



Article

Experimental Analysis of the Behavior of Mirror-like Objects in LiDAR-Based Robot Navigation

Deeptha Damodaran ¹, Saeed Mozaffari ², Shahpour Alirezaee ² and Mohammed Jalal Ahamed ^{1,*}

¹ Department of Mechanical, Automotive and Materials Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada

² Department Electrical and Computer Engineering, University of Windsor, Windsor, ON N9B 3P4, Canada

* Correspondence: m.ahamed@uwindsor.ca

Abstract: Mobile robots are equipped with various sensors to perform object detection, localization, and navigation. Among these sensors, LiDAR (light detection and ranging) is the most widely used sensor for environment map creation. However, LiDAR-based localization is challenging in modern environments containing specular surfaces, such as mirrors and glasses, that cause light reflection, penetration, or diffusion. These conditions make the obtained map inaccurate, unreliable, and noisy. This paper presents the effects of mirror-like objects in various indoor arrangements on 2D LiDAR-based maps. Experiments were conducted using a mobile robot equipped with LiDAR navigating in an environment with several mirrors. Experiments suggest that laser scans may be fully reflected off mirrors, causing no range or intensity data and creating a faulty map. Objects or boundaries within the range of LiDAR may be mapped behind the surface of the mirror, and robot self-detection may occur on the surface of the mirror. This situation exacerbates when more than one mirror is present in the environment. The results presented in this paper can aid the development of LiDAR-based indoor navigation to identify and remove inconsistencies created in LiDAR maps due to mirror-like objects.

Keywords: mirror detection; LiDAR data analysis; robot localization; SLAM



Citation: Damodaran, D.; Mozaffari, S.; Alirezaee, S.; Ahamed, M.J. Experimental Analysis of the Behavior of Mirror-like Objects in LiDAR-Based Robot Navigation. *Appl. Sci.* **2023**, *13*, 2908. <https://doi.org/10.3390/app13052908>

Academic Editors: Yutaka Ishibashi and Marco Troncosi

Received: 8 December 2022

Revised: 9 February 2023

Accepted: 19 February 2023

Published: 24 February 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

Mirror-like objects are prevalent in modern buildings such as museums, offices, lobbies, and hospitals. Such reflective objects pose serious challenges to indoor autonomous navigation systems using simultaneous localization and mapping (SLAM) algorithms, aiming to represent the spatial environment (mapping) while keeping track of position (localization) within the built map. Since mapping and localization are highly correlated in SLAM, inaccurate maps caused by reflective objects adversely affect localization accuracy. SLAM frameworks use sensor technology for data acquisition, including acoustic, visual, and ranging sensors. To create a representation of the environment, occupancy grid maps provide a discretized representation of an environment where each of the grid cells is classified into two categories: occupied or free. Considering each cell as a binary variable, the map is able to estimate the location of an obstacle in the space by computing a posterior approximation for any given cell within the range of the sensor that is collecting data [1].

Among various sensors, LiDAR sensors have received much attention in recent years due to their cost, high accuracy, long scanning range, and high stability [2]. LiDAR uses laser light to measure distance based on the reflective properties of the environment. The light is usually infrared, but can also be in the visible or ultraviolet range of the spectrum. LiDARs are advantageous in industry, as the collected data can be converted into 2D or 3D point clouds and easily integrated with other sensor data. Multiple consecutive LiDAR readings are used for complex applications such as obstacle detection, localization and mapping [3]. In addition, the width of the emitted beam can be made smaller to increase the distance it can travel, allowing LiDARs to detect obstacles over long ranges.

However, there are some limitations associated with LiDARs. External sources that change visual properties affect LiDAR data. In outdoor applications, adverse weather, such as low-hanging clouds and heavy rain or fog, can have a negative impact on LiDAR data collection. One study was performed on ten different LiDARs in different weather conditions, including simulated fog, rain and intense sunlight [4]. LiDAR also has difficulty in some indoor applications. In indoor environments, the presence of walls, furniture, and other obstructions can block laser beams, making it difficult for LiDAR to generate a complete and accurate 3D representation of the environment. To overcome the line-of-sight limitations of LiDARs in indoor environments, fusion of sensors has been suggested in which LiDARs can be combined with other sensors, such as cameras, radar, and ultrasonic sensors, to provide a more complete representation of the environment. The data from these different sensors can be combined to produce a more comprehensive and accurate representation of the environment. SLAM algorithms can use the data collected by LiDAR at different locations to identify objects, even if they are not in the direct line of sight of the sensor. However, SLAM assumes that objects in the environment are diffuse objects [5]. Unlike opaque objects, reflective and transparent objects may lead to ambiguous or erroneous LiDAR perception processes. Mirror-like objects and glass-walled environments distort light beams. Due to data distortion, LiDAR sensors cannot accurately measure the distance to transparent and reflective objects, leading to a potential collision triggered by errors in SLAM map construction.

2. Related Works

Several strategies have been proposed to tackle the difficulties posed by reflective objects in LiDAR-based SLAM. Some researchers have suggested multisensor fusion solutions to provide complementary information. Diosi and Kleeman [6] used LiDAR and sonar sensor fusion to remove specular reflections by detecting all surfaces as solid objects. Singh et al. [7] combined laser and sonar data using a Bayesian filter to estimate the distance to transparent objects more accurately. Yang et al. [8] proposed a similar laser and sonar sensor fusion approach to detect mirrors and windows. Since multisensor fusion approaches need different sensors, this approach has higher computational and financial costs.

Therefore, most previous efforts have relied only on LiDAR sensors and have striven to obtain supplementary information from laser data. In Wang et al. [9], windows were detected by extracting façade planes from LiDAR point clouds by combining bottom-up with top-down strategies. Cluster point clouds were clustered into potential façade regions by using principal component analysis (PCA) in the bottom-up approach. Then, the top-down approach was performed to utilize the random sample consensus (RANSAC) to extract the façade from the potential façade region. Hao et al. [10] proposed a window detection method based on building wall extraction from scene point clouds. The building walls were extracted from the scene point clouds according to a collection of characteristics, then the building facade was sliced both horizontally and vertically to detect window regions. Pu et al. [11] also attempted to detect glass by distinguishing façade features such as walls and roofs. They used the obtained knowledge about these features' sizes, positions, orientations, and topology to detect windows from laser point clouds. The drawback of these methods is that auxiliary information, such as glass frames or walls, cannot be extracted in modern buildings containing frameless glass or glass-walled environments.

To address this issue, some researchers have utilized the inherent properties of glass and mirror objects. Reflection intensity characteristics were studied by Shina et al. [12] to detect glass. They used the specular reflection phenomenon, which occurs when the irradiation angle is close to the perpendicular angle to the glass surface. In this case, the reflection intensity is maximum. They also studied the transmission phenomenon when there is an object in the transmission destination. In this scenario, the strongest peak occurs at the incident angle, and the reflection intensity drops as laser light passes through the glass. Similarly, Wang et al. [13] recognized glass panels based on the specular

reflection of laser beams from glass. Then, the glass detection method was combined with a SLAM algorithm to avoid collisions with glass obstacles. Based on the reflective characteristics of a laser beam, Kim et al. [14] designed a scan-matching algorithm to differentiate different scenarios, such as diffuse reflection, specular reflection, and beam penetration. However, several factors, such as the laser incident angle, affect the received intensity, making intensity-based methods unreliable and error prone. Therefore, some researchers have focused on glass detection alone. Tibebe et al. [15] considered the variation in neighboring LiDAR point clouds to differentiate the pulses that pass through glass and the pulses that directly hit objects. Then, two filters were applied using intensity and range discrepancy to identify the boundary of the glass. Li et al. [16] proposed a method to detect mirrors based on the symmetries of real objects and their images in the mirror. To find symmetrical relationships from the point cloud, the robot itself was considered a reference point, and its position was estimated by the Rao–Blackwellised particle filter (RBPF) SLAM algorithm. Yang et al. [17] also used the geometric property of mirror symmetry. A mirror prediction was represented by a Gaussian function. The uncertainty of a mirror prediction was measured by the iterative closest points (ICPs) algorithm.

