A Safe 2D Grid Map for Indoor Wheelchair Navigation Using the LiDAR Camera System

Ba Viet Ngo
Department of Industrial Electronics - Biomedical Engineering, Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education, Vietnam
vietnb@hcmute.edu.vn

Thanh Hai Nguyen
Department of Industrial Electronics - Biomedical Engineering, Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education, Vietnam
nthai@hcmute.edu.vn (corresponding author)

Thanh Nghia Nguyen
Department of Industrial Electronics - Biomedical Engineering, Faculty of Electrical and Electronics Engineering, Ho Chi Minh City University of Technology and Education, Vietnam
nghiant@hcmute.edu.vn

Minh Chanh Cao
Intel Products Vietnam Co., Ltd., Vietnam
minhchanh89@gmail.com

Received: 16 May 2025 | Revised: 18 July 2025 | Accepted: 27 July 2025

 DOI: https://doi.org/10.48084/etasr.12187

ABSTRACT

Mobile robots are widely used in localization, mapping, and path planning, with map construction being essential for safe navigation. This study presents a method for creating a safe 2D grid map for indoor electric wheelchair navigation using a LiDAR camera system. The process converts a 3D environmental map into a 2D grid map, helping the wheelchair detect obstacles at unsafe heights. First, a 3D environmental map is generated using the LiDAR camera system of an iPhone 12 Pro, capturing 3D point clouds. The Iterative Closest Point (ICP) algorithm merges these point clouds into a complete 3D map. Accuracy is verified by comparing the distances between the real environment and the generated map, with deviations less than 2%. To enhance safety, points exceeding the wheelchair's height limit are removed, forming a refined 3D height map. This map is then projected into a 2D representation and overlaid with a grid, ensuring each cell matches the wheelchair's dimensions. A pixel density approximation refines the grid. The Q-learning method is applied for autonomous wheelchair navigation, proving the system's effectiveness in real indoor environments.

Keywords-safe 2D grid map; 3D point cloud map; smart wheelchair; ICP algorithm; LiDAR camera system

I. INTRODUCTION

Map construction is a critical issue in the field of robotics. When navigating on an existing map, the position and orientation in the data are fixed. However, in practical applications, robots not only need to construct maps for unknown environments but also need to localize themselves within these environments. Therefore, localization, mapping, and navigation are integrated [1, 2], and the 2D grid map is expanded at the end of the established system so that it can be used for subsequent navigation and other applications. In recent years, significant advances have been made in Simultaneous Localization and Mapping (SLAM) within unknown environments, particularly when using laser range finders [3, 4]. In addition, current SLAM algorithms can localize robots within a few centimeters and construct accurate metric maps of the local environment. However, maps generated by laser beams only display obstacles intersecting with the beam's plane. The problem is that obstacles above or below that plane remain undetected.

A 2D grid map is a two-dimensional representation of an environment, with each grid cell representing a specific attribute or characteristic of the environment. Therefore, it can be used as a mobile robot map to store data such as elevation, variance, color, friction coefficient, and navigability [5]. In

practice, there are two approaches to creating grid maps: one involves radar scanning, and the other involves obtaining a 3D point cloud of the environment using a stereo camera and then generating the grid map. In addition, laser-based 2D SLAM algorithms commonly used in the Robot Operating System (ROS) include Hector SLAM and Gmapping [6, 7]. Hector SLAM is an excellent algorithm that relies on high-precision LiDAR data and does not require a rangefinder. Although the core idea of the Gmapping algorithm is particle filtering [8] and also requires a rangefinder, it does not require high-performance sensors. In Karto SLAM [9], the graph-based SLAM method can be used for the localization and mapping of mobile indoor robots. This method has been extended for use in the non-repetitive linear discrete ROS systems with high optimization.

In practice, many studies still examine grid maps for robot navigation. In [10], 3D laser scan sensors were proposed to generate 2D occupancy grid maps, integrating complete obstacle geometry and effective noise filtering from reflective materials such as glass. This method leveraged full 3D data to preserve obstacle shape information and introduced a technique to filter noise reflection based on indoor structural analysis. In [11], the learning method used the maximum expected value to create occupancy maps directly from sensor data. In [12], the Semantic Evidential Grid Map (SEGM) was introduced, based on Dempster-Shafer theory and combining laser scanners and stereo cameras to create more accurate maps representing the environment without losing any details [12]. For building 2D grid maps, the Anytime D (AD) and Dynamic Window Approach (DWA) algorithms have been applied for path planning using LiDAR sensors [12, 13]. In addition, in [14], the ICP algorithm was used to generate 2D grid maps, with line extraction used to correct errors.

