Available online at www.sciencedirect.com
ScienceDirect
Energy Reports 9 (2023) 898–908
www.elsevier.com/locate/egyr

2022 The 3rd International Conference on Power and Electrical Engineering (ICPEE 2022)
29–31 December, Singapore

Research on positioning and mapping algorithm of sliding window optimization for substation monitoring robot

Zhen-yu Liu[a,*], Jing Wang[a], Ze-yu Hui[a], Bin Zhao[b], Fu-li Zhang[a]

[a] *School of Information Science and Engineering, Shenyang University of Technology, Shenyang 110870, China*
[b] *School of Information Science & Engineering. Northeastern University, Shenyang 110819, China*

Received 18 April 2023; accepted 20 May 2023
Available online xxxx

Abstract

This paper proposed a positioning and mapping algorithm of sliding window optimization for substation monitoring robots to solve the problems of low precision location and poor robustness of the existing laser odometer in power inspection outdoor scene mapping. A tightly coupled simultaneous localization and mapping (SLAM) algorithm was proposed based on 16-wire LiDAR and inertial measurement unit (IMU). Firstly, the paper estimated the IMU and corrected the motion distortion of the laser point cloud by linear interpolation. Secondly, scene features were extracted by curvature and classified according to different feature properties. The local map was constructed in the sliding window using the inter-frame matching module. Finally, the joint optimization function was built using the distance and IMU data obtained by matching the frame with the local map. The paper used the KITTI and self-recorded datasets to conduct the experiments. The results show that the improved method's accuracy outperforms the lightweight and ground-optimized LiDAR odometry and Mapping (Lego-LOAM) and LiDAR inertial odometry and mapping (LIO-Mapping) and draws a broad application prospect in power inspection field.

Peer-review under responsibility of the scientific committee of the 2022 the 3rd International Conference on Power and Electrical Engineering, ICPEE, 2022.

Keywords: Simultaneous localization and mapping; LiDAR; IMU tight coupling; Local map

1. Introduction

The manual inspection of substation equipment is not only dangerous to people under high voltage, ultra-high voltage, and bad weather conditions, but also brings certain hidden dangers to the safe operation of the power grid [1]. Oppositely, automatic inspection robot has the advantages of high efficiency, convenience, precision, and safety. Thus, various research has been conducted to adapt inspection robots. Huang et al. designed an automatic reading system with a pan–tilt–zoom camera to solve the cost problem of reading instrument data [2]. Liu et al. proposed an accurate alignment method for PTZ of a substation inspection robot based on a monocular

* Corresponding author.
E-mail address: liuzhenyu@sut.edu.cn (Z.-y. Liu).

https://doi.org/10.1016/j.egyr.2023.05.166

Peer-review under responsibility of the scientific committee of the 2022 the 3rd International Conference on Power and Electrical Engineering, ICPEE, 2022.

camera [3]. Xiao et al. proposed an adaptive pose adjustment method for a multi-DOF guideway inspection robot in a substation [4]. Su et al. developed an intelligent inspection robot for substations to support a real-time deep-learning model and improve detection accuracy and reasoning time [5]. Nevertheless, the system focused more on fault diagnosis. Accurate pose estimation has been used to ensure the robot can calculate its position and motion and provide data support for simultaneous localization and mapping (SLAM) and navigation. Sun et al. proposed an improved particle filter and path-planning algorithm for laser navigation [6]. Nevertheless, the paper did not compare the proposed method's performance with other methods. Chen et al. used geometric and visual features of LiDAR to complete the closed-loop detection, to improve LiDAR SLAM performance [7]. However, while the strict parameters and rules guarantee the accuracy of loop closure detection, some loop closures still need to be included. To realize closed-loop detection, Chen et al. used depth learning to exploit different types of information generated from LiDAR scans [8]. The method outperformed two state-of-the-art methods in two datasets and generalized well to new environments. Karam et al. applied the inertial measurement unit (IMU) to LiDAR-based SLAM and proposed an IMU-SLAM switching strategy to increase the robustness [9]. Zhang et al. divided the simultaneous positioning and mapping problem into two algorithms to reduce the drift of the LiDAR odometer and achieve good matching and registration of point cloud data at different frequencies [10]. However, for the current LiDAR odometer, the low positioning accuracy and poor robustness greatly limit the practical application of inspection robots in the outdoor environment.

In addition, the manual inspection is mainly realized by seeing, touching, listening, smelling, and other senses. Therefore, the automatic inspection robots should integrate sensor technology, motion control, and wireless communication [11]. An inspection robot's system framework mainly includes mobile stations and a base station, which include a driving system, a control system, and a dedicated sensor [12]. Specifically, as to the mobile station, various sensors or detectors are used to realize the inspection of the outdoor high-voltage equipment of the substation and timely detect equipment abnormalities, such as thermal defects of power equipment and foreign body suspension, etc. [13]. However, the basic system framework and core algorithm are different due to the focus and target problems of existing research. Zhang et al. realized robust and low-drift laser optical inertial odometer measurement and mapping by integrating 3D laser scanner, camera, and IMU data [14]. The same goal was achieved using the system architecture of a 3D laser scanner alone in another research [15]. Differently, Yin et al. realized 3D position recognition through a depth fusion network, which improved the accuracy of position recognition and optimized the reasoning time simultaneously [16].

