

# LiDAR-Based 3-D Glass Detection and Reconstruction in Indoor Environment

Lelai Zhou<sup>ID</sup>, Member, IEEE, Xiaohui Sun<sup>ID</sup>, Chen Zhang<sup>ID</sup>, Luyang Cao<sup>ID</sup>, and Yibin Li<sup>ID</sup>

**Abstract**—As an optical sensor, LiDAR cannot effectively detect transparent obstacles in the environment, such as glass parapets and glass doors in buildings. In this article, a glass detection and point cloud reconstruction method relying only on LiDAR is proposed, which can improve the detection success rate for glass and achieve the reconstruction of glass. The algorithm achieves the detection of glass point clouds by reflection intensity and local structure feature of point clouds. Experiments prove that the algorithm can well identify the glass position and reconstruct the point cloud of glass. It lays the foundation for mobile robots to carry out simultaneous localization and mapping (SLAM) and path planning tasks smoothly in the indoor environment with glass.

**Index Terms**—Glass detection, LiDAR, point cloud reconstruction, reflectance intensity, simultaneous localization and mapping (SLAM).

## I. INTRODUCTION

ROBOTICS has achieved rapid development in the last decade, with various models of mobile robots playing an important role in emergency rescue, home services, and environmental exploration. With the development of various sensors and the application of robotics algorithms, the work of mobile robots has gradually involved all aspects of human life. The prerequisite for mobile robots to accomplish their task is the ability to accurately sense and process information about the surrounding environment. Due to the advantages of high accuracy and stability of data, LiDAR is commonly used as the main sensor to collect environmental information in the process of simultaneous localization and mapping (SLAM) and navigation of mobile robots. As an optical sensor, LiDAR relies on emitting and receiving reflected laser beams to acquire environmental information. For some low diffuse reflectance objects, such as glass, mirrors, metal, or tiles, the diffuse reflection of light beams occurring on their surfaces is

Manuscript received 6 December 2023; revised 16 January 2024; accepted 29 January 2024. Date of publication 14 March 2024; date of current version 22 March 2024. This work was supported in part by the Shandong Provincial Natural Science Foundation for Distinguished Young Scholars under Grant ZR2022JQ028, in part by the National Natural Science Foundation of China under Grant 61973191 and Grant 91948201, and in part by the Key Development Program for Basic Research of Shandong Province under Grant ZR2019ZD07. The Associate Editor coordinating the review process was Dr. Yan Zhuang. (*Corresponding author: Lelai Zhou*.)

The authors are with the Center for Robotics, School of Control Science and Engineering, Shandong University, Jinan 250061, China, and also with the Engineering Research Center of Intelligent Unmanned System, Ministry of Education, Jinan 250061, China (e-mail: zhoullelai@sdu.edu.cn; sun\_xh@mail.sdu.edu.cn; zchen@mail.sdu.edu.cn; clyyy@mail.sdu.edu.cn; liyb@sdu.edu.cn).

Digital Object Identifier 10.1109/TIM.2024.3375965



Fig. 1. Glass in the indoor environment.

weak and difficult to receive and recognize by LiDAR. Specularly reflected beams on such surfaces are perceived as return results, i.e., the LiDAR ignores these low diffuse reflectance obstacles and obtains the wrong object position. When facing a common transparent obstacle like glass, the laser beam will pass through the transparent obstacle to directly obtain information of the object on the other side of the transparent obstacle. When mobile robots perform tasks in indoor scenes such as schools, office buildings, and shopping malls, LiDAR-based localization is unreliable in these environments [1], [2]. Robots often face problems such as missing map content and failed path planning, leading to safety problems such as falling and collision of mobile robots [3]. It is noteworthy that modern buildings often use glass materials to improve building lighting and esthetics. The proportion of glass materials used has increased significantly, which further limits the range of activities of mobile robots. These transparent obstacles in buildings are mainly reflected in glass curtain walls, glass doors, glass parapets, and so on, as shown in Fig. 1.

Due to the specular reflection of the glass, beams incident on the glass at normal and nearby angles are picked up by the LiDAR. The point cloud generated from the glass is obtained. The following are a few contributing aspects of the method proposed in this article.

- 1) The method combines the reflection intensity feature and local structure feature to achieve LiDAR-only glass detection and reduce the complexity of the overall system.
- 2) The rich geometric information of 3-D point clouds, which are different from 2-D point clouds, are fully utilized. The extraction method of reflection intensity feature under 3-D distribution is proposed. Four local structure features are proposed to identify the point cloud.

- 3) In the reflection intensity feature part, the concept of divergence is proposed to represent the reflection intensity distribution and extract the intensity peak.
- 4) The algorithm achieves point cloud reconstruction of flat or curved glass, regardless of whether the glass has a boundary or not.

The algorithm works well in indoor environment and obtains high detection efficiency and accuracy.

## II. RELATED WORK

In recent years, research studies on detecting glass during SLAM and navigation of mobile robots have gradually increased. The methods proposed in the past studies are as follows.

### A. Multisensor Fusion

Several attempts to detect glass with multiple sensors have been made. Researchers use the acoustic properties of ultrasound [4], [5] to obtain the correct glass position [6], [7]. In addition to pairing ultrasound with LiDAR, the researchers also fuse ultrasound data with depth camera data [8] to obtain the location of the glass. Relying on different optical properties, a polarization camera is added [9] to supplement the missing glass information.

However, ultrasonic sensors can only measure obstacles in their direction in a small area, which is limited in the 3-D environment. The ultrasonic sensors need to be installed close to each other, which will lead to mutual interference. The multisensor fusion method can make good use of the characteristics of different sensors, but it is complex in terms of installation structure and fusion algorithm. Multiple sensors will make the system less robust and the results less accurate.

### B. Symmetric Features

Yang and Wang [10] detect glass by matching the symmetric features of specular reflection, a simpler way to detect glass is to use a camera to detect laser reflections [11]. In [12], the algorithm estimates the glass by identifying all multireturns for each emitted laser beam to detect specular reflection trajectories and remove erroneous points caused by specular reflections. Specular reflection matching is easy to implement in 2-D point clouds, but not in large-scale 3-D point clouds in real time. LiDAR with multireturns allows simple and fast identification of beams with multiple reflections. The multireturns approach increases efficiency but imposes requirements on the LiDAR model.

### C. Reflection Characteristics

In addition to the fact that specular reflection causes the same point cloud to be distributed on both sides of the glass, it also increases the reflection intensity. Filtering the high reflection intensity from specular reflection and determining the boundary for the glass using multi returns [13] is another way to exploit the characteristics of specular reflection. To further exploit the reflection intensity characteristics, Wang and Wang [14] set three parameters

in the algorithm to filter the reflection intensity feature to detect glass and have combined the algorithm into the SLAM process. An occupied grid improvement algorithm [15] using reflection intensity is proposed to preserve nondiffuse surfaces. In terms of the geometric structure of the point cloud, Cui et al. [16] rely only on the structural features of the point cloud to determine the glass existence as well as the boundary. Tibebu et al. [17] observe that the point cloud passing through the glass is more divergent and have designed two filters to filter the beam passing through the glass to detect the glass. Kim and Chung [18] have achieved localization in a complex glass environment based on different reflective characteristics mentioned above. It has good localization performance but does not enable the reconstruction of point clouds to produce a correct map of the environment. Relying only on the reflection intensity feature or point cloud structure feature may cause problems in practical use, as real glass is easily missed and nontransparent obstacles are incorrectly recognized.

### D. Neural Network

To avoid the errors caused by different signal processing mechanisms of each sensor model or different reflective materials, Jiang et al. [19] input the point cloud reflection intensity, incidence angle, and distance to the neural network to detect the glass. The use of neural networks to process visual images [20] is a more widespread approach compared to using data from LiDAR. Mei et al. [21] proposed GDNet to detect glass in visual images. Neural network methods rely heavily on the training data.