For simple indoor applications, such as robotic vacuum cleaners, as well as more complex outdoor applications, such as self-driving cars, maps are essential for the autonomous operation of robotic systems. In recent years, the process of building 3D environmental maps has become easier with the use of stereo cameras [15] and advanced 3D sensors such as LiDAR, laser scanning [16], smartphone cameras [17], and MS Kinect [18]. Although point clouds obtained from stereo cameras or sensors capture the 3D structure of the environment, they are not directly useful for path planning and navigation algorithms, requiring 2D grid maps as input. Moreover, the discrete point clouds generated by stereo cameras or sensors pose challenges in creating grid maps that clearly define known obstacles and free spaces, which are necessary for reliable path planning. This study proposes a method for constructing a 3D environmental map using a camera and LiDAR system, followed by the conversion of the 3D map into a 2D grid map. Through a threshold-based conversion method, the study successfully transforms 3D point cloud maps into 2D grid maps, enabling pathfinding algorithms and autonomous control for a smart wheelchair.

II. METHOD

A. Constructing a 3D Environmental Map

This paper presents a method for constructing a 3D indoor environmental map using the LiDAR camera system of the iPhone 12 Pro, which allows quick and easy collection of 3D data. Moreover, the LiDAR sensor integrated into the iPhone is significantly more cost-effective than other specialized measurement devices, providing users with a portable scanning tool. This sensor operates in the near-infrared range using a photon detector based on the direct Time-of-Flight (ToF) measurement method, where Vertical Cavity Surface-Emitting Lasers (VCSELs) emit photon beams to measure distances to nearby obstacles. The VCSELs array structure consists of a primary grid with dimensions of 3×3, each containing a sub-grid of 8×8 points, and there is a total of 576 points per photon beam [19]. The operating range of the LiDAR sensor is approximately 5 m, with a 90° Field of View (FoV) and a measurement accuracy of approximately 5 mm. In addition, the point density of the LiDAR sensor is around 7225 depth points/m² at a distance of 25 cm, decreasing to 150 points/m² at a distance of 250 cm. Originally designed for Augmented Reality (AR) applications and enhancing image quality, the depth Application Programming Interface (API) is provided in the Apple ARKit library [20].

The depth APIs in Apple's ARKit library allow users to directly access LiDAR data recorded by the sensor. Figure 1 shows a 3D point cloud captured by the iPhone 12 Pro LiDAR camera system, which typically contains significant noise and errors that can be due to hardware limitations or the scanning process. With up to 7 million points in the point cloud, processing such a large dataset requires considerable time and powerful hardware. Therefore, point cloud filtering is necessary to save resources and processing time, including noise removal, downsampling, and background separation.

(a) (b)

Fig. 1. 3D-point cloud images collected by the iPhone 12 Pro LiDAR camera system: (a) RGB image, (b) 3D point cloud image.

The iPhone 12 Pro camera system scans the environment and generates point clouds to build a 3D environmental map. The generated point clouds are not immediately usable because they contain noise and an excessive number of points, which affect the system's processing speed and execution. In addition, deviations in the points within the clouds lead to inaccuracies during the point cloud registration process for creating the 3D map. Therefore, the initial point clouds need to be processed through a preprocessing block for noise filtering,

downsampling, and background separation. This aims to improve the accuracy and reliability of point clouds, and then a 3D map will be constructed by aligning the point clouds using the ICP method [21].

1) Noise Filtering from the Point Cloud

Noise filtering is used for removing outlier points that affect feature calculations and plane detection. Therefore, a method based on statistical analysis of the neighborhood of a point is employed. For each point, first, the average distance μ_i to its K nearest neighbors is calculated, and then a distribution of these distances is constructed using the following formulas:

$$\bar{\mu} = \frac{\sum_{i=1}^{K} \mu_i}{K} \quad (1)$$

$$\sigma = \sqrt{\frac{\sum_{i=1}^{K} (\mu_i - \bar{\mu})^2}{K-1}} \quad (2)$$

where the global average value $\bar{\mu}$ is used to calculate the standard deviation σ. From (2), the threshold t is calculated as follows:

$$t = \mu + m\sigma \quad (3)$$

where m is the correction factor determined by the user. The average distance μ_i is iteratively calculated and compared with the computed threshold value. If μ_i is greater than the threshold, it is marked as an outlier and described as follows:

$$outlier_i = \begin{cases} true, & \text{if } \mu_i \geq t \\ false, & \text{otherwise} \end{cases} \quad (4)$$