Therefore, based on the above analysis, this paper proposed a positioning and mapping algorithm of sliding window optimization by using the tightly coupled data fusion scheme of LiDAR and IMU to estimate the pose through the data between the two sensors and jointly optimize [17]. Our main contributions were as follows: (1) The structural framework of the system was given, and the algorithm principle was analyzed. (2) The paper proposed an improved ICP point cloud registration scheme. (3) The proposed algorithm was tested on KITTI and Plant datasets.

The remainder of this article is organized as follows. Section 2 describes the system framework and algorithm theory. In Section 3, LiDAR and IMU are analyzed. Section 4 introduces the improved odometer. Section 5 presents the experimental results. Finally, Section 6 concludes.

2. System framework of improved Lego-LOAM algorithm

The accuracy of the odometer is low, especially in extensive scene mapping. The system cannot recognize a loop closure when the robot returns to the origin again, so the global pose error cannot be corrected. Meanwhile, the Lego-LOAM algorithm's overall accuracy is high. Therefore, we use the loop closure detection module to correct the global pose and propose an improved algorithm based on Lego-LOAM.

The framework of the substation detection robot system in this paper is shown in Fig. 1. The IMU data is pre-integrated before the LiDAR data arrive. When the LiDAR data S_j is obtained, the IMU pre-integration transforms, and $T_{L_{j'}}^{W}$, $T_{L_j}^{W}$ are used to linearly interpolate the data to correct the distortion of the point cloud, and the processed point cloud $\tilde{S}_j$ is obtained. The features extracted from $\tilde{S}_j$ are recorded as F_{L_j}. The characteristic points from the oth time to the ith time are estimated according to the posture of IMU, $F_{o,i}$. The pose estimation is performed by correlating the characteristics of the LiDAR at time j and time i to calculate the distance between the two frames, $m_{i,j}$, and construct the constraint relationship. The cost function is constructed by using IMU pre-integration data, Δp_{ij}, Δv_{ij}, Δq_{ij} and the distance between two frames, where Δp_{ij} is the position, Δv_{ij} is the measured speed, Δq_{ij} is the rotation quaternion. The Lego-LOAM is used to improve the pose estimation accuracy.

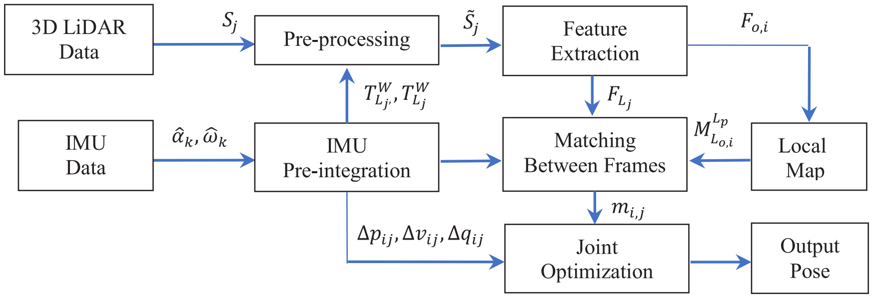

Fig. 1. Improved algorithm odometer framework.

3. Data analysis of LiDAR and IMU

This part describes the models of LiDAR and IMU, the error model, and the calibration method. The definition and relationship of the coordinate system of mobile robots, LiDAR, and IMU, are described. Finally, the synchronization method of the time stamp is described, which provides support for the next chapter.

3.1. LiDAR and IMU external parameter calibration

External parameter calibration calculates a conversion parameter of the coordinate system between LiDAR and IMU. The calibration accuracy directly affects the accuracy of positioning and mapping. Especially in large scenes, the calibration error increases with the system operation time and will be added to the accumulated error. The mobile robot in this project is different from the IMU in position, and the IMU is fixed directly above the laser radar. Theoretically, the external parameters between the sensors can be calibrated, but the positions of the two sensors should be as close as possible. The sensors can record the same data group in two ways and have different representations in their respective coordinate systems. To ensure that LiDAR and IMU accurately represent data group, it is necessary to calculate the transformation matrix between them and put the detected points under the same coordinate system, that is, joint calibration of sensors. Sensor coordinates are shown in Fig. 2.

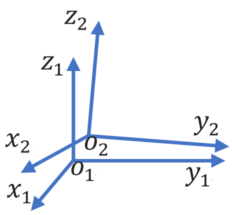

Fig. 2. Sensor coordinates.

