IEEE Access
Multidisciplinary ⁝ Rapid Review ⁝ Open Access Journal

Received 27 April 2024, accepted 19 May 2024, date of publication 27 May 2024, date of current version 3 June 2024.

Digital Object Identifier 10.1109/ACCESS.2024.3405968

RESEARCH ARTICLE

Real-Time Intensity-Based Mirror Detection for Optimizing Point Cloud Data and Occupancy Grid Map

JING ZHU[1], **SIBO WANG**[2], **CANHONG LIN**[2], **AND BANGZHENG YIN**[3]
[1]Lab Center, Guangzhou University, Guangzhou 510006, China
[2]School of Electronics and Communication Engineering, Guangzhou University, Guangzhou 510006, China
[3]Institute of Information Engineering, Guangzhou Railway Polytechnic, Guangzhou 510000, China

Corresponding author: Jing Zhu (zhujing@gzhu.edu.cn)

This work was supported in part by Guangzhou Railway Polytechnic Talent Research Launch Project under Grant GTXYR2005, and in part by the Second Batch of Industry-University Cooperation Collaborative Education Project of the Ministry of Education, in 2018, under Grant 201802212018.

ABSTRACT In recent years, indoor mobile robots have played an increasingly important role in various home, medical, commercial, and industrial applications. However, mirror surfaces commonly found in indoor environments pose challenges to the localization and navigation of indoor mobile robots. In environments with mirror surfaces, robots may misjudge the location of obstacles owing to laser reflections, leading to accidental collisions or mission failure. In this study, a single 2D LiDAR was used to identify the location of the mirrors in the environment, and to optimize the point cloud data and occupancy grid map using the mirror locations. The main innovation is the use of the intensity information of the reflected laser beam and the inherent symmetry of mirrors for real-time detection, including polygonal mirrors and mirrors without contours. Currently, the method has been experimented with in several complex environments, the accuracy of mirror identification exceeds 97%, and effectively modified the erroneous occupancy grid map.

INDEX TERMS Real time, intensity, symmetry, point cloud data, occupancy gird map.

I. INTRODUCTION

Currently, indoor mobile robots rely on Simultaneous Localization And Mapping (SLAM) [1] for localisation and navigation, such as laser SLAM [2], [3] [4], visual SLAM [5], [6], [7], and multi-sensor fusion SLAM [8], [9], [10]. This technology enables robots to construct maps and determine their position in unknown environments in real time. However, SLAM technology is affected by specular reflections, which is a common but challenging problem. Specular reflections can cause point clouds with incorrect distances to be generated in the laser scan data, which can mislead the SLAM system and prevent it from accurately constructing a map or estimating the robot position, as shown in Fig. 1. However, today's architectural styles are undergoing significant changes, and one notable trend is the quest for modernity and innovative design. Architects and designers are increasingly favoring the use of materials and elements, such as mirrors and glass to create the exterior and interior spaces of buildings. Therefore, it is crucial to address the issues of positioning and building maps for indoor mobile robots in mirrored environments.

The SLAM problem for environments with mirrors present has had several solutions proposed, most of them relying on 3D LIDAR, multi-echo sensors or cameras to detect mirrors, which are not only costly but also consume a lot of compute resources.

In this study, an intensity-based real-time 2D LiDAR mirror detection method was proposed with the aim of optimizing real-time acquired point cloud data and occupancy grid map. The method was divided into four stages:(1) Obtaining the position of the mirror in the environment based on the intensity information of the laser beam and the inherent symmetry of the mirror surface. (2) Updating and optimizing the mirror location. (3) The point cloud data were optimized by modifying the distance information of the laser

The associate editor coordinating the review of this manuscript and approving it for publication was Li He.

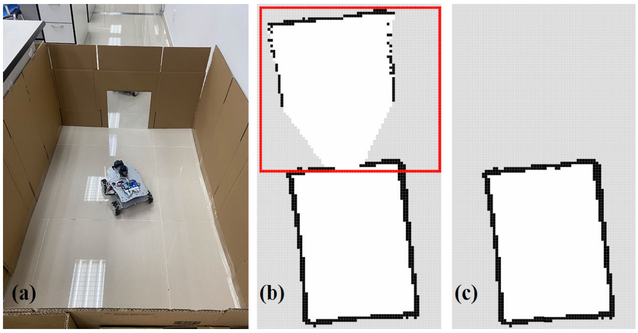


FIGURE 1. (a) The environment used to test the effect of mirrors on the map; (b) The build map in this environment, with incorrect map information in the red box; (c) The correct map after removing the mirrors.

beams directed at the mirrors. (4) Construct and optimize the occupancy grid map using the open-source KartoSLAM algorithm. The innovation of this method is that it can acquire the location of mirrors in real time while optimizing the point cloud data and occupancy grid map. The experimental results show that the method exhibits a good performance in a variety of complex environments, with an accuracy of over 97% for mirror identification. The main innovation is that the method utilizes the intensity information of the reflected laser beam and the inherent symmetry of the mirrors to achieve real-time mirror recognition and remove erroneous map information in real time. It uses only a single 2D LIDAR, eliminating the need for expensive and complex 3D LIDAR and multiple echo sensors, thereby making it more suitable for inexpensive robots. The method performs correctly in the face of polygonal mirrors and mirrors without contours, with an improved percentage of detected mirrors compared to other methods, in real time and at low cost.

Section II reviews state-of-the-art techniques for identifying reflective surfaces.Section III provides the distribution of intensity information of the laser beam directed at the mirror and the symmetry property of the point cloud data generated by the mirror.Sections IV-VII detail methods for identifying mirrors and optimizing point cloud data and occupying grid maps. Section VIII presents the results obtained by applying the proposed method to various challenging environments. Finally, conclusions and future work are presented in Section IX.

II. STATE OF THE ART

Several previous studies discussed the use of a single sensor for mirror detection. These methods can be categorized into four types depending on the detection principle: deep learning based, symmetry property based, bezel detection based, and intensity based. The following is a review of the studies related to each method:

A. BASED ON DEEP LEARNING

Deep learning is an important branch in the field of Artificial Intelligence and has made remarkable progress. There are many deep learning models with excellent feature learning and classification capabilities for image processing and point cloud processing, which have become powerful tools in mirror identification tasks. Jiang et al. [11] proposed a classifier based on a four-layer neural network using only 2D LiDAR, which shows the probability of an object being glass using the distance, intensity, and angle of incidence measured by lidar as inputs. Mei et al. [12] proposed a glass detection network, GDNet-B, which detects glass from images using a single RGB camera. Tao et al. [13] based on classical visual features, proposed a boundary-guided glass image segmentation network based on classical visual features for glass recognition. However, despite their excellent detection results, these methods require expensive computational resources.

B. BASED ON SYMMETRY PROPERTY

The symmetry property causes the point cloud data generated by LiDAR to be symmetrical about the mirror. The mirror position is determined based on this property. Yang and Wang [14] measured the uncertainty in mirror prediction using the Iterative Nearest Points (ICPs) algorithm, which predicts the approximate location of the mirror by setting the mirror length. Li et al. [15] used their own robot as a reference point to obtain the position that is most likely to be the axis of symmetry of the mirror by iterating over a multi-frame point cloud. These two works are unable to achieve a better recognition effect in complex mirror environments because of the lack of determination of the approximate position of the mirror, and thus, they need to set the length of the mirror artificially or to obtain the mirror position through a large number of iterations.

C. BASED ON BEZEL DETECTION

Several approaches have been proposed based on mirror bezel detection to determine mirror location. Pu and Vosselman [16] attempted to differentiate between facade features (e.g., walls and roofs) to detect mirrors, and they introduced knowledge about the size, location, orientation, and topology of the features to identify these mirrored windows in the identified segmented laser point cloud. Hao et al. [17] proposed a point-based slicing-based window localization algorithm to extract architectural walls from the scene point cloud based on the surface area, normal direction, topological relationship, and other features of the ensemble to extract the building wall from the field point cloud. Finally, the building facade was sliced horizontally and vertically to obtain the window region. Wang et al. [18] detected windows by combining bottom-up and top-down strategies to extract facade planes from the LiDAR point cloud. The point cloud was clustered into potential elevation regions using principal component analysis (PCA) in the bottom-up approach. The elevation planes were then extracted from the potential elevation regions using Random Sample Consensus (RANSAC) in the top-up method. However, these efforts had difficulty detecting mirrors that were not embedded in walls.

D. BASED ON INTENSITY

Shiina and Wang [19] experimentally demonstrated that the intensity of the received laser light is maximum when the laser beam irradiation angle is close to the vertical angle of the glass surface and utilized this property of the glass in the environment. Wang and Wang [20] recognized glass based on the intensity of the laser beam reflected by specular surfaces and combined it with existing SLAM algorithms to enable the SLAM system to detect and localize glass in real time. Although these efforts were successful in detecting glass, there were no steps in detecting mirrors or modifying the occupancy grid map.

