https://doi.org/10.7840/kics.2025.50.6.984

GDR: A LiDAR-Based Glass Detection and Reconstruction for Robotic Perception

Akbar Taufiqurrahman◆, Soo-Young Shin°

ABSTRACT

This paper proposes GDR, a LiDAR-based glass detection and reconstruction for robotics perception. This method addresses the challenges LiDAR faces as an optical sensor, particularly its difficulty in achieving comprehensive perception of glass planes in its environment. Using a proposed glass detection method based on the unique characteristics of glass points, GDR separates them from non-glass points in the input point cloud. First, feature extraction is performed based on the characteristic distance values of glass points. Then, feature extraction continues on the intensity values. Based on these two feature extractions, glass points are identified and used by the glass reconstruction method to reconstruct the glass plane in the input point cloud, resulting in a corrected point cloud with a comprehensive perception of the glass planes in the environment. GDR is validated through several experiments, yielding an overall glass plane detection accuracy of 83.67% which outperforms the previous method. Additionally, the overall success rate and accuracy for glass reconstruction are 100% and 96.77%, respectively.

Key Words : Glass detection, glass reconstruction, robot perception, LiDAR, feature extraction.

Ⅰ. Introduction

To be able to perform its automation functions, a robot is required to independently collect data, process information, accurately identify conditions, and make and execute appropriate decisions[1]. One of the first key aspects, and one of the most important, is the ability to perform independent data collection.

Data collection by robots is carried out to obtain information like the state of the robot itself or the conditions of their surrounding environment. This information is then processed to produce decisions and actions that are accurate and fit the actual needs[2].

For example, in the application of robots as unmanned any vehicle (UXV), robots are required to understand their surrounding environment to achieve various automation functions, which typically include robot localization, environment mapping, and robot navigation[3,4]. In such cases, robots are often equipped with a range of sensors such as inertial measurement unit (IMU), camera, global positioning system (GPS), sonar, light detection and ranging (LiDAR), or radar[5,6]. Although these sensors serve similar purposes, each has its own advantages and limitations, making the choice of sensor application dependent.

For example, in terms of environmental conditions, the choice of sensors differs between indoor and out-

※ "This research was supported by the MSIT(Ministry of Science and ICT), Korea, under the ITRC(Information Technology Research Center) support program(IITP-2025-RS-2024-00437190, 50%) supervised by the IITP(Institute for Information & Communications Technology Planning & Evaluation)."

※ "This work was supported by Innovative Human Resource Development for Local Intellectualization program through the Institute of Information & Communications Technology Planning & Evaluation(IITP) grant funded by the Korea government(MSIT) (IITP-2025-RS-2020-II201612, 50%)."

◆ First Author: Kumoh National Institute of Technology, akbartaufiq.atr@gmail.com, 학생회원

° Corresponding Author: Kumoh National Institute of Technology ,wdragon@kumoh.ac.kr, 종신회원

논문번호 : 202411-290-D-RU, Received November 18, 2024; Revised December 18, 2024; Accepted January 9, 2025

door applications. In indoor environments, LiDAR and camera are often preferred over GPS because GPS signals are difficult to obtain due to the lack of satellite visibility. Conversely, in open outdoor environments, GPS is usually favored over LiDAR due to challenges such as the scarcity of geographic features in LiDAR data, which complicates localization, or the high intensity of sunlight that can cause significant noise in camera data. In some cases, specific environmental conditions pose unique challenges to robots, limiting their ability to perform autonomous functions effectively.

In indoor environments, experimental results have shown that in urban buildings with a high prevalence of glass planes, data from camera[7] and LiDAR can become inaccurate in representing the actual surroundings (Figure 1). This is problematic because LiDAR and camera are typically the primary sensors relied upon in such environments. The issue arises because LiDAR, as an optical sensor, relies on light waves for environmental perception[8], and camera capture light reflected from objects in the surroundings. Glass planes naturally refract light, leading to incorrect or incomplete perceptions by LiDAR and camera. Instead of detecting the glass plane itself, these sensors often perceive the objects behind it. This incomplete and incorrect perception of the glass plane, if not addressed, could lead to the robot incorrectly assuming that an area in the environment is free instead of occupied. Naturally, this condition could cause issues in the implementation of autonomous navigation in the robot, such as the possibility of collisions with glass obstacles.