Most of the existing work focuses on glass detection rather than mirror detection. There have been several attempts to solve the problem of specular and transparent object detection using light emitting sensors, such as laser range finders and LiDAR sensors. The main novelty of this research is to study and characterize the obstacle detection problem for reflective or mirror-like objects using 2D LiDAR for indoor navigation applications by running experiments in multimirror environments. Additionally, we proposed a viable solution based on data classification to detect mirrors in the constructed environment map. This can obviate the need for reference point detection, which was needed in the previous methods. We propose a density-based clustering approach to cluster LiDAR data and separate mirror and nonreflective objects based on their impacts on the LiDAR data.

In the following, Section 2 describes the principle of LiDAR and the effect of reflective objects on it. The robot used in this research, experimental setup, and data collection procedure are explained in Section 3. Experimental results and discussion about the effects of mirrors on LiDAR-based navigation methods are presented in Section 4. In Section 5, we propose a potential solution based on a clustering algorithm to differentiate mirror and nonreflective objects according to received LiDAR signals. Our conclusions are drawn in the final section.

3. Problem Statement

LiDARs can take two- or three-dimensional scans. A 2D LiDAR generally spins around an axis repeatedly, emitting a single beam and taking 360° angle scans of the surroundings in a single plane. On the other hand, a 3D LiDAR emits multiple beams while spinning around an axis, allowing it to take more details from the surrounding environment. Due to the nature of the data collected by 3D LiDAR, it is generally used in outdoor environments, while 2D LiDAR is more commonly used in indoor applications. 3D LiDAR is bulkier, more expensive, and more computationally expensive than 2D LiDAR.

The LiDAR transmits a laser beam and can detect a barrier by using a sensor to catch the reflected beam (Figure 1). The distance between the detected obstacle and the receiver is calculated using time-of-flight (ToF). According to Formula (1), the distance can be calculated by measuring the amount of time that it takes for the emitted pulse to reach the object, reflecting some portion of the emitted ray and returning to the receiver lens of the LiDAR. It is assumed that the emitted and reflected rays travel at 3×10^8 m/s.

$$\text{distance} = (\text{ToF} \times \text{speed of flight})/2 \quad (1)$$

Surface properties affect the way that reflected light is scattered, which in turn impacts the way that LiDAR receives light information. There are four key ways that surface properties affect incident light: specular reflection, diffuse reflection, absorption, and transmission. Figure 2 shows the behavior of light according to the surface property.

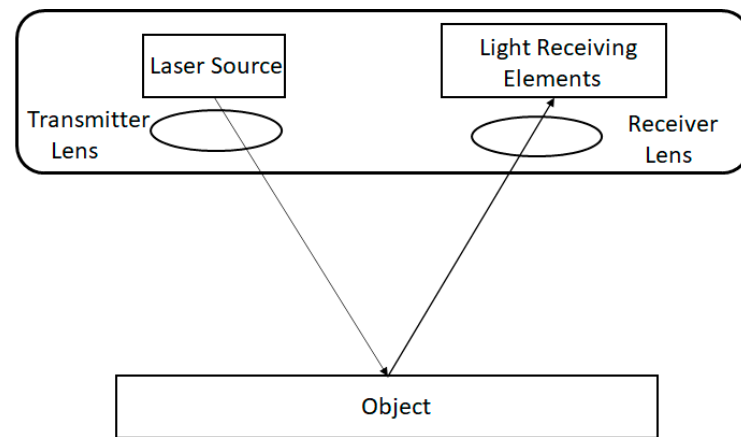


Figure 1. Working principle of a laser-based sensor.

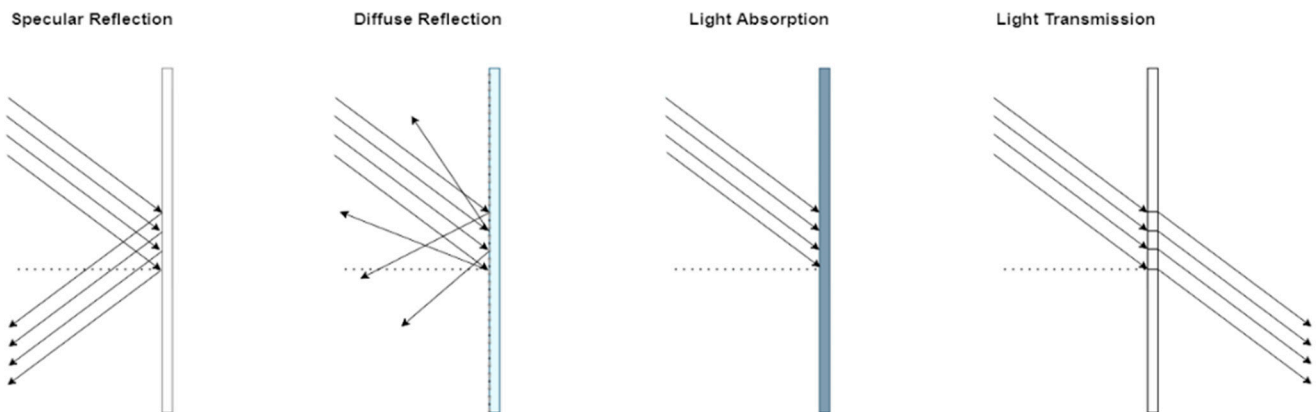


Figure 2. Behavior of Light on various surfaces for specular and diffuse reflection, light absorption and light transmission.

Specular reflection occurs on very smooth surfaces where light is reflected at predictable and consistent angles as the incident light moves along the surface, and these smooth surfaces create a mirror effect. Diffuse reflection occurs on rough surfaces due to inconsistent angles of reflection as the surface moves along the rough surface. Different material properties and surface characteristics affect the absorption and transmission of light on or through a surface. A dark black wall absorbs most of the light as the light meets the surface, whereas a glass wall allows most of the light encountering the wall to transmit through the surface. Most surfaces allow for a combination of reflective behaviors. Diffuse objects can be detected most accurately using LiDAR sensors. Some examples of how different surfaces allow for light reflection are listed in Table 1. Specular and transparent surfaces that allow for large amounts of specular reflection and light transmission distort LiDAR data.

Table 1. Reflective behaviors of various surfaces.

Surface	Example	Primary Light Characteristic	Secondary Light Characteristic
Reflective Surface	Mirror	Specular	Absorption, Diffuse
Diffuse Surface	Concrete	Diffuse	Absorption, Specular
Transparent Surface	Glass	Transmission	Specular, Absorption, Diffuse
Dark Surface	Black	Absorption, Diffuse	Specular
Light Surface	White	Specular, Diffuse	Absorption

4. Methodology and Approach

To investigate the challenges for a LiDAR-based robot localization system, our approach was to use a classical small-scale indoor autonomous robot with a mounted 2D LiDAR. Several mirrors were present in different positions within the developed experimental setup. Four experiments were conducted to investigate the effect of mirror reflection on the map constructed by the SLAM algorithm.

4.1. Tools and Hardware

The ROS (Robot Operating System) is an open-source framework that can be used to build and reuse code between robotic applications. ROS provides many tools, including graphical user interfaces (GUIs), tools for simulation, plotting and visualization, and libraries, language support (C, C++, Python, etc.), and Linux tools, including compilers, debuggers, data loggers, etc. [18]. RViz is a 3D data visualization tool that is used to analyze robot transforms. It can visualize data from both simulations and real-world robots and can capture data individually from all sensors on the robot or robot simulation [19].

The Turtlebot used in this research is a standard ROS platform robot (Figure 3). It is an open-source and low-cost research robot with the capabilities of teleoperation, localization, mapping, navigation, artificial intelligence, and autonomous research. Two-dimensional LiDAR sensors supply reliable measurement data for a whole host of tasks. LDS-01 is a 2D laser scanner capable of sensing 360 degrees that collects a set of data around the robot. Details of the robot and its laser system are shown in Table 2.

