

A Robust 2D Lidar SLAM Method in Complex Environment

Shan HUANG¹, Hong-Zhong HUANG^{1*}, Qi ZENG², and Peng HUANG³

¹Center for System Reliability and Safety, University of Electronic Science and Technology of China, Chengdu 611731, China

²EvenTec Co., Ltd., Chengdu 610097, China

³Jiangxi University of Science and Technology, Ganzhou 341000, China

*Corresponding author: Hong-Zhong HUANG E-mail: hzhuang@uestc.edu.cn

Abstract: The two-dimensional (2D) lidar is a ranging optical sensor that can measure the cross-section of the geometric structure of the environment. We propose a robust 2D lidar simultaneous localization and mapping (SLAM) algorithm working in ambiguous environments. To improve the front-end scan-matching module's accuracy and robustness, we propose performing degeneration analysis, line landmark tracking, and environment coverage analysis. The max-clique selection and odometer verification are introduced to increase the stability of the SLAM algorithm in an ambiguous environment. Moreover, we propose a tightly coupled framework that integrates lidar, wheel odometer, and inertial measurement unit (IMU). The framework achieves the accurate mapping in large-scale environments using a factor graph to model the multi-sensor fusion SLAM problem. The experimental results demonstrate that the proposed method achieves a highly accurate front-end scan-matching module with an error of 3.8% of the existing method. And it can run stably in ambiguous environments where the existing method will be failed. Moreover, it can successfully construct a map with an area of more than 250 000 square meters.

Keywords: Robotics; simultaneous localization and mapping; 2D lidar; multi-sensor fusion

Citation: Shan HUANG, Hong-Zhong HUANG, Qi ZENG, and Peng HUANG, "A Robust 2D Lidar SLAM Method in Complex Environment," *Photonic Sensors*, 2022, 12(4): 220416.

1. Introduction

The two-dimensional (2D) lidar can measure the cross-section of the environmental geometric structure, and it is suited for the indoor environment. Therefore, the 2D lidar has been widely used in indoor robot mapping. The 2D lidar simultaneous localization and mapping (SLAM) algorithm is of great significance to realize the fully autonomous operation of robots, so it has attracted much research attention.

Early research on the 2D lidar SLAM method is mainly based on the filter method. Smith *et al.* [2]

first proposed the extended Kalman filter (EKF) 2D lidar SLAM that used EKF to estimate the posterior distribution of the robot pose and environmental landmarks simultaneously, which performed well in small-scale environments. However, its application in large-scale environments was limited due to the highly computational complexity. To overcome the shortcomings of the EKF 2D lidar SLAM method, Montemerlo *et al.* [3] proposed a method using RBPF (Rao-Blackwellized particle filter) to estimate the pose and environment and further optimize it [4] to achieve better convergence performance. Grisetti *et al.* [5] proposed the most used 2D lidar

Received: 30 November 2021 / Revised: 20 February 2022

© The Author(s) 2022. This article is published with open access at Springerlink.com

DOI: 10.1007/s13320-022-0657-6

Article type: Regular

SLAM-gmapping, which applied the RBPF method to the grid map and used scan matching to improve the convergence speed of RBPF. Kohlbrecher *et al.* [6] proposed a 2D lidar SLAM algorithm that used the Gauss-Newton method for scan matching. This method overcomes the influence of local extrema on the Gauss-Newton method by constructing a multi-resolution map pyramid and can perform well in a small-scale environment. However, it only estimates the current state and cannot correct the past errors. As the map area increases, the probability of error also increases, so filter-based methods cannot provide satisfactory performance in large-scale environments.

The graph optimization method can correct past accumulated errors, working well in large-scale environments. The graph-optimized 2D lidar SLAM method uses a pose-graph to model the robot trajectory and realizes the estimation of all the poses of the robot trajectory through nonlinear least squares [7]. The graph optimization 2D lidar SLAM methods can be classified into front-end matching and back-end optimization.

The mainstream method of scan registration is the iterative closest point (ICP) method [8] and the point-line ICP method [9]. The idea of these two methods is to consider the closest point between the two scans of laser data as the matching point and to obtain the relative pose of the two scans through singular value decomposition (SVD), repeating the process until convergence. Biber *et al.* [10] proposed the normal distribution transform (NDT) method, a piece-wise continuous representation method, representing the search space as a set of Gaussian distributions, where NDT heavily depended on the environment. The above scan-matching algorithms are all local and will be affected by local extreme values that do not converge to the optimal solution. To obtain the optimal solution, Olson [11] proposed the correlative scan match (CSM) method that divided the search space into grids and then enumerated all grids to obtain the optimal pose.

Since CSM enumerates the entire search space, the solution is the optimal solution. At the same time, CSM is accelerated by multi-resolution maps and has good real-time performance. Tagliabue *et al.* [12] proposed a scanning matching algorithm that fused the lidar and IMU, which used ICP for inter-frame matching and explicitly considered the degradation of the environment to improve the algorithm's stability. The proposed scan-matching method in this paper is motivated by this method. Different from the previous work, this paper uses CSM for coarse matching and the Gauss-Newton method for fine matching to obtain a high-precision optimal solution. Meanwhile, we use a global information matrix to analyze the degradation problem to achieve better stability.

The graph optimization method was first proposed by Lu *et al.* [13], and it had good performance in mapping. However, this method's number of state dimensions increases with the map area. Therefore, it has high computational complexity and is almost impossible to use in the actual application. Konolige *et al.* [14] proposed the sparse pose adjustment (SPA) method, which utilized the sparsity of the pose-graph, making the computational complexity of the graph optimization method decrease dramatically. Carlone *et al.* [15] found that graph optimization problems could be linearly approximated in 2D cases, converting nonlinear least-squares problems into linear least-squares problems, and computational complexity was significantly reduced. After that, the graph optimization method replaces the filter-based method as the mainstream method. Google's engineer proposed a 2D lidar SLAM method-cartographer [16], using CSM as a front-end scan-matching and SPA as back-end optimization. At the same time, it adopted a branch-and-bound algorithm to accelerate scan-matching and loop closure detection and achieve high mapping accuracy in both large-scale and small-scale environments. Ren *et al.* [17] proposed a 2D lidar

SLAM algorithm in large-scale environment. This method has improved CSM to give a more robust result in an ambiguous environment. Loop closure has a significant impact on the quality of the map, and the wrong loop closure will lead to the failure of the mapping process. The above method utilizes a robust kernel to mitigate the influence of the incorrect loop closure and could not eliminate the impact of the wrong loop closure. A new key-frame will be added into the pose-graph when the robot walks over a certain distance or rotates over a certain angle. With the extension of the robot trajectory, there are more and more key-frames in the pose-graph, especially when the robot goes back to a particular place, it would further lead to more key-frames and significantly affect the SLAM's performance. Besides, the above methods do not consider the environmental ambiguity, and they would fail to build a map in an ambiguous environment.