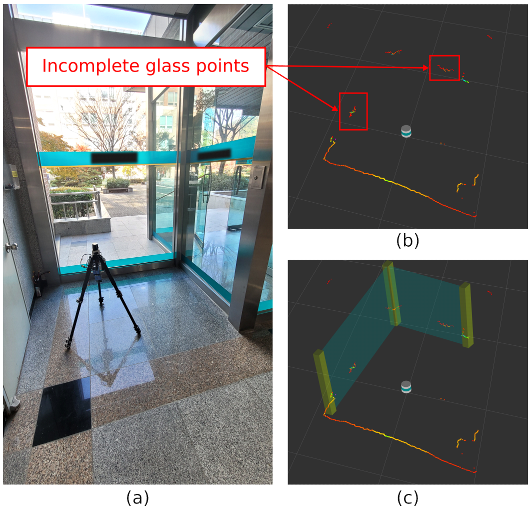


Fig. 1. Incomplete glass plane perception by LiDAR. (a) Real environment view. (b) LiDAR point cloud. (c) LiDAR point cloud with estimated glass plane and glass boundary view.

Furthermore, in localization and mapping systems, additional issues may arise, such as landmark recognition failure or loop closure detection failure, due to the high noise generated by glass objects as laser beams scatter unpredictably[9]. This could result in decreased reliability of the robot's localization results, which in turn could cause the autonomous navigation system on the robot to not perform as expected.

Therefore, a system is needed that can first detect the presence of glass planes in the environment and subsequently reconstruct the glass plane in the perception data, so that the resulting perception is complete, including portions of the glass plane that are inherently undetected by the robot sensor.

Based on that motivation, this paper proposes GDR, a LiDAR-based glass detection and reconstruction, aimed at improving LiDAR perception capabilities in glass-rich environments. GDR addresses the limitations of glass perception by LiDAR through the contributions of this paper as follows:

1. The introduction of two feature extraction method based on the distance and intensity information, for detecting glass planes based on LiDAR data.
2. A simple linear equation-based approach to efficiently reconstruct glass planes in the point cloud data.
3. GDR's LiDAR-based perception-level approach enables its use in various applications that rely on LiDAR point clouds, including localization, mapping, SLAM, and obstacle avoidance for robot navigation systems.

Ⅱ. Related Works

Some researchers have attempted to identify the presence of glass planes using the small number of LiDAR points generated by glass surfaces (glass points). There is a unique characteristic in LiDAR points from smooth objects such as marbles, polished woods, shiny metal planes, and glass planes (Figure 2), where at glass points that are close to the normal angle, a significant spike in intensity value occurs. The difference lies in the height of the spike, the intensity gradient of glass points, which tends to be higher, and the profile width, which is generally narrower than points on other smooth objects. Using this phenomenon, [10][11][12][13] proposed thresholding methods on these three characteristics and attempted to implement them on 2D LiDAR. Meanwhile, [14] using 3D LiDAR, also proposed a thresholding method on the intensity value of glass points, but only on points in areas where spikes occur. The intensity gradient is detected using a divergence method. However, this phenomenon only occurs when the LiDAR laser beam strikes the glass surface at or near the normal angle. The challenge arises because LiDAR mounted on robots are often imperfectly aligned, rarely remaining perpendicular to the glass plane in the environment. Additionally, uneven floors can prevent the laser beam from hitting the glass surface at an angle close enough to the normal angle, resulting in glass points failing to exceed the intensity threshold and thus not being classified as potential glass points. This issue persists even with 3D LiDAR, despite its higher number of channels and smaller gap between them, which theoretically increases the likelihood of the beams hitting glass near its normal angle. Experimental results show that even 64-channel 3D LiDAR sometimes fails to generate glass points with intensity values exceeding the threshold. Consequently, this method does not consistently detect glass surfaces in the environment. If glass detection fails, glass reconstruction that will include glass surfaces as LiDAR perception results cannot be achieved.

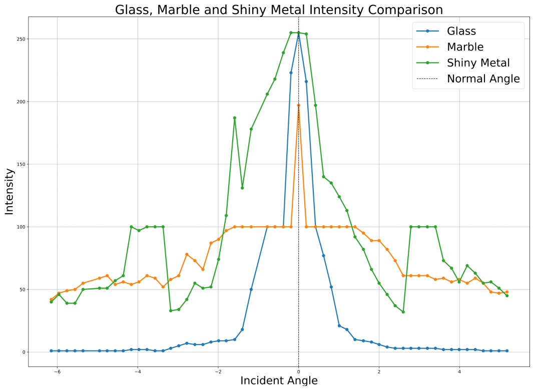


Fig. 2. Intensity value characteristics on glass and other reflective-material objects.