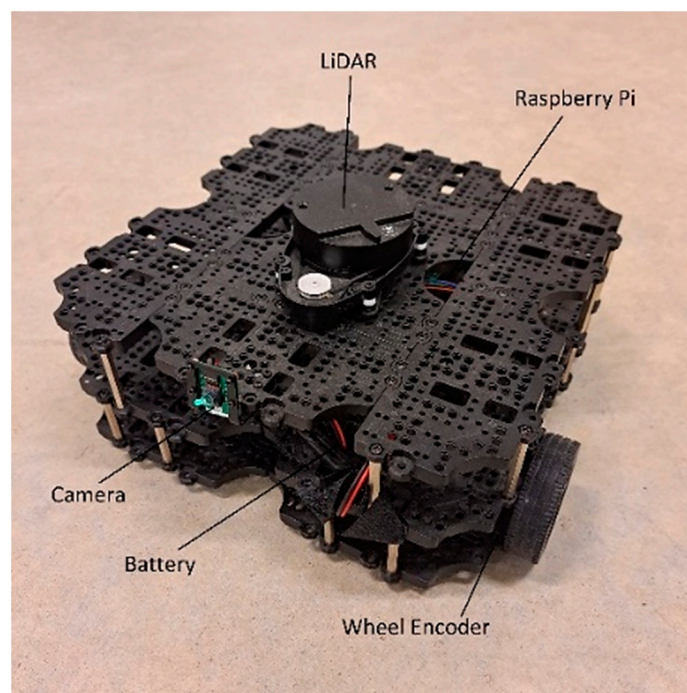


Figure 3. Indoor mobile robot-Turtlebot 3 Waffle Pi.

Table 2. Turtlebot 3 and LiDAR Specifications [20].

Turtlebot 3–Waffle Pi: Specifications	
SBC	Raspberry Pi 3
Embedded Controller	OpenCR
Sensors	Raspberry Pi 3 Camera
	360 LiDAR (LDS-01)
	IMU (3-axis gyroscope, accelerometer, magnetometer)
	LDS Specifications
Detection Distance	120~3500 mm
Distance Precision	± 15 mm $\pm 5.0\%$
Distance Accuracy	± 10 mm $\pm 3.5\%$
Scan Rate	300 ± 10 rpm

4.2. Interface Setup

The Turtlebot has both an SBC (Raspberry Pi) and a controller (OpenCR). The micro-controller is used for communication with sensors and actuators. In this case, the OpenCR interacts with the IMU and the Dynamixel servo motors. This setup is particularly useful for odometry. SBCs are essentially small-scale, fully functioning computers that can run an operating system. This is desirable for use with ROS, as it needs an operating system to run on, so using a raspberry pi allows ROS to be used directly on the robot. The secure shell protocol (SSH) is used to communicate with the SBC of the mobile robot from a PC using remote access. It uses a client-server architecture to allow communication between the two entities by providing a secure connection over a nonsecure network, such as Wi-Fi [21]. In this setup, a Linux command was used to establish the connection between the remote PC and the robot.

Once the robot is accessible remotely, ROS-related packages can be brought up through the roscore command. These packages are a collection of programs and nodes, including an ROS master and ROS parameter server [18]. Enabling the ROS master is a prerequisite to using the system, as it allows nodes to be located and communicate with other nodes. After ROS packages are initiated to interface with the turtlebot, the turtlebot packages need to be brought up as well. These packages are provided through Robotis. Teleop, which is short for “teleoperation”, and provides the ability to control movement of the robot through the keyboard of the remote PC. With respect to the Turtlebot, this node allows a general range of motion in four directions and allows the robot to move with a range of linear and angular speeds. Figure 4 shows the communication between the robot and the PC. The SLAM node can only run after SSH is set up and roscore is enabled.

SLAM uses IMU and LiDAR data to build a continuous map of the environment. GMapping is a grid-based SLAM algorithm that uses particle filter-based adaptive Monte Carlo localization and local pose estimation to create a grid-based map of the environment (Figure 5).

The particle filter used in this algorithm is a Rao Blackwell particle filter. This filter sets the initial belief that in a set of uniform Gaussian distributions, every sample has an associated weight. Random weights determine which states are evaluated, where higher weights are more probable and lower weights are less likely. The joint posterior probability of the position and map is estimated. Sampling occurs to generate new particles, where weighting is recalculated and then resampled, and a map update occurs [20].

While running a SLAM node, to obtain the most accurate data, the robot should be run slowly to obtain as many laser samples as the LiDAR can pick up in a particular area. It is not recommended to drive over the same area more than once while mapping because this

will increase noise in the map. To visualize the LiDAR data, RViz is initiated, and the robot is moved through the system through a teleoperation node. The robot can alternatively be moved using remote control or an object detection and automation strategy. After executing the test strategy, the built map is saved for further analysis.

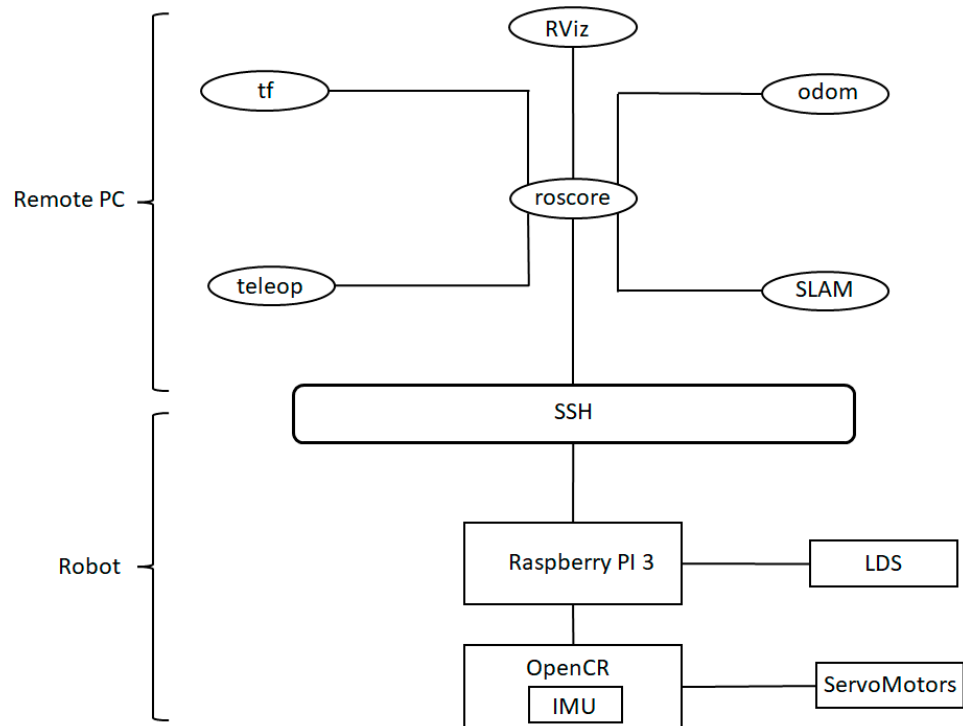


Figure 4. Setup of communication between the mobile robot and remote PC.

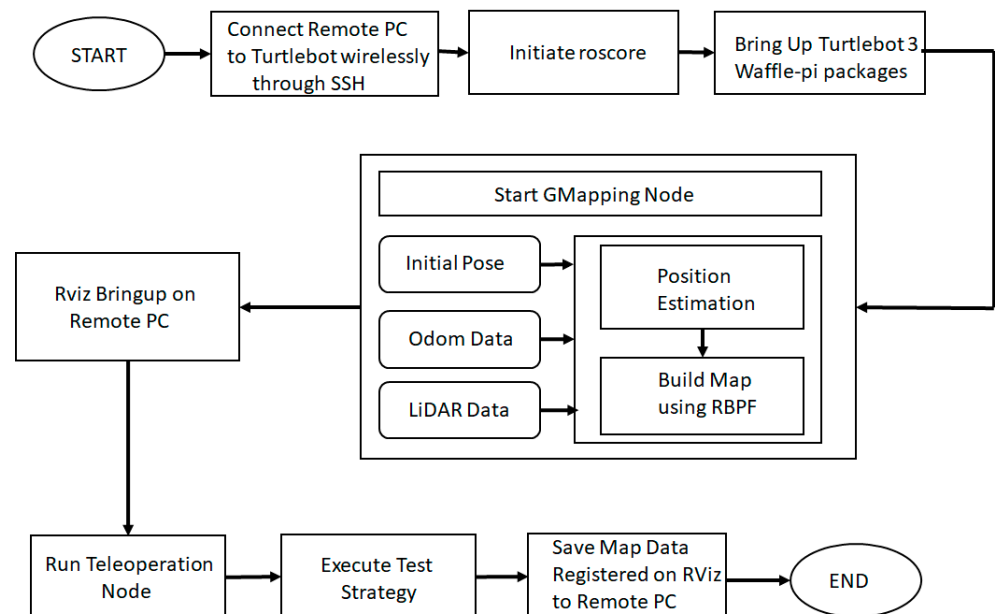


Figure 5. Process of data collection by the mobile robot to build a map using ROS tools.