Considering the advantages and disadvantages of the various methods, we decide to make full use of the rich intensity and geometric structure information of the 3-D point cloud for the detection of glass. Our method is fundamentally different from the features of the related works method. We propose the method of feature extraction with divergence value and four local structure features for glass, which can represent the glass features better. We reconstruct glass based on the environment information.

## III. METHOD

A light beam with glass present in the path contains three superimposed beams when it returns to the starting point. They are diffuse reflections generated by nontransparent obstacles behind the glass, specular and diffuse reflections back from the glass as shown in Fig. 2. These three return beams have different reflection intensities. Usually, a LiDAR will select the beam with max reflection intensity as the result of that laser beam. That is, a beam from an object behind glass or a specular reflected object. However, when the angle of incidence is around  $0^\circ$ , the reflection intensity of the specular reflection from the glass is much greater than the reflection intensity from the object behind the glass. This allows the LiDAR to obtain a point cloud of the glass.

### A. Reflection Intensity Feature

Extremely high reflection intensity can be received when the beam is incident along the glass plane's normal, i.e.,  $0^\circ$ . In a

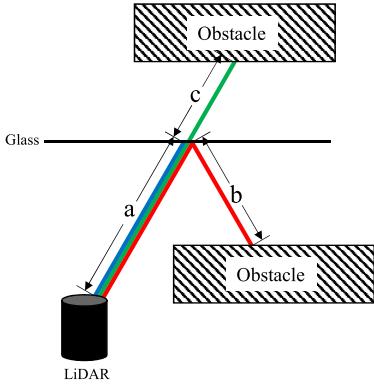


Fig. 2. Return beam received by the LiDAR, the red line is the specular reflected beam of the glass, passing through a distance of  $2(a + b)$ . The blue line is the diffuse reflected beam of the glass, passing through a distance of  $2a$ . The green line is the reflected beam of the nontransparent obstacle, passing through a distance of  $2(a + c)$ .

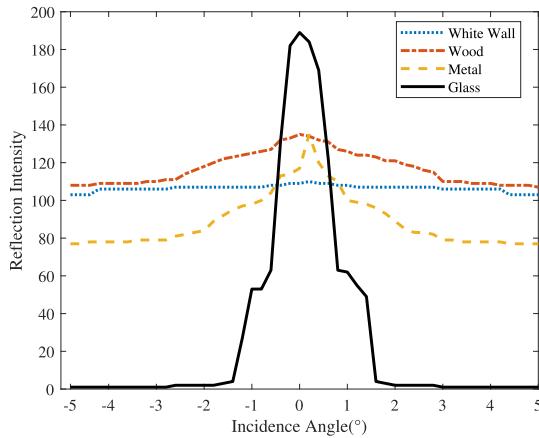


Fig. 3. Relationship curve between reflection intensity and incident angle for different materials. The reflection intensity range is 0–255.

very small range near  $0^\circ$ , the reflection intensity drops rapidly. The reflection curve of wood or metal has a flat decline and less reflective intensity peak. The reflection intensity received at different incidence angles in the horizontal direction is shown in Fig. 3. In addition, according to both theoretical and practical observations, the relationship between the reflection intensity and the incident angle in the vertical direction has the same variation characteristics as that in the horizontal direction, as shown in Fig. 4. It is worth noting that most multiline LiDAR point clouds in the horizontal direction are more denser than those in the vertical direction. Each ring of a 16-line LiDAR scans a circle in the horizontal direction with 1800 points, while the vertical direction only has 16 points. The emission angle interval of the horizontal laser beam is smaller than that of the vertical direction, but this does not affect the objective distribution of the reflection intensity and the reflection intensity of the sampled point cloud data.

Based on the reflection intensity features of the glass point cloud, a filtering algorithm is proposed. The main purpose is to identify whether the distribution of the reflection intensity of the point cloud matches the above intensity distribution. The algorithm views an individual point cloud in the LiDAR view. The LiDAR coordinate system is a Cartesian coordinate

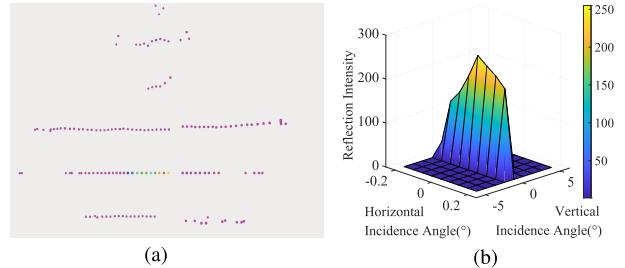


Fig. 4. Glass reflection intensity visualization. (a) Glass point cloud with reflection intensity in the rainbow color scale from purple (0) to red (255). (b) Intensity surface.

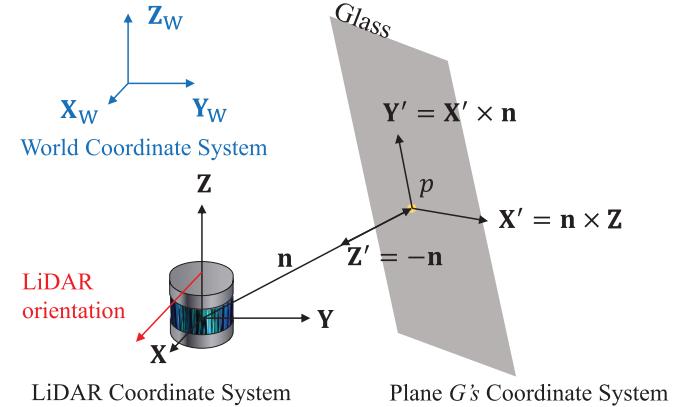


Fig. 5. Relationship between the LiDAR coordinate system and the glass plane coordinate system.

system with the origin at the center of the LiDAR. The LiDAR orientation is along  $\mathbf{x}$ -axis,  $\mathbf{y}$ -axis is perpendicular to the left of the LiDAR orientation, and the  $\mathbf{z}$ -axis is perpendicular to the  $\mathbf{XOY}$  plane upward as shown in Fig. 5.

Before introducing the reflection intensity feature filtering algorithm, the mathematical notation and the parameters in the algorithm are explained.

- 1) *A Single Point:*  $p = [p_x, p_y, p_z]$  denotes a single point in 3-D space.
- 2) *Set of Points:*  $\{\mathbf{P}\}$  contains all points in the point cloud acquired by LiDAR.  $\{\mathbf{P}_q\}$  contains the points in the point cloud that the filtered reflection intensity is over a threshold value, roughly are points with the maximum reflection intensity in a certain area incident at  $0^\circ$ .  $\{\mathbf{P}_u\}$  containing the remaining points that do not meet the threshold.  $\{\mathbf{P}\} = \{\mathbf{P}_q\} + \{\mathbf{P}_u\}$ .  $\{\mathbf{P}_{Subcloud}\}$  is the set of point clouds sampled around  $p \in \{\mathbf{P}_q\}$ , contains  $p \in \{\mathbf{P}_u\}$ .
- 3) *Set of Point Clouds:*  $\{\mathbf{S}\}$  is the set of  $\{\mathbf{P}_{Subcloud}\}$ .
- 4) *Plane Defined by Laser Beam Direction:*  $G$  denotes the plane determined by the following equations:

$$\mathbf{X}' = \mathbf{n} \times \mathbf{Z} \quad (1)$$

$$\mathbf{Y}' = \mathbf{X}' \times \mathbf{n} \quad (2)$$

$$\mathbf{Z}' = -\mathbf{n} \quad (3)$$

- 5) *Reflection Intensity Threshold:* The reflection intensity produced by the glass has a lower limit. It is sufficiently distinguishable from most nonglass objects using this lower limit as a threshold *IntensityThreshold*.