Let the coordinate of a point in space under $(o_1x_1y_1z_1)$ be (a_1, b_1, c_1). The coordinate under $(o_2x_2y_2z_2)$ is (a_2, b_2, c_2), and the transformation matrix under the two coordinate systems is denoted as T, The relationship between the coordinates of two points is shown in formula (1):

$$(a_1, b_1, c_1)^T = T^T (a_2, b_2, c_2) \tag{1}$$

The transformation matrix solution is shown in formula (2):

$$T = \begin{bmatrix} t_{11} & t_{12} & t_{13} & t_{14} \\ t_{21} & t_{22} & t_{23} & t_{24} \\ t_{31} & t_{32} & t_{33} & t_{34} \\ t_{41} & t_{42} & t_{43} & t_{44} \end{bmatrix} = \begin{bmatrix} R & 0 \\ t & 1 \end{bmatrix} \tag{2}$$

where R is the rotation matrix, and t is the translation matrix.

The specific steps are as follows:

(1) Determine the point cloud data of the marker through the LiDAR.

(2) Calculate the original data of IMU, and obtain the attitude angle, and estimate the transformation matrix between the current state and the world coordinate system, and calculate the marker coordinates.

(3) Determine the three-dimensional coordinates of the marker in the LiDAR coordinate system.

(4) Match the data of the markers in the two coordinate systems.

(5) Convert the marker data in IMU coordinates into LiDAR coordinates and compare the two data. The algorithm is iterated continuously, and the external parameters are obtained after convergence.

3.2. Coordinate system definition and transformation

3.2.1. LiDAR coordinate system

The paper expresses the point cloud data scanned by LiDAR in its coordinate system. The global coordinate system is the position of the first frame of the LiDAR after system start-up. When the mobile robot moves, the point cloud data obtained at different times must be converted to the global coordinate system before the carrier state can be estimated. It shows the coordinate transformation in formula (3).

$$T_L^W = \left(tx_L^W, ty_L^W, tz_L^W, \phi x_L^W, \theta y_L^W, \varphi z_L^W\right) = \left(p_L^W, q_L^W\right) \tag{3}$$

where T_L^W is the transformation matrix between the LiDAR and the global coordinate system, and tx_L^W, ty_L^W and tz_L^W are the translation amounts on the three axes respectively, and ϕ_L^W, θ_L^W and φ_L^W respectively correspond to Euler angles, roll angle, pitch angle and yaw angle.

Assuming that X^W and X^L are the states of the object to be measured in the global coordinates and the LiDAR coordinates respectively, thus:

$$X^W = RX^L + q_L^W \tag{4}$$

where R needs to be obtained according to Euler-angle, as shown in formula (5):

$$R = \begin{bmatrix} cos\theta cos\varphi & cos\phi sin\varphi + sin\phi sin\theta cos\varphi & sin\phi sin\varphi - cos\phi sin\theta sin\varphi \\ -cos\theta sin\varphi & cos\phi cos - sin\phi sin\theta sin\varphi & sin\phi cos\varphi + cos\phi sin\theta sin\varphi \\ sin\theta & -sin\phi cos\theta & cos\phi cos\theta \end{bmatrix} \tag{5}$$

Generally, a rotation matrix is not selected for operation in actual operation. Since it needs to represent the rotation of three axes through nine elements, it will consume additional memory when running the system, so it is usually expressed by a quaternion, as shown in formula (6):

$$q = q_0 + q_1\vec{i} + q_2\vec{j} + q_3\vec{k} \tag{6}$$

where i, j and k are complex numbers, and the relationship between them is as follows:

$$\begin{cases} i^2 = j^2 = k^2 = -1 \\ ij = k, ji = -k \\ jk = i, kj = -1 \\ ki = j, ik = -j \end{cases} \tag{7}$$

Assume a unit vector is $\vec{\varepsilon} = [n_x, n_y, n_z]$, and the object's rotation angle is θ around the vector. The quaternion is:

$$q = \left[cos\frac{\theta}{2}, n_x sin\frac{\theta}{2}, n_y sin\frac{\theta}{2}, n_z sin\frac{\theta}{2}\right] \tag{8}$$

The paper can obtain the conversion relationship between LiDAR and the global coordinate system through this conversion. In the experiment, a group of plane points is scanned under the LiDAR coordinates. The plane points under the radar coordinate system can be converted to the global coordinate system through coordinate conversion.