We propose a framework for tightly coupled lidar-wheel-IMU information and realize high accuracy mapping in a large-scale environment to overcome the above problems. And the largest mapping area of the framework is more than 250 000 square meters. The framework introduces the degradation analysis and realizes stable mapping and localization in an ambiguous environment. We use CSM to make the coarse global match and Gaussian-Newton method for fine match and introduce line landmark tracking, which significantly improve the matching accuracy in the structured environment. To eliminate the influence of the wrong loop closure, we propose a two-stage filtering method. First, the max clique method [18] is introduced to select the correct loop closure, which could choose the correct loop closure when the incorrect loop closure rate reaches 90%; second, for the selected loop closures, the odometry check is performed to eliminate the influence caused by incorrect loop closure in SLAM.

This paper is organized as follows. Section 2

describes the system framework. Section 3 introduces front-end module, including scan-matching algorithm, line landmark tracking, and keyframe selection strategy. Section 4 introduces the loop closure selection module. Section 5 describes the factor-graph optimization module. Sections 6 and 7 show experimental results and conclusions.

2. System framework

When the 2D lidar is working, the laser will move in a circular motion with the rotating motor. The laser performs a distance measurement when the motor rotates over a certain angle. When the motor rotates 360 degrees, the 2D lidar obtains a cross-section corresponding to the geometric structure of the environment. The working principle of the 2D lidar is shown in Fig. 1. The blue rectangle represents the obstacles in the environment, the red dot represents the measurement data of the 2D lidar, and the red arrow represents the emitted laser beam. The 2D lidar drives the rotation of the laser by a rotating motor and obtains a cross-section corresponding to the geometric structure of the environment.

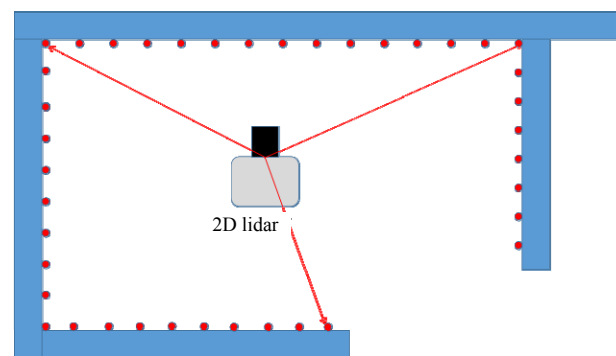


Fig. 1 2D lidar's work principle.

When the lidar data is obtained, we can input it into our SLAM algorithm and fuse it with other sensors to build a map of the environment. The SLAM system includes three modules: front-end scan-matching module, loop closure detection module, and back-end optimization module. The system framework is shown in Fig. 2. The front-end

scan-matching module receives lidar, wheel-odometry, and IMU information. It then estimates the robot movement, in which the line landmark tracking is used to improve the estimation accuracy. Once the robot is translated or rotated over a certain distance, the key-frame selection module will automatically determine whether to generate a new key-frame according to the coverage of the current

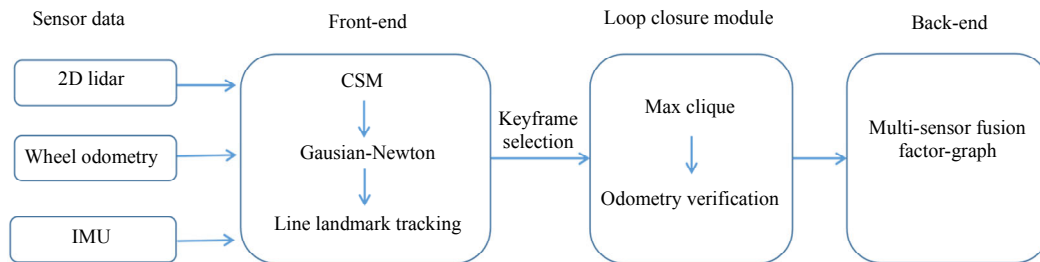


Fig. 2 Framework of SLAM system.

3. Front-end scan-matching module

The front-end scan-matching module aims to estimate the movement of the robot accurately. An accurate movement estimate is beneficial for the quality of mapping and loop closure detection. This paper introduces line landmark tracking. Multiple key-frames are associated with the same line landmark, which introduces additional spatial constraints, increasing the accuracy of the front-end scan-matching result. In addition, we limit the generation of key-frames by analyzing the coverage area. The number of key-frames is only related to the map area and is independent of the robot trajectory's length.

3.1 Scan-matching algorithm

Firstly, we use wheel odometer and IMU information to get the prior pose between two keyframes. Secondly, we use the CSM method to obtain the relative pose between the two keyframes. The relative pose will not be affected by the local extrema, but the grid resolution limits the accuracy, so the relative pose is a sub-optimal solution closest to the optimal solution. Thirdly, we use the Gauss-Newton scan-matching method for further optimization to obtain a high-precision optimal

robot pose. Then, when the new key-frame is obtained, the loop closure module automatically performs loop closure detection and filters the influence of the incorrect loop closure on the SLAM. Finally, the back-end optimization module optimizes the pose graph via factor graph [19] to obtain accurate poses. Three modules are described in detail in the following section.

solution based on the previous step.

The purpose of scan-matching is to find the relative pose between two keyframes. The first keyframe is used to construct a likelihood field map, as shown in Fig. 3. The yellow areas represent the lidar points, and the red areas correspond to the likelihood field. The closer the grid in the likelihood field is to the lidar points, the higher the score is. The black areas indicate the area with a score of 0.

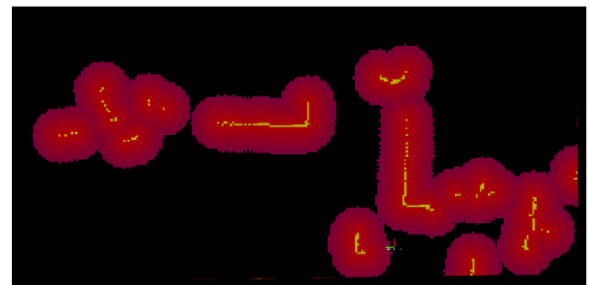


Fig. 3 Likelihood field map.

The lidar points of the second key-frame are projected into the likelihood field map through a pose. If this pose has the largest score among all the poses, this pose is the pose of the second keyframe. We use $\mathbf{X}=(x, y, \theta)=(\mathbf{t}, \theta)$ to represent the pose of the robot and $\mathbf{p}_i=(p_{ix}, p_{iy})^T$ to represent the i th lidar point in the robot coordinate system. The expression of converting the laser point into the likelihood field map through the robot pose is shown in (1).

$$\mathbf{p}'_i = \mathbf{p}'_i(\mathbf{X}) = \begin{pmatrix} p'_{ix} \\ p'_{iy} \end{pmatrix} = \mathbf{R} \cdot \mathbf{p}_i + \mathbf{t} \quad (1)$$

$$\mathbf{R} = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$$

where \mathbf{R} represents the rotation matrix corresponding to the angle of the pose \mathbf{X} .