Tibebu et al. [21] used the standard deviation of the laser beam in the first stage to classify the points hitting an opaque surface and passing through a glass surface. The second stage uses the distance and intensity variations between neighbouring pulses to improve the results of the first stage and to estimate the width of the glass contour and finally to modify the occupancy grid map. Weerakoon et al. [22] were able to detect glass surfaces by examining the intensity profiles of the laser scanning data and identifying intensity peaks. They used the intensity threshold, intensity gradient and profile width as the adjustable parameters. Local and global optimization algorithms were designed to modify the occupancy grid map. Overall, these two methods could locate the glass surface but used more parameters to process the intensity information.

Mora et al. [23] experimentally obtained the intensity values of laser light at different incidence angles for three materials: metal, glass, and marble, acquired the intensity information, fitted the specular equation while the robot was moving, and modified the laser data while modifying the occupancy grid map. Although the method achieved good results in modifying the occupancy grid map, mirror detection and optimization were performed offline, and the entire mirror region could not be directly acquired during the robot motion.

E. WORK OF ARTICLE

In this study, we used a single 2D LiDAR to obtain the position of the mirror surface in the environment by returning the intensity information of the laser beam and the inherent symmetry of the mirror. After obtaining the maximum value of the intensity information, we processed the symmetric point cloud and obtained the entire mirror surface within the observation range in real time as the robot passed over it. This approach eliminated the need for expensive and complex 3D LiDAR or multi-echo sensors. Using the acquired mirror position, we were able to optimize the point cloud data and occupancy grid map in real-time. The flowchart of the proposed method is shown in Fig.2.

III. LASER COLLISION MIRROR FEATURES

A. INTENSITY INFORMATION

According to the principle of light reflection, when light encounters the boundary of a medium, a certain portion of the light is reflected. The emitted laser beam conforms to this property of LiDAR applications. There are two types of reflection: (1) Diffuse reflection: when light hits a rough surface, it is scattered in all the directions. When the LiDAR is in operation, when the emitted laser beam touches a rough surface, a part of the laser ray is reflected back in the original direction and is received by the laser receiver; thus, the LiDAR is able to detect the distance from the obstacle, as shown in Fig.3(a). (2) Specular reflection: when the light meets a smooth surface, the light will reflect at the same angle. In LiDAR applications, after the laser beam reaches a smooth surface, the distance to the obstacle cannot be accurately determined because the laser light is reflected. This is illustrated in Fig.3(b).

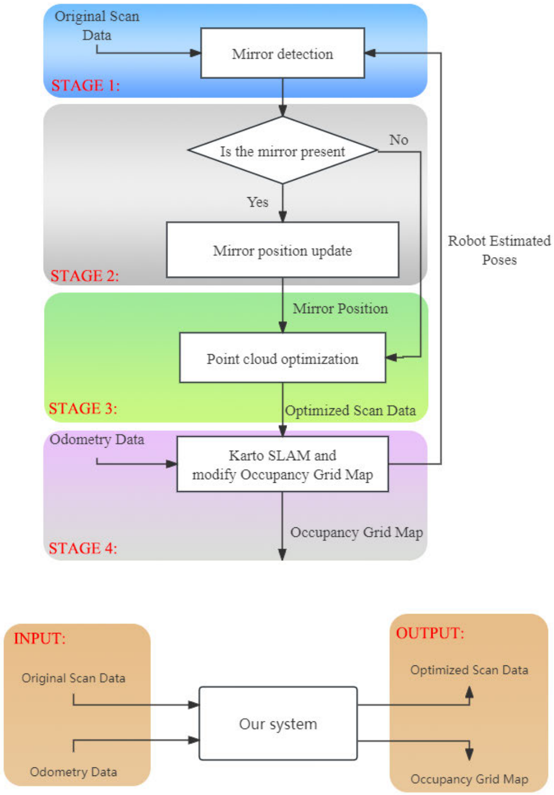


FIGURE 2. A method based on four stages is proposed: mirror detection, mirror update, point cloud data optimisation, running KartoSLAM and modifying the occupancy gird map. The inputs to the method are odometry data and laser scan data, and the outputs are optimised scan data and occupancy gird map.

Based on the principle of reflection, we assume that when the laser beam is incident perpendicular to a smooth surface, all laser rays return and are received by the laser receiver according to the original path, as shown in Fig.3(c). In addition to the correct distance information, extremely high laser intensity was obtained. We tested this in the environment shown in Fig.4, where the LiDAR was placed in front of a mirror, and the intensity distribution of the vertically incident and surrounding laser dots was obtained by adjusting the distance between the LiDAR and the mirror. The LiDAR intensity information used was an 8-bit output, and the range

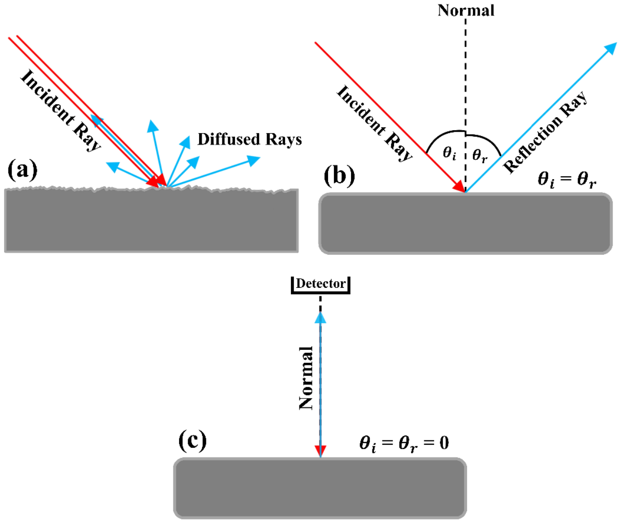


FIGURE 3. (a) Diffuse reflection from rough surfaces; (b) Specular reflection from smooth surfaces; (c) Vertical smooth surface reflection.

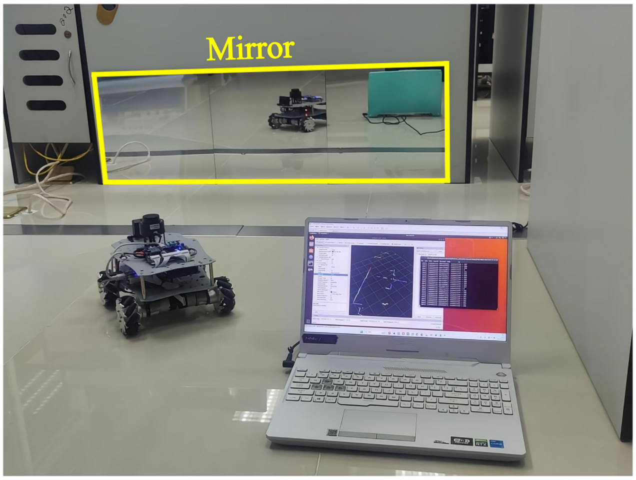


FIGURE 4. Scene settings for measuring the value of the laser intensity at the location of the mirror.

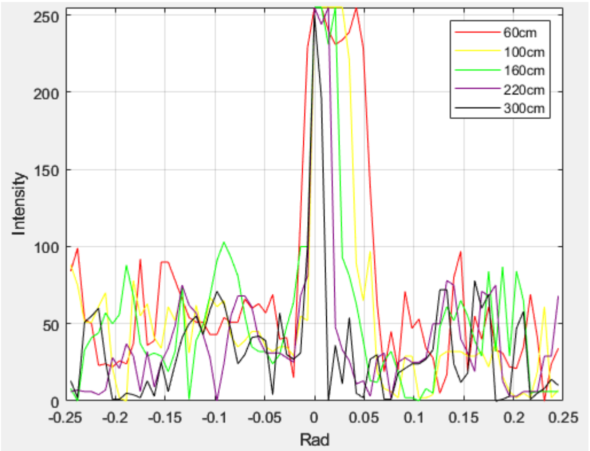


FIGURE 5. Plot of angle of incidence versus intensity values for five different distances facing the mirror.

of the resulting intensity was 0-255, as shown in Fig.5. This experiment successfully verified the hypotheses.

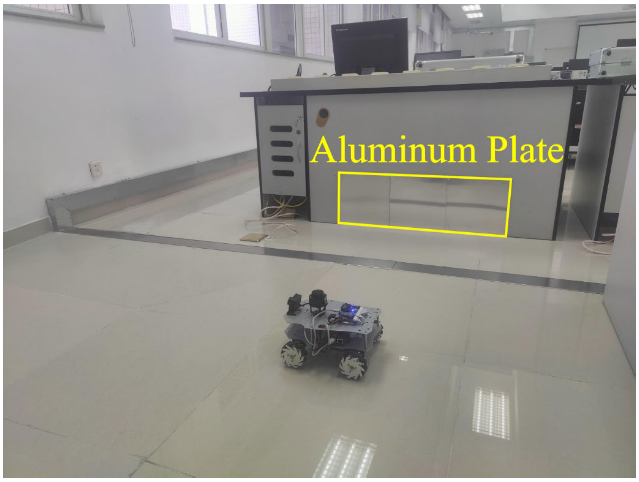


FIGURE 6. Scene setup for measuring the laser intensity value at the location of the aluminium metal plate.