3.2.2. Inertial navigation coordinate system

In the inertial navigation coordinate system, the Z axis is along the Earth's axis of rotation, pointing to due north. The X axis is the intersection of the ecliptic plane of the earth's orbit around the sun and the plane of the Earth's equator Y axis is determined by the right-hand orthogonal criterion. The inertial navigation coordinate system is transformed with the carrier coordinate system. When the system is running, the current state of the carrier can be obtained by solving the transformation matrix between the inertial navigation coordinate system and the carrier coordinate system for mapping. Suppose the transformation matrix of the two coordinate systems is denoted as A, IMU rotates γ, θ and ψ respectively around the X, Z and Y axes at this time, and the two coordinate systems coincide, then A can be solved by formula (9):

$$A = \begin{bmatrix} cos\theta cos\psi & sin\theta & -cos\theta sin\psi \\ -cos\gamma cos\psi sin\theta + sin\gamma sin\psi & cos\gamma cos\theta & cos\gamma sin\psi sin\theta + sin\gamma cos\psi \\ sin\gamma cos\psi sin\theta + cos\gamma sin\psi & -sin\gamma cos\theta & -sin\gamma sin\psi sin\theta + cos\gamma cos\psi \end{bmatrix} \tag{9}$$

The matrix A is an orthogonal matrix. Thus the transposition of the matrix can represent the state transition matrix between the carrier and the IMU. It is shown in formula (10):

$$C = A^T \tag{10}$$

Finally, three Euler angles are obtained, as shown in formula (11) and formula (12):

$$\psi = \begin{cases} arctan\dfrac{-C_{31}}{C_{11}}, (C_{11>0}) \\ arctan\dfrac{-C_{31}}{C_{11}} - \pi, (C_{11} < 0, C_{31} > 0) \\ arctan\dfrac{-C_{31}}{C_{11}} + \pi, (C_{11} < 0), \end{cases} \tag{11}$$

$$\begin{bmatrix} \gamma \\ \theta \end{bmatrix} = \begin{bmatrix} arctan\dfrac{-C_{23}}{C_{22}} \\ arctanC_{21} \end{bmatrix} \tag{12}$$

where C_{ij} represents the direction cosine matrix, and ij represent the elements of the $i$row and the j column.

4. Theory of mileage calculation method

4.1. Point cloud segmentation

The paper will extract some unstable points from features, including occlusion and outlier points. Such feature points will disappear during the movement of the robot, resulting in mismatching in the feature-matching stage and decreasing the accuracy of pose estimation. Here, it uses the clustering method to mark some of the geometric features of the point cloud that may be the same object and uses the marked feature point cloud for processing in the subsequent matching process. It is mainly divided into two parts, as shown in Fig. 3.

For obtaining the features on the same horizontal laser line, the adjacent points with the depth difference within the threshold are firstly selected, and the threshold is determined according to the following formula (13). Where, $\|OF_{i,k}\|$ is the point with a large depth value of the feature point, denoted as d_1, and $\|OF_{i,k+1}\|$ is the point with a small depth value of the feature point, denoted as d_2, α is the resolution between two adjacent feature points.

$$\begin{cases} \Delta d = \|OF_{i,k}\| - \|OF_{i,k+1}\| = d_1 - d_2 \times cos\alpha \\ \beta = arctan\dfrac{d_2 \times sin\alpha}{d_1 - d_2 \times cos\alpha} \end{cases} \tag{13}$$

After rough segmentation, the object's weak points in the LiDAR scanning process can be effectively removed. The different features are classified according to the laser line depth. A threshold parameter is set to determine whether the features of the two adjacent points are divided into two categories or merged into one category.

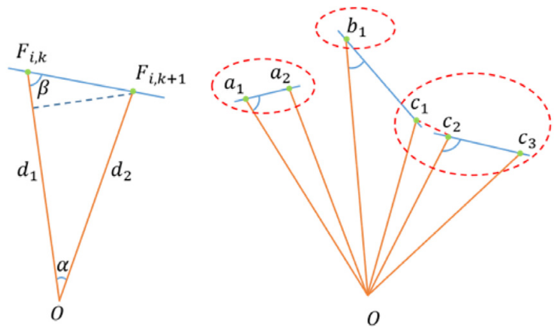

Fig. 3. Point cloud clustering.

4.2. Improved ICP point cloud registration

In the traditional ICP matching algorithm, the point clouds of the front and back frames need to traverse all points to calculate the centroid and match, resulting in a large error in the iterative calculation process, a large amount of calculation, a long time and poor applicability on the mobile robot platform. The preprocessing of point clouds has been described above. Next, the processed point cloud is voxelized to filter out some redundant point clouds and improve matching efficiency. The specific steps are as follows:

(1) Find the boundary on the X, Y and Z axes for the current point cloud data, and record it as $: x_{min}$, $x_{max}, y_{min}, y_{max}, z_{min}, z_{max}$.

(2) Make a difference between the maximum and minimum values on each axis to find the side length of the cube surrounding the point cloud, and record it as: l_x, l_y, l_z.