After getting the coordinates of the lidar point in the likelihood field map, we can query the score of the corresponding grid in the likelihood field map, and the average score of all lidar points is the score of the pose:

$$S(\mathbf{X}) = \frac{1}{n} \sum_{i=1}^n M(\mathbf{p}'_i) \quad (2)$$

where $S(\mathbf{X})$ represents the score of the pose \mathbf{X} , and $M(\mathbf{p})$ represents the likelihood field score of the point \mathbf{p} .

Scan-matching is equivalent to solving (3):

$$\begin{aligned} \mathbf{X}^* &= \arg \max \frac{1}{n} \sum_{i=1}^n M(\mathbf{p}'_i) \\ &= \arg \min \frac{1}{n} \sum_{i=1}^n [1 - M(\mathbf{p}'_i)]. \end{aligned} \quad (3)$$

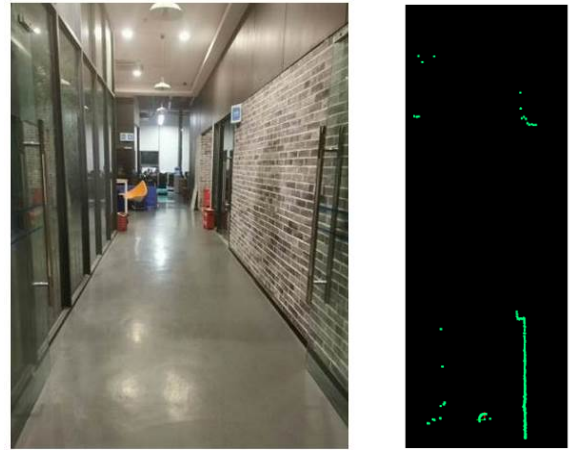
For the CSM method, the solution of (3) is relatively simple. Thus, we directly enumerate all poses and then select the best scoring pose among them, so that the covariance matrix can be directly obtained. The Gauss-Newton method requires a continuous likelihood field. The bi-linear interpolation is used to generate a continuous likelihood field and then is solved iteratively. The result of the scan-matching is shown in Fig.4. The blue points represent the first keyframe, the red points represent the second keyframe using the wheel odometer pose, and the white points represent the second keyframe using the scan-matching pose. It can be seen that the accuracy of the wheel odometry pose is low, and the overlap of the two key-frames is not good; the scan matching pose accuracy is high, and the overlap of the two frames is good.

For a normal environment, the scan-matching method can achieve good results. However, for degraded environments, for example, a long and

straight corridor shown in Fig. 5, the scan-matching pose is not stable.



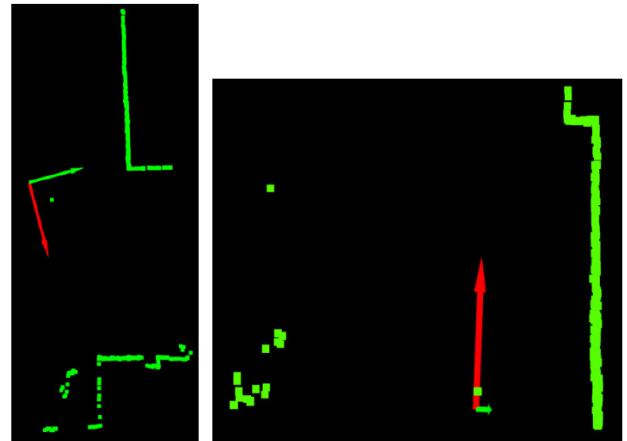
Fig. 4 Scan-matching result.



(a)

(b)

Fig. 5 Long and straight corridor: (a) snapshot of the corridor and (b) correspond lidar points.



(a)

(b)

Fig. 6 Eigenvector of the covariance matrix. The red arrow and green arrow mean the eigenvector, and the length of the arrow is the correspond eigenvalue: (a) eigenvector of normal environment and (b) eigenvector of corridor environment.

As shown in Fig. 5, the robot moves along the corridor, and the data observed by the lidar do not change, that is, the score does not change with the pose. Therefore, the pose is unconstrained along the

corridor, and one of the eigenvalues of the pose covariance matrix will be much larger than the other eigenvalues. However, the two eigenvalues are equal in a normal environment, as shown in Fig. 6.

We use the ratio of the maximum eigenvalue and the minimum eigenvalue of the pose covariance matrix to evaluate the degradation degree of the solution. If it is greater than the threshold, the eigenvector corresponding to the large eigenvalue belong to the degradation direction, and the projection of the solution in this direction needs to be eliminated [20]; otherwise, the solution is accepted directly. Assuming that the dimension of the covariance matrix is n and the number of degenerate eigenvalues of the covariance matrix is m ($0 \leq m \leq n$), three matrices are expressed as follows:

$$\mathbf{V}_d = [\mathbf{v}_1, \dots, \mathbf{v}_m, 0, \dots, 0]^T \quad (4)$$

$$\mathbf{V}_g = [0, \dots, 0, \mathbf{v}_{m+1}, \dots, \mathbf{v}_n]^T \quad (5)$$

$$\mathbf{V}_f = [\mathbf{v}_1, \dots, \mathbf{v}_m, \mathbf{v}_{m+1}, \dots, \mathbf{v}_n]^T \quad (6)$$

where \mathbf{v} means the covariance's eigenvector. Assuming that \mathbf{X}_o is the odometry pose, \mathbf{X}_s is the scan-matching pose, and the final pose is shown in (7):

$$\mathbf{X}_f = \mathbf{V}_f^{-1} \mathbf{V}_d \mathbf{X}_o + \mathbf{V}_f^{-1} \mathbf{V}_g \mathbf{X}_s. \quad (7)$$

According to (7), the final pose in the degenerate direction comes from the odometry pose, and the value of the non-degenerate direction comes from the scan-matching pose.

3.2 Line landmark tracking

The above-mentioned scan-matching algorithm only constructs the spatial constraint between two consecutive keyframes, and it often happens that multiple keyframes observe the same landmark. Therefore, imposing spatial constraints on multiple keyframes through landmarks can greatly improve the accuracy of front-end matching.

The most suitable landmark for the 2D laser SLAM is a 2D line landmark. The line extraction method [21] is first applied to each new key-frame, and a line set L is obtained. For each line $\mathbf{l} = (\theta, d)$ in the set L , the key-frame pose is to project the line

into the world coordinate system and represented as follows:

$$\theta'_i = \theta + \theta_i \quad (8)$$

$$d' = d - \mathbf{t} \cdot [\cos \theta \quad \sin \theta]^T \quad (9)$$

where θ represents the direction vector of the line, d represents the distance from the line to the origin of the coordinate system, θ is the robot orientation, and \mathbf{t} is the robot position.