2) Point Cloud Downsampling

Before downsampling, it is necessary to extract features and annotate them in the obtained point cloud. To analyze the local structure, an algorithm searches for neighboring points for each point, and then a feature extraction method based on the distribution of the 3D components is applied. Finally, downsampling is performed with the lowest deviation ratio, and the neighbor search method is implemented in two steps as follows.

a) Step 1: Subdivide the Space

If a point cloud is large, searching the entire space increases the workload for the hardware, leading to a slowdown in execution. Therefore, subdividing the space can effectively reduce the search range of the entire space and decrease the computational complexity. For point cloud data, the size of the space is fixed, and this space is divided into smaller regions of equal size. The smallest volume of the cube containing the entire point cloud is denoted as V. Thus, the total number of points in the point cloud is N_a. For example, point clouds of irregular shapes within a cube are averaged, with the number of points in each block being n. The length L is then calculated using:

$$L = \sqrt[3]{\frac{V}{N_a}} = \sqrt[3]{\frac{n}{N_a} \times \Delta X \times \Delta Y \times \Delta Z} \quad (5)$$

where $\Delta X = X_{max} - X_{min}$, $\Delta Y = Y_{max} - Y_{min}$, $\Delta Z = Z_{max} - Z_{min}$ are the maximum and minimum values of the OXYZ coordinate system, respectively. Due to the uneven spatial distribution of the point cloud, some blocks at the edges may have uneven sizes. After completion of the subdivision of the space, each point will be assigned to its corresponding subspace.

b) Step 2: K-Nearest Neighbor Search

The K-Nearest Neighbor for each point is calculated using the Euclidean distance formula. Taking a point p at the center and r as the radius of the circle O, the subspaces belonging to O and the K-Nearest neighbors are initialized from the subspaces. When the number of points in the subspace is fewer than K, it indicates that the region being examined is sparse, and the search area needs to be expanded.

3) Point Cloud Registration for Creating a 3D Map

Removal of background from the point cloud map is necessary for creating a 3D map. Therefore, a Plane Fitting Criterion (PFC) is applied to compute the deviation of a point from a plane that approximates its neighboring points [22]. First, a plane is fitted for each point's k-nearest neighbors using Singular Value Decomposition (SVD). Thus, the average distance from the point to the plane is calculated by considering all points in the neighborhood. This is compared with the distance from the point to the plane of the current point, which defines the neighborhood. As the average distance of the neighborhood approaches 0, PFC approaches 1. Inversely, as the distance from the point to the plane of the current point approaches 0, PFC approaches 0. PFC is suitable for detecting outliers when considering noise reconstruction of smooth surfaces, but yields poor results around small features and folds.

An ICP algorithm is commonly used to align point clouds. ICP takes advantage of the transformation connections found in two datasets from separate coordinate systems to match the data [21]. The basic principle is to apply a spatial transformation between the source point cloud P and the target point cloud Q in the overlapping region of the two point clouds to minimize the distance between them. Let T be the transformation matrix consisting of the rotation R and translation t, which transforms the source point cloud P into the target point cloud Q. Thus, the transformed point P' is:

$$P' = RP + t \quad (6)$$

Assume $E(T)$ is the error between Q and P' in rotation and translation. The objective of the ICP algorithm is to minimize $E(T)$ by iteratively updating the transformation parameters R and t. Thus, $E(T)$ is calculated using:

$$E(T) = \sum \|P' - Q\|^2 \quad (7)$$

B. Converting a 3D Map to a Safe 2D Grid Map

In autonomous navigation applications, 3D point clouds are a widely developed form that enables robots to perceive their surrounding environment. However, they are limited in terms of use for autonomous navigation due to the large amount of data that needs to be processed to determine the movement direction of an electric wheelchair for disabled individuals. Therefore, this study proposes a method for constructing a safe 2D grid map converted from a 3D environment map, which is much more commonly used in navigation algorithms. First, from the constructed 3D environmental map, a safe height limit

is set to create a 3D map with the safe height for the wheelchair. Next, this 3D map is converted into a 2D map, and finally, it is transformed into a safe 2D grid map. Pathfinding algorithms can be applied in this safe 2D map, making it easier for the wheelchair to navigate to the desired destination.