(3) Assuming that each voxel contains k point cloud data, the total number of point cloud data at the current time is known to be n, and the voxelized grid side length cell is calculated as shown in formula (14):

$$cell = \sqrt[3]{\frac{k \times l_x \times l_y \times l_z}{n}} \tag{14}$$

(4) Given the side length of each voxel lattice, the total number of voxels is obtained as shown in formula (15):

$$\begin{cases} num = M \times N \times L \\ \begin{cases} M = \left\lfloor \frac{l_x}{cell} \right\rfloor \\ N = \left\lfloor \frac{l_y}{cell} \right\rfloor \\ L = \left\lfloor \frac{l_z}{cell} \right\rfloor \end{cases} \end{cases} \tag{15}$$

where M, N and L respectively correspond to the number of squares in the three axis directions, $\lfloor \blacksquare \rfloor$ represents a downward rounding symbol, num represents the total number of prime elements.

(5) Mark all the voxel squares one by one and calculate the centroid. If the centroid exists, select the point as the centroid. Otherwise, select the point closest to the centroid as the centroid, as shown in formula (16):

$$\begin{cases} \begin{cases} i = \frac{x_i - x_{min}}{cell} \\ j = \frac{y_i - y_{min}}{cell} \\ k = \frac{z_i - z_{min}}{cell} \end{cases} \\ c_{ijk} = \frac{1}{k} \sum_{i=1}^{k} p_i \end{cases} \tag{16}$$

where, i, j, k represent the square where each point cloud is located, c_{ijk} represents the center of mass, k is the total number of point clouds, and P_i represents the data of each point cloud.

The computational cost can be effectively reduced by rasterizing the point cloud voxels, and the initial value obtained by IMU pre-integration can effectively reduce the number of iterations.

4.3. Optimization of lidar odometer

The first reference frame is fixed by fusing the relative position and attitude. Estimated states X_B^W, T_B^L, and relative radar measurements are used to constrain the radar state. Since a single scanning point is not enough to calculate the real relationship, a local map is established. Firstly, the feature points from time p to j are included, where p is the first scan in the local map, and time p to j is the optimization window, as shown in Fig. 4.

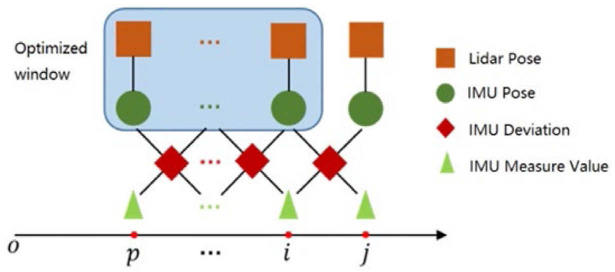

Fig. 4. Optimization window.

The correspondence between the feature point $F_{L\alpha}$ at the current time and the local map can be found by establishing the local map, where $\alpha \in \{p+1, K, i\}$ feature points include plane points and edge points of the previous feature extraction part.

In the optimization window, $T_{L\alpha}^{Lp}$ is used to transform the features in the optimization window to obtain $F_{L\alpha}^{Lp}$ at time p. Since it is a radar state estimated by IMU state, $T_{L\alpha}^{Lp}$ t is expressed as:

$$T_{L\alpha}^{Lp} = T_B^L T_{Bp}^{W^{-1}} T_{B\alpha}^W T_B^{L^{-1}} = \begin{bmatrix} R_{L\alpha}^{Lp} & t_{L\alpha}^{Lp} \\ 0 & 1 \end{bmatrix} \tag{17}$$

For each feature point x_{Lp} in the quasi synthetic plane, a linear equation group is constructed for the existing points, and the plane coefficient $\omega^T x' + d = 0$ can be calculated by solving it. In the formula, ω is the plane normal vector, and d is the intercept in the coordinate system at time p.

Define a distance factor $m = [x, \omega, d]$, record the relative measurement value from the feature point to the face of each $x \in F_{L\alpha}$, and the residual can be expressed as:

$$r_L = (m, T_{Lp}^W, T_{L\alpha}^W, T_B^L) = \omega^T (R_{L\alpha}^{Lp} x + t_{L\alpha}^{Lp}) \tag{18}$$

In formula (18), T_{Lp}^W and $T_{L\alpha}^W$ are the transformation matrix from the frame scanned by the radar at time p and time α to the world coordinate system respectively.

The sliding window helps to limit the amount of calculation. When the new pose constraint appears, the pose constraints at the earliest time in the edge window. There are α_n IMU status information. The overall variables to be estimated are:

$$X = \left[X_{Bp}^W, \ldots, X_{Bj}^W, T_B^L\right] \tag{19}$$

Construct the cost function to solve the global pose estimation of the state variable X, as shown in formula (20):

$$\min_X \frac{1}{2} \left\{ \sum_{m=m_{Lp}}^{m_{Lj}} \|r_L(m, X)\|^2 + \sum_{n=p}^{j-1} \left\| r_B(z_{n+1}^n, X) \right\|^2 \right\} \tag{20}$$

where $r_L(m, X)$ is the residual of relative LiDAR constraint. $r_B(z^n_{n+1}, X)$ is the IMU residual, which can be obtained from the state and pre integration. It can be expressed as:

$$r_B(z^n_{n+1}, X) = \begin{bmatrix} R_n^T(p_{n+1} - p_n - v_n\Delta t - \frac{1}{2}g^W\Delta t^2) - \Delta p \\ R_n^T(v_{n+1} - v_n - g^W\Delta t) - \Delta v \\ 2[\Delta q^{-1} \otimes q_n^{-1} \otimes q_{n+1}]_{xyz} \\ b_{a_{n+1}} - b_{a_n} \\ b_{g_{n+1}} - b_{g_n} \end{bmatrix} \tag{21}$$