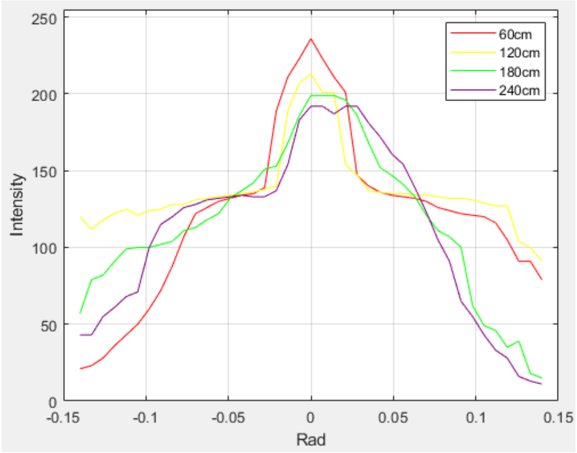


FIGURE 7. Plot of the angle of incidence versus the intensity value at four different distances facing the aluminium metal plate.

In our experiments, we found that the laser beam also returned more intense information when vertically incident on metals with smooth surfaces. Therefore, we performed the above experiment on a metal aluminum plate and obtained an intensity distribution graph as shown in Fig.6 and Fig.7. The probability of misjudging the mirror recognition was reduced by detecting the characteristics of the environment that may produce interference terms prior to the experiment. Comparing the laser intensity distribution graphs corresponding to the mirror and the metal aluminium plate, it could be observed that within 3m, the laser intensity vertically reflected back from the mirror was greater than 250, whereas the maximum intensity generated by the aluminum plate did not exceed 250.

B. SYMMETRY PROPERTY

When mirrors are present in the environment in which the LiDAR is scanned, the laser beam is reflected after hitting the mirror, and the angle of incidence determines the direction of reflection. The reflected laser ray reflected a portion of the laser ray back to the laser receiver when it hit the rough surface of an actual object. As a result, when resolving and processing the received laser ray, the LiDAR system

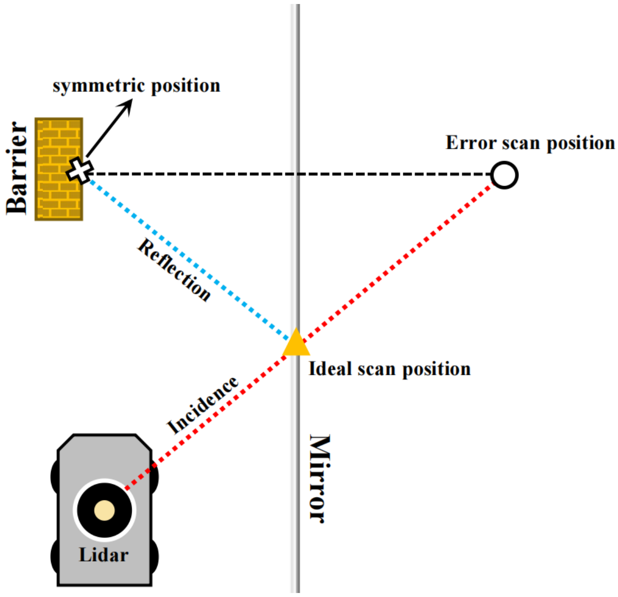


FIGURE 8. After the laser beam is reflected by the mirror, the error scan point generated is symmetrical to an object in the actual environment.

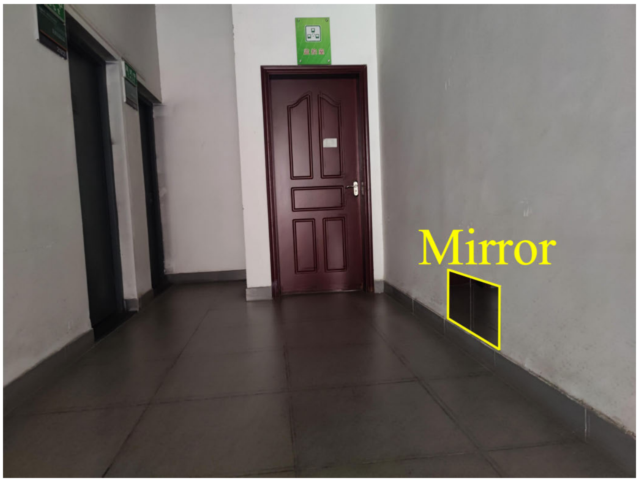


FIGURE 9. Scenario setup for verifying symmetry properties.

mistakenly believes that the reflected ray is caused by a virtual target behind the reflector, and thus incorrectly locates the position of the target on the mirror of the actual object instead of the real position of the mirror, as shown in Fig.8.

Based on this property, we attempted to compute multiple pairs of symmetry points generated by the mirror, obtain the symmetry axis of each pair of symmetry points, and compute the intersection of the line where the laser beam was located with this symmetry axis. We chose to perform the experiment in an environment in which the generated symmetry points were more characteristic, as shown in Fig.9 to reduce the errors that may be introduced by manual labelling, as shown in Fig.10, where the position of the mirror after restoration deviated somewhat from the actual mirror position. Therefore, we need to determine the approximate position of the mirror in advance using the intensity information. The specific details are introduced later.

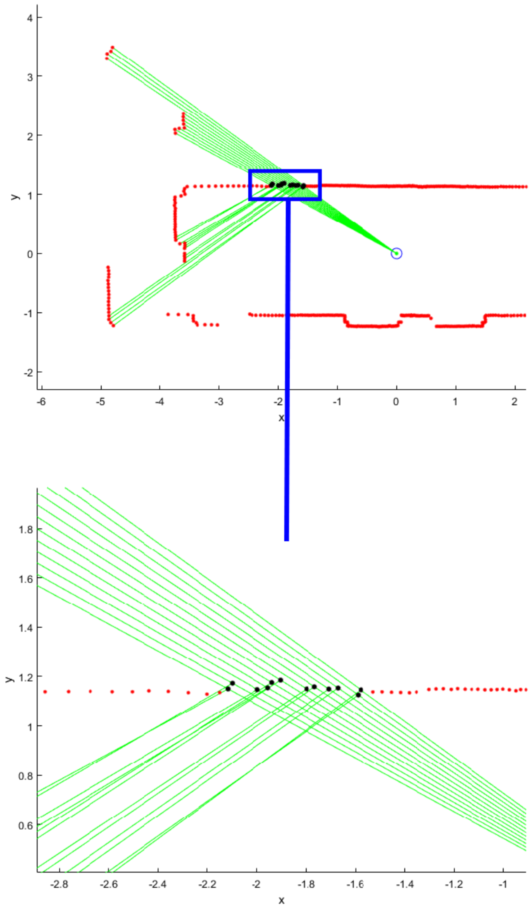


FIGURE 10. Manual labelling of recovered mirror positions (top) and close-up view (bottom).

In the experiment in III-A, we found that the laser beam directed at the polished metal also returned larger laser intensity information. However, when the test was carried out, the laser beam returned the correct distance information to the aluminum plate of the metal placed vertically, as shown in Fig.11. Free electron reflection occurs when the laser beam hit the metal surface and interacts with free electrons. At this point, the direction of the reflected laser beam was opposite to that of the source laser beam, making the output distance information correct. This was different from the specular reflection from a smooth surface. LiDAR could identify vertically placed metals correctly and we did not think it is necessary to optimize them. Therefore, in this study, we optimized the mirrors.

IV. MIRROR POSITION DETECTION

The task in the first stage is to obtain the specific position of the mirror in the environment from the intensity information of the laser beam returned by the lidar and symmetry properties of the mirror. The laser scan data acquired by the robot is based on the lidar coordinate frame, and the lidar coordinate frame keeps changing while the robot is moving, the mirrors in the environment are stationary. Therefore, the last step in this phase is to transform the acquired mirror positions into the world coordinate frame.

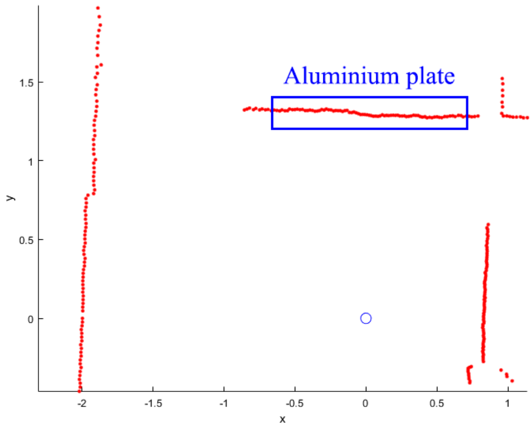


FIGURE 11. Based on the laser scan obtained from the environment in Figure 6, the position where the metal aluminium plate is located has been marked with a blue box, and it can be seen that the output distance information is correct.