4.3. Physical Setup

To test the behavior in a more standardized setup, an environment was constructed using a 12 ft × 6 ft rectangular box with three potential mirror locations. The length of the physical environment was chosen to be more than the LiDAR range. These mirrors were

placed in one or more locations, as shown in Figure 6, and the robot was driven in the same direction (from start position A to end position C).

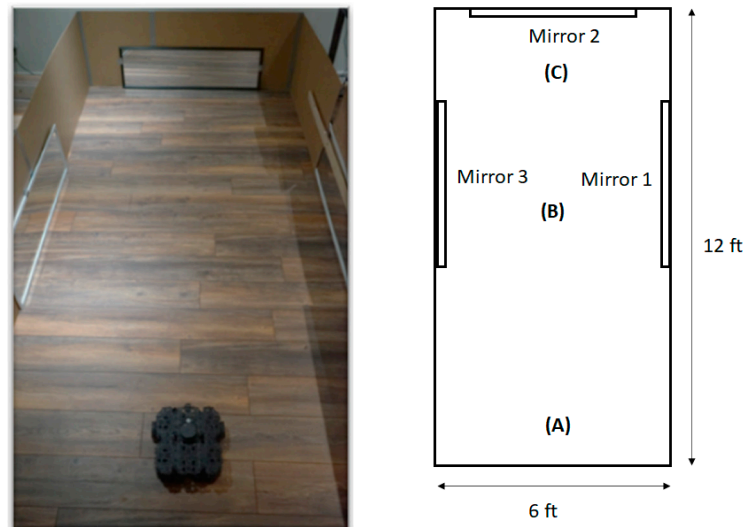


Figure 6. Constructed experimental setup and representation of the various mirror position condition setups with mobile robot locations A, B and C.

5. Experimental Results

This section describes the experimental setup and four test cases. Then, the obtained data are discussed. It should be noted that for each of the following case studies, we repeated the experiment three times. Since the LiDAR-based measurements are highly accurate (Table 2) and the test cases have stationary environments, the repeatability of the collected data in different scenarios is high.

5.1. Data Collection

- Case 1: One frontal mirror (Figure 7a)

At position A, the top borders of the environment have not yet been formed (Figure 8A). This is due to the LiDAR range, which is approximately 10–11 feet. When the Turtlebot moved toward a plane mirror, the mirror was not detected, and some boundary reflections were observed. Being within the detection range of LiDAR, the Turtlebot detects itself when it moves perpendicular to the mirror due to the specular reflection property of the mirror (Figure 2). At position B, the upper boundary is now in range, and the entire experimental area should be mapped. However, the mirror location is still not detected, and the area within the field of view between the mirror location tapered to the LiDAR is still unaccounted for, showing up as an unmapped area in the RViz visualization tool (Figure 8B). At position C, the robot has completed its course. As the mirror is a plane mirror, the reflections of some parts of the boundaries are equidistant from the surface of the mirror as the physical boundary (Figure 8C). This situation is associated with the light transmission property (Figure 2). In other words, the LiDAR system wrongly recognized the reflective surface with a transparent surface. Another interesting observation is that the Turtlebot itself was detected on the surface of the mirror when the robot was in the normal range to the mirror. However, some parts of the mirror remained undetected.

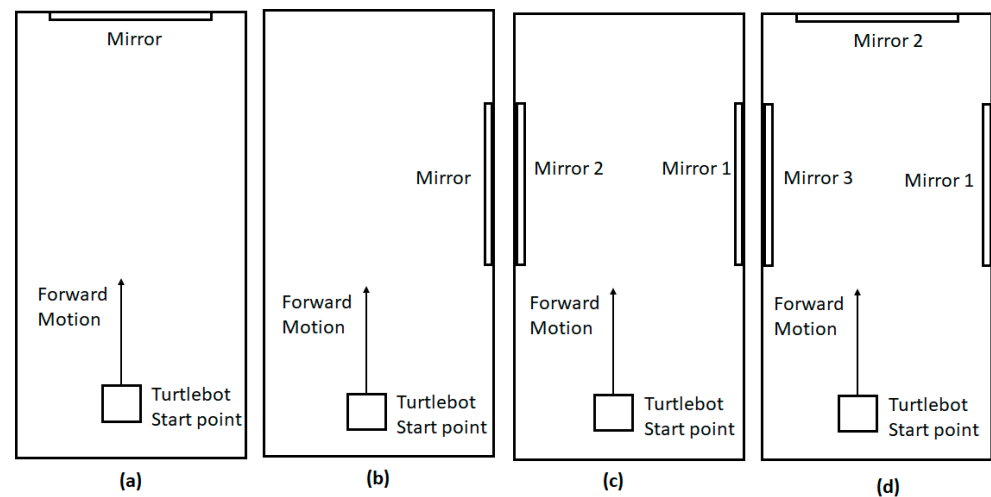


Figure 7. Schematic showing the representation of test cases 1–4 with single or multiple mirrors. (a) One frontal mirror. (b) One side mirror. (c) Two parallel side mirrors. (d) One frontal mirror and two side mirrors.

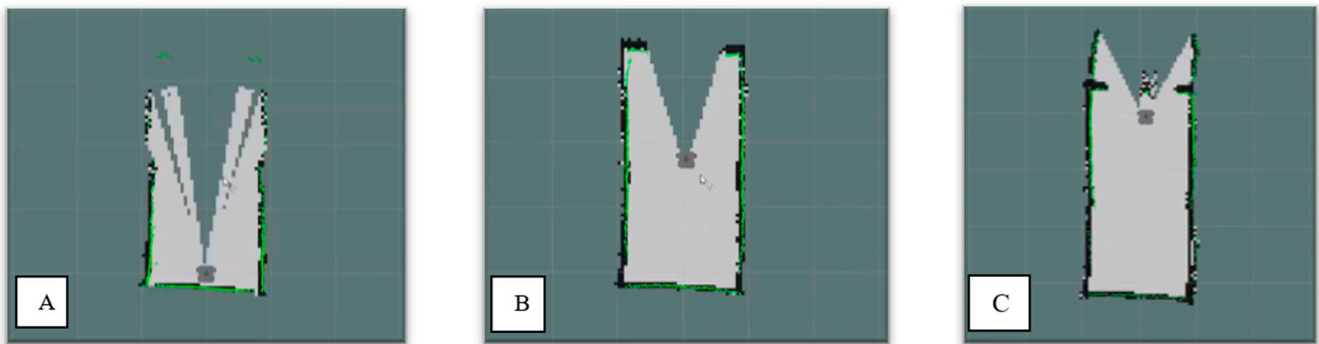


Figure 8. RViz map of test case 1 at positions (A–C) where the mobile robot travels toward a plane mirror.

- Case 2: One side mirror (Figure 7b)

At position A (Figure 9A), it can be seen that the top of the boundary is still out of range, and some of the areas between the LiDAR and the mirror within the field of view of the LiDAR are undetected. A small amount of boundary reflection is seen on the other side of the mirror. At position B, there is no more negative space within the test boundary (Figure 9B). A significant amount of the reflection of the left boundary wall is now detected behind the surface of the mirror at the same distance from the mirror as the physical wall. Similar to case 1, specular reflection of the mirror was misinterpreted as light transmission through transparent medium; it was expected to observe the robot on the surface of the mirror when it passes. This observation was seen in some runs of this test. At position C, the reflection of the left wall boundary is detected behind the mirror and is well defined on the RViz visualization tool (Figure 9C). Some of the bottom boundary reflections are also seen. The robot travels slowly to increase the number of LiDAR scans. However, by observing the amount of detected boundary reflection, it is possible that if the robot slowed down further, complete boundary reflection would have been seen.

