applied sciences

MDPI

Article

MIM_SLAM: A Multi-Level ICP Matching Method for Mobile Robot in Large-Scale and Sparse Scenes

Jingchuan Wang [1,2,*], Ming Zhao [1,2] and Weidong Chen [1,2]

[1] Department of Automation, Shanghai Jiao Tong University, Shanghai 200240, China; mzhao1993@sjtu.edu.cn (M.Z.); wdchen@sjtu.edu.cn (W.C.)

[2] Laboratory of System Control and Information Processing, Ministry of Education of China, Shanghai 200240, China

* Correspondence: jchwang@sjtu.edu.cn; Tel: +86-21-3420-4513

Received: 19 October 2018; Accepted: 27 November 2018; Published: 30 November 2018

Abstract: In large-scale and sparse scenes, such as farmland, orchards, mines, and substations, 3D simultaneous localization and mapping are challenging matters that need to address issues such as maintaining reliable data association for scarce environmental information and reducing the computational complexity of global optimization for large-scale scenes. To solve these problems, a real-time incremental simultaneous localization and mapping algorithm called MIM_SLAM is proposed in this paper. This algorithm is applied in mobile robots to build a map on a non-flat road with a 3D LiDAR sensor. MIM_SLAM's main contribution is that multi-level ICP (Iterative Closest Point) matching is used to solve the data association problem, a Fisher information matrix is used to describe the uncertainty of the estimated pose, and these poses are optimized by the incremental optimization method, which can greatly reduce the computational cost. Then, a map with a high consistency will be established. The proposed algorithm has been evaluated in the real indoor and outdoor scenes as well as two substations and benchmarking dataset from KITTI with the characteristics of sparse and large-scale. Results show that the proposed algorithm has a high mapping accuracy and meets the real-time requirements.

Keywords: data association; 3D-SLAM; localization; mapping

1. Introduction

In recent years, the simultaneous localization and mapping (SLAM) based on a 3D LIDAR sensor have become an important topic in robotics due to the rise of autonomous driving technology. Meanwhile, the building of a map has become the basis for an autonomous mobile robot to complete tasks such as inspection and autonomous navigation [1] in some harsh environments.

The research on SLAM can be traced back to Smith et al. [2] of Stanford University in the 1980s. They published a seminal paper on SLAM, and later generations have done considerable work on it. Among them, the algorithms such as Extended Kalman Filters [3], Extended Information Filters [4], and Rao-Blackwellized Particle Filters [5] based on a filtering idea all adopt the Bayesian state estimation theory to estimate the posterior probability of the system's state. However, the essence of this kind of algorithm is only optimizing the local state of the system. As the distance traveled by the robot increases, it is difficult to establish a high consistency map. Another kind of algorithm is the graph optimization that was first proposed by Lu and Milio [6]. The essence of this algorithm is maintaining all the observation and space constraints between observations, then using the maximum likelihood method to estimate the pose of the robot. Better mapping results can be obtained in large-scale scenes using the global optimization method.

Although the SLAM problem has been thoroughly researched, there are still many problems to be solved, including the fact that computational complexity will increase from two-dimensions to three-dimensions, and how to ensure a consistent map with the increase of the size of mapping and reliable data association with less information in sparse environments. Combining these issues, Zlot [7] utilizes a rotating 2D LIDAR sensor to achieve high-precision map construction in a wide mine, but it cannot be applied online. Olufs [8] proposes a data association method in sparse environments, but it is only suitable for solving the localization problem under known maps. Huang [9] proposes a sparse local submap joining filter (SLSJF) for map-building in large-scale environments. However, this depends on environmental features. Moosmann [10] and Nüchter [11] propose a real-time SLAM framework, and the experiments in some scenarios have achieved good results, but the results of the mapping are not tested for large-scale and sparse environments. Zhang [12] estimates the pose of the robot by extracting the feature points in the scene, but it is difficult to establish a closed-loop map as the map size expands. Wang [13] proposes a multi-layer matching SLAM in large-scale and sparse environments, but it only solves the data association in a 2D scene. The algorithm will be ineffective when the robot drives on a non-flat road. Therefore, this paper is dedicated to solving the following two problems:

- In sparse environments, due to the scarce environmental information perceived by a 3D LIDAR sensor and the non-flat road, data association is easy to fall into a local minimum. Liu [14] proposes a LiDAR SLAM method in natural terrains which fuses multiple sensors including two 3D LiDAR. Therefore, the robustness of data associated with a lower accumulated error should be strengthened.
- In large-scale environments, it is difficult to obtain a reasonable pose estimation when the robot returns to a region it has previously explored and then generates an inconsistent map. Liang [15] addresses the laser-based loop closure problem by fusing visual information. Hess [16] can achieve real-time mapping in indoor scenes, and it may fail in the closing loop due to the incremental computation in large-scale outdoor scenes. How to reduce the computational complexity of graph optimization is also a problem that needs to be improved.

To solve these problems, we propose a real-time incremental SLAM algorithm called the MIM_SLAM based on a multi-level ICP matching method. This paper is organized as follows. We continue in the next section with an overview of the algorithm. Section 3 details the process of the MIM_SLAM algorithm. Then, we follow up with the experimental results and analysis in Section 4 and finally provide a summary in Section 5.

2. Algorithm Overview

Figure 1 shows the overall MIM_SLAM algorithm framework. The MIM_SLAM algorithm simplifies the SLAM problem into the data association and the incremental pose graph optimization. Firstly, the data association problem is solved by the multi-level ICP matching method which includes matching the time-neighbor frame, matching the current frame with the map, and matching the current frame with an area-neighbor keyframe. Then the uncertainty of the estimated pose is described via the Fisher information matrix. After the above steps, this algorithm can obtain the transformation matrix T_{ij} and covariance matrix Σ_{ij} between pose x_i and pose x_j, which can be saved in the pose graph. Finally, the incremental QR decomposition method [14] is used to optimize the pose graph. In the next section, the MIM_SLAM algorithm will be described in detail.

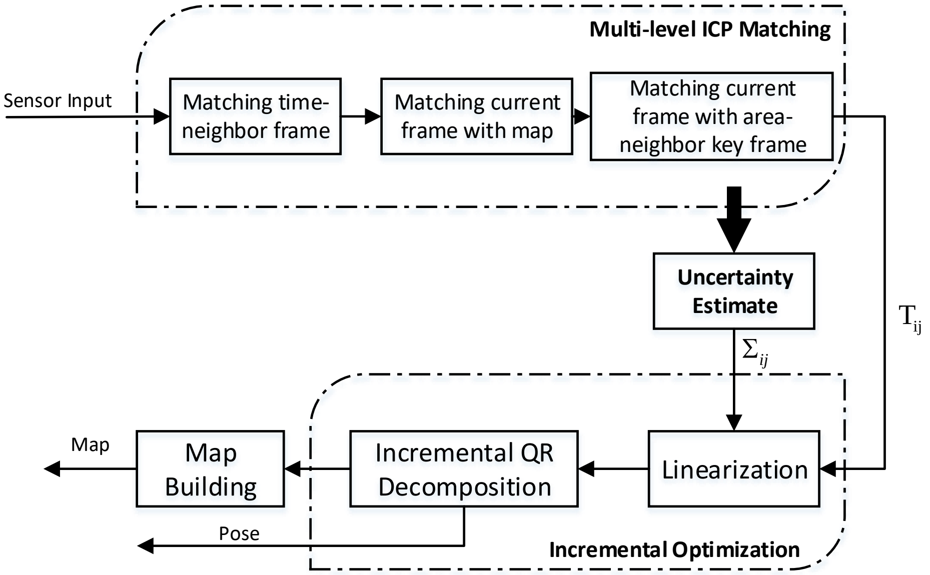

Multi-level ICP Matching
Sensor Input
Matching time-neighbor frame
Matching current frame with map
Matching current frame with area-neighbor key frame
Uncertainty Estimate
T_{ij}
Σ_{ij}
Map
Map Building
Incremental QR Decomposition
Linearization
Pose
Incremental Optimization

Figure 1. The proposed MIM_SLAM algorithm framework.

