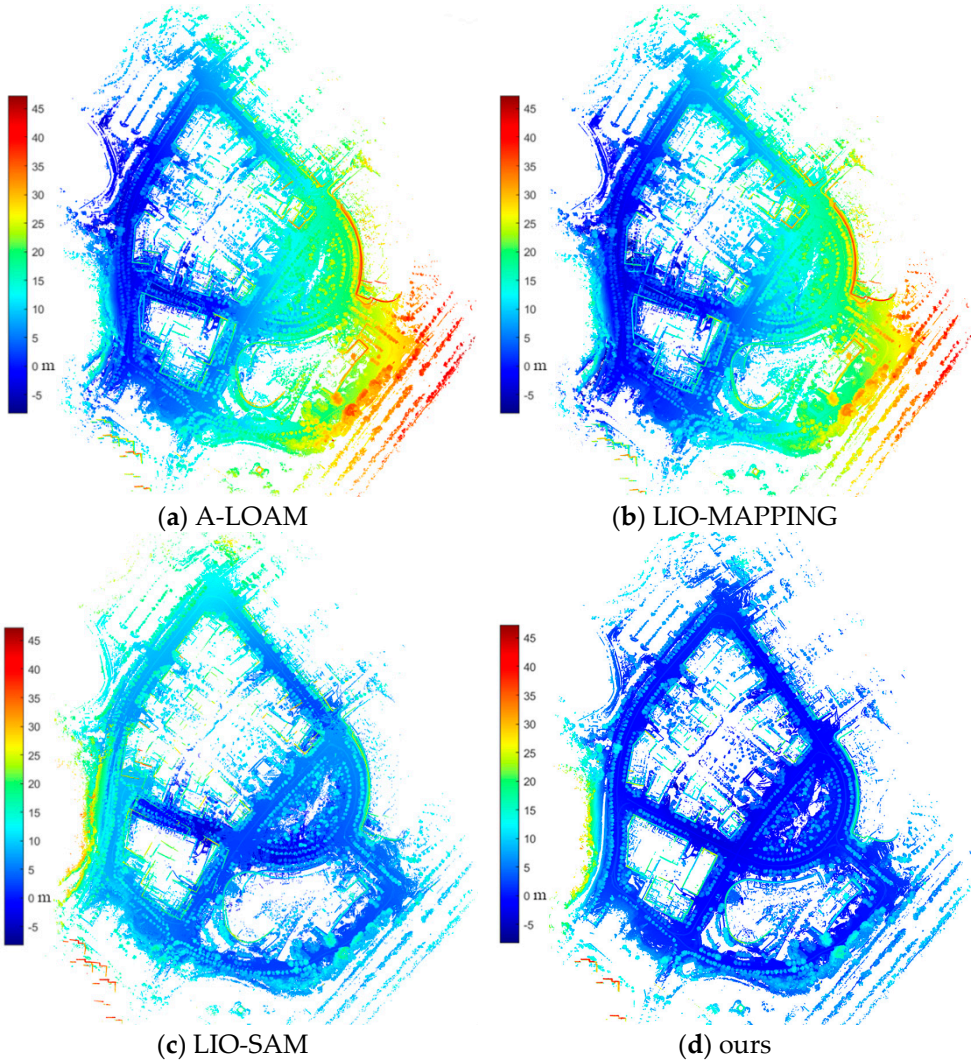


(**a**) (**b**)

Figure 5. Satellite image of BLOCK and trajectory of different methods in BLOCK. (**a**) Satellite image of BLOCK; (**b**) Horizontal trajectory of different methods in BLOCK.

The results in BLOCK show that the translation and rotation error of lio-only was 6.24 m and 1.70 degrees, which indicates that LIO will produce unacceptable errors in a large-scale environment. It was necessary to reduce cumulative errors by constraint factors. Compared with LIO, the lio-ground method used the ground constraint factor to significantly reduce the translation error in the altitude direction from 5.65 m to 1.21 m, while the rotation error was reduced from 1.70 to 1.06 degrees. Because there were multiple repeated paths in BLOCK, lio-loop reduced the translation error of the trajectory in three directions from 6.24 m to 4.09 m and the rotation error from 1.70 to 1.19 degrees through loop closure factors. Under the function of the GPS constraint factor, the horizontal translation error of lio-gps was limited to about 1.11m, close to the ground-truth trajectory. However, the translation error in the altitude direction was only reduced from 5.65 m to 4.74 m, which is related to the measurement accuracy of GPS in the altitude direction. Table 4 and Figure 5b show that different constraint factors positively affect the LIO. The ours-finale indicates the final result of our method with all factors working together, and it is clear that our method can maintain high precision even in a large-scale environment. In BLOCK, the translation error of our method was 0.941 m, while the rotation error was 0.733 degrees. As a comparison, the point cloud maps in BLOCK generated by A-LOAM, LIO-MAPPING, LIO-SAM, and our method are shown in Figure 6. This figure shows that the three methods

compared with ours have multiple ghosting and irregular roads, which is a phenomenon of inaccurate positioning. In contrast, our point cloud map shows more complete buildings and trees, preserving more details of the environment.