We planarise the mirror, and based on the principle described earlier that a laser beam illuminating the mirror at a near-vertical angle returns high-intensity information, we can effectively locate the straight-line position where the mirror is located. For a given moment t in the lidar coordinate frame, set up an intensity threshold to determine whether the mirror exists and traverses the detection of the intensity value of each laser beam. If the intensity value of the laser is greater than the threshold, then it is judged that lidar detects the existence of the location of the mirror. Based on the detection point $s_{i,t}$, the equation for the line where the mirror is located $y = k_{M,t}x + b_{M,t}$ can be obtained, as shown in Fig.12 and Eq.1 to 5:

$$\theta_i = \theta + i \cdot \Delta\theta \tag{1}$$

$$x_{i,t} = r_{i,t} \cdot \cos\theta_i \tag{2}$$

$$y_{i,t} = r_{i,t} \cdot \sin\theta_i \tag{3}$$

$$k_{M,t} = -\frac{x_{i,t} - x_{lidar}}{y_{i,t} - y_{lidar}} \tag{4}$$

$$b_{M,t} = y_{i,t} - x_{i,t} \cdot k_{M,t} \tag{5}$$

where $\left(x_{i,t}, y_{i,t}\right)$ is the Cartesian coordinates of the detection point $s_{i,t}$, (x_{lidar}, y_{lidar}) is the origin of the lidar coordinate frame, i is the index of the laser beam corresponding to that detection point, θ is the starting angle of the laser scan data, $\Delta\theta$ is the angular resolution, and $r_{i,t}$ is the distance measurement of the laser beam corresponding to that detection point.

After successfully obtaining the equation for the straight line where the mirror is located, the next step is to obtain lidar data points that are symmetrical about the straight line based on the symmetry of the lidar data points at both ends of the mirror. Because LiDAR generates small position fluctuations when scanning or detecting a target, it is numerically impossible to achieve a complete symmetry of the two measurement points with respect to the line. The two measurement points are allowed to be symmetrical within a certain error margin. Specifically, assuming that the two lidar data points $s_{n,t}$ and $s_{m,t}$ are symmetrical with respect to the line, they must satisfy the following three conditions:

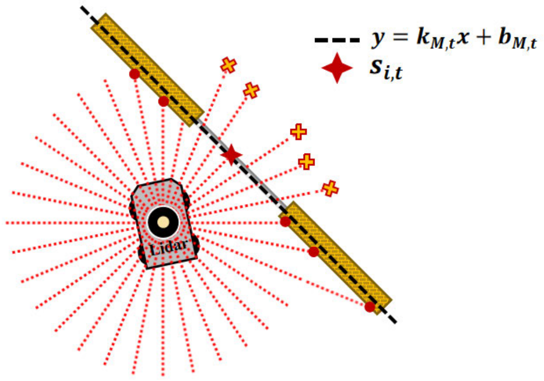


FIGURE 12. Mirror line position acquisition based on $s_{i,t}$.

(1) For the slope set, the minimum threshold k_{min} and the maximum threshold k_{max}, satisfy $k_{min} < k_{M,t} \cdot k_{c,t} < k_{max}$, which indicates that the line connecting the two lidar data points is perpendicular to the straight line where the mirror is located.

(2) Satisfy $d_{n,t} \cdot d_{m,t} < 0$, indicating that the two lidar data points are not on the same side of the straight line where the mirror is located.

(3) Set a threshold value of d_{min}, satisfying $\left|d_{n,t}\right| > d_{min}$ and $\left|d_{m,t}\right| > d_{min}$, which indicates that the two lidar data points are not on a straight line where the mirror is located.

where $k_{c,t}$ is the slope of the line connecting the two lidar data points, and $d_{n,t}$ and $d_{m,t}$ are the distances from the two lidar data points to the straight line on which the mirror is located (no absolute values are taken so as to make it easier to differentiate their orientations with respect to the straight line), and the following are the formulas for their calculation, as seen in Eq.6 to 8:

$$d_{n,t} = \frac{k_{M,t} \cdot x_{n,t} - y_{n,t} + b_{M,t}}{\sqrt{k_{M,t}^2 + 1}} \tag{6}$$

$$d_{m,t} = \frac{k_{M,t} \cdot x_{m,t} - y_{m,t} + b_{M,t}}{\sqrt{k_{M,t}^2 + 1}} \tag{7}$$

$$k_{c,t} = \frac{y_{n,t} - y_{m,t}}{x_{n,t} - x_{m,t}} \tag{8}$$

Based on the above requirements, multiple pairs of symmetry points are obtained after traversing all lidar data at moment t. We selected the point furthest from the origin of the lidar coordinate frame among the symmetry point pairs, connected it to the origin of the lidar coordinate frame, calculated the intersection points of the line segment with the line where the mirror was located, and stored these intersection points. Eventually, we obtained a group of points on the line where the mirror was located. The number of points is the same as the number of symmetry pairs, and most of them are located in the actual mirror position in the lidar coordinate frame. There may be some symmetric structures in the real environment that cause some real lidar data points

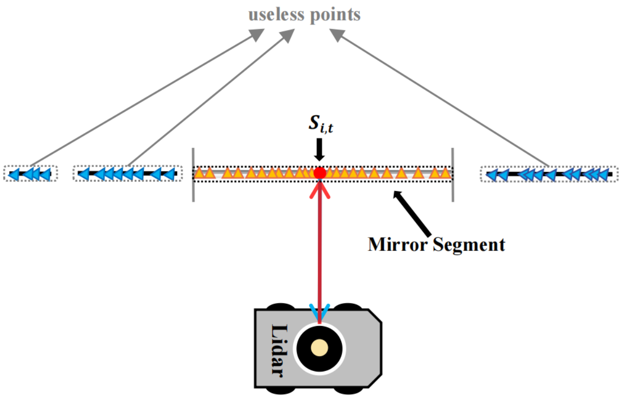


FIGURE 13. The orange triangles in the diagram mark valid decision points, and the line segments in which they are located are judged to be mirror segments.

to be symmetric about the straight line of the mirror, which in turn causes some of the computed intersection points not to be located at the actual mirror location. Therefore, we clustered intersection points. If the distance between two intersection points in the Cartesian coordinate system is less than the threshold d_{limit}, we determine them as points on the same line segment to ensure that all intersection points are located on any one of the line segments. Finally, we obtained the line segment where the detection point $s_{i,t}$ was located as a mirror line segment, as shown in Fig.13.

Once we have the first and last coordinates of the mirror line segments $\left(x_{M,t,start}, y_{M,t,start}\right)$ and $\left(x_{M,t,end}, y_{M,t,end}\right)$, they are transformed to the world coordinate frame according to the lidar pose (x_t, y_t, θ_t) to obtain the absolute position of the mirror in the environment, as shown in Eq.9 to 12:

$$X_{M,start} = x_{M,t,start} \cdot \cos\theta_t - y_{M,t,start} \cdot \sin\theta_t + x_t \tag{9}$$

$$Y_{M,start} = x_{M,t,start} \cdot \sin\theta_t + y_{M,t,start} \cdot \cos\theta_t + y_t \tag{10}$$

$$X_{M,end} = x_{M,t,end} \cdot \cos\theta_t - y_{M,t,end} \cdot \sin\theta_t + x_t \tag{11}$$

$$Y_{M,end} = x_{M,t,end} \cdot \sin\theta_t + y_{M,t,end} \cdot \cos\theta_t + y_t \tag{12}$$

where $\left(X_{M,start}, Y_{M,start}\right)$ and $\left(X_{M,end}, Y_{M,end}\right)$ are the first and last coordinates of the mirror in the world coordinate frame, respectively.

The advantage of this mirror detection method is that once the robot passes over the mirror, as shown in Fig.14, the entire mirror can be instantly detected within the LiDAR scanning range. Simultaneously, polygonal mirrors can be split into multiple combinations of line segments with different orientations, as shown in Fig.15.

V. MIRROR LOCATION UPDATE

For the mirror detection method introduced in Section IV, when the robot is working, 1) multiple lidar data points satisfying the intensity threshold may be detected at the same moment, and the line segments generated based on them may represent a single mirror or multiple mirrors. 2) For lidar data points meeting the intensity threshold detected at different moments, the line segments generated based on them may also represent the same mirror. Therefore, the task in the second stage was to determine whether the line segments generated based on these points were the same. Otherwise, a completely new mirror is created in the world coordinate system; if so, the mirror line segment is updated.

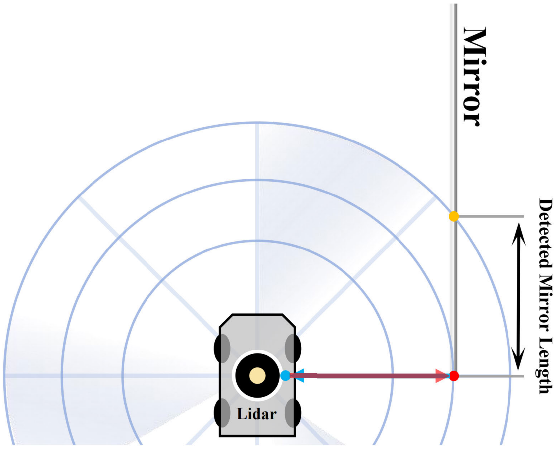


FIGURE 14. For longer mirrors, the method can detect the maximum mirror length within the radar scanning range.