3. Algorithm Description

3.1. SLAM as an Incremental Optimization Problem

This part analyzes the SLAM problem from the probabilistic model and transforms the problem into a least squares problem based on the maximum a posteriori estimate. Then, the incremental QR decomposition method [17] is used to solve the least squares problem.

We denote robot pose by $X = \{x_i\}$ with $i \in 0 \dots M$, the landmark by $L = \{l_j\}$ with $j \in 1 \dots N$, control input given through wheeled odometry and IMU by $U = \{u_i\}$ with $i \in 1 \dots M$, landmark measurement by $Z = \{z_i\}$ with $i = 1 \dots M$. The SLAM problem is equivalent to estimating the posterior probability distribution:

$$\begin{aligned} & P(X, L|Z, U) \\ & = P(L|Z, X, U)P(X|Z, U) \\ & = P(L|Z, X)P(X|Z, U) \end{aligned} \tag{1}$$

From Equation (1), it can be seen that the landmark L depends on the observations and robot pose sequences, so the SLAM problem is further simplified to the pose estimation problem utilizing Baye's rule $P(X|Z, U) = P(Z|X, U)P(X|U)/P(Z|U)$. In this paper, we just consider the online SLAM problem [4], which only involves estimating the posterior over the momentary pose, this is $P(x_M|z_{1:M}, u_{1:M})$.

$$\begin{aligned} & P(x_M|z_{1:M}, u_{1:M}) \\ & = \frac{P(z_{1:M}|x_M, u_{1:M})P(x_M|u_{1:M})}{P(z_{1:M}|u_{1:M})} \\ & = \frac{P(z_M|x_M, z_{1:M-1}, u_{1:M})P(x_M|z_{1:M-1}, u_{1:M})}{P(z_M|z_{1:M-1}, u_{1:M})} \\ & = \eta P(z_M|x_M)P(x_M|z_{1:M-1}, u_{1:M}) \\ & = \eta P(z_M|x_M)\int P(x_M|x_{M-1}, z_{1:M-1}, u_{1:M})P(x_{M-1}|z_{1:M-1}, u_{1:M})dx_{M-1} \\ & = \eta P(z_M|x_M)\int P(x_M|x_{M-1}, u_M)P(x_{M-1}|z_{1:M-1}, u_{1:M-1})dx_{M-1} \\ & = \eta P(x_0)\prod_{i=1}^{M} P(x_i|x_{i-1}, u_i)P(z_i|x_i) \end{aligned} \tag{2}$$

In Equation (2) line 4 to 5, it is derived using the theorem of total probability. $P(x_0)$ is a prior on the initial state, $P(x_i|x_{i-1}, u_i)$ is the motion model, and $P(z_i|x_i)$ is the measurement model.

To estimate the pose of the robot, we transform the SLAM problem into the least-squares problem based on the maximum a posteriori estimate. The maximum a posteriori estimate X^* and L^* for the trajectory and map are obtained by minimizing the negative log of the joint probability from Equation (2):

$$\begin{aligned} X^* = -\underset{X}{\operatorname{argmin}} \eta \Bigg\{ & \sum_{i=1}^{M} \log P(x_i | x_{i-1}, u_i) \\ & + \sum_{i=1}^{M} \log P(z_i | x_i) \Bigg\} \end{aligned} \tag{3}$$

$$L^* = \{(x_1^*, z_1), \ldots, (x_i^*, z_i)\} \tag{4}$$

Both of the motion model and measurement model are generally assumed such that they meet the Gaussian distribution. Then, both of them can be converted to the following form:

$$x_j \sim N(x_i \oplus T_{ij}, \Sigma_{ij}) \tag{5}$$

where the operator $\oplus$ denotes the coordinate transformation. T_{ij} and Σ_{ij} represent respectively the transformation matrix and covariance matrix between pose x_i and pose x_j, which will be obtained through the multi-level ICP matching method and uncertainty estimation method in the next two paragraphs. Additionally, it can be written as follows:

$$\begin{cases} x_i = f(x_{i-1}, u_i) + m_i \\ z_i = g(x_i) + n_i \end{cases} \tag{6}$$

where m_i is the motion noise, and n_i is the measurement noise. This leads to the following nonlinear least squares problem:

$$\begin{aligned} X^* = \underset{X}{\operatorname{argmin}} \Bigg\{ & \sum_{i=1}^{M} \| f(x_{i-1}, u_i) - x_i \|_{m_i}^2 \\ & + \sum_{i=1}^{M} \| g(x_i) - z_i \|_{n_i}^2 \Bigg\} \end{aligned} \tag{7}$$

Since the above model has non-linear functions that are not easy to solve, they must be linearized, as Gauss-Newton [18] and Levenberg-Marquardt [19] has done. It can eventually be transformed into the general least-squares problem as follows, and the specific derivation can refer to the literature [14].

$$\theta^* = \underset{\theta}{\operatorname{argmin}} \| A\theta - b \|^2 \tag{8}$$

where the vector θ contains all pose variables, and the matrix A is a large and sparse Jacobian matrix.

The problem in Equation (8) is apparently solved by the Cholesky decomposition method, but the essential problem is the number of calculations involved in solving the information matrix $A^T A$. Thus, applying standard QR decomposition to matrix A:

$$A = Q \begin{bmatrix} R \\ 0 \end{bmatrix} \tag{9}$$

Apply Equation (9) to the least squares problem in Equation (8). As Q is an orthonormal matrix, $Q^TQ = I$ (I is identity matrix). Thus, an additional term Q^T is added to the second line in (10), which doesn't change the length of the vector $A\theta - b$.

$$\begin{aligned} \|A\theta - b\|^2 &= \|Q\begin{bmatrix} R \\ 0 \end{bmatrix}\theta - b\|^2 \\ &= \|Q^TQ\begin{bmatrix} R \\ 0 \end{bmatrix}\theta - Q^Tb\|^2 \\ &= \|\begin{bmatrix} R \\ 0 \end{bmatrix}\theta - \begin{bmatrix} d \\ e \end{bmatrix}\|^2 \\ &= \|R\theta - d\|^2 + \|e\|^2 \end{aligned} \tag{10}$$

where $[d, e]^T = Q^Tb$ is defined, and (10) obtain the minimum $\|e\|^2$ if and only if $R\theta = d$. Now, the SLAM problem has been solved by the above analysis. However, as a new observation arrives, the Jacobian matrix A does not significantly change. Thus the results of the QR decomposition at previous times can be used as the iterative initial value and the process is incremental, which can significantly reduce computational complexity.