- 6) *Filter Range*: In reality missing glass point clouds can occur only at a certain distance and have an impact on tasks such as SLAM and navigation. *DistanceThreshold* is set to reduce the number of point cloud that need to be processed.
- 7) *The Deletion Range of the Peak Point at Close Range*: In general, the reflection intensity of glass is generally greater than the reflection intensity of other objects in the environment, so points with smaller reflection intensity values in the *RangeThreshold* range near the points with larger reflection intensity values are removed.
- 8) *Divergence threshold DivThreshold*: Divergence of a vector field is used to describe the degree to which the vector field flux behaves like a source at a given point in space. The point with peak reflection intensity can be considered as the source of the reflection intensity for each part of the local point cloud. The divergence value at that point represents the rate of gradient decrease around that point. That is, the larger the divergence value, the larger the peak reflection intensity and the faster it decreases.

The pseudocode of the filtering algorithm for reflection intensity is presented below as Algorithm 1.

---

#### Algorithm 1 Reflective Intensity Filter

---

**Input:**  $\{\mathbf{P}\}$

```

1: for each point  $p \in \{\mathbf{P}\}$  do
2:    $Norm \leftarrow$  Distance from  $p$  to the LiDAR;
3:    $Intensity \leftarrow$  Reflection intensity of  $p$ ;
4:   if  $Norm < DistanceThreshold$  and  $Intensity >$ 
     $IntensityThreshold$  then
5:      $\{\mathbf{P}_q\} \leftarrow p$ ;
6:   end if
7: end for
8: Sorting  $\{\mathbf{P}_q\}$  by Intensity;
9: for each  $p \in \{\mathbf{P}_q\}$  do
10:   $Distance_i \leftarrow$  Distance between  $p$  and each previous
    point  $p_i \in \{\mathbf{P}_q\}$ ;
11:  if any  $Distance_i < RangeThreshold$  then
12:    Delete  $p \in \{\mathbf{P}_q\}$ ;
13:  end if
14: end for
15: for each  $p \in \{\mathbf{P}_q\}$  do
16:   Get  $\mathbf{T}$  and the parameters of  $G$ ;
17:   Get  $\{\mathbf{P}_{Subcloud}\}$ ;
18:   Get IntensityMatrix;
19:   Get GradientMatrix, DivMatrix;
20:    $Div \leftarrow$  The divergence value of the center position of
    DivMatrix;
21:    $\{\mathbf{S}\} \leftarrow \{\mathbf{P}_{Subcloud}\}$ ;
22:   if  $Div < DivThreshold$  then
23:     Delete  $p \in \{\mathbf{P}_q\}$ ,  $\{\mathbf{P}_{Subcloud}\} \subset \{\mathbf{S}\}$ ;
24:   end if
25: end for

```

---

Lines 1–7 of the code in Algorithm 1 are the initial processing of the LiDAR point cloud  $\{\mathbf{P}\}$  to obtain a collection of point clouds that may be glass. The distance between

$p \in \{\mathbf{P}_q\}$  and the origin of LiDAR coordinate system is  $Norm = (p_x^2 + p_y^2 + p_z^2)^{1/2}$ .

Lines 15–25 of the code are the reflection intensity filtering process. Line 16 obtains the transformation matrix  $\mathbf{T}$  and the parameters of the glass plane  $G$  in which the point  $p \in \{\mathbf{P}_q\}$  is located. The parameters  $A, B, C$  of the 3-D plane equation  $Ax + By + Cz + D = 0$  are  $p_x, p_y, p_z$  of the plane normal vector  $\mathbf{n} = [p_x, p_y, p_z]$ , respectively.  $D$  is the distance of point  $p \in \{\mathbf{P}_q\}$  from the LiDAR, which is  $Norm$ . Since the point  $p \in \{\mathbf{P}_q\}$  with the maximum reflected intensity of the glass is obtained by incidence along the normal of the glass plane, it can be considered that the line between the point  $p \in \{\mathbf{P}_q\}$  and the LiDAR is the normal vector  $\mathbf{n}$  of the glass plane  $G$ . The direction of  $\mathbf{n}$  points from the LiDAR to the point  $p \in \{\mathbf{P}_q\}$ . The coordinate systems  $\mathbf{X}', \mathbf{Y}', \mathbf{Z}'$  of the plane  $G$  are related to the LiDAR coordinate systems  $\mathbf{X}, \mathbf{Y}, \mathbf{Z}$  as in (1)–(3). Fig. 5 shows the schematic. The transformation matrix  $\mathbf{T}$  is obtained based on this relationship. The planes obtained by the above method correspond to those in which the actual glass is located in reality.

Let the base  $\mathbf{X}', \mathbf{Y}', \mathbf{Z}'$  of the plane  $G$  coordinate system be represented under the LiDAR coordinate system as  $\mathbf{X}' = [ex_1, ex_2, ex_3], \mathbf{Y}' = [ey_1, ey_2, ey_3], \mathbf{Z}' = [ez_1, ez_2, ez_3]$ . The  $p \in \{\mathbf{P}_q\}$  in the LiDAR coordinate system is  $p = [p_x, p_y, p_z]$ . The transformation matrix between the two coordinate systems is obtained

$$\mathbf{T} = \begin{bmatrix} ex_1 & ey_1 & ez_1 & p_x \\ ex_2 & ey_2 & ez_2 & p_y \\ ex_3 & ey_3 & ez_3 & p_z \\ 0 & 0 & 0 & 1 \end{bmatrix}. \quad (4)$$

All points  $p \in \{\mathbf{P}_u\}$  in the vicinity of each  $p \in \{\mathbf{P}_q\}$  are collected to form a point cloud  $\{\mathbf{P}_{Subcloud}\}$ . The range is a cylindrical range with the upper and lower column surfaces parallel to the plane  $G$ , as shown in Fig. 6(a). The geometric center of the cylinder is the point  $p \in \{\mathbf{P}_q\}$ , the radius of the column surface is  $R$ , and the height of the cylinder is  $H$ . Points  $p \in \{\mathbf{P}_{Subcloud}\}$  are projected on plane  $G$ , and reflection intensity features are extracted for filtering.

The discrete intensity features are sampled by a grid *IntensityMatrix*. *IntensityMatrix* centered at  $p \in \{\mathbf{P}_q\}$ , with rows and columns along  $x'$ -axis and  $y'$ -axis directions, respectively. Project  $p \in \{\mathbf{P}_{Subcloud}\}$  onto the plane  $G$ , and fill the reflection intensity value of each point into the grid by position, keeping the maximum intensity value when multiple reflections are located in one cell. After that, the *GradientMatrix* of the *IntensityMatrix* in  $x'$ -axis and  $y'$ -axis directions are calculated and the divergence of each cell is calculated to generate the *DivMatrix*. Determine whether the divergence at  $p \in \{\mathbf{P}_q\}$  meets the threshold *DivThreshold*, and if not, delete  $p \in \{\mathbf{P}_q\}$  and  $\{\mathbf{P}_{Subcloud}\} \subset \{\mathbf{S}\}$ .

#### B. Local Structure Feature

After the above reflection intensity filtering, there will still be nontransparent obstacles left behind. Some local structure features are extracted. Those features need to satisfy structural requirements for further filtering. The local structure

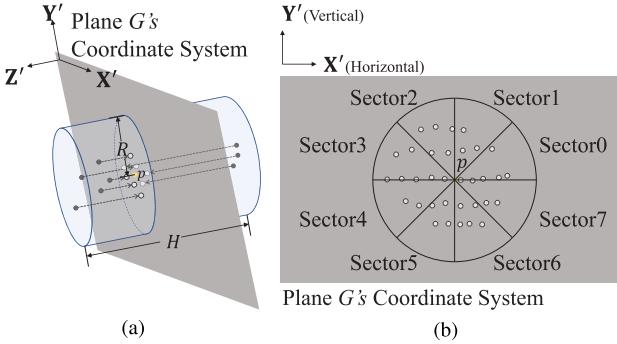


Fig. 6. Schematic of projecting a point cloud and extracting local structural features. (a) Points within the cylindrical sampling range form  $\{P_{\text{Subcloud}}\}$  and are projected onto the plane. (b) Number of points in the sector.