Then, its relationship with the existing straight line is analyzed. For two straight lines A and B , they are viewed as the same line when their parameter differences satisfy the following condition:

$$|\theta_A - \theta_B| \leq \delta\theta, \quad |d_A - d_B| \leq \delta d. \quad (10)$$

The two key-frames can be connected by the line. If the current line cannot be associated with any known line, it is viewed as a new one and saved to the currently known line set for data associated with subsequent key-frames.

3.3 Key-frame selection

Generally, when the robot moves over a certain distance or a certain angle, it needs to insert a key-frame. In addition to the movement distance, we performed an additional check step. We find all other key-frames near the current key-frame to form a key-frame set and use this set to generate a likelihood field sub-map and then obtained current key-frame's in this sub-map. If the score is greater than the threshold, the current key-frame do not need to be inserted; if the score is less than the threshold, the current key-frame needs to be inserted into the pose-graph.

The above check step avoids adding duplicate key-frames in the constructed area. The number of key-frames in the pose graph only relates to the map area, which is also demonstrated in Fig. 7. Comparing the result in Fig. 7(b) with that in Fig. 7(a), although the robot moves back and forth in the same environment, the number of key-frames is not increased after reaching a certain level, which reduces the computational complexity of SLAM and improves the real-time performance of SLAM.

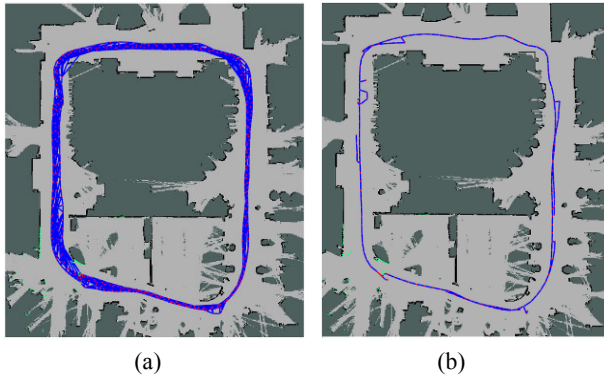


Fig. 7 Pose-graph. The red point is the robot pose and the blue line is the robot trajectory: (a) pose graph without check step and (b) pose graph with check step.

4. Loop closure detection module

Although scan-matching and line landmark tracking can greatly improve the accuracy of front-end scan-matching, the cumulative error still cannot be eliminated. With the extension of the robot trajectory, the cumulative error will become larger and larger. Loop closure detection is a good way to eliminate accumulated errors. Loop closure detection detects that the robot returns to the

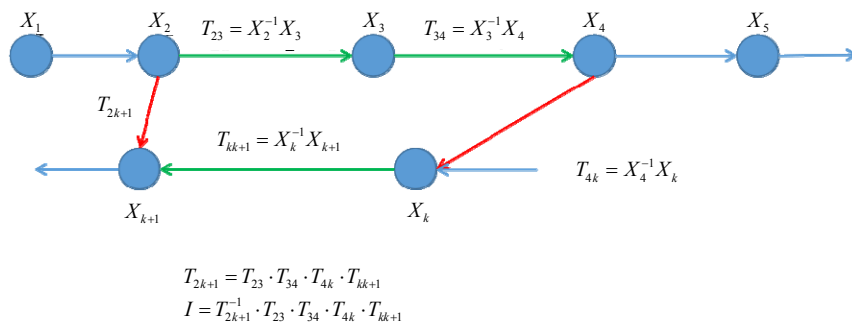


Fig. 8 Loop closure. The blue circles are the robot pose and arrows are the spatial constraint.

If the two-loop closures are wrong and the magnitude of the error is the same, the pose integral of the quadrilateral formed along the two incorrect loop closures is also equal to the identity matrix. Therefore, when the pose integral of the quadrilateral formed by two loop closures is equal to the identity matrix, the two-loop closures may be right or wrong. But these two loop closures are either right or wrong at the same time, so we can consider the two-loop closures to be compatible. Due to the existence of noise, the pose integral of the compatible loop closures cannot be strictly equal to the identity matrix. The two-loop closures LP_{ik}

previously visited area and can directly calculate the relative pose between the current key-frame and the past key-frame, and add the loop closure constraint to the pose graph to achieve the goal of eliminating accumulated errors. The disadvantage of the 2D lidar is that there is insufficient information, so it is easy to get the incorrect loop closures. And the incorrect loop closure has a huge impact on the mapping. To remove the incorrect loop closure, we use the maximum clique method and the odometry post-check to select the correct loop closures.

4.1 Max clique

An intuitive idea does not immediately decide when a loop closure is detected but stores the loop closure in the buffer and waits until enough loop closure is accumulated. After that, these loop closures are checked. Two loop closures form a quadrilateral on the pose graph. If both are correct, the pose integration along this quadrilateral results in an identity matrix, as shown in Fig. 8.

and LP_{jl} are considered compatible if they satisfy the constraint of (11).

$$e_{ikjl} = \left| \mathbf{T}_k^i \cdot (\mathbf{T}_j^i \cdot \mathbf{T}_l^j \cdot \mathbf{T}_k^l)^{-1} \right|_{\Sigma} \leq \Delta. \quad (11)$$

where Δ is the predefined threshold value.

That is, the Mahalanobis distance of the pose integral of the compatible loop closures is less than the threshold, and the two loops are considered to be compatible.

If the two-loop closures are correct, the two-loop closures must be compatible. Next, we need to assume that all correct loop closures are compatible, and wrong loop closures may or may not be

compatible, but there is a high probability that they are incompatible. This assumption makes sense because there are many reasons for incorrect loop closure, which can be considered random, and the probability that the error amplitudes of two wrong loop closures are exactly equal is very small. Based on this assumption, we divide all loop closures into multiple sets, and loop closures in one set are all compatible. The number of elements in the incorrect loop closure set is relatively small. The number of elements in the correct loop closure set is relatively large, so it can be concluded that the largest compatible subset of all loop closures is the correct loop closure set.

For each loop closure, LP_i is in the loop set LP , and we use a binary variable s_i to indicate whether the loop closure is correct. “1” indicates correct, “0” indicates incorrect, and \mathbf{S} indicates a vector composed of all binary variables. Then, the correct loop closure set selection can be expressed as the constraint optimization problem described in (12):

$$\mathbf{S}^* = \arg \max_{\mathbf{S} \in \{0, 1\}^n} \|\mathbf{S}\|_0, \text{ s.t. } e_{u,v} s_u s_v \leq \Delta. \quad (12)$$

The constrained optimization problem described in (12) is converted into a graph theory problem and then is solved by the graph theory. We regard a loop closure as a vertex. If two loop closures are compatible, an edge between them is connected. As a result, all vertices and edges form a graph G , as shown in Fig. 9.

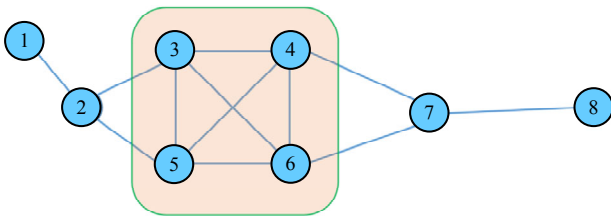


Fig. 9 Max clique. The nodes are loop-closure and dark-areas are the max-clique.