3.2. The Multi-Level ICP Matching Method

Before detailing this method, it is useful to understand the ICP algorithm. The standard ICP [20] (Iterative Closest Point) algorithm firstly establishes the point-to-point correspondence between the two point clouds by the nearest neighbor principle and then establishes the matching error function.

Finally, it computes a transformation matrix so that the error function is minimized. In the past 30 years, many ICP variant algorithms [21] have been proposed. In this paper, an ICP variant algorithm that calls point-to-plane ICP [22] is used. Compared with the standard ICP, the robustness and accuracy are better. This algorithm is listed as Algorithm 1.

Algorithm 1: Point-to-plane ICP.

```
Input: Two point cloud: $A = \{a_i\}$, $B = \{b_i\}$; An initial transformation: $T_0$
Output: Transformation $T_{AB}$ which aligns $A$ and $B$; Fitness Score: $Score_{fitness}$
$T \leftarrow T_0$
while not converged do
    for $i \leftarrow 1$ to $N$ do
        $m_i \leftarrow$ FindClosestPointInA$(T \cdot b_i)$
        if $\|m_i - T \cdot b_i\| \leq d_{\max}$ then $w_i \leftarrow 1$
        else $w_i \leftarrow 0$
        end
    end
    $T \leftarrow \text{argmin}\left\{\sum_i w_i \|\eta_i \cdot (T \cdot b_i - m_i)\|^2\right\}$
end
$T_{AB} \leftarrow T$
$Score_{fitness} = \sum_i w_i \|\eta_i \cdot (T \cdot b_i - m_i)\|^2$
```

The observation of the 3D LiDAR sensor is denoted as z_k at time k. There are data associations between all previous observations $z_{1:k-1}$ and the current observation included in z_k. A traditional scan matching method associates z_k with z_{k-1} at last time, or associates z_k with all previous observations $z_{1:k-1}$. These two methods either have a substantial accumulated error, or have large amounts of calculations which cannot meet online applications. In this paper, the data association is divided into two categories: continuous in time and continuous in area.

For observations that are continuous in the time, the method firstly utilizes the ICP algorithm to associate z_k and z_{k-1}, and a rough estimation of the current robot's pose can be obtained. The initial value of ICP is obtained by using a Kalman filter to merge the wheeled odometry with IMU. Because the inter-frame matching cannot obtain the accurate pose estimation, the next step will match the current frame with the map to further eliminate the accumulated error.

As shown in Figure 2, M_i denotes the global map at time i and T_i^W denotes the pose of the robot in the world coordinate system at time i. T_{i+1}^L denotes the transformation matrix from time $i+1$ to time i (that is, the output of inter-frame matching). z_{i+1} denotes the observation at time $i+1$. The process for matching the current frame with the map is as follows:

- Finding the nearest neighbor in M_i for each point in z_{i+1}, and saving it as m_{i+1}. We use the combination of octree and approximate nearest neighbor algorithm in PCL [23] for speeding up.
- Taking z_{i+1} and m_{i+1} as the input of Algorithm 1, the transformation matrix T_{opt} can be obtained after registering the two point clouds.
- The pose of the robot at time $k+1$ is $T_{i+1}^W = T_i^W \cdot T_{i+1}^L \cdot T_{opt}$.

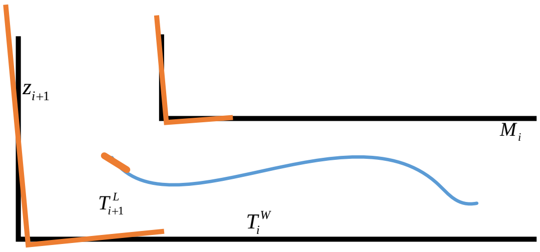
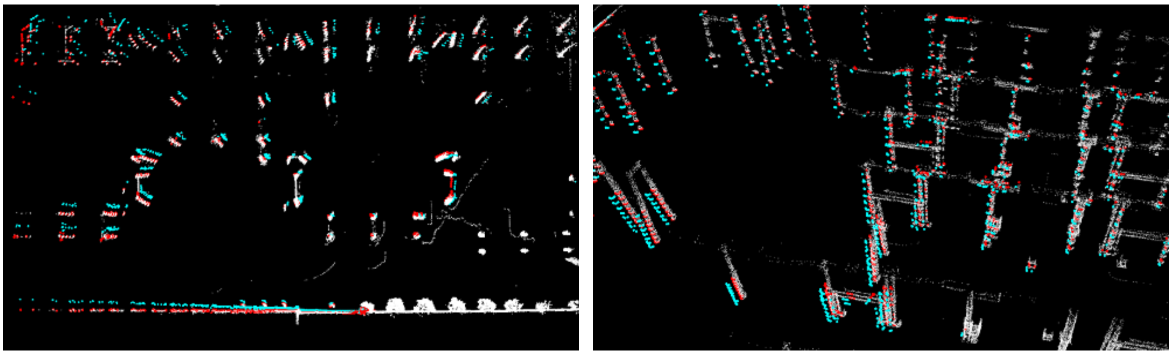


Figure 2. The matching process of the current frame and map.

The effectiveness of the above scan matching method is verified as shown in Figure 3. The white point clouds denote the map that SLAM has established. The blue point clouds denote the point clouds after inter-frame matching. The redpoint clouds denote the point clouds after matching the current frame with the map. It can be seen that the accumulated error in the process of inter-frame matching is obviously eliminated.



(**a**) Scene 1 (**b**) Scene 2

Figure 3. An example of the matching process.

For observations that are continuous in the area, the robot returns to a region that it has previously explored. In particular, in large-scale and sparse environments, the accumulative error cannot be completely eliminated when the robot travels a long distance if only the above two matching processes are considered. So the following method matches the current frame z_i with the area-neighbor keyframe

z_τ based on a priori of pose estimation. If the matching score is less than a certain threshold, the pose constraint between x_i and x_τ will be established. The process is listed as Algorithm 2. The initial parameter is: $L = \{(0, x_0, z_0)\}, d = 0$.

Algorithm 2: Matching the current frame with an area-neighbor keyframe.

```
Input: Current pose and observation: $x_i, z_i$
Output: Transformation $T_{i\tau}$ which aligns $z_i$ and $z_\tau$
calculate $d = d + \|x_i - x_{i-1}\|$
if $d < d_{thre}$ then
        return null
end
foreach $(k_\tau, x_\tau, z_\tau)$ in $L$ do
    if $k - k_\tau > k_{thre}$ then
        if $\|x_t - x_\tau\| < x_{thre}$ then
            put $z_t$ and $z_\tau$ into Algorithm 1, get $T$ and $Score_{fitness}$
            if $Score_{fitness} < S_{thre}$ then
                $T_{i\tau} \leftarrow T$
                return $T_{i\tau}$
            end
        end
    end
end
$L = L \cup (k_i, x_i, z_i), d = 0, k = k + 1$
return null
```

3.3. Uncertainty Estimation

After multi-level ICP matching, the transformation matrix T_{ij} between pose x_i and pose x_j can be obtained, but we still need to know the exact uncertainty estimation, namely the covariance matrix Σ_{ij}. However, an incorrect covariance matrix may damage the established map.