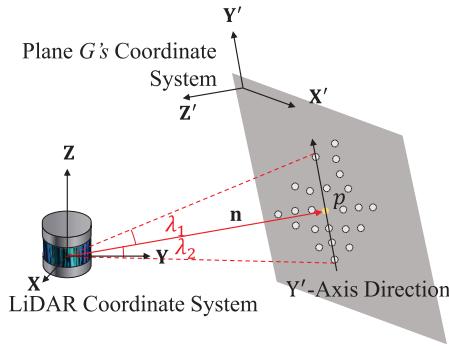


Fig. 7. Pitch angle feature in the algorithm.

features and the corresponding structural requirements are as follows.

- 1) *Number of Points in the Sector:* The overall shape of the points projected onto the plane  $G$  is observed as a circle centered at  $p \in \{P_q\}$  or an ellipse with the long axis in  $x'$ -axis direction. Fig. 6(b) shows the sectors divided at fixed angular intervals centered on  $p \in \{P_q\}$ . The left and right sides are divided by  $y'$ -axis direction. LiDAR samples densely in the horizontal direction and sparsely in the vertical direction. Sectors containing the maximum number of points should be in  $x'$ -axis direction.
- 2) *Dispersion:*  $p_j$  denotes the points  $p \in \{P_{\text{Subcloud}}\}$  projected onto the plane  $G$ . Dispersion feature is the variance of the distance between the  $p_j$  and  $p \in \{P_q\}$ . Variance should be within a certain range.
- 3) *Pitch Angle:* For a  $p \in \{P_q\}$ , find the two points of its  $\{P_{\text{Subcloud}}\}$  at the uppermost and lowermost positions in  $y'$ -axis direction of its glass plane  $G$ . The angle between the line of the LiDAR to the two points and the normal to the glass plane is calculated which can be understood as the pitch angle in the usual sense.  $\lambda_1$  denotes elevation angle and  $\lambda_2$  denotes depression angle as shown in Fig. 7. Take the absolute values of  $\lambda_1$  and  $\lambda_2$ . The feature is used to describe the shape of the point cloud.
- 4) *Point Cloud Scale:* The point cloud scale is the number of points within the  $\{P_{\text{Subcloud}}\}$ . Too small a scale is generally caused by noise.

The pseudocode of filtering algorithm for local structure is presented below as Algorithm 2.

#### Algorithm 2 Local Structure Filter

**Input:**  $\{P_q\}$   $\{P_{\text{Subcloud}}\}$

```

1: for each  $\{P_{\text{Subcloud}}\} \subset \{S\}$  do
2:   Extracting structural features;
3: end for
4: for each  $p \in \{P_q\}$  do
5:   if Any of the structural features do not meet the
      structural requirements then
6:     Delete  $p \in \{P_q\}$ ,  $\{P_{\text{Subcloud}}\} \subset \{S\}$ ;
7:   end if
8: end for

```

#### C. Reconstruction

After the reflection intensity feature filtering and local structure feature filtering, the obtained point cloud is the one generated by the glass reflection, and its plane is determined by (1)–(3).

1) *Flat Glass With Frame:* Most of the glass in indoor environment has fixed boundaries. As an example of finding the left boundary, let the elevation angle  $\lambda_1$  and the depression angle  $\lambda_2$  be the upper and lower standards. The reconstruction algorithm takes  $p \in \{P_q\}$  as the starting point, finds the continuous point cloud in the environment along the negative direction of  $x'$ -axis of plane  $G$  as the left boundary. The continuous point cloud satisfies the elevation angle greater than  $\lambda_1$  and the depression angle greater than  $\lambda_2$ . Then the algorithm finds the upper and lower boundaries along  $y'$ -axis from the left and right boundaries, respectively. At this step, two upper boundaries and two lower boundaries are obtained, and the common part of the boundaries is taken.

The laser beam within the boundary will be modified. The position of the laser beam hitting point contained within the boundary are modified to the plane  $G$ . In this process, the conversion of the plane equations on Cartesian and spherical coordinate systems is involved, as in the following equation:

$$\rho = -\frac{D}{A \cos(\phi) \cos(\theta) + B \cos(\phi) \sin(\theta) + C \sin(\phi)} \quad (5)$$

where  $A, B, C, D$  are from the plane equation  $Ax + By + Cz + D = 0$ . The origin of the spherical coordinate system is at the center of LiDAR as shown in Fig. 8.  $p_g = [p_{gx}, p_{gy}, p_{gz}]$  denotes a point on the plane  $G$ . The vector  $\mathbf{v}_{pg} = [p_{gx}, p_{gy}, p_{gz}]$  represents the vector pointing from the origin of the coordinate system to the point  $p_g$ .  $\rho$  is the length of the vector  $\mathbf{v}_{pg}$ .  $\theta$  is the angle between the projection of the vector  $\mathbf{v}_{pg}$  in the  $xoy$  plane and the positive direction of the  $x$ -axis.  $\phi$  is the complementary angle of the vector  $\mathbf{v}_{pg}$  to the positive direction of the  $y$ -axis.

2) *Curved Glass or Frameless Flat Glass:* If the boundaries are not found on both sides of the glass point cloud, the historical positional information is used to reconstruct the glass point cloud. Historical positional information means the transformation matrix  $\mathbf{T}$  between the glass point cloud and LiDAR

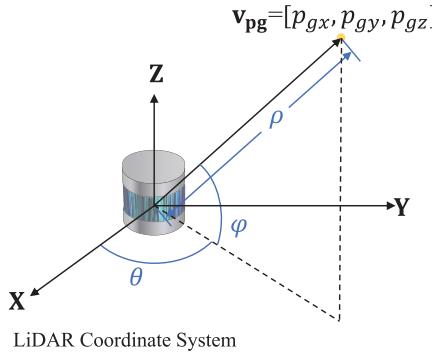


Fig. 8. Conversion of Cartesian and spherical coordinate systems.

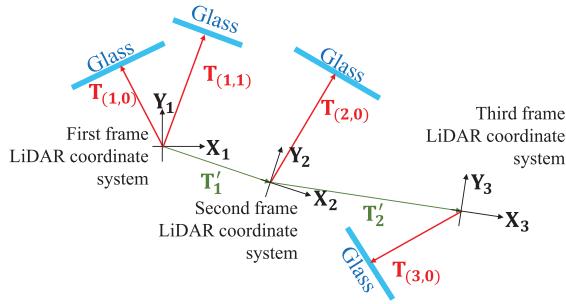


Fig. 9. Relationship between the coordinate systems of each frame.

at the current frame. LiDAR collects 10 frames per second of the point cloud of the entire environment. A frame contains the complete environmental point cloud data at the current moment in time.  $T$  is further written as  $T_{(i,j)}$ .  $i$  means the  $i$ -th frame in which the current glass point cloud is located.  $j$  means the order of the  $j$ -th frameless glass point cloud under the current frame.  $T'_i$  denotes the transformation matrix between the frame  $i$  and frame  $i - 1$  point clouds.

Using the initial robot pose as the world coordinate system, the schematic of the world coordinate system, the LiDAR coordinate system of each frame, the glass plane coordinate system and its transformation matrix are shown in Fig. 9. The viewpoint in the figure is oriented in the negative direction of  $z$ -axis of the world coordinate system.

The positional relationship between any two glass planes is obtained through the individual transformation matrix. Starting from the first laser point cloud frame, the following operations are performed for each newly obtained frameless glass point cloud thereafter.

The normal of the current glass plane  $G$  divides the left and right sides. Left side is the negative direction of  $x'$ -axis. The origin of the coordinate system is the point  $p_o$ . The quadtree is used to find the two points  $p_l$ ,  $p_r$ . Points  $p_l$  and  $p_r$  are the coordinate origins of the previously preserved frameless glass plane.  $p_l$  is the nearest point to  $p_o$  on the left side, and the distance between them is  $d_l$ . The same rule goes for the right side shown in Fig. 10.