Furthermore, [15] proposed another method using the standard deviation of distance values from LiDAR points. This method assumes that points behind glass objects exhibit greater noise in their distance values compared to those not obscured by glass. By applying a threshold, points with a standard deviation exceeding the threshold are categorized as being behind glass. Glass reconstruction then begins and ends at the points where the standard deviation exceeds the threshold. However, this method is not entirely reliable, as glass detection depends on the presence of objects behind the glass, which is not always the case in real world environments. Additionally, errors in glass reconstruction may occur when objects are positioned in front of the glass, causing the reconstruction to begin or end at the object in front rather than at the true boundary of the glass. As a result, this method is also unable to correctly reconstruct the point cloud.

Other researchers have proposed combining RGB and depth information from RGB-D camera[16,17], sensor fusion between RGB-D camera and LiDAR[18], using polarization camera and LiDAR[19], or sonar/ultrasonic sensors with LiDAR[20,21]. However, methods that rely on RGB-D camera have significant limitations, primarily due to the minimal distance information between the robot and the glass plane. This limitation makes it difficult to estimate the position and profile of glass planes in the environment. Additionally, camera naturally depend heavily on lighting conditions, and there is no guarantee that lighting will always be sufficient to support glass detection via camera. In contrast, glass detection methods using ultrasonic sensors are unaffected by lighting conditions. However, these methods face the common drawback of a narrower field of view (FOV) compared to LiDAR. This limitation increases the difficulty of estimating the position and profile of glass planes and reduces the likelihood of successfully detecting them. Moreover, the use of multiple sensors adds computational complexity to the robot's system, slowing down autonomous processes and potentially impairing its perception capabilities.

To address these issues, this paper proposes a LiDAR-based glass detection method that is more robust to changes in lighting conditions, offers a wider FOV compared to other sensors, and provides more comprehensive depth information about the glass plane. This enables more accurate estimation of the glass plane's position and profile. Additionally, a new distance feature and intensity feature extraction approach is proposed to overcome the limitations of previous methods, which have not adequately considered the effects of uneven floor contours or the loss of LiDAR's vertical perpendicularity to the glass plane.

Ⅲ. Proposed Method

In this chapter, a detailed explanation will be provided on how GDR performs glass detection and glass reconstruction on LiDAR point cloud inputs, up to the generation of a corrected point cloud. As illustrated in Figure 3, the objective of GDR is to receive LiDAR point cloud input, process it, and produce are fined, corrected point cloud.

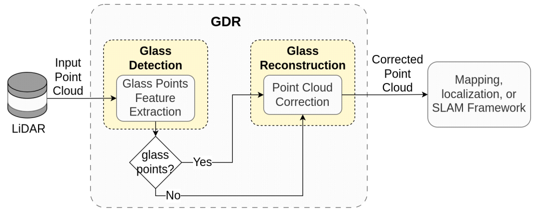


Fig. 3. GDR's system model.

3.1 Glass Detection

The process begins with the Glass Detection module, which comprises two feature extractors based on the distance and intensity information of the input point cloud. Based on those extracted features, glass points are identified, representing the presence of glass planes in the environment. The glass detection starts with distance feature extraction which essentially assesses the variance of a group of points that are considered clustered. In Figure 2, it is evident that only a few glass points are generated by the LiDAR. In this condition, since the glass points are primarily located near the normal angle, the variance of the distance values from these points tends to be lower compared to the points from many other objects with different materials. This occurs because the distance values of points on the other objects are typically more varied compared to those on a glass plane, where the points tend to have more similar distance values to each other.

The first step in this feature extractor involves grouping the LiDAR points from the input point cloud P_S, which contains a set of points p, into several clusters C, with some noise points N. The clustering process is based on the spatial positions of the points in 2D space. This system employs the DBSCAN clustering method for grouping points. DBSCAN is chosen because, unlike clustering methods such as K-Means, it does not require a parameter k, allowing the clustering results to yield n clusters that reflect the true structure of the data, especially for 2D LiDAR data.