In this paper, the inversion of the Fisher information matrix is used as the covariance. The Fisher information matrix is defined as the function of the expected measurement and the surface slope scanned by the laser sensor [24]. Liu [24] and Wang [25] give the derivation of the Fisher information matrix based on a two-dimensional probability grid. It is now extended to three-dimensions. The Fisher information matrix is discretized:

$$\hat{L}(p) = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \left(\frac{\Delta r_{iE}}{\Delta p}\right)^T \left(\frac{\Delta r_{iE}}{\Delta p}\right) \tag{11}$$

$$\frac{\Delta r_{iE}}{\Delta p} = \left[\frac{\Delta r_{iE}}{\Delta x}, \frac{\Delta r_{iE}}{\Delta y}, \frac{\Delta r_{iE}}{\Delta z}, \frac{\Delta r_{iE}}{\Delta \varphi}, \frac{\Delta r_{iE}}{\Delta \psi}, \frac{\Delta r_{iE}}{\Delta \theta}\right] \tag{12}$$

where $p = [x, y, z, \varphi, \psi, \theta]$ is the pose of the robot, and σ_{i}^2 is the noise variance of the i-th 3D laser range finder scan ray. N is the total number of scan rays. r_{iE} is the expected distance from the robot to the nearest obstacle along the i-th scan ray. To calculate Equation (12) efficiently, the point cloud map is converted into a three-dimensional grid map. Thus r_{iE} can be computed as follows:

$$r_{iE} = \frac{\sum\limits_{j=1}^{s} r_{ij}\mu_{ij}}{\sum\limits_{j=1}^{s} \mu_{ij}} \tag{13}$$

where r_{ij} is the distance between the robot and the j-th voxel along the direction of an i-th scan ray. μ_{ij} is the occupancy value of the corresponding voxel. s is the sequence number of the ending voxel. By combining Equation (11) and Equation (12), we obtain:

$$\hat{L}(p) = \sum_{i=1}^{N} \frac{1}{\sigma_i^2} \begin{bmatrix} \frac{\Delta r_{iE}^2}{\Delta x^2} & \frac{\Delta r_{iE}^2}{\Delta x \Delta y} & \frac{\Delta r_{iE}^2}{\Delta x \Delta z} & \frac{\Delta r_{iE}^2}{\Delta x \Delta \varphi} & \frac{\Delta r_{iE}^2}{\Delta x \Delta \psi} & \frac{\Delta r_{iE}^2}{\Delta x \Delta \theta} \\ \frac{\Delta r_{iE}^2}{\Delta x \Delta y} & \frac{\Delta r_{iE}^2}{\Delta y^2} & \frac{\Delta r_{iE}^2}{\Delta y \Delta z} & \frac{\Delta r_{iE}^2}{\Delta y \Delta \varphi} & \frac{\Delta r_{iE}^2}{\Delta y \Delta \psi} & \frac{\Delta r_{iE}^2}{\Delta y \Delta \theta} \\ \frac{\Delta r_{iE}^2}{\Delta x \Delta z} & \frac{\Delta r_{iE}^2}{\Delta y \Delta z} & \frac{\Delta r_{iE}^2}{\Delta z^2} & \frac{\Delta r_{iE}^2}{\Delta z \Delta \varphi} & \frac{\Delta r_{iE}^2}{\Delta z \Delta \psi} & \frac{\Delta r_{iE}^2}{\Delta z \Delta \theta} \\ \frac{\Delta r_{iE}^2}{\Delta x \Delta \varphi} & \frac{\Delta r_{iE}^2}{\Delta y \Delta \varphi} & \frac{\Delta r_{iE}^2}{\Delta z \Delta \varphi} & \frac{\Delta r_{iE}^2}{\Delta \varphi^2} & \frac{\Delta r_{iE}^2}{\Delta \varphi \Delta \psi} & \frac{\Delta r_{iE}^2}{\Delta \varphi \Delta \theta} \\ \frac{\Delta r_{iE}^2}{\Delta x \Delta \psi} & \frac{\Delta r_{iE}^2}{\Delta y \Delta \psi} & \frac{\Delta r_{iE}^2}{\Delta z \Delta \psi} & \frac{\Delta r_{iE}^2}{\Delta \varphi \Delta \psi} & \frac{\Delta r_{iE}^2}{\Delta \psi^2} & \frac{\Delta r_{iE}^2}{\Delta \psi \Delta \theta} \\ \frac{\Delta r_{iE}^2}{\Delta x \Delta \theta} & \frac{\Delta r_{iE}^2}{\Delta y \Delta \theta} & \frac{\Delta r_{iE}^2}{\Delta z \Delta \theta} & \frac{\Delta r_{iE}^2}{\Delta \varphi \Delta \theta} & \frac{\Delta r_{iE}^2}{\Delta \psi \Delta \theta} & \frac{\Delta r_{iE}^2}{\Delta \theta^2} \end{bmatrix} \tag{14}$$

According to the Cramér–Rao Bound theory [26], the lower bound of covariance can be determined by the inversion of the Fisher information matrix. Finally, the uncertainty of the pose estimation can be obtained:

$$\mathrm{cov}(p) = \hat{L}^{-1}(p) \tag{15}$$

4. Experimental Results and Analysis

As shown in Figure 4, the mobile robot called SmartGuard [27] is used to verify the proposed algorithm. It is a completely autonomous robotic system that can inspect substation equipment. SmartGuard is equipped with a wheeled odometry, an IMU, and a 3D LIDAR (RS-Lidar-16). The RS-Lidar-16 can measure out to 150 m with a high precision, ±2 cm, and it has a $+15$ to -15-degree vertical field of view. It continuously scans the 360-degree surrounding environment at a 10 Hz frame rate and at 300,000 points/sec. The capability of the computer that runs the SLAM algorithm is as follows: Intel Core i5-6300HQ CPU 2.3 Hz and 8 G DDR3 RAM.

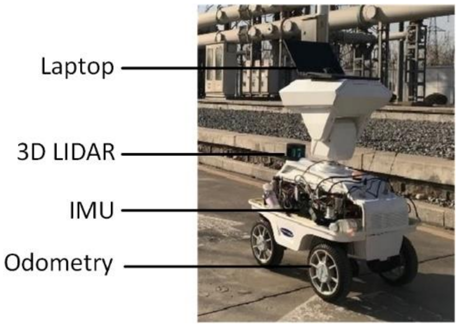
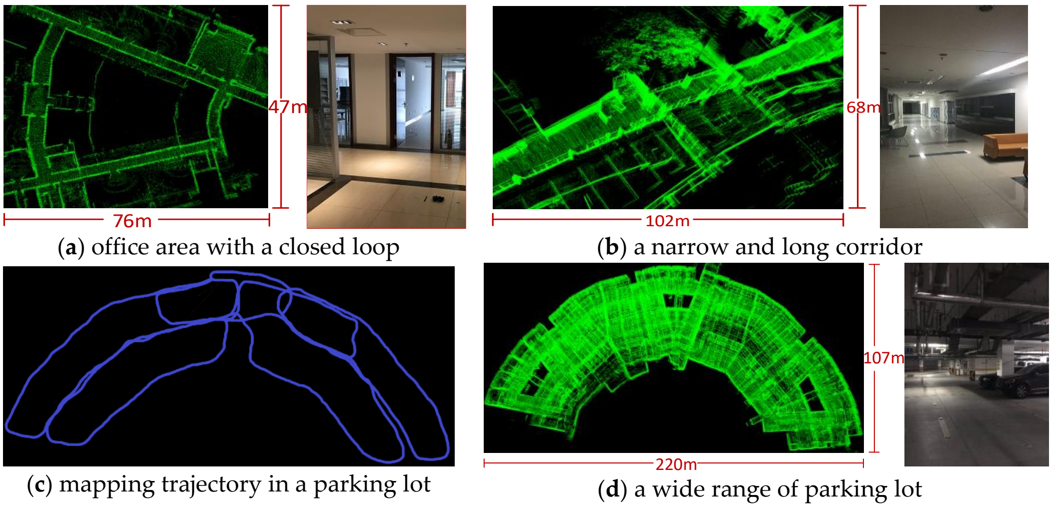


Figure 4. The mobile robot (SmartGuard) for experiments.