The cost function in the form of nonlinear least squares can be solved by Gauss Newton method in the form of $H\delta X = -b$, which is solved by Google's nonlinear optimization library Ceres Solver.

5. Experimental results and analysis

The hardware and software configuration of the experimental platform is shown in Table 1.

Table 1. Hardware and software configuration.

Software configuration		Hardware configuration	
Operation system	Ubuntu 18.04	Notebook	i7 processor 8G memory
ROS	ROS Melodic	Mobile platform	Mobile robot
Development environment	Vs code	LiDAR	RS-LiDAR-16
Point cloud processing	Cloud compare	Ins	Xsens MTi-G-710

5.1. Analysis of pre-processing results

In this stage, we mainly deal with the point cloud distortion caused by the movement of the mobile robot and provide an excellent prior condition for the subsequent point cloud processing.

According to the self-recorded dataset of the parking lot of Shenyang SIASUN phase III company, the paper tests the algorithm and shows the result of point cloud processing before and after correction in Fig. 5. The comparison between Fig. 5(a) and (b) shows that the preprocessing algorithm proposed in this paper can effectively reduce the distortion of point clouds.

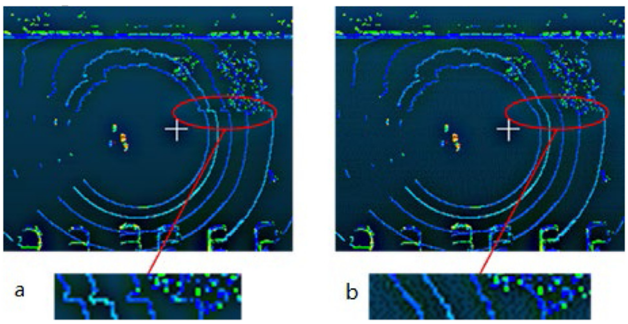

Fig. 5. (a) point cloud preprocessing before correction; (b) point cloud preprocessing after correction.

5.2. Analysis of feature extraction and ground segmentation results

Fig. 6 shows the results of different features after point cloud segmentation. Fig. 6(a) shows the feature extraction result for the point cloud after pre-processing. Fig. 6(b) shows the effect of ground feature extraction, where different colors represent the reflectance of the laser-scanning object. Fig. 6(c) is the feature extracted from the ground points.

In each evenly divided area, only the points with excellent smoothness are calculated and classified according to the ground features. The edge features such as tree trunks, road signs, and wall corners are classified through

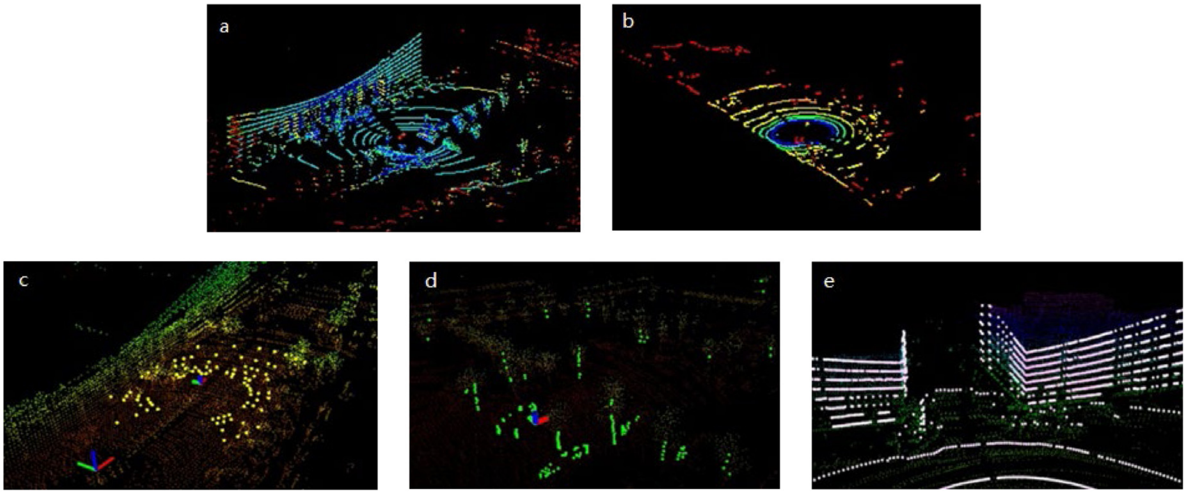

Fig. 6. (a) single frame point cloud feature; (b) ground point cloud; (c) ground characteristics; (d) angular characteristics; (e) edge characteristics. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the clustering of point clouds, as shown in Fig. 6(d). The green feature points are the extracted edge features. The classified surface features are shown in Fig. 6(e).