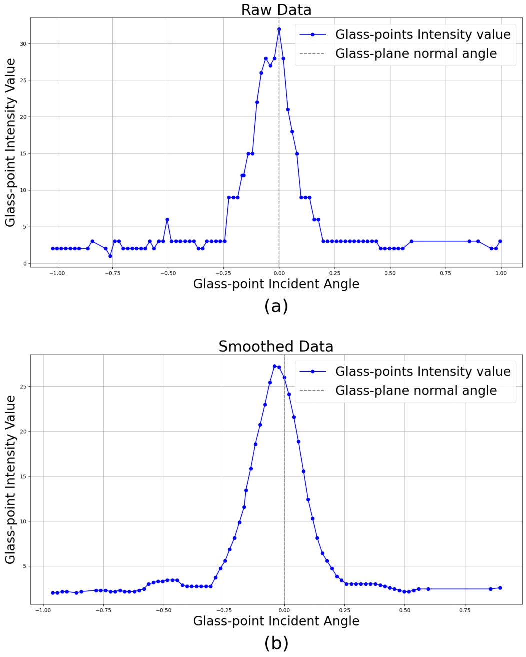
Next, for each cluster, the variance of the distance values for all points within it is calculated. If this value is less than the variance threshold σ^2_{tres}, the cluster is categorized as candidate glass points and added to P_{cg}. If the value exceeds σ^2_{tres}, the points in that cluster are categorized as non-glass points and will not be processed further in the subsequent feature extractor.

Next, in the intensity feature extractor, glass points are identified based on their intensity values. As shown in Figure 4, the intensity values on a glass plane exhibit two types: values within the intensity spike and values outside of it. The values within the intensity spike are further divided into two segments: those before the peak point, p_{peak}, which is the point with the highest intensity, and those after the peak point. As seen in Figure 4a, the values before the peak tend to follow an upward trend, with a nearly always positive gradient. Conversely, values after the peak generally show a downward trend, with a nearly always negative gradient. Additionally, intensity values outside the spike are almost always more numerous than those within the spike. Based on these characteristics, the intensity feature extractor will refine the candidate glass points produced by the distance feature extractor, denoted as P_{cg}.

The first step is to smooth the intensity values of points within each cluster in P_{cg} using a moving average algorithm (as shown in Figure 4b). This smoothing stabilizes the intensity values without excessively altering them, making the gradient analysis of the intensity spike easier to perform.

Following this, the minimum intensity value i_{min} and the maximum intensity value i_{max} (intensity range) are identified. Based on this range, a set of k bins are created to map points according to their intensity values (as shown in Figure 5a). Experimental results indicate that using $k = 5$ bins effectively separate the points within a cluster into two groups: those within the intensity spike and those outside it.

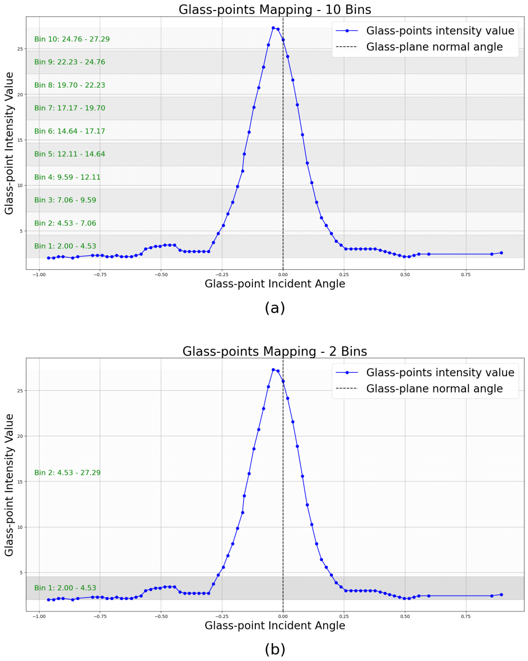
Once the 5 bins are created and the points in each clusters in p_{cg} are mapped into them, the bin containing the lowest values within the cluster is retained, while the remaining 4 bins are merged into one. This results in only two bins, $B1$ and $B2$, effectively separating the points within the intensity spike from those outside of it (as shown in Figure 5b).



(a)

(b)

Fig. 4. Glass-points-intensity smoothing. (a) Raw data, before smoothing. (b) Smoothed data, after smoothing.



(a)

(b)

Fig. 5. Glass points mapping. (a) Points mapped into 10 bins. (b) Points mapped into 2 bins, where the first bin unchanged, and the other bins are merged into one.

Based on these two bins, the number of points in $B1$ is compared to the number of points in $B2$. If the number of points in $B1$ is greater than that in $B2$, the cluster will proceed to further processing. Conversely, if the number of points in $B1$ is less than that in $B2$, the points in the cluster will be classified as non-glass points and will not undergo further processing.