To optimize the movement path, it is necessary to identify the gaps through which the wheelchair can pass. Since these gaps must meet the safe height and width requirements, the safe height of the constructed 3D environmental map must be established. This 3D map removes points with a height lower than the selected height threshold and only retains points with a height greater than or equal to this threshold. The threshold will be chosen to be greater than or equal to the height of the user while sitting in the wheelchair.

The 3D map with a safe height is converted into a 2D map using a geometric projection to project the 3D points onto a 2D plane. An orthogonal projection is used because it preserves the distances and angles between the points in the 3D point cloud when converting to 2D. The shape and size of the obstacles in the point cloud are preserved as they are projected onto the 2D plane. In particular, a point $A(a_x, a_y, a_z)$ in the 3D space projected onto the 2D plane will be the point $P_a(P_{ax}, P_{ay})$ according to the following formula:

$$P_a \begin{bmatrix} P_{ax} \\ P_{ay} \\ 0 \end{bmatrix} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix} * P; \quad \text{with } P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} \tag{8}$$

With the 2D map converted from the 3D environmental map with a safe elevation, it still cannot be used for automatic wheelchair navigation. Therefore, this 2D map needs to be converted into a safe 2D grid map based on the wheelchair's length and width through the following steps.

- Step 1: Divide the 2D map into square grids of size $u \times v$, where each square has a size of $(m \times n)$ corresponding to the perimeter of the wheelchair.
- Step 2: In the 2D map, the black pixel points represent obstacles, and the white pixel points represent gaps. The density of black pixels in each grid cell is calculated using:

$$\sigma_i = \frac{\sum_{x=i\times n}^{(i\times n)+(n-1)} \sum_{y=i\times m}^{(i\times m)+(m-1)} A(x, y)}{m \times n} \tag{9}$$

where $\sigma_i \left(i = 0, \ldots, \frac{u}{m} \times \frac{v}{n} - 1\right)$ is the density of black pixels in the i^{th} grid cell, m and n are the dimensions of one grid cell, and A is the 2D map image after being divided into square grids.

- Step 3: Compare the density of black pixels σ_i of each grid cell with the threshold δ for determining whether the grid cell is an obstacle or empty space. The safe 2D grid map is represented by a value of 0 for obstacle cells and 1 for empty cells, according to:

$$GridCell_i = \begin{cases} 1, & \sigma_i \leq \delta \\ 0, & \sigma_i > \delta \end{cases} \tag{10}$$

Figure 2 illustrates the process of converting a 2D map to a safe 2D grid map for wheelchair users. Figure 2(a) shows a grid with the size of $u \times v$ created based on the actual experimental environment size and the wheelchair's circumference. Figure 2(b) describes the 2D map overlayed with a grid of square cells, and Figure 2(c) depicts the result of converting the 2D map into a safe 2D grid map.

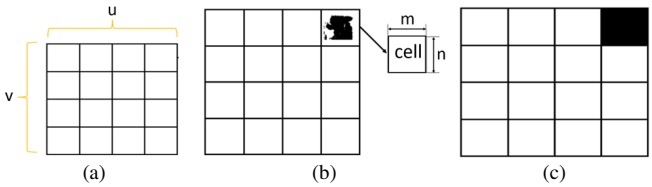


Fig. 2. Converting a 2D map to a safe 2D grid map: (a) Square grid with the size of $u \times v$, (b) Grid overlayed on the 2D map, (c) 2D grid map.

III. RESULTS AND DISCUSSION

A. *Result of Constructing the Indoor 3D Environment Map*

The high-quality camera of the iPhone 12 Pro has enabled the collection of realistic and detailed 3D point cloud images, which are used to create the 3D environment map. Figures 3 and 4 show the point cloud images collected at two different locations in the environment where the map is to be built. In these, Figure 3(a) and Figure 4(a) are RGB images, Figure 3(b) and Figure 4(b) are depth images, and Figure 3(c) and Figure 4(c) are 3D point cloud images. The point cloud images from different positions demonstrate that the camera system collects 3D point cloud data of the environment with good and reliable quality.

Fig. 3. Collected images at the first location: (a) Environment image, (b) Depth image, (c) 3D environment point cloud image.