Then, obtaining the correct loop closure set is equivalent to solving the largest complete subgraph of the graph G and the dark area in Fig. 9 is the

largest complete sub-graph. Because the correct loop closures must be compatible, they will form a complete sub-graph. According to the above assumption, the correct loop closure set must be the largest compatible subset, so it is equivalent to the largest complete sub-graph. Therefore, solving (12) is transformed into finding the largest complete sub-graph of the graph G . In the graph theory, the search for the largest complete sub-graph is called the max clique problem. The max clique problem is a non-deterministic polynomial hard problem. Although it is not guaranteed to find the optimal solution, it can give an excellent sub-optimal solution. This paper uses the method proposed by Pattabiraman *et al.* [22].

4.2 Odometry check

Although the loop closures obtained with the maximum clique algorithm are likely to be correct, we add an odometry verification step based on the maximum clique selection to further improve the robustness of loop closure detection. The wheel odometry is highly accurate in the short term. Therefore, the spatial constraint accuracy between adjacent nodes in the pose graph is very high. When a loop closure is obtained, the loop closure is temporarily added to the pose graph, and then the global optimization is performed. After optimization, the relative pose between adjacent nodes is compared with the relative pose before optimization. If the difference is less than the threshold, the loop closure is correct and the loop closure can be accepted directly; otherwise, the loop closure is wrong and is rejected, and the state before the optimization is restored.

5. Factor-graph optimization

We need to fuse the information of multiple constraints to obtain a high-precision robot trajectory when obtaining a correct loop closure. We use factor-graph to model graph optimization problems. The factor graph is shown in Fig. 10.

There are five factors in the factor graph: prior factor, wheel odometry and IMU factor, lidar odometry factor, loop closure factor, and line landmark factor.

(1) The prior factor is used to fix the first node of the pose graph. We regard the first node of the pose graph as the origin of the world coordinate system, so it needs to be fixed by a priori factor.

(2) Wheel odometer and IMU factor are obtained from the wheel odometer and IMU. The relative pose between two key-frames can be calculated through integration, and the relative pose can provide spatial constraints between the two key-frames.

(3) The lidar odometry factor comes from the front-end scan-matching module. The front-end scan-matching module calculates the relative pose between the two key-frames through the CSM and Gauss-Newton method.

(4) The loop closure factor comes from the loop closure detection module, and the loop closure constraint that passes the verification will be added to the factor graph.

(5) The line landmark factor comes from the line landmark tracking in the front-end scan-matching module, and multiple key-frames are connected through the line landmark factor.

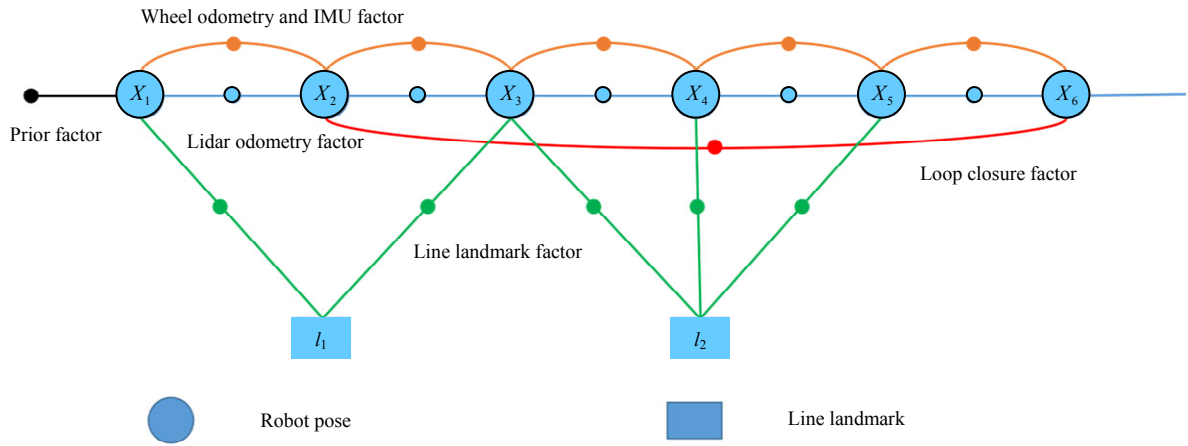


Fig. 10 Factor-graph.

The factor graph can be expressed as a nonlinear least-squares problem:

$$\mathbf{X}^*, \mathbf{l}^* = \arg \min_{\mathbf{X}, \mathbf{l}} \sum_i \left(\|\mathbf{r}_{\text{wo}}\|_{\Sigma_{\text{wo}}}^2 + \|\mathbf{r}_{\text{lo}}\|_{\Sigma_{\text{lo}}}^2 + \|\mathbf{r}_{\text{lc}}\|_{\Sigma_{\text{lc}}}^2 + \|\mathbf{r}_{\text{lp}}\|_{\Sigma_{\text{lp}}}^2 \right) + \|\mathbf{r}_{\text{prior}}\|_{\Sigma_{\text{prior}}}^2 \quad (13)$$

where \mathbf{r}_{wo} represents the error function corresponding to the wheel odometry factor; \mathbf{r}_{lo} represents the error function corresponding to the laser odometry; \mathbf{r}_{lc} represents the error function corresponding to the loop closure constraint factor; \mathbf{r}_{lp} represents the error function corresponding to the line landmark factor; $\mathbf{r}_{\text{prior}}$ represents the error function corresponding to the prior factor; Σ

represents the covariance matrix corresponding to the error function.

For the nonlinear least-squares system described in (13), we use the Levenberg-Marquardt method to solve it.

The error function of the wheel odometry factor is defined as follows:

$$\mathbf{r}_{\text{wo}} = T2V(\mathbf{O}_{i,j}^{-1} \cdot \mathbf{X}_i^{-1} \cdot \mathbf{X}_j), \quad |i-k| > k \quad (14)$$

where $\mathbf{O}_{i,j}$ represents the inter-frame constraint measured by the wheel odometry, and the wheel odometry constraint only exists between adjacent keyframes. $T2V(*)$ represents the function that extract the robot pose from the transformation matrix and is expressed as follow:

$$\mathbf{T} = \begin{bmatrix} \cos \theta & -\sin \theta & x \\ \sin \theta & \cos \theta & y \\ 0 & 0 & 1 \end{bmatrix} \quad (15)$$

$$T2V(\mathbf{T}) = (x, y, \theta).$$

The error function of the lidar odometry factor is defined as follows:

$$\mathbf{r}_{lo} = T2V(\mathbf{LO}_{i,j}^{-1} \cdot \mathbf{X}_i^{-1} \cdot \mathbf{X}_j), \quad j = i+1 \quad (16)$$

where $\mathbf{LO}_{i,j}$ represents the inter-frame constraint obtained by the front-end scan-matching module, and the laser odometry constraint also only exists between adjacent key-frames.