4.1. Indoor and Outdoor Mapping Test

To verify the mapping performance, the proposed algorithms have been tested in several different indoor and outdoor environments (Figure 5). Figure 5a shows that the proposed MIM_SLAM algorithm can build a consistency map with lower accumulated errors after a 180 m long loop. Figure 5b shows that the consistency map can also be generated from a narrow and long corridor. As shown in Figure 5c,d, in the wide range of the parking lot, the robot traveled approximately 1600 m at a speed of 0.6 m/s; the trajectory is shown in Figure 5c. The map shown in Figure 5c is still accurate and clear.



(**a**) office area with a closed loop (**b**) a narrow and long corridor

(**c**) mapping trajectory in a parking lot (**d**) a wide range of parking lot

Figure 5. The maps generated in the indoor and outdoor environments.

4.2. Large-Scale and Sparse Scenes Test

Two substations are selected to verify the validity of the MIM_SLAM algorithm in the larger-scale and sparse environments. A general substation comprises thousands of square meters, and its electrical equipment needs to include a sufficient distance for electrical safety. We first tested in substation A shown in Figure 6. During the experiment, the robot traveled approximately 600 m at a speed of 0.6 m/s and the 3D LIDAR sensor collected a total of 10,080 frames. The plan and the environment of substation A are shown in Figure 6. The red line denotes the robot's trajectory ABCDA. The point cloud maps established by the MIM_SALM and the LOAM [12] which has been considered state-of-the-art in LiDAR SLAM are shown in Figure 7.

Intuitively, the MIM_SLAM algorithm has a high consistency map compared with the LOAM. Due to the accumulated error, the estimated pose has a great deviation when the robot travels from area A along the red track shown in Figure 6, and then returns back to area A. The LOAM cannot effectively handle the data association at this moment, which leads to the wrong association and damage to the established map. In this paper, the multi-level ICP matching method takes into account this situation and generates a reliable constraint between the two poses. The incremental optimization method is used to optimize the global pose. Thus the final map has a high consistency.

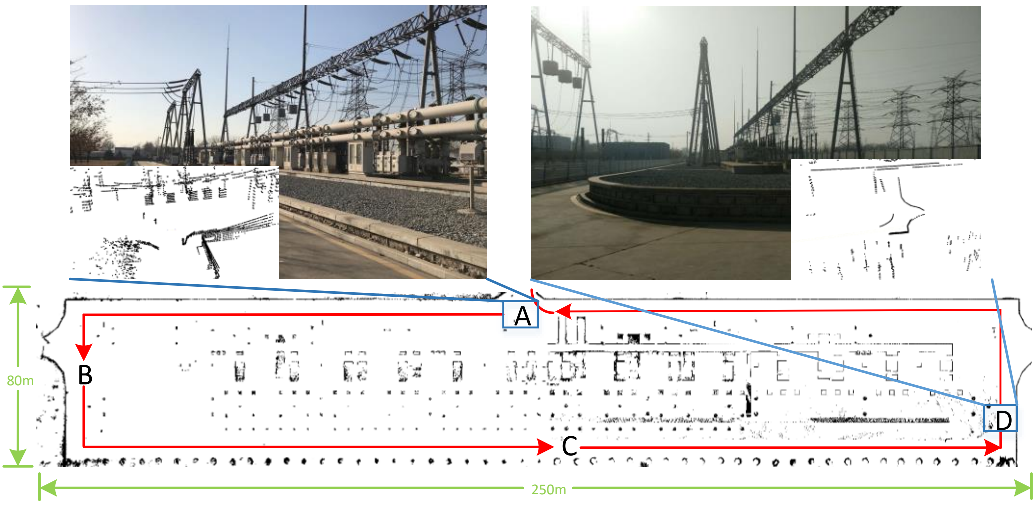


Figure 6. The plan and the environment of substation A.

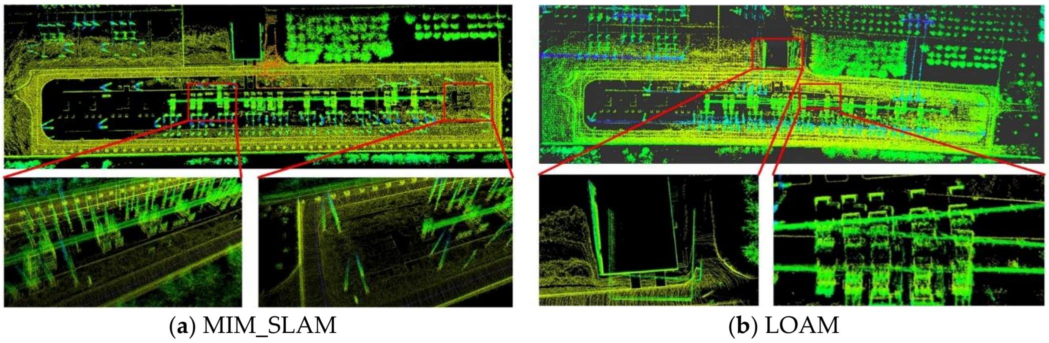


Figure 7. The point cloud maps established by the MIM_SLAM and the LOAM [12] algorithms.

Next, we tested it in substation B. Compared with substation A, this was larger. Meanwhile, the electrical equipment is placed more sparsely, and less information is gathered by the 3D LIDAR sensor. During the experiments, the robot traveled approximately 1450 m at a speed of 0.6 m/s, and the LIDAR sensor collected a total of 24,220 frames. The plan and the environment of substation B are shown in Figure 8. The red line denotes the robot's trajectory ABCDEABFGDC. The point cloud maps established by the MIM_SLAM algorithm and the LOAM are shown in Figure 9.

It can be seen from the observation of the two scenes given in Figure 8 that the electrical equipment is distributed more sparsely, and more than 70% of the laser rays reach the maximum measurement range. As shown in Figure 9a, both contour lines of the map established by the MIM_SLAM algorithm are at right angles at areas F and G. Additionally, when the robot returned back to the area C, the details of the point cloud map at area C showed that the telegraph poles and the road edge are clearly visible, and there is no incorrect point cloud accumulation. Because the LOAM cannot effectively process the data association of the closed-loop area, the performance is very poor in this situation, which has more closed-loop areas in large-scale environments, as shown in Figure 9b.