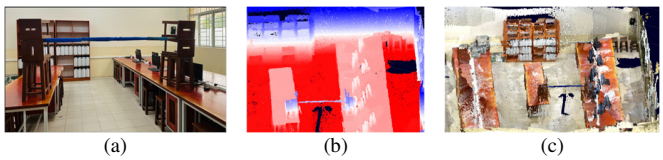
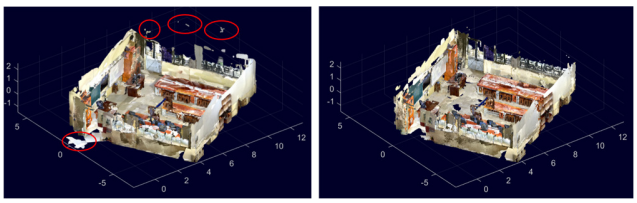


Fig. 4. Collected images at the second location: (a) Environment image, (b) Depth image, (c) 3D environment point cloud image.

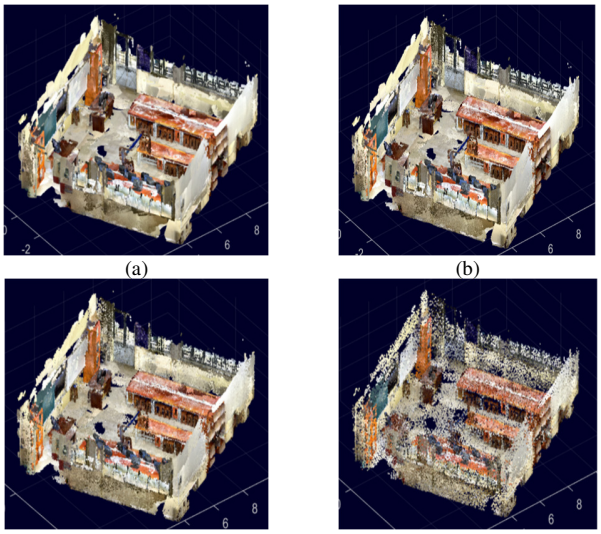
The point cloud maps obtained from the LiDAR camera system typically contain noise, much of which originates from hardware limitations and errors during the environmental scanning process. Therefore, noise filtering is required to achieve better-quality point clouds. Figure 5 illustrates the results of the 3D point cloud maps before and after noise filtering. Figure 5(a) shows the initial 3D point cloud image with noisy and outlier points, as indicated by the red circle. Figure 5(b) shows the 3D point cloud image after filtering, where noisy and outlier points have been completely removed.

For the point cloud obtained with nearly 7 million points, processing them requires a significant amount of computation time and powerful hardware. Therefore, it is necessary to downsample to save resources and processing time. In this study, the 3D point cloud is downsampled to approximately 1 million points, ensuring both quality and processing time efficiency. Figure 6 shows the results of downsampling the 3D point cloud at different thresholds. Figure 6(a) shows the original 3D point cloud with a total of 7 million points. These point clouds are very sharp, but the processing time is very slow. Figure 6(b) shows the 3D point cloud downsampled to 1.3 million points while maintaining quality and faster processing time. Figure 6(c) shows the 3D point cloud downsampled to 500 thousand points, where it is obvious that the cloud no longer maintains quality, and the reliability is low. Figure 6(d) shows the 3D point cloud downsampled to 250 thousand points, with its cloud being very blurry and unusable.



(a) (b)

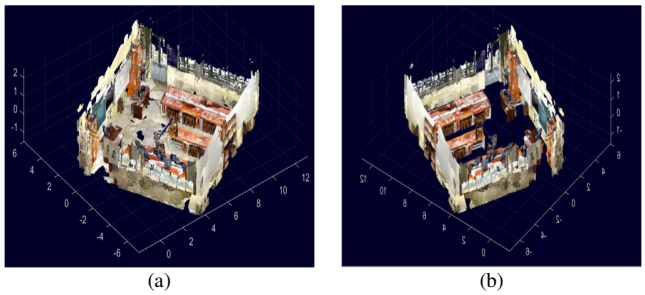
Fig. 5. Point cloud maps before and after preprocessing: (a) Initial 3D point cloud map, (b) 3D point cloud map after filtering.