In the left part of the point cloud reconstruction, the point  $p_o$  and the point  $p_l$  are set as the two ends of the reconstructed point cloud shape. Two planes intersect at the point  $p_m$  viewing toward the negative direction of  $z$ -axis of the world coordinate system. The second-order Bessel curve drawn with

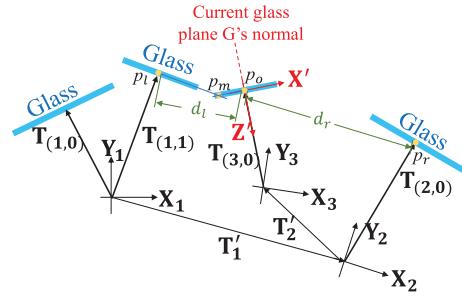


Fig. 10. Find the nearest points on both sides of the plane  $G$ .

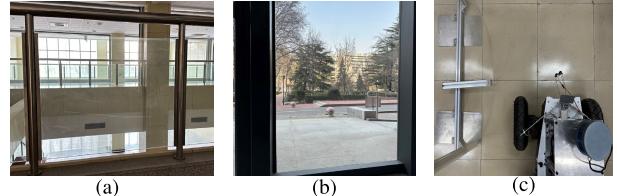


Fig. 11. Experimental environments. (a) Glass parapet. (b) Glass wall. (c) Curved surface.

$p_o$ ,  $p_l$  and  $p_m$  as the vertices is used as the shape of the glass point cloud, as in (6). The tree structure is used to store each completed curve shape. The nodes of the tree are the curve endpoints, the tree branches are represented by the second-order Bessel curve equation. The final point cloud of curved glass is reconstructed using the nodes and second-order Bessel curves stored in the above tree structure

$$p(t) = (1-t)^2 p_l + 2t(1-t)p_m + t^2 p_o, \quad t \in [0, 1]. \quad (6)$$

#### IV. EXPERIMENTS

The LiDAR used in the experiment is Robosense RS-LiDAR-16. It is a 16-line LiDAR with  $360^\circ$  horizontal angle measurement and  $-15^\circ$  to  $15^\circ$  vertical angle measurement, 320 000 points per second output. The ADLINK ROScube Pico TGL with Intel Core i5 CPU is used to process the LiDAR data. We use LiDAR to scan the glass in the indoor environment at a distance of about 1 m. In the curved frameless obstacle reconstruction, the transparent acrylic plate is selected as the obstacle. Three experimental environments are shown in Fig. 11.

##### A. Transparent Obstacle Recognition

1) *Recognition*: In the process of reflection intensity filtering, the reflection intensity distribution and central divergence value are shown in Fig. 12. The threshold values in Algorithm 1, in the experimental scenario, are given as: *IntensityThreshold* = 80 means that points with a reflection intensity greater than 80 will be added to the candidate point queue; *DistanceThreshold* = 2 refers to the LiDAR's two-meter range; *RangeThreshold* = 15 means that the top of the cone is located in LiDAR and the top angle is  $15^\circ$ ; *DivThreshold* = 70 means the divergence value threshold is 70. For the glass parapet, the original point cloud of LiDAR is shown in Fig. 13(a). The coordinate system appearing in the figure represents the position of the LiDAR, the red axis is the

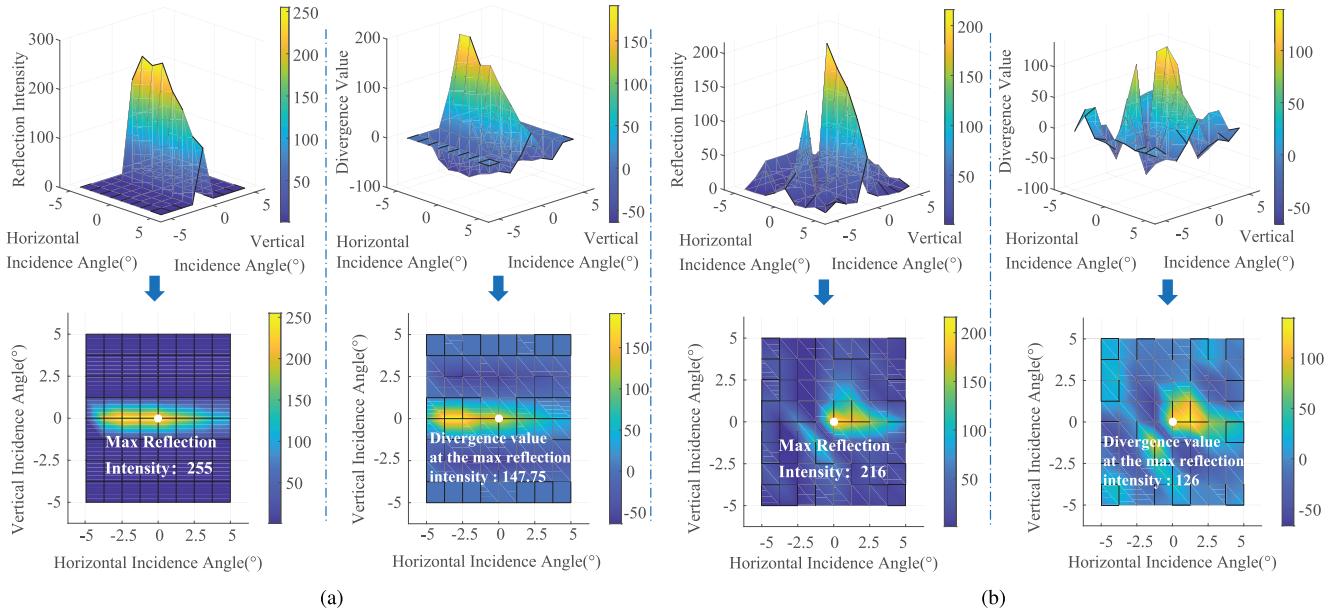


Fig. 12. Reflectance intensity distribution and divergence value. (a) Glass parapet. (b) Metal frame.

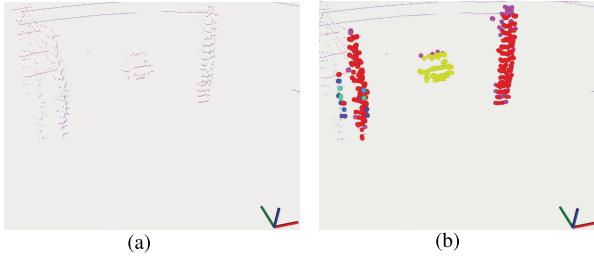


Fig. 13. Point cloud for the glass parapet. The points marked in red are the point clouds with reflection intensity greater than the threshold value incorporated into the suspected glass set, which is  $\{\mathbf{P}_{\text{Subcloud}}\}$ . The point cloud recognized as glass is marked in yellow. (a) Raw point cloud from LiDAR. (b) Point cloud of the glass recognized by our method.

orientation of the LiDAR. According to the rainbow color scale representing the reflection intensity from 0 to 255, the purple points are the points with a reflection intensity close to 0, and the red points are the points with a reflection intensity close to the upper limit of 255. Because the point cloud generated by glass reflection is too sparse, it is often considered noise and not considered in SLAM or path planning algorithms.

The reflection intensity distribution and central divergence value of the glass reflection point cloud are shown in Fig. 12(a). The features of the metal frame next to the glass are shown in Fig. 12(b). It can be seen that the maximum reflection intensity of both glass and metal frame is very high, generally exceeds the set reflection intensity threshold. A fairly typical reflection intensity characteristic of glass is shown in Fig. 12(a), while the characteristics of the metal frame are very similar to it. Both glass and metal frames are recognized as suspected glass point clouds and are marked, and then local structure features need to be extracted.