The error function of the loop closure constraint factor is defined as follows:

$$\mathbf{r}_{lc} = T2V(\mathbf{T}_{i,j}^{-1} \cdot \mathbf{X}_i^{-1} \cdot \mathbf{X}_j), \quad |i-k| > k \quad (17)$$

where $\mathbf{T}_{i,j}$ represents the relative pose between the key-frame i and key-frame j . Unlike the odometry factor, the loop closure constraint only exists on two key-frames with a certain difference in time, and the subscript difference of the two key-frames must be greater than k .

The error function corresponding to the line landmark factor is as follows:

$$\mathbf{r}_{lp} = f(\mathbf{X}, \mathbf{l}) - \mathbf{l}_w, \quad \mathbf{l} \in \mathbf{l}_w \quad (18)$$

where \mathbf{l}_w represents the line in the world coordinate system, \mathbf{l} represents the line observed by the key-frame i , which is the same line as \mathbf{l}_w , and $f(\mathbf{X}, \mathbf{l})$ represents the projection of the line \mathbf{l} through the pose \mathbf{X} . For details, please refer to (8) and (9). Therefore, all the key-frames that see the line \mathbf{l}_w can be correlated with each other.

The prior factor is used to fix the coordinate system of the map, we choose to fix the pose of the first key-frame:

$$\mathbf{r}_{prior} = T2V(\mathbf{X}_1) \quad (19)$$

The error function described in (19) is equivalent to setting the pose of the first key-frame to $(0, 0, 0)$.

6. Experiments

To verify the effectiveness of the proposed method, we conducted some experiments to

compare them with state-of-the-art 2D lidar SLAM algorithm-cartographer. Although cartographer's original paper was published in 2016, its open-source code has been updated until now, and it is still the state-of-the-art 2D SLAM algorithms. For comparison, we modified the visualization part of the cartographer and produced the same visualization effect as the proposed method. The experimental platform is shown in Fig. 11. The experimental platform was a differentially driven wheeled robot equipped with various sensors, such as a wheeled odometer, IMU, and 2D lidar. Its processor was a Pentium quad-core processor with 8 GB of memory.

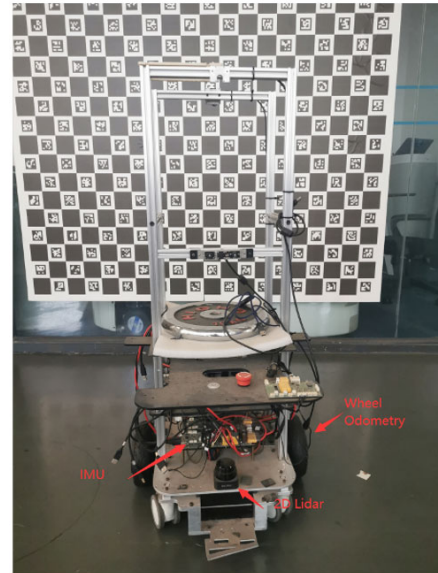


Fig. 11 Test platform. The test platform is equipped with three types of sensors: 2D lidar, wheel odometry, and IMU.

To verify the effectiveness of the proposed method, we conducted sufficient experiments on the front end of the SLAM method, the loop detection method, and the construction of large-scale maps. We recorded the original sensor data-set through the experimental platform. Then, the two algorithms were running based on the data-set. Finally, we compared the results of the two algorithms.

6.1 Front-end experiments

To test the front-end's performance. We conducted a mapping test in an environment that did

not require a loop closure. In this environment, the maps obtained by the cartographer algorithm and the proposed method are shown in Figs. 12(a) and 12(b), respectively.

The map as shown in Fig. 12 is generated after the robot walks from left to right. Since the map does not have a loop structure, the quality of the map is completely determined by the performance of the front-end matching algorithm. It can be seen from Fig. 12 that the map constructed by the proposed method can maintain the shape of a straight line, while cartographer's map becomes an arc. We used the degree of deviation of the straight

line as the accuracy evaluation index of the algorithm. The deviation was defined as the maximum distance that the built map deviated from the reference line. The deviation of the proposed method was 0.07 m and the deviation of the cartographer was 1.82 m, so the deviation of the proposed method was only 3.8% of the deviation of the cartographer. The reason was that the line landmark track module could build constraints over multiple frames and improve state estimation accuracy. The experimental results showed that the line landmark track module could significantly improve the accuracy of the front-end module.

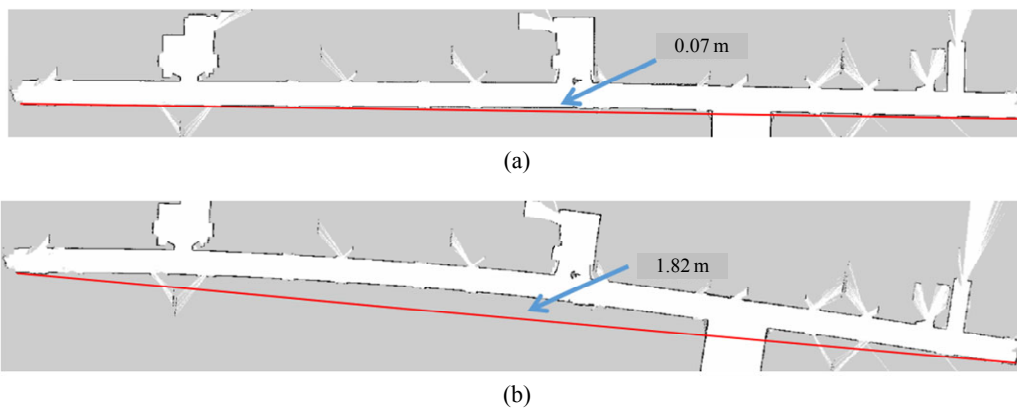


Fig. 12 Front-end map: (a) map built by cartographer and (b) map built by proposed method.

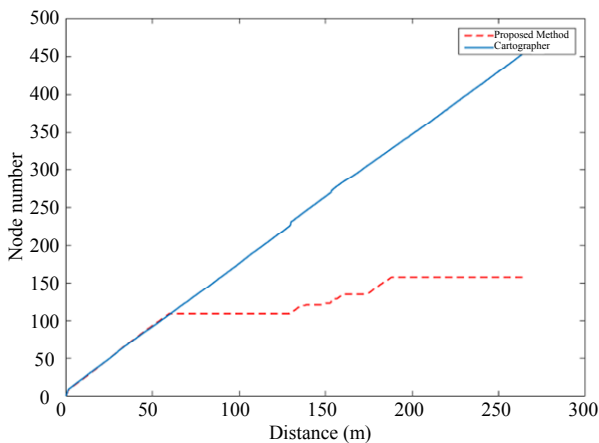


Fig. 13 Pose-graph nodes. The relationship between the number of nodes in the pose graph and the running distance.