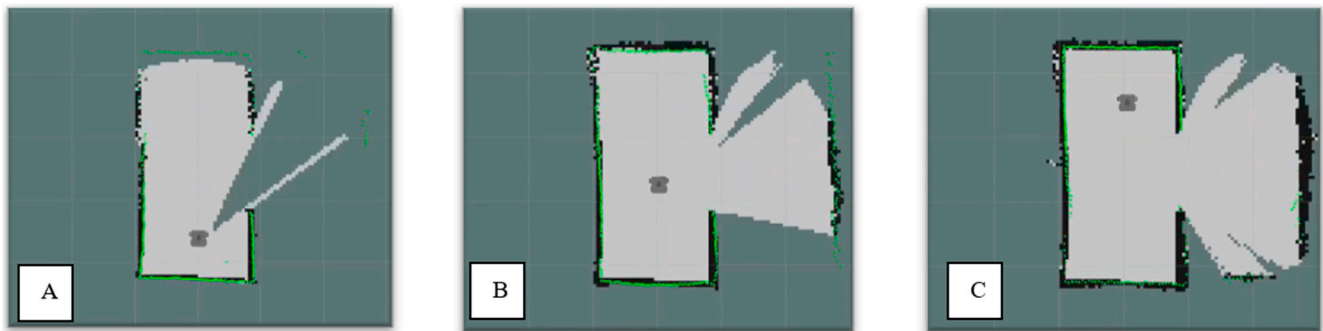


Figure 9. RViz map of test case 2 at positions (A–C) where the mobile robot travels along a plane mirror.

- Case 3: Two side mirrors (Figure 7c)

When the Turtlebot traveled in a straight line between two parallel mirrors, the maps of the boundary wall were seen on both sides behind both parallel mirrors; however, one mapped reflection was lower in map point density than the other (Figure 10A). The Turtlebot was continuously detected on the surface of one of the mirrors. At position B, the reflection of the boundary walls could be seen on both sides behind both mirrors (Figure 10B). It can be assumed that both mirrors map the reflection of opposite walls. In other words, the faulty interpretation of LiDAR to recognize reflective objects with transparent objects can be seen on both sides of the physical setup. Another interesting observation is that the robot was detected on only one of the mirrors. This was continuously corrected as the robot traveled along the mirrors and did not leave a map boundary after crossing them. At position C, solid boundaries were seen behind both mirrors. The mirror reflections of the bottom boundaries were partially formed as well (Figure 10C). The left boundary reflection map was slightly less prominently mapped than the right side. A potential cause for this could be the disturbance caused by the moving Turtlebot map. Inconsistent lighting could also cause the difference in mapping between the two boundary reflection maps.

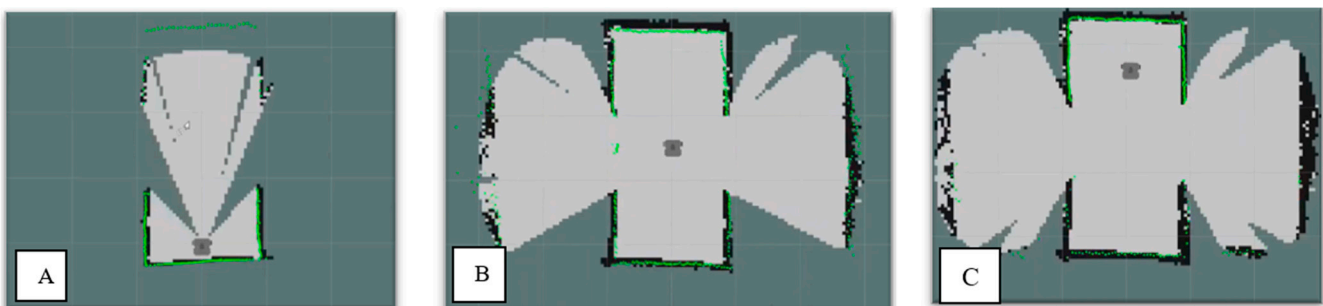


Figure 10. RViz map of test case 3 at positions (A–C) where the mobile robot travels between two parallel plane mirrors.

- Case 4: One front mirror and two side mirrors (Figure 7d)

Similar to the first three experiments, at point A, there was negative space in the field of view between the LiDAR and the mirror locations (Figure 11A). At point B, the reflections of the boundaries of the 12-foot sides of the environment were mapped behind the mirrors as in previous tests (Figure 11B). With respect to the mirror at the top of the environment, there was only negative space in the field of view between the LiDAR and the mirror. However, unlike case 3, where the Turtlebot was continuously mapped in the left mirror, the Turtlebot was continuously mapped in the right mirror in case 4. At position C, the top mirror also allowed for some of the reflections of the side boundaries to be mapped while also mapping the robot reflection on the surface of the mirror, normal to the robot

(Figure 11C). It also left some negative space where no boundaries were in range. Solid reflections of the left and right boundaries were mapped behind both mirrors due to the faulty light transmission property. The right reflected boundary map was less dense than the left. This is likely due to disturbance caused by the robot mapping and correction as it passed through the parallel mirrors.

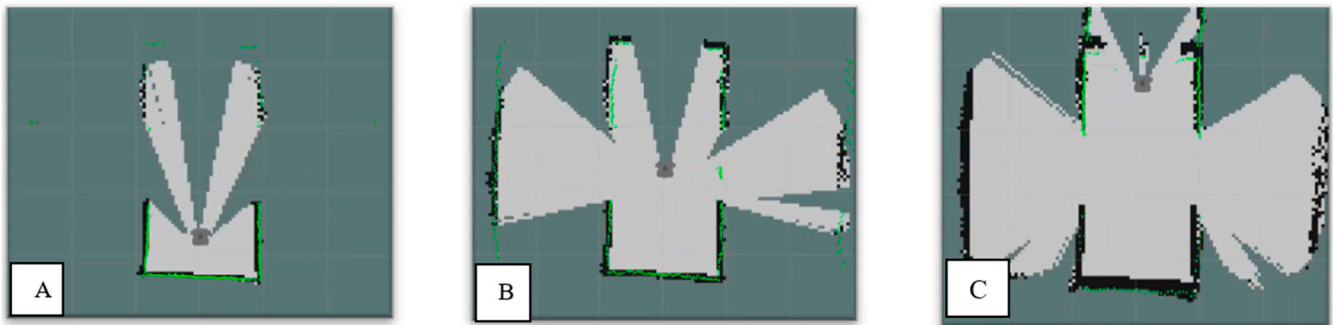


Figure 11. RViz map of test case 4 at positions (A–C) where the mobile robot travels in a multiple mirror environment with three plane mirrors.

To reinforce the understanding of detecting the robot on the surface of the mirror, a stationary test was performed by placing the robot in front of a single mirror, running the SLAM algorithm and viewing the map on the RViz visualization tool. As seen in Figure 12, a solid collection of map points was mapped on the surface of the mirror. The remaining length of the mirror was not mapped, but the reflected portion of the opposite boundary was mapped behind the mirror.

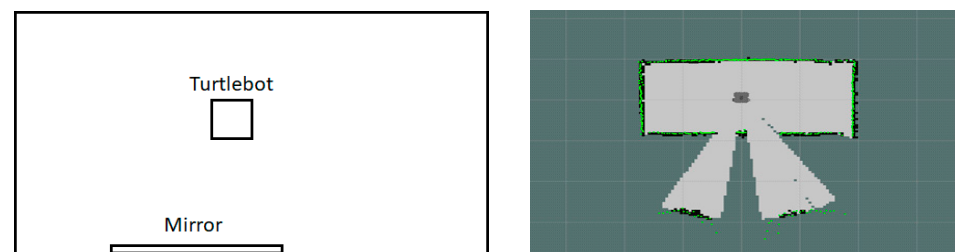


Figure 12. RViz map of test positions (A–C) where the mobile robot is stationary in front of a plane mirror.

According to Figure 13, there are some negative spaces where mirrors were seen consistently over multiple experiments, both before and after the mirrors. This is consistent with the uniform reflection of all LiDAR scan points from the start to end of the mirror. It is possible to detect a reflection caused by any boundary that is out of the LiDAR range. Alternatively, LiDAR was unable to receive any intensity from any other nearby reflected objects. All scans that lie outside the length of the mirror are unaffected by having a mirror in the environment. That is, only the points between the mirror and the LiDAR are affected. Beyond the mirror, all LiDAR readings are false.

False detection within the mirror was seen consistently when the opposite boundary was in range (Figure 14). This is consistent with the law of reflection for a plane mirror, stating that for reflection off a plane reflective surface, the angle of incidence is equal to the angle of reflection. Thus, it can be assumed that any other diffuse object placed at a distance from the mirror where the reflection of the object would be within the range of the LiDAR would also be mapped.