5.3. Accuracy analysis of odometer

5.3.1. KITTI data

The KITTI00 dataset is used for testing, and the accuracy of the odometer output trajectory and the actual value is evaluated for Lego-LOAM [18], LIO-Mapping [19], and the improved algorithm. The absolute position error (APE) and root mean square error (RMSE) is selected as the positioning accuracy evaluation indicators, and the EVO plug-in is used to evaluate the trajectory. Fig. 7 is a map of the KITTI00 built in the improved scheme, wherein the red triangle is the starting point.

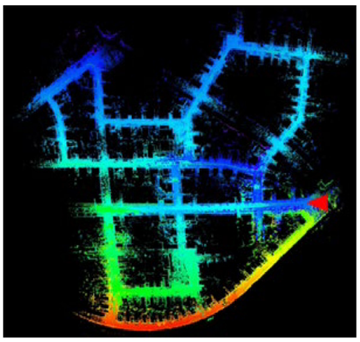

Fig. 7. KITTI00 mapping. . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 2 shows the absolute position and attitude errors of the three schemes under the kitti00 dataset. The minimum mean absolute error of the improved algorithm is 1.14 m compared with the two comparison algorithms. It is 60.1% and 34.2% higher than the result of the Lego-LOAM algorithm (2.86 m) and the LIO-Mapping algorithm (1.53 m), respectively. In comparing root mean square error, the improved algorithm is slightly lower than the

Table 2. Absolute pose error (m).

Algorithm	Average value	Maximum	RMSE
Lego-LOAM	2.86	5.11	3.02
LIO-Mapping	1.53	3.94	1.14
Improved	1.14	2.73	1.25

LIO-Mapping algorithm using the same tight coupling framework, but it is still better than the Lego-LOAM algorithm.

5.3.2. Plant data

Next, the improved algorithm's sliding window optimization pose module is tested. The dataset is selected from the factory office building with a total length of 943 m in outdoor scenes. By recording and displaying the optimized posture output trajectory, the test results of three schemes are shown in Fig. 8. It can be seen that the cumulative error of the Lego-LOAM algorithm increases gradually during slam operation in large scenes, as shown in Fig. 8(a). Moreover, the original loop detection module fails to perform a closed-loop optimization posture due to the significant drift, as shown in the circle. Fig. 8(b) and (c) show the LIO-Mapping trace and the improved algorithm trace, respectively. Because these two algorithms adopt a tightly coupled framework, they can still maintain an excellent ability to construct the graph and strong robustness when running long distances. By comparing the starting point and ending point trajectories of the two images shown in the circle part in Fig. 8(b) and (c), the improved algorithm is slightly better than LIO-Mapping in terms of cumulative drift processing.

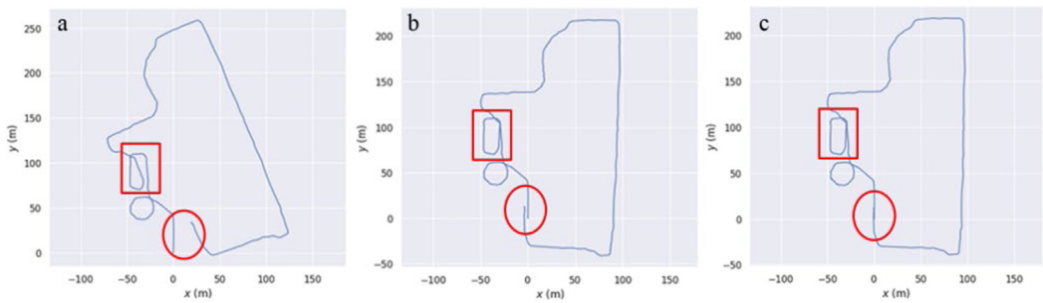

Fig. 8. (a) Lego-LOAM trajectory; (b) LIO-Mapping trajectory; (c) improved algorithm trajectory.

By calculating the position and pose changes of the odometer at the adjacent time stamp and the position and pose changes of the actual value at the adjacent time stamp, the obtained results are compared to obtain the relative position and pose error. In 943 m Park, the maximum drift error of the improved algorithm is less than 1 m. In addition, the square parts in Fig. 8(a) to (c) demonstrate that the improved algorithm is better than the other two algorithms. It can be seen that the tightly coupled odometer framework based on the optimization window can complete the localization and mapping work with a low drift during SLAM operation in large-scale outdoor scenes.