(a) (b) (c) (d)

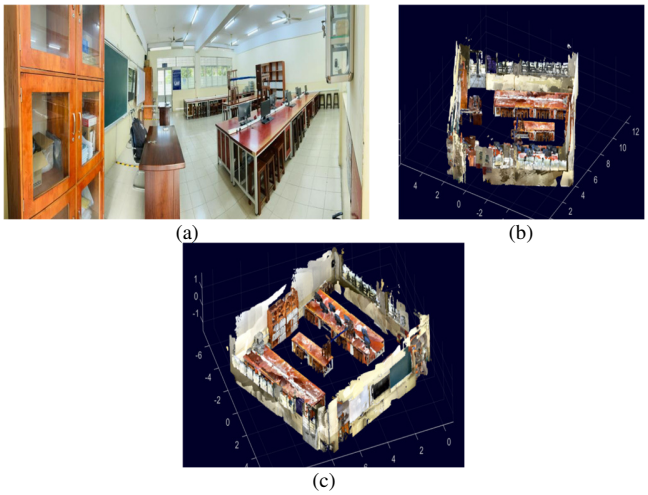
Fig. 6. 3D point cloud maps before and after downsampling: (a) Map with 7 million points, (b) Map with 1.3 million points, (c) Map with 500 thousand points, (d) Map with 250 thousand points.

After noise filtering and downsampling, the background is removed from the 3D point clouds. Figure 7 illustrates the result of the background subtraction for the 3D point cloud, where Figure 7(a) shows the 3D point cloud before background subtraction and Figure 7(b) shows the 3D point cloud after background subtraction. Figure 8 represents the results of constructing the 3D map of the indoor environment, where Figure 8(a) describes the environment that needs to be mapped in this study, with dimensions of 960×960 cm. The 3D point cloud data is recorded using the iPhone 12 Pro LiDAR and camera system at a frame rate of 30 fps. For safety, one can calculate to move the wheelchair at an average speed of 3 km/h in the indoor environment to begin the mapping process. During the mapping process, if an error occurs, the system will display a warning and then pause the mapping process. Therefore, the wheelchair can return to the initial position, where a valid frame can be viewed. During the mapping process, it is observed that errors often occur in areas with no texture or blurry movement. The images of the experimental environment map from various viewpoints are relatively clear, as shown in Figure 8(b, c), demonstrating that the system can accurately recreate the 3D model of the indoor environment with good quality and reliability.



(a) (b)

Fig. 7. Point cloud maps before and after background subtraction: (a) Point cloud before background subtraction, (b) Point cloud after background subtraction.



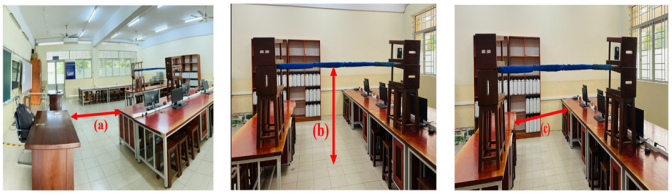
(a) (b) (c)

Fig. 8. Constructing the 3D map from different perspectives: (a) Environment used for 3D mapping, (b) First perspective, (c) Second perspective.

B. Evaluating the Accuracy of the 3D Map

To evaluate the accuracy of the 3D map, this study calculated the deviation by comparing the estimated distance between obstacles in the 3D map with the actual distance in the real environment. This study utilized several MATLAB tools to measure the distance between obstacles in the 3D environment map. Distance measurements between two obstacles were taken at three different positions to assess the accuracy of the

map, as shown in Figure 9. Table I presents the results, showing that the constructed 3D environment map has an average deviation rate of less than 2% compared to the real environment.



(a) (b) (c)

Fig. 9. Three positions for measuring the distances between two obstacles in the environment: (a) Distance between two tables, (b) Distance from the floor to the barrier, (c) Distance between two rows of tables.