Next, the intensity values of points in $B2$ are checked for an increasing trend, with a consistently positive gradient before reaching the peak, followed by a decreasing trend with a consistently negative gradient after the peak. If this condition is met, the points in the cluster are categorized as glass points and added to P_g. If not, they are classified as non-glass points and will not undergo further processing. Thus, the glass points are gathered in P_g. $P_g$is then passed to the glass reconstruction module for the glass plane reconstruction process.

3.2 Glass Reconstruction

After obtaining the clusters of glass points in P_g from the glass detection module, the glass reconstruction module reconstructs the glass plane based on these clusters of glass points. This process begins by selecting the point with the highest intensity value, p_{peak}, in each cluster. Each point in the 2D LiDAR data inherently contains information on distance/range and azimuth angle, which represents a vector from the LiDAR origin point to the given point. Assuming that p_{peak} in each glass point cluster represents the most accurate measurement, glass plane reconstruction is based on p_{peak}.

For each cluster of glass points, a vector $\overrightarrow{v_{peak}}$ is defined based on p_{peak}. Then, a perpendicular vector $\overrightarrow{v_{perp}}$ is defined at p_{peak}, positioned exactly at this point. Thus, p_{peak} becomes the intersection point between $\overrightarrow{v_{peak}}$ and $\overrightarrow{v_{perp}}$.

The goal of the Glass Reconstruction module is to correct points that should lie on the glass plane but are instead positioned behind it or have infinite values (inf/ 0) due to the laser beam not returning to the LiDAR. To determine whether a LiDAR point belongs on the glass plane, a check is conducted starting from the outermost point of the glass point cluster and extending to neighboring points. This evaluation is based on the Cartesian coordinate system, although 2D LiDAR points are originally defined within a spherical coordinate system.

As illustrated in Figure 6, the point p_{check}, shown in blue, is the point under inspection, and the vector $\overrightarrow{v_{perp}}$ serves as an estimated profile of the glass plane. So next, based on the vector $\overrightarrow{v_{check}}$, which is defined based on p_{check}, find the intersection point p_{int} between $\overrightarrow{v_{check}}$ and $\overrightarrow{v_{perp}}$. This intersection point p_{int} is assumed to be the position where p_{check} should ideally be located. In other words, p_{int} is the point on the LiDAR laser beam defined as $\overrightarrow{v_{check}}$, representing the correct measurement from the laser beam $\overrightarrow{v_{check}}$ and lying on the actual profile of the glass plane.

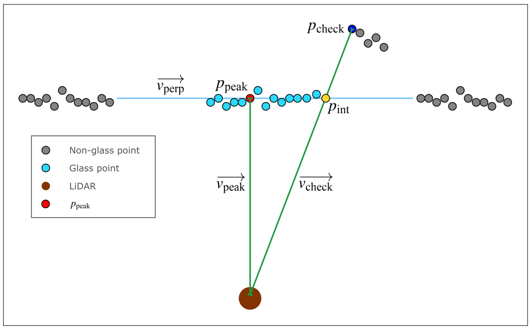

p_{check}
$\overrightarrow{v_{perp}}$
p_{peak}
p_{int}
Non-glass point
Glass point
LiDAR
p_{peak}
$\overrightarrow{v_{peak}}$
$\overrightarrow{v_{check}}$

Fig. 6. Glass Reconstruction: point correction mechanism.

Once p_{int} is identified, the distance from the origin to p_{check}, denoted as d_{check}, is compared with the distance from the origin to p_{int}, denoted as d_{int}. If d_{check} is greater than d_{int}, then p_{check} will be replaced with p_{int}. If d_{check} is less than d_{int}, it will be checked whether p_{check} belongs to another cluster from the DBSCAN clustering results, C. If so, p_{check} will be considered as a glass boundary, and the checking will be stopped. If p_{check} belongs to N, which represents noise in the DBSCAN clustering results, then p_{check} will be replaced with p_{int}, and the process will continue until the glass boundary is found.

Ⅳ. Experiments and Results

4.1 Experiment Details

To implement and further test the proposed method, a 2D LiDAR, the Oradar MS500 (Figure 7), which includes intensity information, is mounted on a tripod to capture point cloud data of a campus building with numerous glass walls and glass doors. An Intel NUC 13 Pro minicomputer is used for processing, featuring an Intel Core i5 1360P processor, 64 GB of RAM, and 2 TB of SSD storage.

Fig. 7. 2D LiDAR: Oradar MS500.