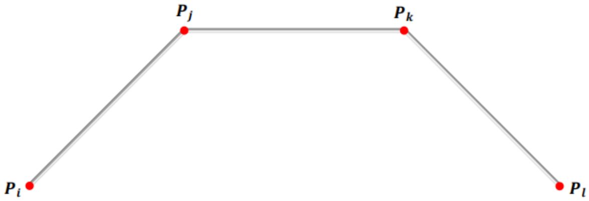


FIGURE 15. For the polygonal mirrors in the figure, three mirror line segments, P_iP_j, P_jP_k, and P_kP_l, will be determined during detection.

A. MULTIPLE MIRROR DETECTION

We considered the mirrors computed in the two cases to be different mirrors. Specifically, these two cases are (1) two mirror line segments that are approximately on a straight line and do not overlap. (2) two mirror line segments on different straight lines that do not intersect. The judgement of these two cases is explained in the following description:

(1) For the newly detected mirror line segment, we determine whether it is approximately a straight line by the minimum distance from the origin of the lidar coordinate frame to the straight line where the segment is located and the direction angle of the segment, as shown in Fig.16. The specific calculation formula is given by Eq.13 to 16:

$$d_{\perp} = \sqrt{x_{\perp}^2 + y_{\perp}^2} \tag{13}$$

$$\theta = \arctan\left(-\frac{x_{\perp}}{y_{\perp}}\right) \tag{14}$$

$$\left|d_{\perp,new} - d_{\perp,old}\right| < \varepsilon_{length} \tag{15}$$

$$\left|\theta_{new} - \theta_{old}\right| < \varepsilon_{angle} \tag{16}$$

where ε_{length} is the tolerated error in length, and ε_{angle} is the tolerated angle error.

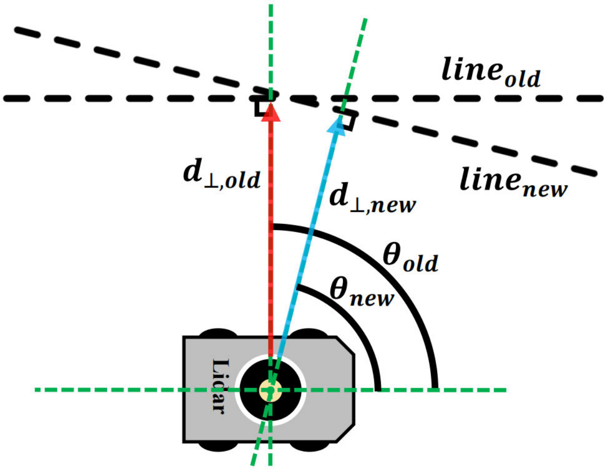


FIGURE 16. The labels θ_{new}, θ_{old}, $d_{\perp,new}$ and $d_{\perp,old}$ in the figure are the variables that determine whether two mirror segments are in the same straight line or not.

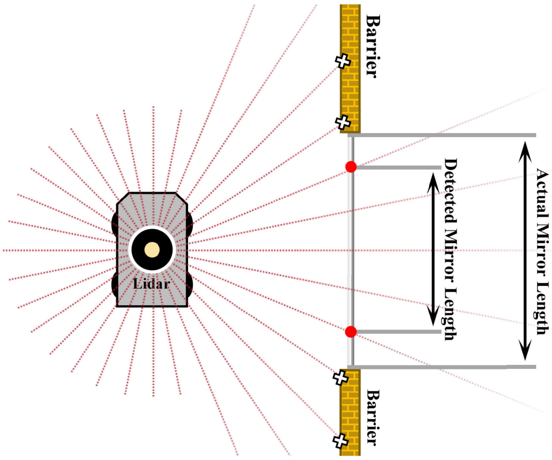


FIGURE 17. Comparison between the detected and actual mirror lengths by this method.

For two mirror line segments that are approximately on a straight line, it is determined that the segments do not coincide if $x_{new,max} < x_{old,min}$ or $x_{new,min} > x_{old,max}$.

(2) The same method as in (1) was used to determine whether the two mirror line segments were on different straight lines. When they are not in a straight line, they are naturally in different lines. We used the Bentley-Ottmann [24] algorithm to determine whether they were disjointed or not.

Here, we consider the case in which the old and new mirror line segments are on different straight lines and intersect because of error detection. Once this happens, we do not process the obtained mirror segments, but wait for the input of the correct segments.

B. SINGLE MIRROR UPDATE

Because lidar detects the same position at different moments there may be small position fluctuations, resulting in a small gap in the slope and intercept of the calculated specular line segment for the same lidar data point. At the same time, because of the effect of the lidar angular resolution, the measurement points located farther away from the lidar were sparser, resulting in the actual length of the acquired mirror line segment being smaller than the actual length of the mirror, as shown in Fig.17. Therefore, for the same mirror line segments $P_{i,t-1}P_{j,t-1}$ and $P_{a,t}P_{b,t}$ acquired at different moments, we updated and merged them to obtain new mirror line segments.

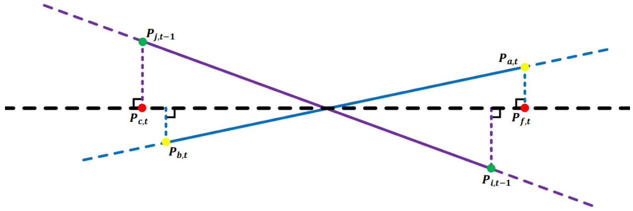


FIGURE 18. Line segment updating using least squares.

Data points are first collected on a straight line where $P_{i,t-1}P_{j,t-1}$ and $P_{a,t}P_{b,t}$ are located, denoted as $(x_{1p,t-1}, y_{1p,t-1})$ and $(x_{2q,t}, y_{2q,t})$, where p and q denote the p^{th} and q^{th} data points on the two straight lines, respectively. For the updated mirror straight line $y = k_{update}x + b_{update}$, the following error function is available:

$$E(k_{update}, b_{update}) = \sum_{p}\left(y_{1p,t-1} - \left(k_{update}x_{1p,t-1} + b_{update}\right)\right)^2 + \sum_{q}\left(y_{2q,t} - \left(k_{update}x_{2q,t} + b_{update}\right)\right)^2 \tag{17}$$

Minimising the error function gives the best-fit k_{update} and b_{update}.

After obtaining the updated mirror straight line, the first and last endpoints of $P_{i,t-1}P_{j,t-1}$ and $P_{a,t}P_{b,t}$ are made to be the perpendiculars of the updated specular straight line, the intersection of the perpendiculars and the straight line is taken as the updated endpoint, and the two farthest points of the updated mirror line segment are taken as the new endpoints. As shown in Fig.18, the updated mirror line segment is $\hat{P}_{c,t}\hat{P}_{f,t}$.

By updating the mirrors, not only can the computational increase in point cloud optimization due to mirror stacking be reduced, but the length of the obtained mirrors can also be constantly approximated to the actual length, thus improving the detection accuracy.

VI. POINT CLOUD DATA OPTIMIZATION

For all the mirrors obtained in the second phase, which are in the world coordinate frame, we want to stop outputting incorrect distance information because of the mirrors during the lidar detection. Therefore, in the third stage, we modified the point cloud data returned by the lidar at the current moment based on the mirrors.

To reduce the computational effort, we first computed the indices n_{start} and n_{end} of the coordinates of the first and last coordinates of the mirror line segments in the laser scan data at the current moment k. We obtained the updated mirror line segments based on the world coordinate system in V. Let the current lidar pose be (x_k, y_k, θ_k). Next, we converted it to the lidar coordinate frame, as shown in Eq.18 and 21:

$$\begin{aligned}(x_{M,\text{start}}, y_{M,\text{start}}) &= \big((X_{M,\text{start}} - x_k)\cos\theta_k - (Y_{M,\text{start}} - y_k)\sin\theta_k, \\ &\quad (X_{M,\text{start}} - x_k)\sin\theta_k + (Y_{M,\text{start}} - y_k)\cos\theta_k\big) \end{aligned} \tag{18}$$

$$\begin{aligned}(x_{M,\text{end}}, y_{M,\text{end}}) &= \big((X_{M,\text{end}} - x_k)\cos\theta_k - (Y_{M,\text{end}} - y_k)\sin\theta_k, \\ &\quad (X_{M,\text{end}} - x_k)\sin\theta_k + (Y_{M,\text{end}} - y_k)\cos\theta_k\big) \end{aligned} \tag{19}$$

where $(x_{M,start}, y_{M,start})$ and $(x_{M,end}, y_{M,end})$ are the first and last coordinates of the mirror line segment based on the lidar coordinate frame at the current moment k after conversion, and $(X_{M,start}, Y_{M,start})$ and $(X_{M,end}, Y_{M,end})$ are the first and last coordinates of the mirror line segment based on the world coordinate frame.

$$n_{start} = (int)\frac{\arctan\frac{y_{M,start}}{x_{M,start}} - \theta}{\Delta\theta} \tag{20}$$

$$n_{end} = (int)\frac{\arctan\frac{y_{M,end}}{x_{M,end}} - \theta}{\Delta\theta} \tag{21}$$

For each mirror line segment, we calculated the angular range in the lidar coordinate frame. For any laser beam within that range, if the laser scanning line segment produced intersects with the mirror line segment, the laser coordinates produced by the laser beam are replaced with the intersection coordinates; otherwise, the original laser coordinates are retained. If a laser scanning line segment intersects multiple mirror line segments, we select the intersection coordinates closest to the origin of the lidar coordinate frame for replacement, i.e., the distance value corresponding to the laser in the laser scan data is modified to the value of the distance from the origin of the lidar coordinate frame to that intersection, as shown in Fig.19.