To verify the effectiveness of the keyframe selection strategy, we also counted the number of nodes in the pose-graph during the running of the two algorithms. The number of nodes in the pose graph is shown in Fig. 13. It can be seen from Fig. 13

that the number of nodes in the cartographer method increases with the length of the robot trajectory. The number of nodes in the proposed method will stabilize after the robot trajectory reaches a certain distance and will not increase. Although the robot continues to move on the map, no more nodes will be added to the pose graph. This means that with an increase in the robot's moving distance, the memory consumption of the cartographer will increase until all memory is consumed, and the memory consumption of the proposed method will be stabilized. The experimental results showed that the proposed keyframe selection strategy can significantly reduce the memory consumption of the algorithm.

To test the robustness of the proposed method in an ambiguous environment, we conducted a mapping test in an ambiguous environment. The

maps built by the cartographer algorithm and the proposed algorithm in the ambiguous environment are shown in Figs. 14(a) and 14(b), respectively. It can be seen from Fig. 14 that there is degradation in the environment, the cartographer has ambiguities in the front-end matching, and the map appears inconsistent in the corridor shown in the red box. The proposed method

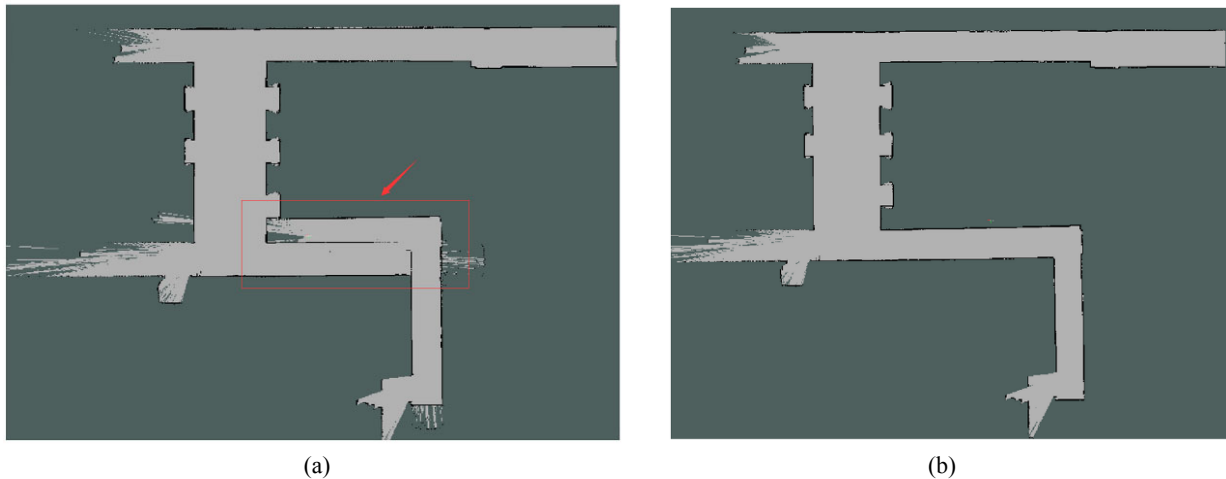


Fig. 14 Map of degraded map: (a) map built by cartographer and (b) map built by the proposed method.

6.2 Loop closure experiments

To test the robustness of the proposed loop closure detection method, a mapping test was finished in an ambiguous environment. The correct map of the environment is shown in Fig. 15.

The map constructed by the cartographer algorithm and the proposed algorithm in the ambiguous environment is shown in Figs. 16(a) and 16(b), respectively. It can be seen from Fig. 16 that the map constructed by the proposed method is clear and consistent. The map constructed by the cartographer has inconsistency in the part shown in the red circles. The reason is that the test environment has ambiguities in the place, and loop closure detection is prone to errors. Therefore, the cartographer method was accepted an incorrect loop closure and then resulted in an inconsistent map. The proposed method successfully removed the incorrect loop closure and obtained a consistent map in the ambiguous environment. The experimental

introduced degradation analysis to deal with the degradation situation, even in an ambiguous environment, and a globally consistent map could be constructed. The experimental results showed that the degradation analysis module could greatly improve the algorithm's robustness to run stably in an ambiguous environment.

results showed that the proposed max-clique loop closure selection method could select correct loop closure from the set containing many incorrect loop closures, which significantly improved the algorithm's stability.



Fig. 15 Correct map.

6.3 Large-scale mapping experiment

To verify the mapping ability of the proposed method in a large-scale environment, a mapping test was conducted in a low-level parking lot with an area of more than 250 000 square meters, and the constructed map is shown in Fig. 17. It can be seen

that despite the large area of the parking lot, the proposed method still successfully constructed the map of the parking lot, and the constructed map was clear and consistent. The experimental results demonstrated that the proposed method could accurately build maps in a large-scale environment. In the process of building the map, due to excessive

memory consumption, the cartographer suffered a program crash and its map is not shown here. The reason is that the proposed method consumes much less memory than the cartographer. Therefore, the proposed method can build a map of the large-scale environment, and the cartographer will crash due to excessive memory consumption.

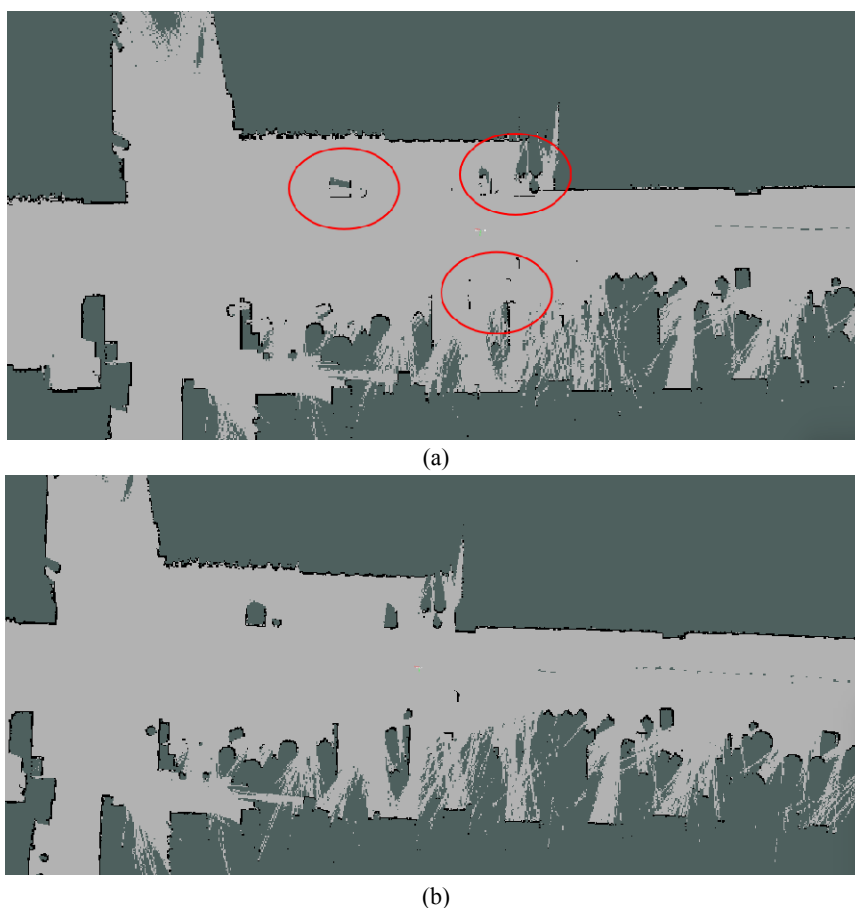


Fig. 16 Loop closure test: (a) the map constructed by cartographer and (b) the map constructed by proposed method.