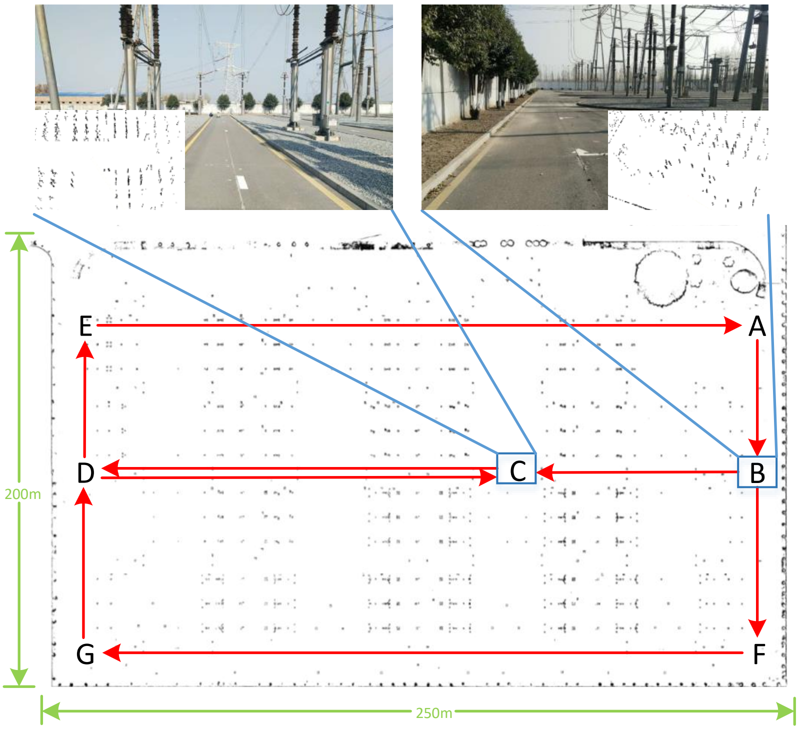
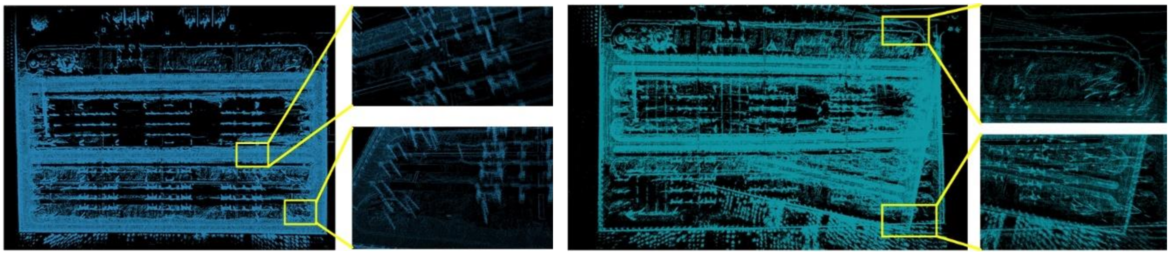


Figure 8. The plan and the environment of substation B.



(**a**) MIM_SLAM (**b**) LOAM

Figure 9. The point cloud maps established by the MIM_SLAM and the LOAM algorithms.

4.3. Mapping Accuracy Test

In order to analyze the mapping accuracy quantitatively, we selected two substations with large-scale and sparse characteristics which are shown in Figures 6 and 8, also the standard ICP [20] and the LOAM [12] algorithm was selected to compare with the MIM_SLAM algorithm. In the experiments, the robot is controlled running at a speed of 0.6 m/s on a flat road along a straight line with a length of 10 m. The accuracy of mapping is defined as the robot's localization error ε [28], expressed as follows:

$$\begin{gathered} \varepsilon = \tfrac{1}{N}\sum_{i=0}^{N-1} \|e_i\| \\ e_i = (x_0 \oplus T_i) - (x_0^* \oplus T_i^*) \end{gathered} \tag{16}$$

where x_0 and x_0^* denote the initial pose. $\oplus$ is the standard motion composition operator. T_i denotes the transformation matrix of the estimated pose x_i relative to the initial pose x_0. T_i^* denotes the transformation matrix of the true pose x_i^* relative to the initial pose x_0^*. $\|\cdot\|$ is the 2-Norm used in this paper. For the robot to navigate in three-dimensional environments, e_i is represented as $(x_e, y_e, z_e, \varphi_e, \psi_e, \theta_e)$. Then ε can be divided into two parts: translation error and rotation error.

$$\varepsilon = \frac{1}{N}\sum_{i=0}^{N-1} \|trans(e_i)\| + \frac{1}{N}\sum_{i=0}^{N-1} \|rot(e_i)\| \tag{17}$$

Since it is difficult to obtain the true pose of the robot, 20 points are selected from the robot's trajectory. The first point is taken as the initial pose. For the next 19 points, the difference value of the localization results relative to the initial pose is calculated. The difference value of the true pose can be measured by the tape. In Table 1, the localization error is tested with the standard ICP, the LOAM and the MIM_SLAM algorithms under substation A and B. It can be seen that the MIM_SLAM algorithm is more precise. If there is a closed-loop area on the robot's path, the MIM_SLAM algorithm will be better than the LOAM. The processing time per frame using different approaches is calculated and shown in Table 2. MIM_SLAM has a similar time consuming compared to the LOAM and meets the real-time requirements. In Section 4.4, this will be further analyzed in the benchmark dataset.

Table 1. The localization error results of different approaches/scenes.

	Standard ICP	**LOAM**	**MIM_SALM**
		Trans. Error (unit: m)	
A	0.213 ± 0.148	0.064 ± 0.057	0.052 ± 0.043
B	0.282 ± 0.177	0.075 ± 0.066	0.066 ± 0.062
		Rot. Error (unit: deg)	
A	2.5 ± 1.9	1.8 ± 1.2	1.6 ± 1.4
B	3.1 ± 2.2	2.5 ± 2.1	2.2 ± 1.8

Table 2. The average time consumption per frame for different approaches/scenes.

	Standard ICP	LOAM	MIM_SALM
		Processing Time (unit: s)	
A	0.0405	0.1036	0.0886
B	0.0416	0.1067	0.0932