(**a**) A-LOAM (**b**) LIO-MAPPING

(**c**) LIO-SAM (**d**) ours

Figure 6. Point cloud maps were generated by different methods in BLOCK. The point cloud colors represent altitude value.

5. Conclusions and Future Work

This paper proposed a robust simultaneous localization and mapping method. It used the factor graph to fuse multi-modality sensor measurements, including five types of factors: IMU preintegration factor, lidar odometry factor, ground factor, loop closure factor, and GPS factor. The proposed method used tightly coupled lidar and inertial to obtain low-drift lidar odometry factors by factor graph optimization. The optimized lidar odometry factors were then added to a global factor graph, together with ground, loop closure, and GPS factors, to obtain accurate keyframe pose estimation and save the point cloud of each keyframe to build a point cloud map after global factor graph optimization.

In order to compress the trajectory drift along the altitude direction, which is mainly caused by lidar measurement noise in outdoor non-plane environments, this paper used low-drift lidar odometry to recover a global ground. The ground factor was then constructed using the robot pose invariant relative to the ground. The results of different experimental environments show that the trajectory estimation error of our method grew slowly. In a general ground robot working environment such as BLOCK, the ground factor can positively constrain the trajectory drift in the altitude direction. Compared with ad-

vanced lidar SLAM, our method achieved comparable results, and even performed better in some large-scale and complex environments.

One limitation of our method is that the translation drift in the altitude direction can be compressed through the ground in a flat environment, which remains challenging in some environments where the ground is uneven. In the future, we plan to add vision sensors to our methodology framework to improve the stability and performance of the system.

Author Contributions: Conceptualization, B.S. and R.X.; methodology, D.L. and R.X.; software, R.L. and D.L.; validation, B.S., R.L. and D.L.; data curation, D.L.; formal analysis, D.L. and B.S.; writing—original draft preparation, D.L. and R.L.; writing—review and editing, R.X., B.S. and R.L. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: Not applicable.