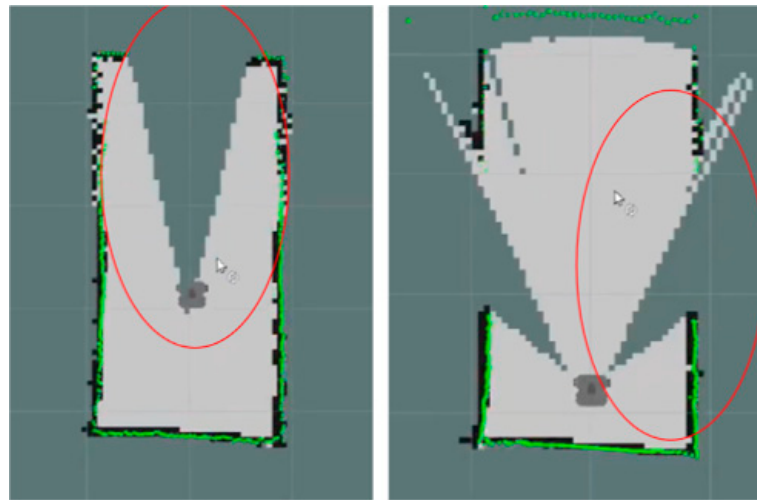


Figure 13. Observation of negative space in mirrored environments. Negative spaces are shown with red ovals.



Figure 14. Observation of the reflected boundary detected behind the surface of the mirror.

It is also interesting that the robot detected itself on the surface of the mirror. According to the previous observation, all obstacles were detected at the correct location in the mirror image; however, in the case of the robot itself, from the collected data it can be inferred that the robot was detected on the surface of the mirror rather than a few feet behind the mirror where their mirror image should have been (Figure 15). It can be inferred that the mirror acted as a diffuse surface.

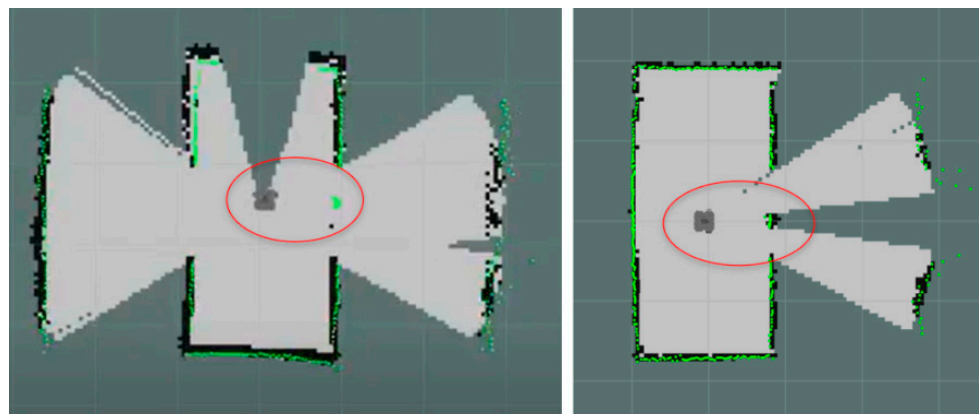


Figure 15. Observation of the robot detecting itself on the surface of the mirror.

5.2. Data Analysis

To better understand the effect of mirrors, we analyzed the LiDAR range and intensity parameters separately. Our laser, LDS-01, has a 1° resolution, which means that a full rotation of the laser will take 360 scan points. The robot receives both range and intensity data to build the map. Equation (2) shows the relationship between sending and receiving intensities.

$$\text{Intensity}_{\text{received}} = \text{Intensity}_{\text{sent}} / \text{Distance}^2 \quad (2)$$

The unit of range is meters (m), and the unit of intensity is Watts/meter (W/m). For better visualization, in the following figures the intensity has been scaled down by a fraction of 600, and it is assumed that the robot is located at the origin of a normalized map. The beam position corresponding to 0 or 360 degrees relates to the position when the LiDAR points to the front of the robot. Figures 16 and 17 show some data collected for test case 2. This is a good example to show some normal boundary conditions that depend on the distance of the LiDAR from the boundary. This value was in the range of 3000 to 4000 W/m when the range was approximately 0.5 m from the boundary. In the case of a single mirror, and when the robot travels in a single direction, very clear data distribution was seen in the range data (Figure 17). However, some collected intensity data were inconsistent with Equation (2). In other words, in Figure 18, the intensity and range values are zero for the mirror (305–330 degrees). Figures 18 and 19 show some normal boundary conditions, some negative space, and some false detection when the robot position is close to the start position of test case 3. Normal boundary conditions in comparison to false boundary detection and robot self-detection are shown in Figures 20 and 21 when the robot is in the middle position of test case 4. Comparing Figures 19 and 21 shows that when the robot is seen in several mirrors, the range and intensity values become more erratic.

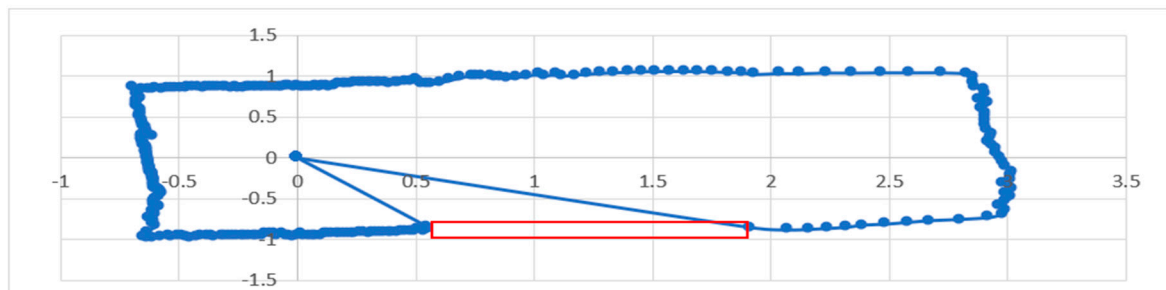


Figure 16. 360 laser scan points for range mapping radially around the robot. The red box shows the side mirror location.

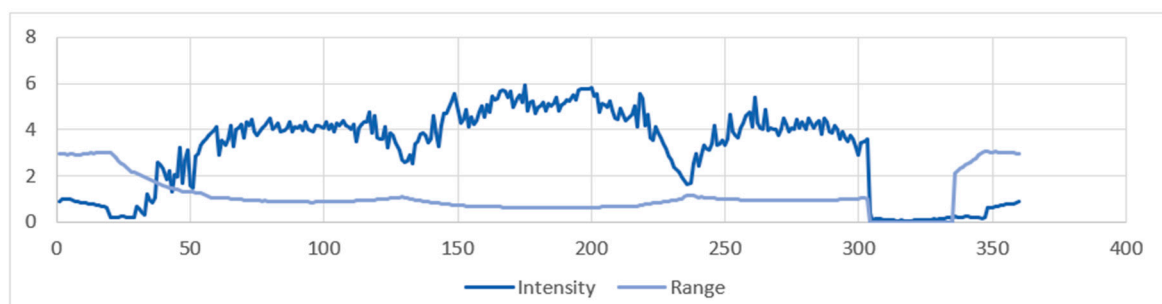


Figure 17. The 360 received intensity laser scan points mapped linearly for the Figure 16.



Figure 18. The 360 range mapping laser scan points mapped radially around the robot. The red boxes show the locations of side mirrors.

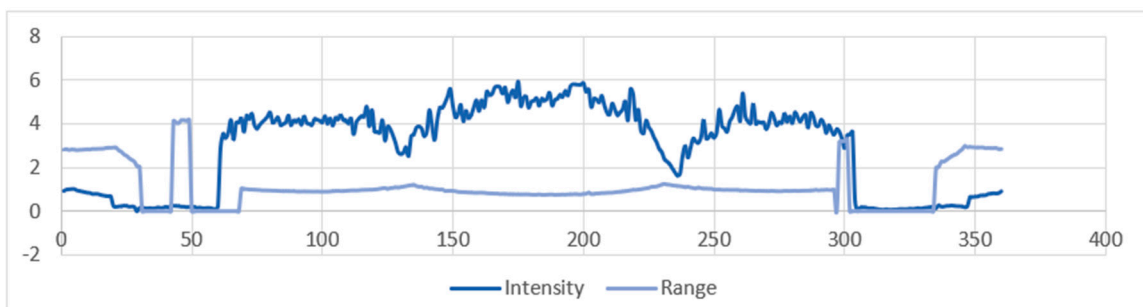


Figure 19. The 360 received intensity laser scan points mapped linearly for the Figure 18.