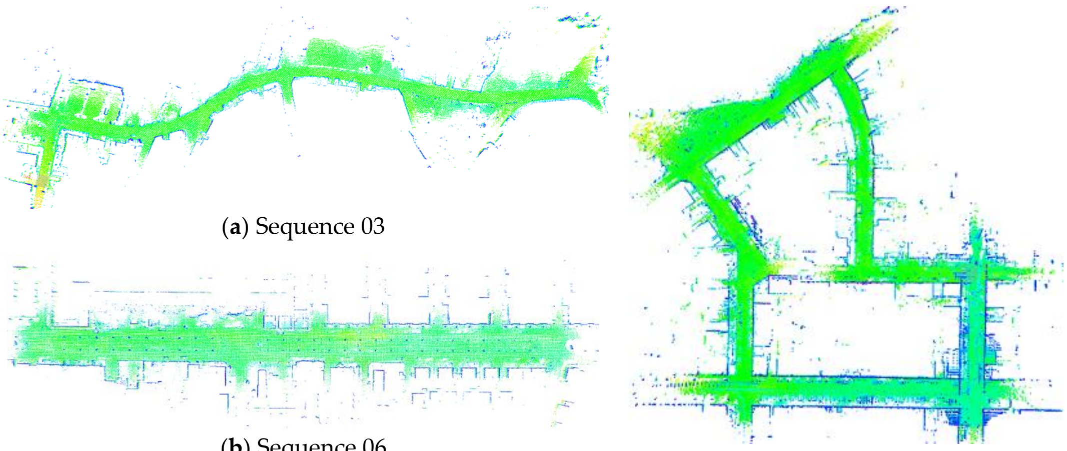
4.4. Benchmarking Datasets Test

Then, datasets from the KITTI odometry benchmark [29] are used to evaluated MIM_SLAM. The datasets take advantage of the autonomous driving platform Annieway to develop novel challenging real-world computer vision benchmarks. The autonomous driving platform is equipped with a 360° Velodyne HDL-64E laser scanner and two high-resolution color and grayscale video cameras. Accurate ground truth is provided by a GPS localization system. We selected three typical types of environments: "urban" with building around (sequence 07), "country" on small roads with vegetation in this scene (sequence 03), and "highway" where roads are wide and lower dynamic (sequence 06).

(1) Sequence 03: This dataset is designed to verify that MIM_SLAM can achieve a low drift pose estimation in sparse vegetation environment. The mapping result is shown in Figure 10a and the trajectory and the ground truth are shown in Figure 11a. Due to the scarce stable features in this scene, there is a bit of position deviation after 420 m of traveling compared with the ground truth in Figure 11a. To evaluate the pose estimation accuracy, we use the evaluation method in the KITTI odometry benchmark which calculated the translational and rotational errors for all possible subsequences of length (100, 200, . . . , 800) meters. As is shown in Figure 12a, both LOAM and MIM_SLAM achieve lower drift values compared with the standard ICP method, and MIM_SLAM is slightly worse than LOAM.

(2) Sequence 06: We use this dataset to verify that MIM_SLAM can effectively process the data association of the closed-loop area shown in Figures 10 and 11. From Figure 12b, it can be seen that the translational and rotational errors are lower compared to the standard ICP, and slightly better than LOAM.

(3) Sequence 07: This urban road scene is highly dynamic and large-scale. The vehicle travels approximately 660 m at a speed of 6.2 m/s. The mapping result is shown in Figure 10c and the trajectory and the ground truth are shown in Figure 11c. Intuitively, the details of the point cloud map are clear. The estimated pose and the ground truth are almost overlapping. The pose estimation error is shown in Figure 12c quantitatively.

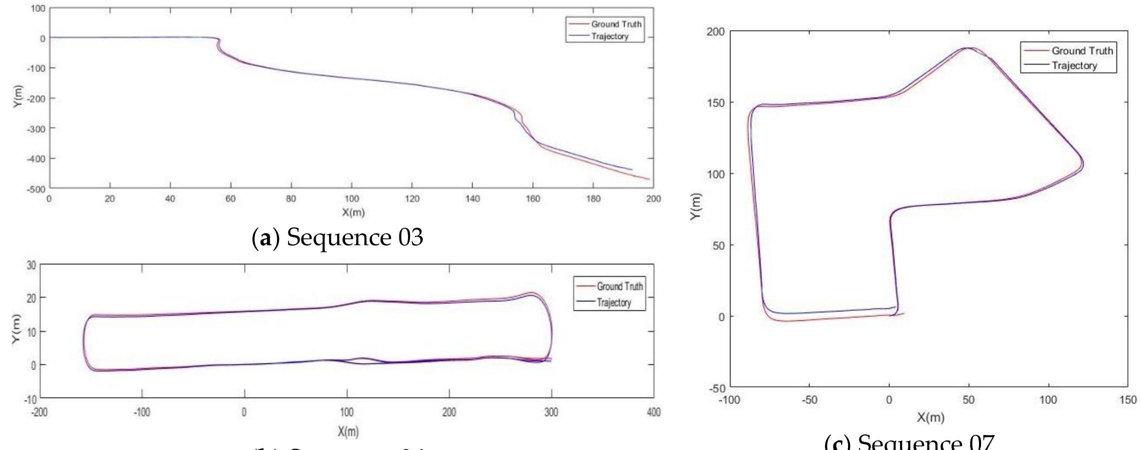


(**a**) Sequence 03

(**b**) Sequence 06

(**c**) Sequence 07

Figure 10. The mapping results in the three sequences.

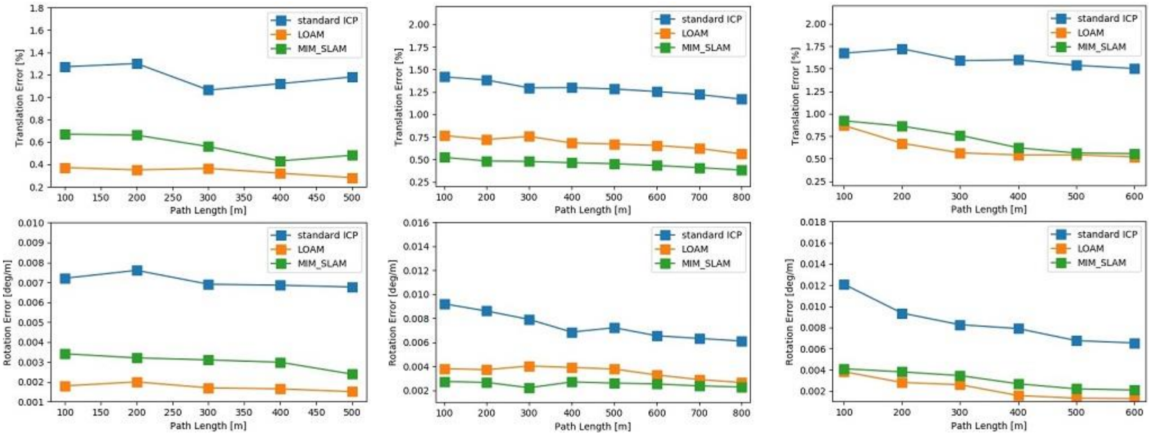


(**a**) Sequence 03

(**b**) Sequence 06

(**c**) Sequence 07

Figure 11. The trajectory produced by MIM_SLAM in the three sequences.



(**a**) Sequence 03 (**b**) Sequence 06 (**c**) Sequence 07

Figure 12. The pose estimation error in the three sequences.

5. Conclusions

For large-scale and sparse environments, we propose a novel MIM_SLAM algorithm in this paper, which simplifies the SLAM problem to the least-squares optimization problem. The contribution of this paper is that the multi-level ICP matching method is proposed to solve the data association problem and the uncertainty estimation is handled by the Fisher information matrix. It considers the accumulated errors in two aspects: matching between time-neighbor frames (building a low-drift LiDAR odometry) and matching between area-neighbor frames (dealing with the data association at a revisited area). Moreover, multi-level matching and incremental optimization reduce the computational complexity while ensuring mapping accuracy, we can achieve accurate mapping and real-time applications. Additionally, its application is not only limited to mobile robots, and it can potentially be extended to other vehicles, e.g., UAVs. Experimental results show that it can effectively build a high consistency map with a smaller amount of environmental information and the large-scale scenes, and it also achieves similar or better accuracy compared with the standard ICP and the state-of-the-art LOAM algorithm in the KITTI dataset. Additionally, there are some limitations in high dynamic scenes. Our future work will focus on the construction of a more robust data association method, we will integrate the output of our multi-level ICP matching with an IMU in a Kalman filter to further reduce the accumulated error.