The sector and pitch angle features of the glass point cloud and the metal frame point cloud are shown in Fig. 14. The left and right sides are divided by  $y'$ -axis direction. When

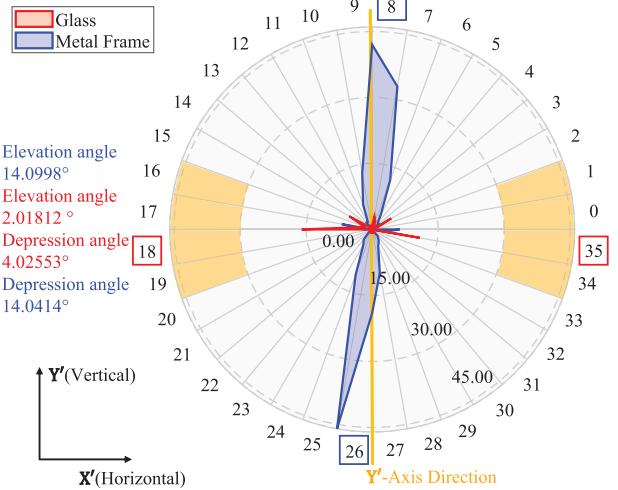


Fig. 14. Sector and pitch angle features.

the peak areas on the left and right sides appear in the shaded marked range, respectively, the features are considered to meet the structural requirements. The shaded ranges correspond to sectors 16, 17, 18, and 19 on the left and sectors 34, 35, 0, and 1 on the right. The sector features of glass and metal frames are distinctly different. The left and right peaks of glass are located in sectors 18 and 35, in the horizontal direction. The peaks of metal frame are located in sectors 26 and 8. It is not in the horizontal direction, so the metal frame does not conform to the local structure feature. It can also be seen that the pitch angle of the glass point cloud is smaller than that of the metal frame. The glass point cloud is recognized by our method. The point cloud after filtering by the algorithm is shown in Fig. 13(b).

2) *Comparative Experiment:* We chose the method proposed in [16] as the comparative method, considering the

publication time, technical relevance, and representativeness of each related works. In [16], its method is simpler to recognize the glass point cloud. For part of a suspected glass point cloud in a single frame, a complete and continuous boundary is needed to determine that the point cloud at that location belongs to glass. Then subsequent frames are needed to determine the plane on which the glass is located. It is necessary to use at least two frames of the point cloud to determine a plane of suspected glass. We propose evaluation indexes for a glass recognition method as frame recognition rate  $R_f$ , glass recognition rate  $R_g$ , and correction rate  $R_c$

$$R_f = \frac{F_s}{F_t} \quad (7)$$

$$R_g = \frac{N_{gs}}{N_{gt}} \quad (8)$$

$$R_c = \frac{N_c}{N_t}. \quad (9)$$

$F_s$  is the frame that recognizes the glass.  $F_t$  is the total number of frames with glass present.  $N_{gs}$  is the number of glasses recognized.  $N_{gt}$  is the total number of glasses.  $N_c$  is the number of point clouds correctly recognized as glass.  $N_t$  is the number of all point clouds recognized as glass.  $R_f$  and  $R_g$  reflect how much glass is missed by the method on a single frame and global level, respectively. Low  $R_f$  and  $R_g$  indicate that there is a lot of glass that is recognized as a common obstacle.  $R_c$  reflects the robustness of the method. Low  $R_c$  indicates that there are many nonglasses recognized as glass. The three indicators together evaluate a method.  $R_f$  and  $R_g$  should be increased while ensuring a lower limit on  $R_c$ .

The experimental datasets were collected from the experimental environment shown in Fig. 11(a), which was a circular corridor of the building. The LiDAR device is a Robosense RS-LiDAR-16. The experimental datasets include data from multiple experiments located at various distances from the glass and in different lighting conditions. In the datasets, the number of captured glasses  $N_{gt}$  is 86, where the number of frames containing glasses  $F_t$  is 1079. In the comparative experiment, the comparative method was not able to recognize frameless glass, which is illustrated in a separate row of the table. In the same glass parapet scene, the recognition results of our method and the comparative method are shown in Fig. 15. For the point cloud at the edge of the boundary, our method can successfully recognize the glass point cloud. However, the method in [16] assumes that the point cloud is not a glass point cloud. The method needs the glass part of the point cloud from the other frames to recognize the glass. The results of the two methods are shown in Table I. The first two rows are the results of two methods in the same scene in Fig. 11. The third row shows the results for the case where the glass boundary is incomplete.

3) *Ablation Study*: In the glass parapet scenario of Fig. 11(a), we make an ablation study to verify that reflection intensity feature and local structure feature are necessary in our method. The reflection intensity feature is removed and the recognition results are shown in Fig. 16(a). The point cloud representing the correct glass in Fig. 16(a) is located next to the LiDAR. The yellow point cloud that appears as a straight

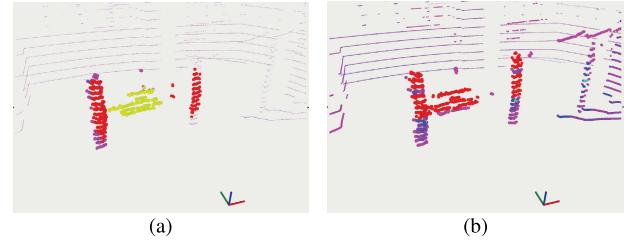


Fig. 15. Glass point cloud recognized by two methods in the same scene. (a) Glass recognized by the method in this article. (b) Glass point cloud recognized by the comparative method.

TABLE I  
COMPARISON OF EVALUATION INDEXES OF THE TWO METHODS

Method	$R_f$	$R_g$	$R_c$
Our method	82.95%	89.54%	83.87%
Comparative method(Complete Glass Boundary)	61.43%	81.25%	58.4%
Comparative method(Incomplete Glass Boundary)	0%	0%	0%

TABLE II  
EVALUATION INDEXES OF ABLATION STUDY

Conditions	$R_f$	$R_g$	$R_c$
No reflection intensity feature	81.28%	88.37%	35.06%
No local structure feature	80.82%	95.35%	13.06%

line in the foreground of Fig. 16(a) is a white wall. The yellow point cloud is far apart from the point clouds in the upper and lower rows. It conforms to the local structure feature. It is recognized as a glass point cloud. When the reflection intensity features are removed from our method, it cannot recognize the glass point cloud correctly.

When the local structure feature is removed, the recognition results are shown in Fig. 16(b). Metal frames on both sides with high reflective intensity peaks due to specular reflection. The distribution of reflected intensity is similar to that of glass, i.e., it is gradually reduced from a point of peak value to the surrounding diffusion. The method considers the point cloud at the metal frames to be a glass point cloud. The method without a local structure feature will incorrectly recognize glass point clouds.

A quantitative analysis of the ablation study is shown in Table II. Through the analysis of the results, only relying on reflection intensity feature or local structure feature can achieve a satisfactory recognition effect for glass. However, too many nontransparent obstacles are recognized as glass, which makes the overall effect worse, i.e.,  $R_c$  decreases. By introducing the recognition of one feature based on the results of the recognition of another feature, it is possible to increase  $R_c$ . It also reduces the  $R_f$  for glass. Compared to the negative impact on  $R_f$ , the combined application of the two features is more critical to the overall recognition improvement.

### B. Reconstruct Point Cloud

The glass parapets and glass curtain walls in the above experiments have fixed boundaries. The algorithm finds their

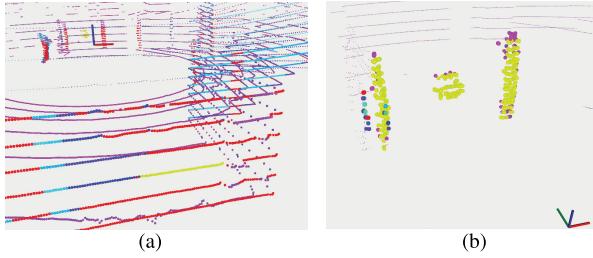


Fig. 16. Results of the ablation study. (a) No reflection intensity feature. (b) No local structure feature.