Fig. 17 Parking lot map constructed by proposed method.

7. Conclusions

This paper proposes a robust 2D lidar SLAM method for complex environments. The constructed multi-sensor tightly-coupled 2D lidar SLAM framework is suitable for multi-sensor and multi-feature fusion. First, we introduce line landmark tracking and degradation analysis and improve the matching accuracy of the front-end and the stability in the complex environment. Then, we introduce the maximum clique method and odometry check and achieve the correct loop closure selection in the presence of lots of outliers. The experiments validate our approach on real-world datasets. A detailed evaluation and comparison with existing methods are conducted to demonstrate the effectiveness of the proposed method. Our future work is to introduce semantic features to make the mapping process reliable, more accurate and human-friendly.

Acknowledgment

This work was supported by National Key Research and Development Program of China (Grant No. 2017YFB1301300).

Open Access This article is distributed under the terms of the Creative Commons Attribution 4.0 International License (<http://creativecommons.org/licenses/by/4.0/>), which permits unrestricted use, distribution, and reproduction in any medium, provided you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons license, and indicate if changes were made.

References

- [1] J. J. Leonard and H. F. Durrant-Whyte, "Simultaneous map building and localization for an autonomous mobile robot," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Osaka, Japan, Nov. 1991, pp. 1442–1447.
- [2] R. Smith, M. Self, and P. Cheeseman, "Estimating uncertain spatial relationships in robotics," in *Autonomous Robot Vehicles*, New York, USA, 1990, pp. 167–193.
- [3] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM: A factored solution to the simultaneous localization and mapping problem," in *Proc. of Theaaai National Conference on Artificial Intelligence. American Association for Artificial Intelligence*, Edmonton, Canada, Jul. 2002, pp. 593–598.
- [4] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, "FastSLAM 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges," in *International Joint Conference on Artificial Intelligence*, Acapulco, Mexico, Aug. 2003, pp. 1151–1156.
- [5] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with Rao-Blackwellized particle filters," *IEEE Transactions on Robotics*, 2007, 23(1): 34–46.
- [6] S. Kohlbrecher, O. V. Stryk, J. Meyer, and U. Klingauf, "A flexible and scalable SLAM system with full 3D motion estimation," in *IEEE International Symposium on Safety, Security, and Rescue Robotics*, Kyoto, Japan, Nov. 2011, pp. 155–160.
- [7] R. Kümmerle, G. Grisetti, H. Strasdat, K. Konolige, and W. Burgard, "G2o: A general framework for graph optimization," in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May. 2011, pp. 3607–3613.
- [8] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 1992, 14(2): 239–256.
- [9] A. Censi, "An ICP variant using a point-to-line metric," in *IEEE International Conference on Robotics and Automation (ICRA)*, Pasadena, USA, May. 2008, pp. 19–25.
- [10] P. Biber and W. Strasser, "The normal distributions transform: A new approach to laser scan matching," in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Las Vegas, USA, Oct. 2003, pp. 2743–2748.
- [11] E. B. Olson, "Real-time correlative scan matching," in *IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May. 2009, pp. 4387–4393.
- [12] A. Tagliabue, J. Tordesillas, X. Cai, A. Santamaria-Navarro, J. P. How, L. Carlone, *et al.*, "LION: lidar-inertial observability-aware navigator for vision-denied environments," in *International Symposium on Experiment Robotics*, La Valletta, Malta, Nov. 2020, pp. 380–390.
- [13] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, 1997, 4(4): 333–349.
- [14] K. Konolige, G. Grisetti, R. Kummerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient sparse pose adjustment for 2d mapping," in *IEEE International*

- Conference on Intelligent Robots and Systems (IROS)*, Taipei, China, Oct. 2010, pp. 22–29.
- [15] L. Carlone, R. Aragues, J. A. Castellanos, and B. Bona, “A linear approximation for graph-based simultaneous localization and mapping,” in *Robotics: Science and Systems*, Sydney, Australia, Jul. 2012, pp. 41–48.
- [16] W. Hess, D. Kohler, H. Rapp, and D. Andor, “Real-time loop closure in 2d lidar SLAM,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May. 2016, pp. 1271–1278.
- [17] R. Ren, H. Fu, and M. Wu, “Large-scale outdoor SLAM based on 2D lidar,” *Electronics*, 2019, 8(6): 6–13.
- [18] J. G. Mangelson, D. Dominic, R. M. Eustice, and R. Vasudevan, “Pairwise consistent measurement set maximization for robust multi-robot map merging,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Prague, Australia, May 2018, pp. 2916–2923.
- [19] M. Kaess, H. Johannsson, R. Roberts, V. Ila, J. Leonard, and F. Dellaert, “iSAM2: Incremental smoothing and mapping with fluid relinearization and incremental variable reordering,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, May 2011, pp. 3281–3288.
- [20] J. Zhang, M. Kaess, and S. Singh, “On degeneracy of optimization-based state estimation problems,” in *IEEE International Conference on Robotics and Automation (ICRA)*, Stockholm, Sweden, May 2016, pp. 809–816.
- [21] A. Schaefer, D. Büscher, L. Luft, and W. Burgard, “A maximum likelihood approach to extract polylines from 2D laser range scans,” in *IEEE International Conference on Intelligent Robots and Systems (IROS)*, Madrid, Spanish, Oct. 2018, pp. 4766–4773.
- [22] B. Pattabiraman, M. M. A. Patwary, A. H. Gebremedhin, W. Liao, and A. Choudhary, “Fast algorithms for the maximum clique problem on massive graphs with applications to overlapping community detection,” *Internet Mathematics*, 2014, 11(4–5): 421–448.
- [23] Y. F. Li, H. Z. Huang, J. Mi, W. Peng, X. Han, “Reliability analysis of multi-state systems with common cause failures based on Bayesian network and fuzzy probability,” *Annals of Operations Research*, 2022, 311(1): 195–209.
- [24] Y. F. Li, Y. Liu, T. Huang, H. Z. Huang, J. Mi, “Reliability assessment for systems suffering common cause failure based on Bayesian networks and proportional hazards model,” *Quality and Reliability Engineering International*, 2020, 36(7): 2509–2520.