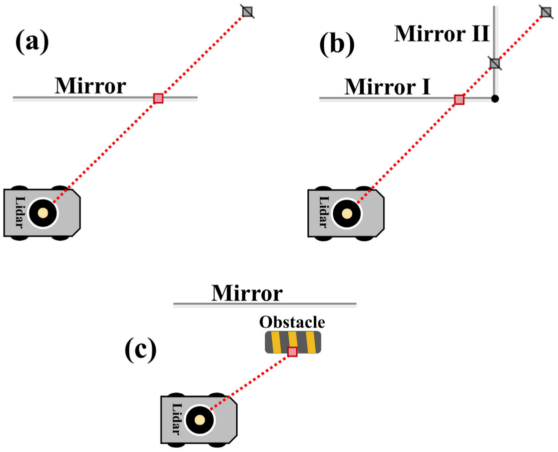


FIGURE 19. Three modifications of the laser ray collision with mirrors: (a) the laser ray intersects with a single mirror, modifying the scanning position to mark the red intersection; (b) the laser ray intersects with multiple mirrors, modifying the scanning position to the nearest intersection with a radar coordinate system, which is marked with a red dot in the figure; (c) the laser ray does not intersect with mirrors, retaining the original distance data.

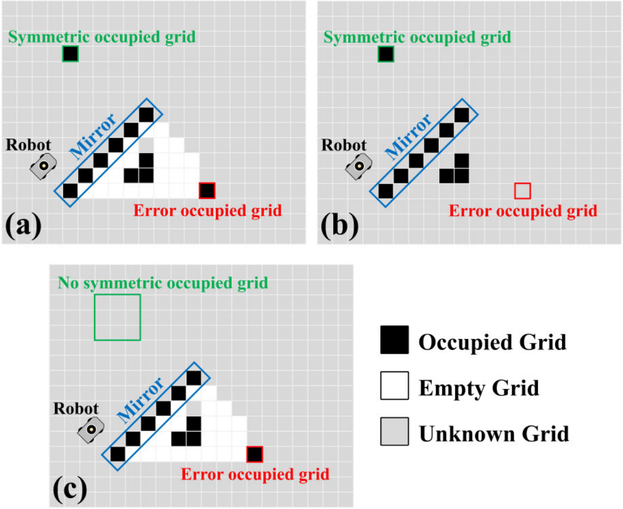


FIGURE 20. For a single error occupied grid, the map modification principle: (a) before modification; (b) after modification; (c) No occupied grid and error occupied grid symmetry about mirrors.

VII. OCCUPY GRID MAP OPTIMIZATION

In the fourth stage, we use the open-source SLAM algorithm Karto to implement map building, which is different from the global matching methods of ICP [25] and NDT [26], which adopt a sub-map based point cloud matching approach to model and match each sub-map, thus reducing the computational complexity and improving the efficiency of matching. Optimized laser scan data were used as the input. First, the output of the position transformation between two consecutive moments was used as the input to the IV. When the robot has not yet passed through the mirror, lidar incorrectly constructs the occupancy grid map through the mirror. When the robot detects the mirror, the optimized laser scan data no longer produce incorrect map information. Therefore, we only needed to modify the previously generated erroneous occupancy grid map at the moment of the mirror detection. We searched for all occupation grids that were symmetric about the mirror position within the lidar scanning range, obtained the symmetric occupation grid that was far away from the robot, and set it to unknown. Simultaneously, the free region in the line connecting it to all the points on the mirror is unknown. As shown in Fig.20(a)(b). The points on the mirror were obtained using the Bresenham line drawing algorithm.

This optimization method allows the map data to be modified in real time, although it has some drawbacks: (1) it may set a small number of regions that are not generated by the mirrors as unknown; (2) the method will not optimize for incorrectly occupied grids generated by the mirrors if no occupied grid symmetrical to them is generated outside the mirrors, as shown in Fig.20(c); and (3) for the actual region that has been set to unknown, the robot will need to rescan the region to restore it to free. However, in practice, these three

TABLE 1. Experiment 1: Mirror recognition results.

	Li [15]	Mora [23]	Ours
Mirror1(60cm) accuracy	94.3%	98.5%	98.2%
Real time	Yes	No	Yes
Maximum time consumption(ms)	5.77		3.46

TABLE 2. Experiment 2.1: Mirror recognition results.

	Li [15]	Mora [23]	Ours
Mirror1(80cm) accuracy	87.4%	95.3%	97.1%
Mirror2(40cm) accuracy	93.9%	98.1%	98.2%
Mirror3(30cm) accuracy	94.7%	98.8%	99.1%
Real time	Yes	No	Yes
Maximum time consumption(ms)	8.62		5.11

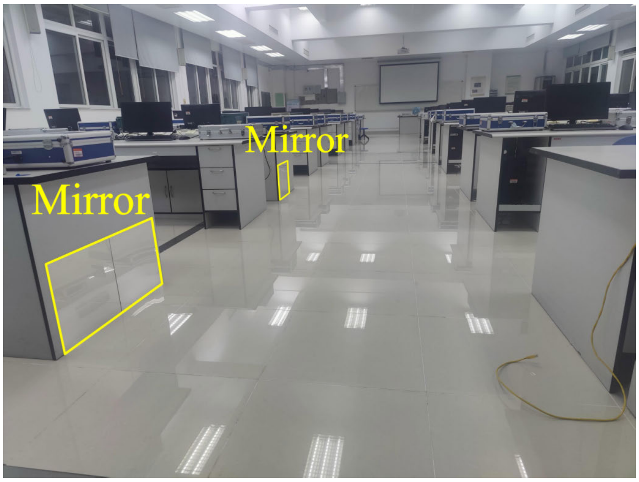


FIGURE 21. Experiment 2.1: Experimental environment.

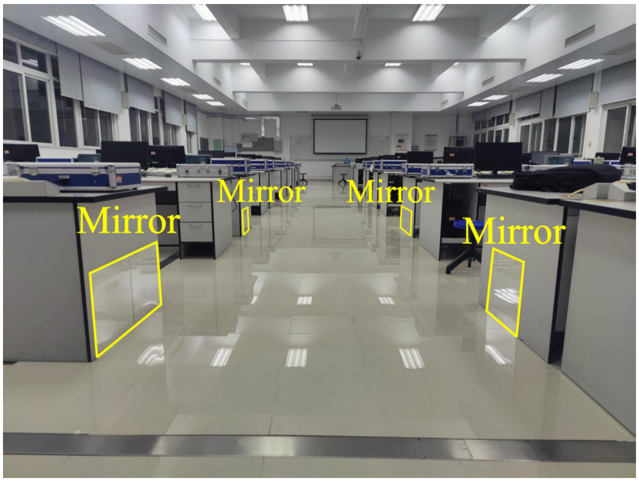


FIGURE 22. Experiment 2.2: Experimental environment.

shortcomings have a small impact on the overall optimization results. This method achieved good optimization results.

VIII. EXPERIMENTAL RESULTS

In this section, four environments with mirrors are built and the performance of our method is compared with the method proposed by Mora et al. [23] and Li et al. [15] in these environments.In the demonstration of performance, the accuracy is the ratio of the length of the mirror calculated by the algorithm to the actual length of the mirror.

TABLE 3. Experiment 2.2: Mirror recognition results.

	Li [15]	Mora [23]	Ours
Mirror1(80cm) accuracy	87.2%	95.9%	97.7%
Mirror2(40cm) accuracy	93.8%	96.6%	98.2%
Mirror3(30cm) accuracy	93.6%	96.4%	98.8%
Mirror4(60cm) accuracy	89.4%	95.7%	97.8%
Mirror5(30cm) accuracy	94.3%	96.9%	98.9%
Real time	Yes	No	Yes
Maximum time consumption(ms)	10.62		4.93

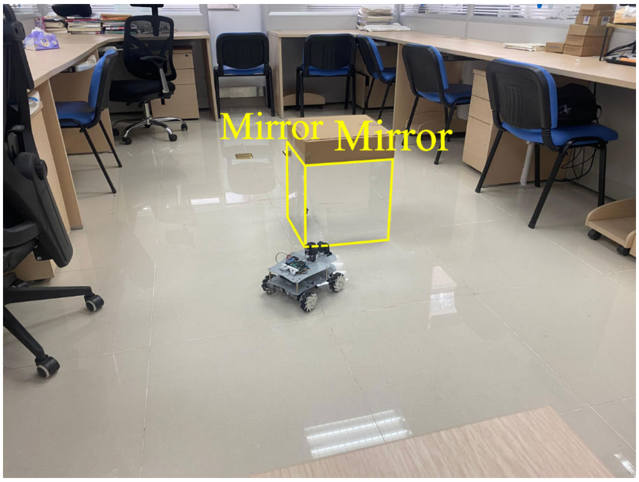


FIGURE 23. Experiment 3: Experimental environment.