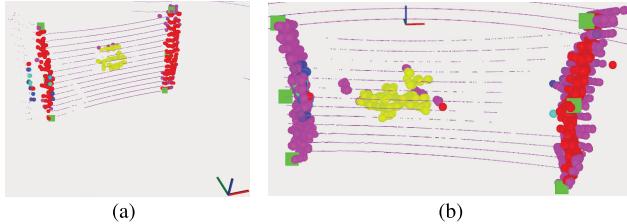


Fig. 17. Point cloud reconstruction results. The green marker in the figure is the boundary position found by the algorithm. These markers form a rectangular range. (a) Glass parapet. (b) Glass curtain wall.

TABLE III  
GLASS POINT CLOUD SIZE AND EVALUATION  
INDEXES AT DIFFERENT DISTANCES

Distance(m)	Glass point cloud size(mm)	$R_f$	$R_g$	$R_c$
0.5	27	78.78%	89.65%	85.71%
0.75	93	86.86%	91.34%	84.62%
1.25	150	78.69%	85.19%	81.81%
1.5	213	79.66%	79.31%	75.00%

boundaries and reconstructs the point cloud. The reconstruction for the glass parapet is shown in Fig. 17(a), and that for the glass curtain wall is in Fig. 17(b). The hit position of all laser beams that pass through within the range will be modified to the plane where the glass is located.

The recognition and reconstruction experiments are added at different distances to demonstrate that our method has no restriction on distance. The  $R_f$ ,  $R_g$ ,  $R_c$ , and glass point cloud sizes at different distances during the execution of our method are shown in Table III. The glass point cloud size is expressed as the distance of the point cloud in the y'-axis direction. The point cloud size is proportional to the distance. At different distances, our method can accurately recognize and reconstruct glass, as shown in Fig. 18.

In the curved framless obstacle reconstruction, the reconstructed results are shown in Fig. 19. The local structure features of the acrylic plate point cloud conform to the features presented in Section III-B. Local structure feature. Its reflection intensity feature divergence value is lower than that of the glass point cloud. The recognition and reconstruction results obtained for the actual glass scene will be better than those under the acrylic plate substitution. A sufficiently large sampling density can ensure that the reconstructed glass point cloud conforms to the actual shape.

The processing time for the point cloud is related to the point cloud scale. In the experimental scenario of this article,

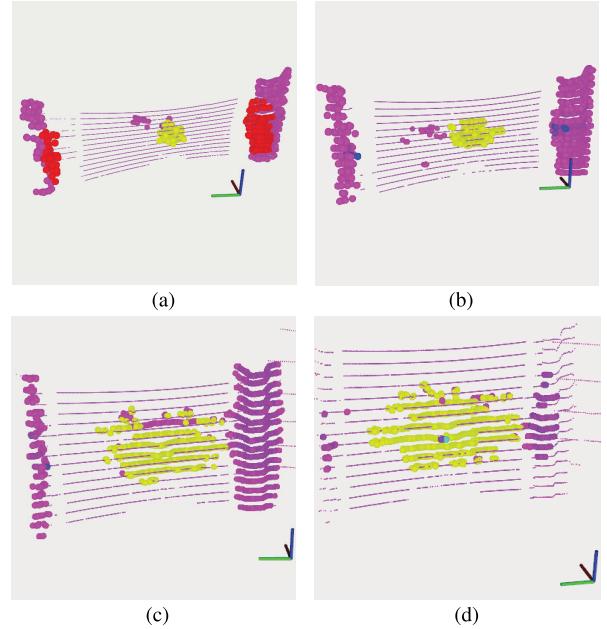


Fig. 18. Recognition and reconstruction results of LiDAR at different distances from glass. (a) 0.5 m. (b) 0.75 m. (c) 1.25 m. (d) 1.5 m.

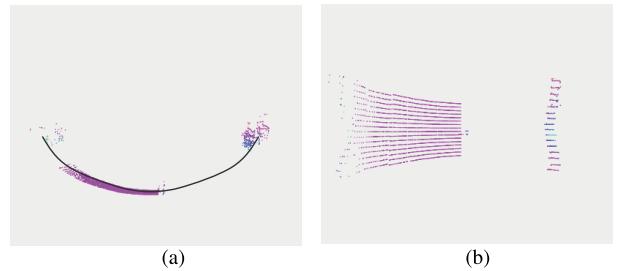


Fig. 19. Point cloud reconstruction result of the curved frameless obstacle. The black line represents the realistic shape of the surface. (a) Top view. (b) Front view.

the processing time for a single-frame point cloud is 12.7 ms per frame on average.

### C. 3-D SLAM

The complete point cloud output after reconstruction can be directly applied to 3-D SLAM algorithms for path planning and navigation. We run our method and perform the SLAM algorithm in the experimental environment shown in Fig. 20. Fig. 20(a) shows the experimental environment, which extensively uses glass parapets. Fig. 20(b) shows the planar map of the environment. The black dashed line represents the glass parapet in the environment. The trajectory of the LiDAR movement is shown by the long blue dashed arrow. The yellow arrow in the figure represents the viewpoint shown in Fig. 21.

The results of SLAM are shown in Fig. 21, with different colors representing different heights. The red line in the figure represents the metal frame used to hold the glass in place. LiDAR moves from the left side of the figure to the right of the metal frame marked by the red line. A clear distinction can be obtained from comparing the results in marked portions presented in Fig. 21(a) and (b). It can be found that in Fig. 21(a), less point cloud can be sampled from the glass

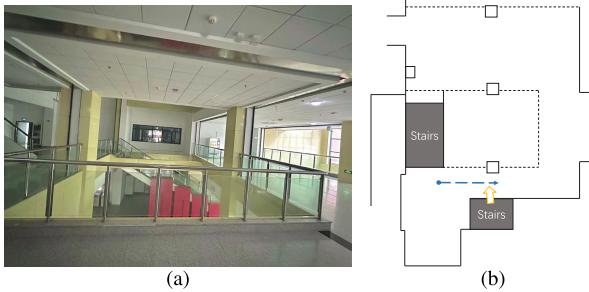


Fig. 20. Experimental environment of SLAM. (a) Photograph of the experimental environment. (b) Planar map.

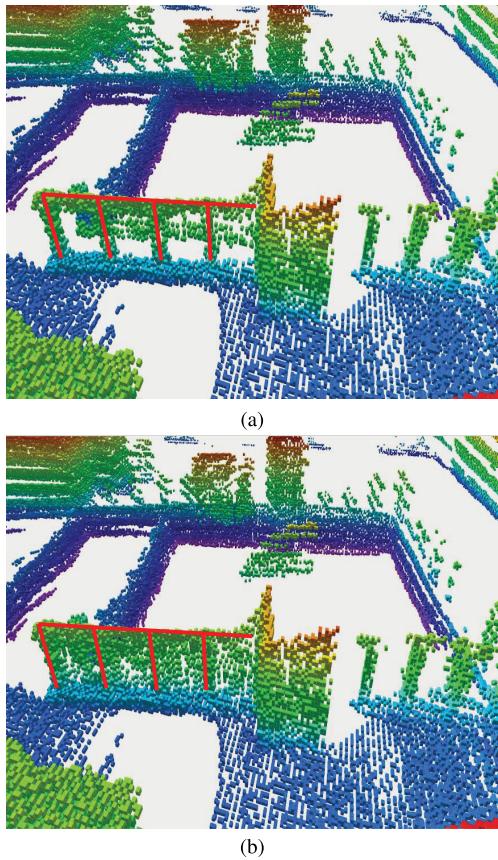


Fig. 21. SLAM effect demonstration, a significant difference in the glass point cloud in the portion marked by the red line. (a) Original SLAM results without the proposed method. (b) SLAM results after glass reconstruction.

at the height of the LiDAR. No point cloud is formed for other heights of glass. This is due to the fact that the LiDAR receives a small amount of specular reflection points from the glass at  $0^\circ$  of incidence. This is usually not enough to construct a complete map of the environment. Path planning on this incomplete map will also cause the mobile robot to collide with the glass. Our method filters the glass point cloud by reflection intensity feature and local structure feature. The reconstructed point cloud is integrated into the SLAM process. The output complete map can be used for mobile robots to perform various tasks safely. After the identification and reconstruction of the glass, the glass in the same position in Fig. 21(b) is completed.