Author Contributions: This study was completed by the co-authors. J.W. conceived and led the research. The major experiments and analyses were undertaken by J.W. and M.Z. W.C. supervised and guided this study. J.W., M.Z. and W.C. wrote the paper. All the authors have read and approved the final manuscript.

Funding: This research was partly funded by the National Natural Science Foundation of China (Grant No. 61773261) and the National Key R&D Program of China (Grant No. 2017YFC0806501).

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

References

1. Martínez, J.L.; Morán, M.; Morales, J.; Reina, A.J.; Zafra, M. Field Navigation Using Fuzzy Elevation Maps Built with Local 3D Laser Scans. *Appl. Sci.* **2018**, *8*, 397. [CrossRef]
2. Smith, R.; Cheeseman, P. On the Estimation and Representations of Spatial Uncertainty. *Int. J. Robot. Res.* **1987**, *5*, 56–68. [CrossRef]
3. Shinsuke, M.M.D.; Noboru, I.M.D.; Yoshito, I.M.D. Estimating Uncertain Spatial Relationships in Robotics. *Mach. Intell. Pattern Recognit.* **2013**, *5*, 435–461.
4. Thrun, S.; Liu, Y.; Koller, D.; Ng, A.Y.; Ghahramani, Z.; Durrant-Whyte, H. Simultaneous localization and mapping with sparse extended information filters. *Int. J. Robot. Res.* **2004**, *23*, 693–716. [CrossRef]
5. Montemerlo, M.S. Fastslam: A factored solution to the simultaneous localization and mapping problem with unknown data association. *Arch. Environ. Contam. Toxicol.* **2015**, *50*, 240–248.
6. Gutmann, J.S.; Konolige, K. Incremental mapping of large cyclic environments. In Proceedings of the 1999 IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '99), Monterey, CA, USA, 8–9 November 2002; pp. 318–325.
7. Zlot, R.; Bosse, M. Efficient large-scale 3D mobile mapping and surface reconstruction of an underground mine. In *Field and Service Robotics*; Springer: Berlin/Heidelberg, Germany, 2014; pp. 479–493.
8. Olufs, S.; Vincze, M. An efficient area-based observation model for monte-carlo robot localization. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009), St. Louis, MO, USA, 10–15 October 2009; pp. 13–20.
9. Huang, S.; Wang, Z.; Dissanayake, G. Sparse local submap joining filter for building large-scale maps. *IEEE Trans. Robot.* **2008**, *24*, 1121–1130. [CrossRef]
10. Moosmann, F.; Stiller, C. Velodyne slam. In Proceedings of the 2011 IEEE Intelligent Vehicles Symposium (IV), Baden, Germany, 5–9 June 2011; pp. 393–398.
11. Nüchter, A.; Lingemann, K.; Hertzberg, J.; Surmann, H. 6D SLAM-3D mapping outdoor environments. *J. Field Robot.* **2007**, *24*, 699–722. [CrossRef]
12. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-time. In Proceedings of the 2014 Robotics: Science and Systems, Berkeley, CA, USA, 12–16 July2014; Volume 2.
13. Wang, J.; Li, L.; Zhe, L.; Weidong, C. Multilayer matching SLAM for large-scale and spacious environments. *Int. J. Adv. Robot. Syst.* **2015**, *12*, 124. [CrossRef]
14. Liu, Z.; Chen, H.; Di, H.; Tao, Y.; Gong, J.; Xiong, G.; Qi, J. Real-Time 6D Lidar SLAM in Large Scale Natural Terrains for UGV. In Proceedings of the 2018 IEEE Intelligent Vehicles Symposium (IV), Changshu, China, 26–30 June 2018; pp. 662–667.
15. Liang, X.; Chen, H.; Li, Y.; Liu, Y. Visual laser-SLAM in large-scale indoor environments. In Proceedings of the IEEE International Conference on Robotics and Biomimetics, Qingdao, China, 3–7 December 2017; pp. 19–24.
16. Hess, W.; Kohler, D.; Rapp, H.; Andor, D. Real-time loop closure in 2D LIDAR SLAM. In Proceedings of the IEEE International Conference on Robotics and Automation, Stockholm, Sweden, 16–21 May 2016; pp. 1271–1278.
17. Kaess, M.; Ranganathan, A.; Dellaert, F. iSAM: Incremental smoothing and mapping. *IEEE Trans. Robot.* **2008**, *24*, 1365–1378. [CrossRef]
18. Hartley, H.O. The modified Gauss-Newton method for the fitting of non-linear regression functions by least squares. *Technometrics* **1961**, *3*, 269–280. [CrossRef]
19. Moré, J.J. The Levenberg-Marquardt algorithm: Implementation and theory. In *Numerical Analysis*; Springer: Berlin/Heidelberg, Germany, 1978; pp. 105–116.
20. Besl, P.J.; Mckay, N.D. Method for registration of 3-D shapes. *IEEE Trans. Pattern Anal. Mach. Intell.* **2002**, *14*, 239–256. [CrossRef]

21. Li, L.; Liu, J.; Zuo, X.; Zhu, H. An Improved MbICP Algorithm for Mobile Robot Pose Estimation. *Appl. Sci.* **2018**, *8*, 272. [CrossRef]
22. Chen, Y.; Medioni, G. Object modelling by registration of multiple range images. *Image Vis. Comput.* **1992**, *10*, 145–155. [CrossRef]
23. Rusu, R.B.; Cousins, S. 3D is here: Point Cloud Library (PCL). In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1–4.
24. Liu, Z.; Chen, W.; Wang, Y.; Wang, J. Localizability estimation for mobile robots based on probabilistic grid map and its applications to localization. In Proceedings of the 2012 IEEE Conference on Multisensor Fusion and Integration for Intelligent Systems (MFI), Hamburg, Germany, 13–15 September 2015; pp. 46–51.
25. Wang, Y.; Chen, W.; Wang, J.; Wang, H. Active global localization based on localizability for mobile robots. *Robotica* **2015**, *33*, 1609–1627. [CrossRef]
26. Bobrovsky, B.; Zakai, M. A lower bound on the estimation error for certain diffusion processes. *IEEE Trans. Inf. Theory* **1976**, *22*, 45–52. [CrossRef]
27. Wang, B.; Guo, R.; Li, B.; Han, L.; Sun, Y.; Wang, M. SmartGuard: An autonomous robotic system for inspecting substation equipment. *J. Field Robot.* **2012**, *29*, 123–137. [CrossRef]
28. Burgard, W.; Stachniss, C.; Grisetti, G.; Steder, B.; Kümmerle, R.; Dornhege, C.; Ruhnke, M.; Kleiner, A.; Tardös, J.D. A comparison of SLAM algorithms based on a graph of relations. In Proceedings of the 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2009), St. Louis, MO, USA, 10–15 October 2009; pp. 2089–2095.
29. Geiger, A. Are we ready for autonomous driving? The KITTI vision benchmark suite. In Proceedings of the 2012 Computer Vision and Pattern Recognition, Providence, RI, USA, 16–21 June 2012; pp. 3354–3361.