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

References

1. De Winter, A.; Baldi, S. Real-Life implementation of a GPS-Based path-following system for an autonomous vehicle. *Sensors* **2018**, *18*, 3940. [CrossRef] [PubMed]
2. Hayajneh, M.; Al Mahasneh, A. Guidance, Navigation and Control System for Multi-Robot Network in Monitoring and Inspection Operations. *Drones* **2022**, *6*, 332. [CrossRef]
3. Zhang, J.; Liu, R.; Yin, K.; Wang, Z.; Gui, M.; Chen, S. Intelligent collaborative localization among air-ground robots for industrial environment perception. *Ind. Electron.* **2018**, *66*, 9673–9681. [CrossRef]
4. Liu, R.; Zhang, G.; Wang, J.; Zhao, S. Cross-modal 360° depth completion and reconstruction for large-scale indoor environment. *Intell. Transp. Syst.* **2022**, *23*, 25180–25190. [CrossRef]
5. Pomerleau, F.; Colas, F.; Siegwart, R.; Magnenat, S. Comparing ICP variants on real-world data sets Open-source library and experimental protocol. *Auton. Robot.* **2013**, *34*, 133–148. [CrossRef]
6. Szymon, R.; Levoy, M. Efficient variants of the ICP algorithm. In Proceedings of the Third International Conference on 3-D Digital Imaging and Modeling, Montreal, QC, Canada, 28 May 2001. [CrossRef]
7. Kümmerle, R.; Grisetti, G. g2o: A general framework for graph optimization. In Proceedings of the IEEE International Conference Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011. [CrossRef]
8. Kim, G.; Kim, A. Scan context: Egocentric spatial descriptor for place recognition within 3d point cloud map. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018. [CrossRef]
9. Laconte, J.; Deschênes, S.P.; Labussière, M.; Pomerleau, F. Lidar measurement bias estimation via return waveform modelling in a context of 3D mapping. In Proceedings of the IEEE International Conference Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019. [CrossRef]
10. Kaess, M.; Ranganathan, A.; Dellaert, F. iSAM: Incremental smoothing and mapping. *Robot* **2008**, *24*, 1365–1378. [CrossRef]
11. Kaess, M.; Johannsson, H.; Roberts, R.; Ila, V.; Leonard, J.; Dellaert, F. iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering. In Proceedings of the IEEE International Conference Robotics and Automation (ICRA), Shanghai, China, 9–13 May 2011. [CrossRef]
12. Deschaud, J.E. IMLS-SLAM: Scan-to-model matching based on 3D data. In Proceedings of the IEEE International Conference Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018. [CrossRef]
13. Zhang, J.; Singh, S. Low-drift and real-time lidar odometry and mapping. *Auton. Robot.* **2017**, *41*, 401–416. [CrossRef]
14. Demir, M.; Fujimura, K. Robust localization with low-mounted multiple lidars in urban environments. In Proceedings of the Intelligent Transportation Systems Conference (ITSC), Auckland, New Zealand, 27–30 October 2019. [CrossRef]
15. Gao, Y.; Liu, S.; Atia, M.M.; Noureldin, A. INS/GPS/LiDAR integrated navigation system for urban and indoor environments using hybrid scan matching algorithm. *Sensors* **2015**, *15*, 23286–23302. [CrossRef] [PubMed]
16. Yang, S.; Zhu, X.; Nian, X.; Feng, L.; Qu, X.; Ma, T. A robust pose graph approach for city scale LiDAR mapping. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018. [CrossRef]
17. Ye, H.; Chen, Y.; Liu, M. Tightly coupled 3D lidar inertial odometry and mapping. In Proceedings of the IEEE International Conference Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019. [CrossRef]
18. Shan, T.; Englot, B.; Meyers, D.; Wang, W.; Ratti, C.; Rus, D. Lio-sam: Tightly-coupled lidar inertial odometry via smoothing and mapping. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Las Vegas, NV, USA, 24 October 2020–24 January 2021. [CrossRef]
19. Zheng, F.; Tang, H.; Liu, Y. Odometry-vision-based ground vehicle motion estimation with SE (2)-constrained SE (3) poses. *Cybernetics* **2018**, *49*, 2652–2663. [CrossRef] [PubMed]
20. Zheng, F.; Liu, Y. Visual-odometric localization and mapping for ground vehicles using SE (2)-XYZ constraints. In Proceedings of the IEEE International Conference Robotics and Automation (ICRA), Montreal, QC, Canada, 20–24 May 2019. [CrossRef]

21. Koide, K.; Miural, J.; Menegatti, E. A portable three-dimensional LIDAR-based system for long-term and wide-area people behavior measurement. *Int. J. Adv. Robot. Syst.* **2019**, *16*, 1729881419841532. [CrossRef]
22. Geneva, P.; Eckenhoff, K.; Yang, Y.; Huang, G. Lips: Lidar-inertial 3D plane slam. In Proceedings of the International Conference on Intelligent Robots and Systems (IROS), Madrid, Spain, 1–5 October 2018. [CrossRef]
23. Wei, X.; Lv, J.; Sun, J.; Pu, S. Ground-SLAM: Ground Constrained LiDAR SLAM for Structured Multi-Floor Environments. *arXiv* **2021**. [CrossRef]
24. Zhang, J.; Zhang, C.; Wu, J.; Jin, J.; Zhu, Q. LiDAR-Inertial 3D SLAM with Plane Constraint for Multi-story Building. *arXiv* **2022**. [CrossRef]
25. Fischler, M.A.; Bolles, R.C. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM* **1981**, *24*, 381–395. [CrossRef]
26. Moore, T.; Stouch, D. A generalized extended kalman filter implementation for the robot operating system. In *Proceedings of the Intelligent Autonomous Systems 13: Proceedings of the 13th International Conference IAS-13, Padova, Italy, 15–18 July 2014*; Springer International Publishing: Cham, Switzerland, 2016; Volume 302, pp. 335–348. [CrossRef]
27. Geiger, A.; Lenz, P.; Stiller, C.; Urtasun, R. Vision meets robotics: The kitti dataset. *Int. J. Robot. Res.* **2013**, *32*, 1231–1237. [CrossRef]
28. Yin, J.; Li, A.; Li, T.; Yu, W.; Zou, D. M2dgr: A multi-sensor and multi-scenario slam dataset for ground robots. *IEEE Robot. Autom. Lett.* **2021**, *7*, 2266–2273. [CrossRef]

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.