To evaluate the capability of the glass detection module in identifying glass planes, several experimental scenarios were conducted and compared with the previous method[11]. First, the LiDAR was moved parallel to the glass plane, creating a lateral movement, while detecting the presence of the glass plane in real time. This scenario was designed to highlight the potential for glass detection failures due to the robot moving over uneven floor contours, which may cause the LiDAR's position to deviate from being perpendicular to the normal angle of the glass plane, as illustrated in Figure 8a.

Second, the LiDAR will be rotated multiple times in the same position, creating a rotational movement. This is intended to introduce the possibility of glass detection failure due to imperfections in the LiDAR installation on the robot or specific conditions affecting the robot, which may cause the LiDAR to deviate from a 0-degree vertical angle in the robot' s body frame, as illustrated in Figure 8b.

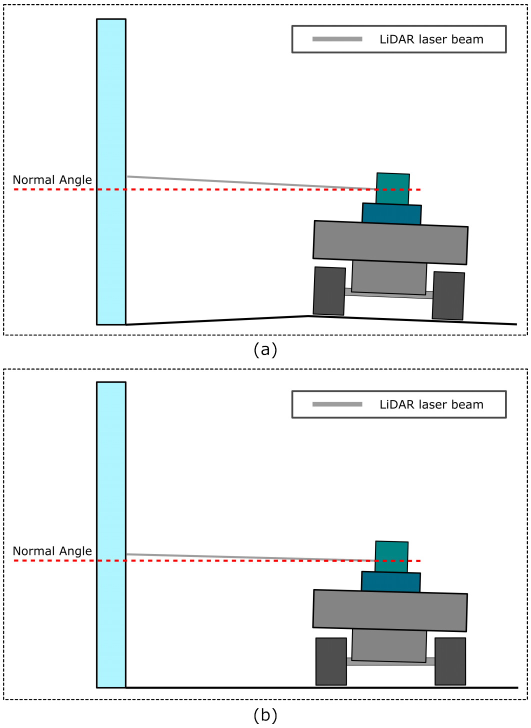


Fig. 8. LiDAR misalignment against glass planes. (a) The effect of uneven floor contour on laser beams position relative to the glass-plane normal angle. (b) The effect of the robot body's tilt on the laser beams position relative to the glass-plane normal angle.

Third, the LiDAR will be positioned at varying distances from the glass plane, forming a longitudinal movement. This experiment aims to assess the tested method's ability to handle changes in the distance between the LiDAR and the glass plane.

Next, to test the capability of the Glass Reconstruction module in reconstructing a glass plane, several trials will be conducted on different types and conditions of glass surfaces. First, the LiDAR will be positioned close to a wide glass wall, and glass detection will be performed in real time. Second, the LiDAR will be positioned close to a glass door. Following this, glass plane reconstruction will be carried out based on the detected glass points. Based on that, the accuracy of the glass plane reconstruction results will be checked against the actual glass plane. Then, the success rate of the reconstruction process on the obtained glass points will be calculated.

4.2 Results

Based on the experiments conducted, as shown in Table 1, GDR achieved a fairly high accuracy in glass detection under various LiDAR positioning conditions, often detecting glass points with intensity values that do not exceed the fixed thresholds set by previous method [11]: 78.46% accuracy in lateral movement, 83.48% in rotational movement, and 89.07% in longitudinal movement. Since GDR does not directly apply a threshold on intensity values when detecting glass points, it is able to maintain good accuracy across these varied conditions.

Next, regarding the Glass Reconstruction module,

Table 1. Comparison of Glass Detection Accuracy.

Movement	Accuracy	
	Previous [11] (%)	GDR (%)
Lateral	51.53	**78.46**
Rotational	46.93	**83.48**
Longitudinal	76.05	**89.07**

Table 2. Results of GDR's Glass Reconstruction Test.