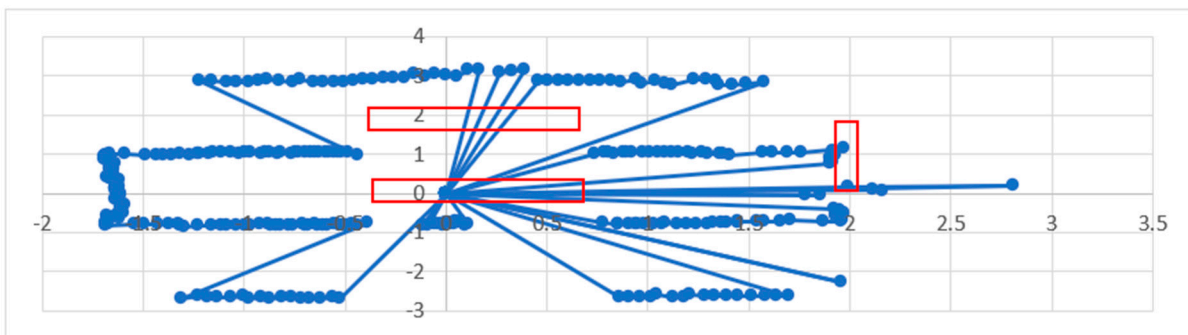


Figure 20. The 360 laser scan points for range mapping radially around the robot. The red boxes show the locations of side mirrors and the frontal mirror.

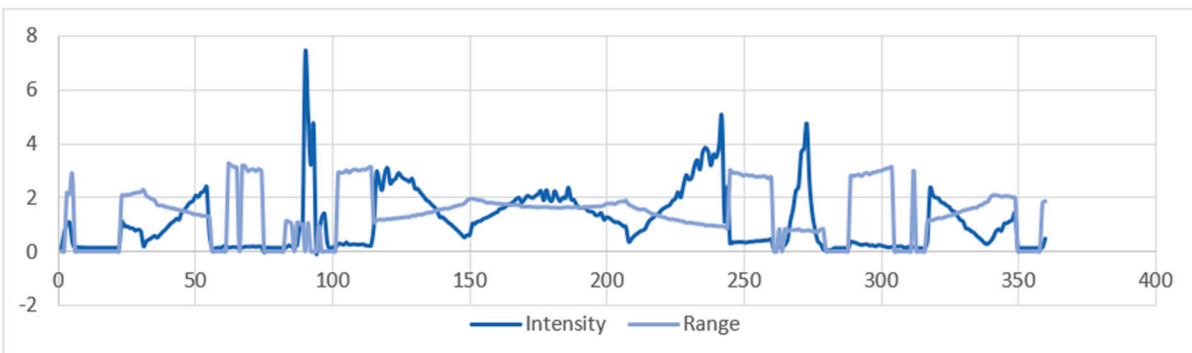


Figure 21. The 360 received intensity laser scan points mapped linearly for the Figure 20.

6. Potential Solutions

In the previous sections we discussed the challenges imposed by reflective objects on the SLAM algorithm. However, there are some workable solutions that can be adopted to alleviate these problems. First, sensor fusion and the complementary properties of sensors can be utilized to detect mirrors. Unlike LiDAR, which fails to detect reflective objects, sonar sensors can detect mirrors and windows. However, the combination of LiDAR and sonar sensors adds to the robot costs, and fusion of two individual occupancy grid maps boosts computation. Additionally, sonar sensors suffer from frequency inference of external ultrasound sources or crosstalk. To obviate the need for additional sensors, symmetries of real objects and their images in the mirror can be used. This method requires a reference object. The robot itself can be considered the reference object, and by moving it, the mirror symmetry between the robot and its images can be utilized to detect the mirror location. Robot self-detection can facilitate this process (Figure 15). Another solution is machine learning employment. Instead of using classical techniques, where the features of the range and intensity data should be manually designed, the end-to-end learning pipeline of deep learning methods can be utilized. Deep learning methods can be utilitarian to extract distinguishing features from reflective and nonreflective objects, leading to more accurate mirror detection results. However, these methods require a massive amount of training data, and the whole learning process should be repeated when the robot working environment is changed. To address this issue, unsupervised machine learning algorithms can be employed.

Clustering is an ideal candidate because it does not require a learning process and can be implemented in real time. Various clustering techniques have been proposed that cluster the input data based on distribution, partition, or density [22]. Density-based spatial clustering of applications with noise (DBSCAN) [23] is able to cluster LiDAR data into mirror and nonreflective sets because they have different properties. Figure 22 shows the proposed solution for clustering mirrored points and removing them in a postfiltering process. The LiDAR scan readings are used as input to the DBSCAN clustering algorithm. Each outlier property can be categorized based on range and intensity readings. Diffuse surface properties remain within a normal range of range and intensity. Negative space properties, caused by mirrors, should provide no range readings and very low or no intensity readings. Self-detection properties can be characterized with very high-intensity readings. Mirror reflection properties mostly include an area of no range or intensity. These properties can be fed into the algorithm to remove clusters that are affected by mirror reflection.

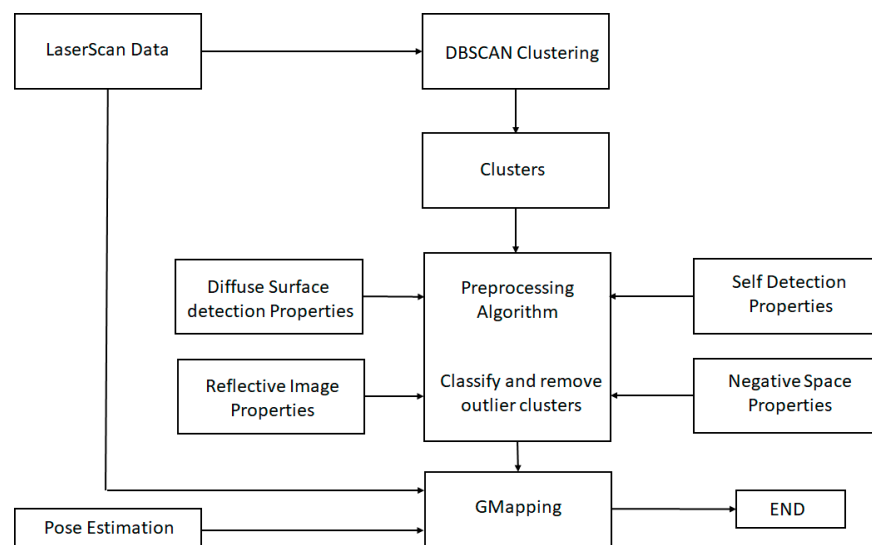


Figure 22. Proposed Preprocessing Algorithm.

The next step would be to implement a postprocessing algorithm (Figure 23). Assuming that all mirrors are plane mirrors, the border conditions of all empty map scans are considered. Using ROS parameters to update the SLAM node, the new map should connect the boundary points. Implementing preprocessing and postprocessing algorithms along with continuous map updates should enable all mirrors to be mapped as diffuse objects in real time.

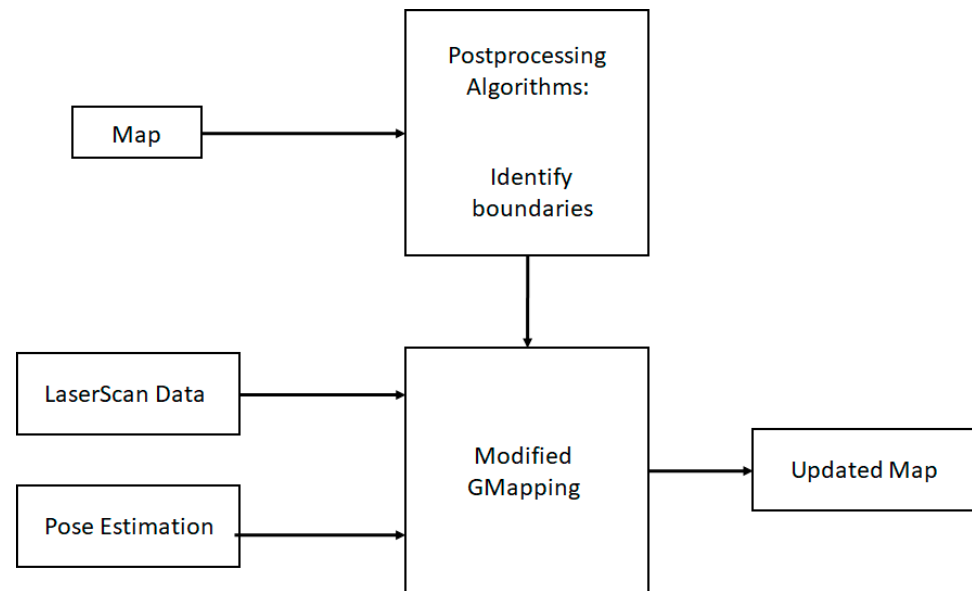


Figure 23. Proposed Postprocessing Algorithm.