TABLE 4. Experiment 3: Mirror recognition results.

	Li [15]	Mora [23]	Ours
Mirror1(40cm) accuracy	94.1%	97.3%	98.8%
Mirror2(40cm) accuracy	94.5%	97.7%	98.2%
Mirror3(40cm) accuracy	95.2%	96.8%	99.0%
Mirror4(40cm) accuracy	93.4%	97.4%	98.6%
Real time	Yes	No	Yes
Maximum time consumption(ms)	7.32		6.48

A. EXPERIMENT 1: DARKROOM CORRIDOR, ELECTRONIC INFORMATION LABORATORY BUILDING

In order to verify the performance of the method, we first tested it in a simple environment, the darkroom corridor of the Electronic Information Laboratory Building, where a 60cm*30cm mirror was attached to one of the side walls, as shown in Fig.9. By allowing the robot to move slowly along the specified path, a map constructed using the point cloud before optimization and a map constructed using the point cloud after optimization were acquired, as shown in Fig.24(a).

In this experiment, our main purpose was to debug some parameters in the method and find the relatively optimal parameters for mirror identification, after continuous testing, we set the intensity threshold in IV to 250, k_{min} to -1.07, k_{max} to -0.93, d_{min} to 0.08, and d_{limit} to 0.03. In the test, it was found that if the range from k_{min} to k_{max} is too large, it will lead to an incorrect match between symmetric points

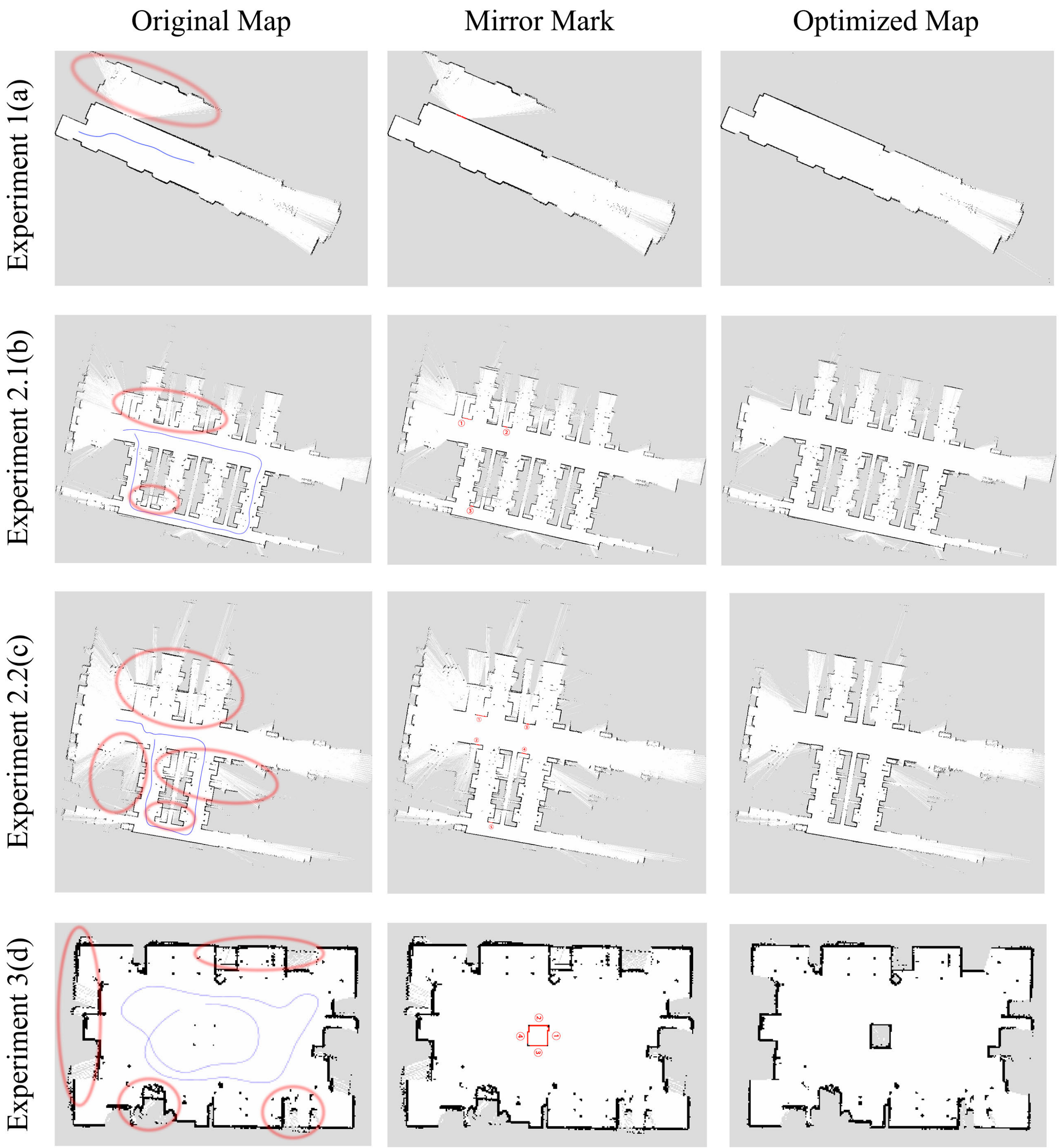


FIGURE 24. Experimental results in four experimental environments: the first column is the original map, where the blue curve indicates the robot's movement path and inside the red circle is incorrect map information; the second column is the markers of the mirror's location; the third column is the optimised map.

and a small range will lead to some symmetric points being difficult to detect. mismatch between them, and a range that is too small can make some symmetry points difficult to detect. If d_{min} is small, the two very close points on the line where the mirror is located will be judged as symmetry points, whereas a larger range will make it difficult to detect some of the symmetry points that are actually generated by the mirror.

B. EXPERIMENT 2: CLASSROOM, ELECTRONIC INFORMATION LABORATORY BUILDING

In Experiment 1, we ensured that the robot was equipped with basic mirror recognition and map building optimization capabilities. In Experiment 2, we tested the robot in a more complex environment by choosing a classroom in the Electronics and Information Lab Building, a relatively spacious environment with more obstacles and the presence of multiple symmetric objects. We conducted two experiments in this environment to verify the effectiveness of the proposed scheme for multiple mirror detection.

1) EXPERIMENT 2.1: TWO MIRRORS ON THE SAME LINE

We placed a mirror of 80cm*40cm and a mirror of 40cm*40cm on the same straight line, as well as a mirror of 30cm*30cm further away, as shown in Fig.21. Through testing, we set the parameters ε_{length} and ε_{angle} to 0.15 and 2Â°. The robot moved slowly along the blue trajectory, the experimental results are shown in Fig.24(b).

2) EXPERIMENT 2.2: TWO MIRRORS PLACED FACE TO FACE

Four mirrors were placed face-to-face in the path of the robot's movement: 80cm*40cm, 40cm*40cm, 60cm*30cm, and 30cm*30cm. also, a 30cm*30cm mirror was placed farther away, as shown in Fig.22. Through testing, the mirror was correctly identified and optimised for the point cloud and map as the robot moved. Some of the incorrectly occupied grids caused by mirrors were not completely removed from the occupancy grid map, which we analyzed. The possible reasons for this are described in Section VII, but the optimization effect of the method can still be clearly seen, as shown in Fig.24(c).

C. EXPERIMENT 3: STUDENT OFFICE, ELECTRONIC INFORMATION LABORATORY BUILDING

To verify that our mirror recognition method can detect mirrors lacking a bezel in a symmetric environment, a third experiment was conducted, in which a small student office was selected. Eight highly symmetric desks existed in this area. To reduce the size of the build, we placed cardboard at the entrance of the area. We built a cardboard box with dimensions 50cm*40cm*40cm and pasted four mirrors with dimensions 40cm*40cm around it as shown in Fig.23. The robot builds a map along the blue trajectory. It can be seen that it walked around the cardboard box twice, successfully detecting the four mirrors during the first lap and eliminating the resulting erroneous builds. However, some of the free regions that were recognised were incorrectly set as unknown. Therefore, during the second lap, the robot could detect these regions by scanning them and resetting them free. The experimental results are shown in Fig.24(d).