	Glass Wall (%)	Glass Door (%)
Success rate	100	100
Accuracy	95.65	97.88

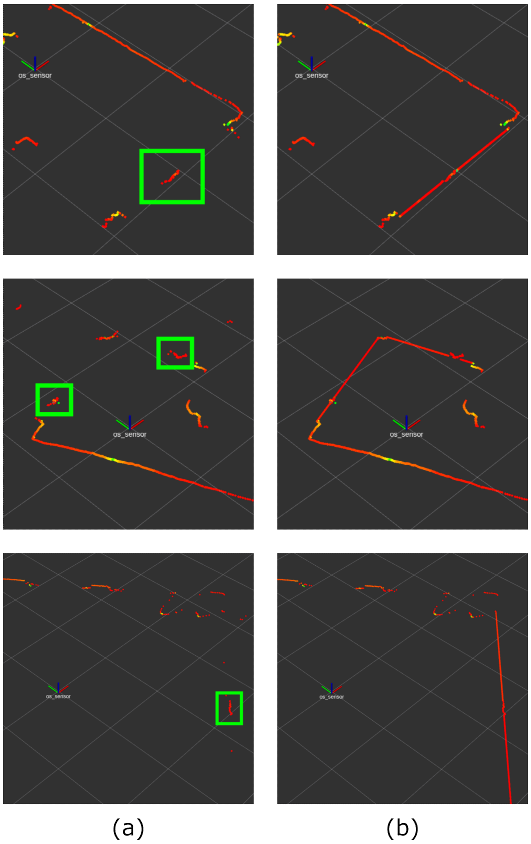


Fig. 9. Glass Reconstruction result visualized on RViz. Some points marked with green box are the detected glass points. (a) Input point cloud. (b) Corrected point cloud.

as shown in Figure 9, the input point cloud (Figure 9a) was corrected successfully by the module (Figure 9b), and the corrected point cloud matches the actual condition of the glass plane in the environment. In Table 2, it can be seen that for both the glass wall and glass door, the reconstruction process achieved a high success rate, with a small margin of error related to the estimation inaccuracies of the glass plane profile.

Ⅴ. Conclusion and Future Work

This paper proposed GDR, a LiDAR-based glass detection and reconstruction for robotics perception. GDR enhances LiDAR perception in areas with numerous glass planes, generating a corrected point cloud applicable to various robotics applications, such as robot localization, environment mapping, SLAM, and navigation. Experimental results demonstrate that GDR effectively tackles challenges related to LiDAR misalignment with glass planes, particularly in environments with uneven floor contours, suboptimal LiDAR positioning, or specific robot conditions. These factors have previously reduced detection accuracy in methods that relied on direct thresholding of glass point intensity values. GDR's Glass Detection tests showed an overall accuracy of 83.67%, which outperforms the previous method[11], while the Glass Reconstruction module achieved accuracy and success rates of 100% and 96.77%, respectively.

In the future, the proposed Glass Detection method will be further developed for full implementation in robotics applications using 3D LiDAR. This implementation will focus on robots requiring 3D perception, such as quadcopters or robotics manipulators, and will aim to optimize the reconstruction method for greater computation efficiency.

References

[1] M. Chaudhary, N. Goyal, A. Benslimane, L. K. Awasthi, A. Alwadain, and A. Singh, "Underwater wireless sensor networks: Enabling technologies for node deployment and data collection challenges," *IEEE Internet of Things J.*, p. 1, 2022. (https://doi.org/10.1109/jiot.2022.3218766)

[2] Z. Wei, et al., "UAV assisted data collection for internet of things: A survey," *IEEE Internet of Things J.*, p. 1, 2022. (https://doi.org/10.1109/jiot.2022.3176903)

[3] G. Bresson, Z. Alsayed, L. Yu, and S. Glaser, "Simultaneous localization and mapping: A survey of current trends in autonomous driving," *IEEE Trans. Intell. Veh.*, vol. 2, no. 3, pp. 194-220, Sep. 2017. (https://doi.org/10.1109/tiv.2017.2749181)

[4] A. R. Vidal, H. Rebecq, T. Horstschaefer, and D. Scaramuzza, "Ultimate SLAM? Combining events, images, and IMU for robust visual SLAM in HDR and high-speed scenarios," *IEEE Robotics and Automat. Lett.*, vol. 3, no. 2, pp. 994-1001, Apr. 2018.

(https://doi.org/10.1109/lra.2018.2793357)

[5] M. B. Alatise and G. P. Hancke, "A review on challenges of autonomous mobile robot and sensor fusion methods," *IEEE Access*, vol. 8, pp. 39830-39846, 2020. (https://doi.org/10.1109/access.2020.2975643)

[6] M. A. K. Niloy, et al., "Critical design and control issues of indoor autonomous mobile robots: A review," *IEEE Access*, vol. 9, pp. 35338-35370, 2021. (https://doi.org/10.1109/access.2021.3062557)

[7] H. Mei, et al., "Don't hit me! Glass detection in real-world scenes," *2020 IEEE/CVF Conf. CVPR*, Jun. 2020. (https://doi.org/10.1109/cvpr42600.2020.00374)