## V. CONCLUSION

Modern buildings have widely used glass materials, whose optical properties make it difficult for LiDAR to detect effectively. Mobile robots face difficulties in accomplishing various tasks in indoor environments. The method proposed in this article relies only on a single sensor of LiDAR. It can be applied to the point clouds generated by various models of LiDAR. It is able to detect the glass and its corresponding planes by extracting filtering reflection intensity feature and local structural feature. The missing points are finally reconstructed at the corresponding positions of the output point cloud. The modified output point cloud has the correct environmental point cloud and can be applied to 3-D SLAM to build a complete map of the indoor glass environment.

## REFERENCES

- [1] S. Yuan et al., “DSmT-based ultrasonic detection model for estimating indoor environment contour,” *IEEE Trans. Instrum. Meas.*, vol. 69, no. 7, pp. 4002–4014, Jul. 2020.
- [2] J. Meng, S. Wang, G. Li, L. Jiang, Y. Xie, and C. Liu, “Accurate LiDAR-based localization in glass-walled environment,” in *Proc. IEEE/ASME Int. Conf. Adv. Intell. Mechatronics (AIM)*, Boston, MA, USA, Jul. 2020, pp. 534–539.
- [3] L. Xia, D. Meng, J. Zhang, D. Zhang, and Z. Hu, “Visual-inertial simultaneous localization and mapping: Dynamically fused point-line feature extraction and engineered robotic applications,” *IEEE Trans. Instrum. Meas.*, vol. 71, pp. 1–11, 2022.
- [4] T. Zhang, Z. J. Chong, B. Qin, J. G. M. Fu, S. Pendleton, and M. H. Ang, “Sensor fusion for localization, mapping and navigation in an indoor environment,” in *Proc. Int. Conf. Humanoid, Nanotechnol., Inf. Technol., Commun. Control, Environ. Manage. (HNICEM)*, Nov. 2014, pp. 1–6.
- [5] H. Wei, X. Li, Y. Shi, B. You, and Y. Xu, “Fusing sonars and LRF data to glass detection for robotics navigation,” in *Proc. IEEE Int. Conf. Robot. Biomimetics (ROBIO)*, Kuala Lumpur, Malaysia, Dec. 2018, pp. 826–831.
- [6] S.-W. Yang and C.-C. Wang, “Dealing with laser scanner failure: Mirrors and windows,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2008, pp. 3009–3015.
- [7] H. Wei, X.-E. Li, Y. Shi, B. You, and Y. Xu, “Multi-sensor fusion glass detection for robot navigation and mapping,” in *Proc. WRC Symp. Adv. Robot. Autom. (WRC SARA)*, Beijing, China, Aug. 2018, pp. 184–188.
- [8] M. Ye, Y. Zhang, R. Yang, and D. Manocha, “3D reconstruction in the presence of glasses by acoustic and stereo fusion,” in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2015, pp. 4885–4893.
- [9] E. Yamaguchi, H. Higuchi, A. Yamashita, and H. Asama, “Glass detection using polarization camera and LRF for SLAM in environment with glass,” in *Proc. 21st Int. Conf. Res. Educ. Mechatronics (REM)*, Cracow, Poland, Dec. 2020, pp. 1–6.
- [10] 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.
- [11] H. Sudo, S. Yukushige, S. Muramatsu, K. Inagaki, D. Chugo, and H. Hashimoto, “Detection of glass surface using reflection characteristic,” in *Proc. 47th Annu. Conf. IEEE Ind. Electron. Soc.*, Toronto, ON, Canada, Oct. 2021, pp. 1–6.
- [12] J.-S. Yun and J.-Y. Sim, “Virtual point removal for large-scale 3D point clouds with multiple glass planes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 43, no. 2, pp. 729–744, Feb. 2021.
- [13] X. Zhao, Z. Yang, and S. Schwertfeger, “Mapping with reflection-detection and utilization of reflection in 3D LiDAR scans,” in *Proc. IEEE Int. Symp. Saf., Secur., Rescue Robot. (SSRR)*, Nov. 2020, pp. 27–33.
- [14] X. Wang and J. Wang, “Detecting glass in simultaneous localisation and mapping,” *Robot. Auto. Syst.*, vol. 88, pp. 97–103, Feb. 2017.
- [15] P. Foster, Z. Sun, J. Jin Park, and B. Kuipers, “VisAGGE: Visible angle grid for glass environments,” in *Proc. IEEE Int. Conf. Robot. Autom.*, May 2013, pp. 2213–2220.
- [16] G. Cui, M. Chu, W. Wangjun, and S. Li, “Recognition of indoor glass by 3D LiDAR,” in *Proc. 5th CAA Int. Conf. Veh. Control Intell. (CVCI)*, Oct. 2021, pp. 1–4.

- [17] 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.
- [18] J. Kim and W. Chung, "Localization of a mobile robot using a laser range finder in a glass-walled environment," *IEEE Trans. Ind. Electron.*, vol. 63, no. 6, pp. 3616–3627, Jun. 2016.
- [19] 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)*, Taipei, Taiwan, Dec. 2017, pp. 405–410.
- [20] J. Lin, Z. He, and R. W. H. Lau, "Rich context aggregation with reflection prior for glass surface detection," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Nashville, TN, USA, Jun. 2021, pp. 13410–13419.
- [21] H. Mei et al., "Don't hit me! Glass detection in real-world scenes," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jun. 2020, pp. 3684–3693.



**Lelai Zhou** (Member, IEEE) received the B.Eng. degree in mechanical engineering and automation from the University of Science and Technology, Beijing, China, in 2006, the M.Eng. degree in mechanical design and theory from Beihang University, Beijing, in 2009, and the Ph.D. degree in mechanical engineering from Aalborg University, Aalborg, Denmark, in 2012.

He is currently a Professor with the School of Control Science and Engineering, Shandong University, Jinan, China. He has authored or coauthored over 70 journal and conference articles. His main researches are robotic arm, quadruped robot, human–robot interaction, and multi-robot collaboration.

Dr. Zhou was a recipient of the Shandong Provincial Natural Science Foundation for Distinguished Young Scholars in 2022.



**Xiaohui Sun** received the B.Eng. degree in automation from Yanshan University, Qinhuangdao, China, in 2021. He is currently pursuing the M.Eng. degree in mobile robots with the School of Control Science and Engineering, Shandong University, Jinan, China.

His research interests include LiDAR point cloud, path planning, and autonomous exploration in mobile robots.



**Chen Zhang** received the B.Eng. degree in automation and the Ph.D. degree in pattern recognition and intelligent systems from Shandong University, Jinan, China, in 2016 and 2023, respectively.

He is currently a Post-Doctoral Researcher with the School of Control Science and Engineering, Shandong University. His current research interests include motion and path planning, decision-making, and multi-agent collaborative control in robotics.



**Luyang Cao** received the bachelor's degree in electrical engineering and automation from the China University of Geosciences, Beijing, China, in 2022. He is currently pursuing the master's degree in mobile robotics with the School of Control Science and Engineering, Shandong University, Jinan, China.

His research interests include path planning and decision-making in robotics.



**Yibin Li** received the B.Eng. degree in automation from Tianjin University, Tianjin, China, in 1982, the M.Eng. degree in electrical automation from the Shandong University of Science and Technology, Qingdao, Shandong, China, in 1990, and the Ph.D. degree in automation from Tianjin University, Beijing, China, in 2008.

From 1982 to 2003, he was with the Shandong University of Science and Technology. Since 2003, he has been the Director of the Center for Robotics, Shandong University, Jinan, China. His research interests include robotics, intelligent control theories, and computer control systems.