D. DATA RESULTS

The mirror detection results are summarized in Table 1234. The detection accuracy was obtained from the mirror length output using the method and the actual mirror lengths. Overall, the results were greater than 97% accurate, although the results would be better in a more confined and simpler environment. We compared our results with those of state-of-the-art work. The results show that the proposed method improves recognition accuracy and real-time performance. In addition, our method can make the occupancy grid map more accurate which will greatly aid robot localization and path planning.

IX. CONCLUSION AND FUTURE WORK

This paper describes an efficient mirror-detection method that uses a single 2D LiDAR. By analyzing the intensity values of the returned laser scan data, it is possible to determine the linear position of the mirror in the environment. Next, the exact location of the mirror was back-calculated by processing the symmetric point cloud generated by the mirror. Finally, the point cloud data and occupancy grid map were modified in real time to prevent the robot from being disturbed by the erroneous environmental information generated by the mirrors while working.

To validate the performance of the method, we constructed three complex environments and compared it with those proposed by Mora et al. [23] and Li et al. [15]. The experimental results show that the proposed method improves recognition accuracy and real-time performance.

Although the current method has improved in accuracy and real-time, the method still has some shortcomings in modifying the occupancy grid map, it is not able to completely eliminate the erroneous map information generated by mirrors, which is due to the fact that the map information at the two ends of the mirrors is not completely symmetrical to each other due to obstacle obstructions or the blind spot of the lidar itself before the mirrors are recognised. In addition, in degraded environments with a lack of features such as long corridors and lobbies, the robot's localization will be most affected by erroneous radar information before the mirror is recognized, and the point cloud's features based on the symmetry of the mirror will be blurred, which will affect the effectiveness of mirror detection.

These problems may be solved by processing multiple frames of previous lidar scan data simultaneously at each moment in time, however, this results in a significant consumption of computational resources.

In future work, we expect to calculate the probability of identifying each lidar scan point in front of the mirror as incorrect, which can be used to minimize the building of incorrect map and also to make the robot's localization more accurate.

REFERENCES

[1] M. F. Ahmed, K. Masood, V. Fremont, and I. Fantoni, "Active SLAM: A review on last decade," *Sensors*, vol. 23, no. 19, p. 8097, Sep. 2023.

[2] K. Sugiura and H. Matsutani, "A universal LiDAR SLAM accelerator system on low-cost FPGA," *IEEE Access*, vol. 10, pp. 26931–26947, 2022.

[3] M. Tu, P. Zeng, Q. Wu, T. Jing, Y. Tian, Y. Luo, and W. Mao, "Lidar SLAM based on particle filter and graph optimization for substation inspection," *IEEE Access*, vol. 10, pp. 127540–127549, 2022.

[4] J. Guivant, E. Nebot, and S. Baiker, "Localization and map building using laser range sensors in outdoor applications," *J. Robotic Syst.*, vol. 17, no. 10, pp. 565–583, Oct. 2000.

[5] P. Su, S. Luo, and X. Huang, "Real-time dynamic SLAM algorithm based on deep learning," *IEEE Access*, vol. 10, pp. 87754–87766, 2022.

[6] X. Liu, J. Zhang, B. Wang, and Q. Jiang, "A dense visual SLAM method in dynamic scenes," *IEEE Access*, vol. 11, pp. 138530–138539, 2023.

[7] A. M. Barros, M. Michel, Y. Moline, G. Corre, and F. Carrel, "A comprehensive survey of visual SLAM algorithms," *Robotics*, vol. 11, no. 1, p. 24, Feb. 2022.

[8] C. Campos, R. Elvira, J. J. G. Rodríguez, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM3: An accurate open-source library for visual, visual–inertial, and multimap SLAM," *IEEE Trans. Robot.*, vol. 37, no. 6, pp. 1874–1890, Dec. 2021.

[9] G. Peng, Y. Zhou, L. Hu, L. Xiao, Z. Sun, Z. Wu, and X. Zhu, "VILO SLAM: Tightly coupled binocular Vision–Inertia SLAM combined with LiDAR," *Sensors*, vol. 23, no. 10, p. 4588, May 2023.

[10] C. Tian, H. Liu, Z. Liu, H. Li, and Y. Wang, "Research on multi-sensor fusion SLAM algorithm based on improved gmapping," *IEEE Access*, vol. 11, pp. 13690–13703, 2023.

[11] J. Jiang, R. Miyagusuku, A. Yamashita, and H. Asama, "Glass confidence maps building based on neural networks using laser range-finders for mobile robots," in *Proc. IEEE/SICE Int. Symp. Syst. Integr. (SII)*, Dec. 2017, pp. 405–410.

[12] H. Mei, X. Yang, L. Yu, Q. Zhang, X. Wei, and R. W. H. Lau, "Large-field contextual feature learning for glass detection," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 45, no. 3, pp. 3329–3346, Mar. 2023.

[13] Y. Tao, H. Gao, Y. Wen, L. Duan, and J. Lan, "Glass recognition and map optimization method for mobile robot based on boundary guidance," *Chin. J. Mech. Eng.*, vol. 36, no. 1, p. 74, Jun. 2023.

[14] S.-W. Yang and C.-C. Wang, "On solving mirror reflection in LiDAR sensing," *IEEE/ASME Trans. Mechatronics*, vol. 16, no. 2, pp. 255–265, Apr. 2011.

[15] Z. Li, M. Huang, Y. Yang, Z. Li, and L. Wang, "A mirror detection method in the indoor environment using a laser sensor," *Math. Problems Eng.*, vol. 2022, pp. 1–12, Jun. 2022.

[16] S. Pu and G. Vosselman, "Knowledge based reconstruction of building models from terrestrial laser scanning data," *ISPRS J. Photogramm. Remote Sens.*, vol. 64, no. 6, pp. 575–584, Nov. 2009.

[17] W. Hao, Y. Wang, W. Liang, X. Ning, and Y. Li, "Slice-based window detection from scene point clouds," in *Proc. Int. Conf. Virtual Reality Visualizat. (ICVRV)*, Oct. 2018, pp. 35–39.

[18] R. Wang, F. P. Ferrie, and J. Macfarlane, "A method for detecting windows from mobile LiDAR data," *Photogramm. Eng. Remote Sens.*, vol. 78, no. 11, pp. 1129–1140, Nov. 2012.

[19] T. Shiina and Z. Wang, "An indoor navigation algorithm incorporating representation of quasi-static environmental object and glass surface detection using LRF sensor," in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Dec. 2017, pp. 2508–2514.

[20] X. Wang and J. Wang, "Detecting glass in simultaneous localisation and mapping," *Robot. Auto. Syst.*, vol. 88, pp. 97–103, Feb. 2017.

[21] H. Tibebu, J. Roche, V. De Silva, and A. Kondoz, "LiDAR-based glass detection for improved occupancy grid mapping," *Sensors*, vol. 21, no. 7, p. 2263, Mar. 2021.

[22] L. Weerakoon, G. S. Herr, J. Blunt, M. Yu, and N. Chopra, "Cartographer_glass: 2D graph SLAM framework using LiDAR for glass environments," 2022, *arXiv:2212.08633*.

[23] A. Mora, A. Prados, P. González, L. Moreno, and R. Barber, "Intensity-based identification of reflective surfaces for occupancy grid map modification," *IEEE Access*, vol. 11, pp. 23517–23530, 2023.

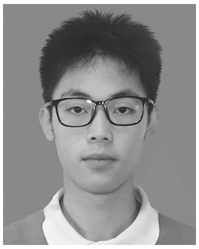
[24] J.-D. Boissonnat and F. P. Preparata, "Robust plane sweep for intersecting segments," *SIAM J. Comput.*, vol. 29, no. 5, pp. 1401–1421, Jan. 2000.

[25] P. Dellenbach, J.-E. Deschaud, B. Jacquet, and F. Goulette, "CT-ICP: Real-time elastic LiDAR odometry with loop closure," in *Proc. Int. Conf. Robot. Autom. (ICRA)*, May 2022, pp. 5580–5586.

[26] S. Bouraine, A. Bougouffa, and O. Azouaoui, "Particle swarm optimization for solving a scan-matching problem based on the normal distributions transform," *Evol. Intell.*, vol. 15, no. 1, pp. 683–694, Mar. 2022.

JING ZHU received the degree from South China Normal University, in 2005. She is currently pursuing the master's degree in optics. She was with Guangzhou University. She is a Lecturer. Her research interests include mobile robots and image-processing technology.

SIBO WANG was born in Guangdong, China, in 2003. He is currently pursuing the degree with the School of Electronics and Communication Engineering, Guangzhou University. His areas of research interests include robot position and robot navigation. At present, the mirror recognition method of single 3D LiDAR and the improvement of the traditional NDT point cloud registration method are being studied.



CANHONG LIN was born in Guangdong, China, in 2004. He is currently pursuing the degree with the School of Electronics and Communication Engineering, Guangzhou University. His areas of research interest includes multiple LiDAR slams.

BANGZHENG YIN received the degree from South China Normal University, in 2004. He is currently pursuing the master's degree in optics. He was with Guangzhou Railway Polytechnic. He is an Associate Professor. His research interests include mobile robots and artificial intelligence.