6. Conclusion

This paper designed the overall framework of the SLAM system for a mobile robot. The hardware and software composition, working principles of the mobile robot experimental platform, laser, and IMU used in this paper were described, and the ROS operating system was also described. This paper mainly introduced the construction of a tight coupling odometer positioning scheme based on laser IMU and described in detail the correction of point cloud distortion, feature extraction and classification, point cloud matching, and sliding window optimization. The substation equipment monitoring robot can automatically carry out the global path planning in the substation and independently complete the inspection of the substation equipment with the help of LiDAR and GPS global positioning system. Besides, it can complete the image inspection of the substation equipment, the equipment instrument's automatic identification, and the primary equipment's infrared detection, etc.

Furthermore, it can record the equipment information and provide an abnormal alarm. The monitoring task navigation experiment compared the improved algorithm with mainstream algorithms using the KITTI dataset. The results showed that the improved method possessed accuracy advantages. Since automatic inspection robots significantly reduce personnel costs and increase error detection efficiency, the system draws a broad application prospect in the power inspection field.

Declaration of competing interest

The authors declare no conflict of interest.

Data availability

The authors do not have permission to share data

Foundation

Major Science and Technology Innovation Project of Shandong Province (2019JZZY010128); Natural Science Foundation of Liaoning Province (20180520022).

References

[1] Wang C, Li Z, Kang Y, et al. Applying SLAM algorithm based on nonlinear optimized monocular vision and IMU in the positioning method of power inspection robot in complex environment. Math Probl Eng 2022;2022:1–14.
[2] Huang J, Wang J, Tan Y, Wu D, et al. An automatic analog instrument reading system using computer vision and inspection robot. IEEE Trans Instrum Meas 2020;69(9):6322–35.
[3] Liu J, Nie M, Wu H, Mai X. An image-based accurate alignment for substation inspection robot. In: 2018 International conference on power system technology. POWERCON, 2018, p. 4113–7.
[4] Xiao Y, Yan Y, Yu Y, Yu YS, et al. Research on pose adaptive correction method of indoor rail mounted inspection robot in GIS substation. Energy Rep 2022;8:696–705.
[5] Su L, Yang X, Cao B, et al. Development and application of substation intelligent inspection robot supporting deep learning accelerating. J Phys Conf Ser 2021;1(1754):012170.
[6] Weiwei Sun, Han Wang, Min Huang. Laser navigation system of mobile robot based on robot operating system. Sci Technol Eng 2019;19(202019):203–10.
[7] Chen S, Zhou B, Jiang C, et al. A lidar/visual SLAM backend with loop closure detection and graph optimization. Remote Sens 2021;13(14):2720.
[8] Chen X, Läbe T, Milioto A, et al. OverlapNet: Loop closing for LiDAR-Based SLAM. Proceeding of robotics: Science and systems. RSS, Corvalis, USA: RSS; 2020, XVI.009.
[9] Karam S, Lehtola V, Vosselman G. Strategies to integrate IMU and LiDAR SLAM for indoor mapping. ISPRS Ann Photogramm, Remote Sens Spatial Inf Sci 2020;1:223–30.
[10] Zhang J, Singh S. Low drift and real time LiDAR odometry and mapping. Auton Robots 2017;41(2):401–16.
[11] Quang HD, et al. Design a nonlinear MPC controller for autonomous mobile robot navigation system based on ROS. Int J Mech Eng Robot Res 2022;11(6):379–88.
[12] Jia S, Liu C, Wu H, et al. A cross-correction LiDAR SLAM method for high-accuracy 2D mapping of problematic scenario. ISPRS J Photogramm Remote Sens 2021;171(5):367–84.
[13] Zhou B, He Y, Qian K, et al. S4-SLAM: A real-time 3D LiDAR SLAM system for ground/water surface multi-scene outdoor applications. Auton Robots 2021;45(1):77–98.
[14] Zhang J, Singh S. Laser–visual–inertial odometry and mapping with high robustness and low drift. J Field Robotics 2018;35(82018):1242–64.
[15] Zhang J, Singh S. Low drift and real time lidar odometry and mapping. Auton Robots 2017;41(2):401–16.
[16] Yin P, Xu L, Zhang J, et al. Fusion VLAD: A multi-view deep fusion networks for viewpoint-free 3D place recognition. IEEE Robot Autom Lett 2021;6(2):2304–10.
[17] Bavle H, Puente PDeLa, How JP, et al. VPS-SLAM: visual planar semantic SLAM for aerial robotic systems. IEEE Access 2020;8:60704–18.
[18] Shan T, Englot B. Lego-LOAM: Lightweight and ground-optimized lidar odometry and mapping on variable terrain. In: IEEE/RSJ International conference on intelligent robots and systems. IROS, 2018, p. 4758–65.
[19] Ye H, Chen Y, Liu M. Tightly coupled 3D lidar inertial odometry and mapping. In: IEEE International conference on robotics and automation. ICRA, 2019, p. 3144–50.