[8] Q. Mo, Y. Zhou, X. Zhao, X. Quan, and Y. Chen, "A survey on recent reflective detection methods in simultaneous localization and mapping for robot applications," in *Proc. ISAS*, Jun. 2023. (https://doi.org/10.1109/isas59543.2023.10164614)

[9] K. Liu and M. Cao, "DLC-SLAM: A robust LiDAR-SLAM system with learning-based denoising and loop closure," *IEEE/ASME Trans. Mechatronics*, pp. 1-9, 2023. (https://doi.org/10.1109/tmech.2023.3253715)

[10] X. Wang and J. Wang, "Detecting glass in simultaneous localisation and mapping," *Robotics and Autonomous Syst.*, vol. 88, pp. 97-103, Feb. 2017. (https://doi.org/10.1016/j.robot.2016.11.003)

[11] L. Weerakoon, G. S. Herr, J. Blunt, M. Yu, and N. Chopra, "Cartographer_glass: 2D graph SLAM framework using LiDAR for glass environments," *arXiv preprint arXiv:2212.08633,* Dec. 2022. (https://doi.org/10.48550/arxiv.2212.08633)

[12] A. Mora, A. Prados, P. Gonzalez, L. Moreno, and R. Barber, "Intensity-based identification of reflective surfaces for occupancy grid map modification," *IEEE Access*, vol. 11, pp. 23517-23530, Jan. 2023. (https://doi.org/10.1109/access.2023.3252909)

[13] A. Taufiqurrahman and S. Shin, "Lidar-based glass reconstruction for urban robotics application," in *Proc. 2024 KICS Summer Conf.*, pp. 890-891, Jun. 2024, Available: https://www.dbpia.co.kr/Journal/articleDetail?nodeId=NODE11906231

[14] L. Zhou, X. Sun, C. Zhang, L. Cao, and Y. Li, "LiDAR-based 3D glass detection and reconstruction in indoor environment," *IEEE Trans. Instrumentation and Measurement*, p. 1, Jan. 2024. (https://doi.org/10.1109/tim.2024.3375965)

[15] H. Tibebu, J. Roche, V. D. Silva, and A. Kondoz, "LiDAR-based glass detection for improved occupancy grid mapping," *Sensors*, vol. 21, no. 7, p. 2263, Mar. 2021. (https://doi.org/10.3390/ s21072263)

[16] Y. Tao, H. Gao, Y. Wen, L. Duan, and J. Lan, "Glass recognition and map optimization method for mobile robot based on boundary guidance," *Chinese J. Mechanical Eng.*, vol. 36, no. 1, Jun. 2023. (https://doi.org/10.1186/s10033-023-00902-9)

[17] Y. Zhao, H. Li, S. Jiang, H. Li, Z. Zhang, and H. Zhu, "Glass detection in simultaneous localization and mapping of mobile robot based on RGB-D camera," in *Proc. ICMA*, Aug. 2023. (https://doi.org/10.1109/icma57826.2023.10216171)

[18] J. Lin, Y. H. Yeung, and R. W. H. Lau, "Depth-aware glass surface detection with cross-modal context mining," *arXiv preprint arXiv:2206.11250,* 2022. (https://arxiv.org/abs/2206.11250)

[19] E. Yamaguchi, H. Higuchi, A. Yamashita, and H. Asama, "Glass detection using polarization camera and LRF for SLAM in environment with glass," in *Proc. Int. Conf. REM*, Dec. 2020. (https://doi.org/10.1109/rem49740.2020.9313933)

[20] 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 *2014 Int. Conf.*

HNICEM, Nov. 2014. (https://doi.org/10.1109/hnicem.2014. 7016188)

[21] 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. ROBIO*, Dec. 2018. (https://doi.org/10.1109/robio.2018.8664805)

Akbar Taufiqurrahman

Mar. 2022 : Telkom University, computer science, B. Sc.

Feb. 2024 : Kumoh National Institute of Technology, IT Convergence Engineering, M. Eng.

<관심분야> Robotics, UAV, autonomous mobile robot, machine learning, deep learning.

[ORCID:0009-0004-7623-5778]

Soo-Young Shin

1999년 2월 : 서울대학교 전기공학부 졸업

2001년 2월 : 서울대학교 전기공학부 석사

2006년 2월 : 서울대학교 전기공학부 박사

2010년~현재 : 국립금오공과 대학교 전자공학부 교수

<관심분야> 미래 무선 통신, 신호 처리, 무인 이동성, 양자 신경망, 사물인터넷, 인공지능/딥러닝 응용 프로그램 등

[ORCID:0000-0002-2526-2395]