7. Conclusions

A potential fully autonomous robot was implemented using Turtlebot along with the ROS ecosystem to perform experiments in an environment with mirror objects. When a mirror was in the range of the LiDAR, a negative space or undetected area was seen in the range between the LiDAR and mirror due to complete reflection of all the laser scans. While traveling alongside a mirror, detection was seen behind the surface of the mirror when diffuse surfaces were in the range of the LiDAR. The robot detected and mapped the mirror reflection of the opposite wall to the mirror in test cases 2–4. The robot was also observed to detect itself in the mirror at all points where it was perpendicular to the mirror. This suggests that the mirror acts as a diffuse object in this situation. Additionally, it was noticed that several times, in the case of two parallel mirrors, the robot was detected only in one mirror. This infers that those inconsistencies in light reflection on the mirror may affect laser scans as well. This phenomenon was seen in test cases 3 and 4.

To identify and remove inconsistencies in the generated maps caused by mirrors, the symmetrical property of the mirror can be used. This approach requires a reference object, which can be the robot itself. By moving the robot, the mirror symmetry between the robot and its image can be utilized to locate the mirror. Self-detection of the robot can simplify this process. The second solution is using supervised machine learning to differentiate reflective and nonreflective objects. This method requires numerous training samples, locating the robot at different positions and orientations and collecting LiDAR data. This method has some drawbacks. Preparing training sets is a tedious and time-consuming task. Additionally, in the case of a changing working environment or the LiDAR, the whole data collection process should be repeated. Therefore, we propose the DBSCAN clustering algorithm to cluster diffuse and reflective surfaces according to the range and intensity information of LiDAR. A postprocessing step is needed to remove the reflective surfaces from the created map.

Author Contributions: All authors contributed equally. All authors have read and agreed to the published version of the manuscript.

Funding: Natural Sciences and Engineering Research Council (NSERC) of Canada discovery grants program and Canada Foundation for Innovation (CFI) JELF grant.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The data related to this paper are available on request from the corresponding author.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Su, P.; Luo, S.; Huang, X. Real-Time Dynamic SLAM Algorithm Based on Deep Learning. *IEEE Access* **2022**, *10*, 87754–87766. [CrossRef]
2. Khan, M.U.; Zaidi, S.A.A.; Ishtiaq, A.; Bukhari, S.U.R.; Samer, S.; Farman, A. A Comparative Survey of LiDAR-SLAM and LiDAR based Sensor Technologies. In Proceedings of the 2021 Mohammad Ali Jinnah University International Conference on Computing (MAJICC), Karachi, Pakistan, 15–17 July 2021; pp. 1–8. [CrossRef]
3. Hasan, M.S.M.S.; Rahman, M.M.U.; Hossain, M.A. Recent Advances in LiDAR-based Obstacle Detection and Localization for Autonomous Vehicles. *IEEE Access* **2020**, *8*, 227482–227502.
4. Carballo, A.; Lambert, J.; Cano, A.; Wong, D.R.; Narksri, P.; Kitsukawa, Y.; Takeuchi, E.; Kato, S.; Takeda, K. LIBRE: The Multiple 3D LiDAR Dataset. In Proceedings of the 2020 IEEE Intelligent Vehicles Symposium, Las Vegas, NV, USA, 19 October–13 November 2020; pp. 1094–1101.
5. Zhao, X.; Yang, Z.; Schwertfeger, S. Mapping with Reflection—Detection and Utilization of Reflection in 3D Lidar Scans. In Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics, Abudhabi, United Arab Emirates, 4–6 November 2020; pp. 27–33. [CrossRef]
6. Diosi, A.; Kleeman, L. Advanced Sonar and Laser Range Finder Fusion for Simultaneous Localization and Mapping. In Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems, Sendai, Japan, 28 September–2 October 2004.
7. Singh, R.; Nagla, K. Multi-data sensor fusion framework to detect transparent object for the efficient mobile robot mapping. *Int. J. Intell. Unmanned Syst.* **2019**, *7*, 2–18. [CrossRef]
8. Yang, S.-W.; Wang, C.-C. Dealing with laser scanner failure: Mirrors and windows. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008; pp. 3009–3015. [CrossRef]
9. Wang, R.; Bach, J.; Ferrie, F.P. Window detection from mobile LiDAR data. In Proceedings of the 2011 IEEE Workshop on Applications of Computer Vision, WACV, Washington, DC, USA, 5–7 January 2011; pp. 58–65. [CrossRef]
10. Hao, W.; Wang, Y.; Liang, W.; Ning, X.; Li, Y. Slice-Based Window Detection from Scene Point Clouds. In Proceedings of the 2018 International Conference on Virtual Reality and Visualization (ICVRV), Qingdao, China, 22–24 October 2018; pp. 35–39. [CrossRef]
11. Pu, S.; Vosselman, G. Knowledge based reconstruction of building models from terrestrial laser scanning data. *ISPRS J. Photogramm. Remote Sens.* **2009**, *64*, 575–584. [CrossRef]
12. Shiina, T.; Wang, Z. An indoor navigation algorithm incorporating representation of Quasi-Static Environmental Object and glass surface detection using LRF sensor. In Proceedings of the 2017 IEEE International Conference on Robotics and Biomimetics (ROBIO), Macau, Macao, 5–8 December 2017; pp. 2508–2514. [CrossRef]
13. Wang, X.; Wang, J.G. Detecting glass in Simultaneous Localisation and Mapping. *Rob. Auton. Syst.* **2017**, *88*, 97–103. [CrossRef]
14. Kim, J.; Chung, W. Localization of a Mobile Robot Using a Laser Range Finder in a Glass-Walled Environment. *IEEE Trans. Ind. Electron.* **2016**, *63*, 3616–3627. [CrossRef]
15. Tibebe, H.; Roche, J.; De Silva, V.; Kondo, A. LiDAR-Based Glass Detection for Improved Occupancy Grid Mapping. *Sensors* **2021**, *21*, 2263. [CrossRef]
16. Li, Z.; Huang, M.; Yang, Y.; Li, Z.; Wang, L. A Mirror Detection Method in the Indoor Environment Using a Laser Sensor. *Math. Probl. Eng.* **2022**, *2022*, 9621694. [CrossRef]
17. Yang, S.-W.; Wang, C.-C. On Solving Mirror Reflection in LIDAR Sensing. *IEEE/ASME Trans. Mechatron.* **2011**, *16*, 255–265. [CrossRef]
18. Joseph, L.; Cacace, J. Introduction to ROS. In *Mastering ROS for Robotics Programming*; Packt: Birmingham, UK, 2018.
19. Pyo, Y.; Cho, H.; Jung, R.; Lim, T. Mobile Robots. In *ROS Robot Programming*; ROBOTIS Co., Ltd.: Seoul, Republic of Korea, 2017; pp. 279–308.
20. Robotis. Turtlebot 3, Robotis. 2022. Available online: <https://www.robotis.us/turtlebot-3/> (accessed on 10 July 2022).
21. Zhang, X.; Lai, J.; Xu, D.; Li, H.; Fu, M. 2D Lidar-Based SLAM and Path Planning for Indoor Rescue Using Mobile Robots. *J. Adv. Transp.* **2020**, *2020*, 8867937. [CrossRef]

22. Ahmad, A.; Khan, S.S. Survey of State-of-the-Art Mixed Data Clustering Algorithms. *IEEE Access* **2019**, *7*, 31883–31902. [[CrossRef](#)]
23. Khan, K.; Rehman, S.U.; Aziz, K.; Fong, S.; Sarasvady, S. DBSCAN: Past, present and future. In Proceedings of the Fifth International Conference on the Applications of Digital Information and Web Technologies (ICADIWT 2014), Chennai, India, 17–19 February 2014; pp. 232–238.

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.