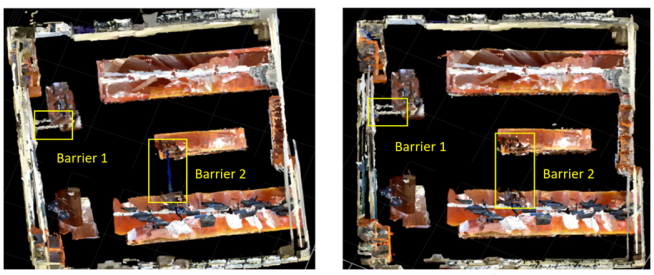
TABLE I. DISCREPANCY RESULTS IN THE CONSTRUCTED 3D ENVIRONMENT MAP

Figure	The distance measured on the 3D map (cm)	The actual distance (cm)	Deviation (%)
Figure 9(a)	122.00	120.12	1.57
Figure 9(b)	153.00	151.16	1.22
Figure 9(c)	169.00	165.72	1.98

C. Building a Safe 2D Grid Map

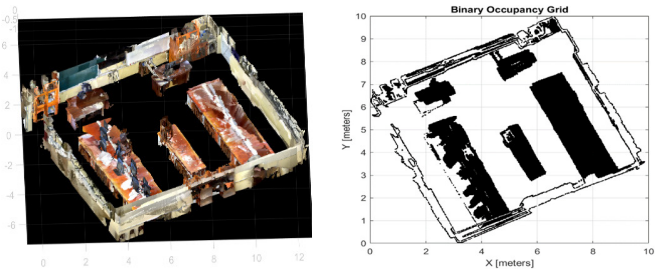
The environment for constructing the 3D map for the experiment includes two horizontal barriers with heights of 77 (Barrier 1) and 151.16 cm (Barrier 2), as shown in Figure 10(a). The safe height threshold δ for wheelchair users is set at 150 cm. After setting the safe height limit for the wheelchair users, all points with a height greater than 150 cm will be removed, such as the horizontal barrier in blue between two rows of the tables with a height of 151.16 cm, which has been removed, as shown in Figure 10(b).

Using geometric projection, the 3D map with safe height was converted into a 2D map, as shown in Figure 11. An electric wheelchair with dimensions of 100×80 cm is used to build the map, and then experiments are performed. The safe size of the wheelchair was chosen to be 120×120 cm to ensure that it will not collide with obstacles during movement. The experimental environment has dimensions of 960×960 cm, the result being that a grid of cells with dimensions 8×8 is superimposed on the 2D map, and the pixel density within each cell is calculated to convert it into a 2D grid map.



(a) (b)

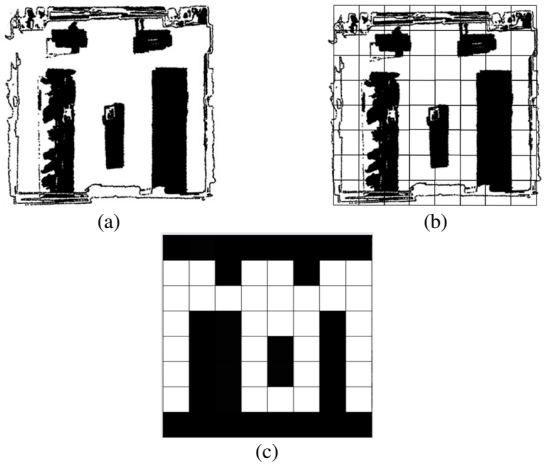
Fig. 10. 3D maps before and after setting the safe height: (a) Initial 3D map, (b) 3D map after setting the safe height.



(a) (b)

Fig. 11. The result of converting from a 3D map with the safe height to a 2D map: (a) the 3D map with the safe height, (b) the 2D map obtained.

Figure 12 shows the result of converting the 2D map to a safe 2D grid map, where Figure 12(a) is the original 2D map and Figure 12(b) is the 2D map with an 8×8 grid overlay. Moreover, Figure 12(c) is the safe 2D grid map for the wheelchair, converted from the 2D map. The black squares on the grid map represent obstacles, areas where the wheelchair cannot move through. The white squares on the map represent safe empty spaces through which the wheelchair can pass.



(a) (b) (c)

Fig. 12. Results of converting from the 2D map to a 2D grid safe map: (a) 2D map, (b) 2D map with an overlay of the square grid, (c) Safe 2D grid map.

D. Application of the Safe 2D Grid Map for Controlling the Smart Wheelchair

To evaluate the practical application of the constructed 2D grid safety map, this study applied a real-world wheelchair control model based on the Q-learning method. Figure 13 illustrates the real-world wheelchair control model using Q-learning. Specifically, by selecting a start point and a destination on the 2D grid safe map (referred to as states), the Q-learning method generates a sequence of predicted actions a (up, down, left, right). These actions and the wheelchair's orientation d on the 2D grid safety map are used in a block to convert into actual actions a_{xl} (forward, backward, left-forward, right-forward) for the wheelchair, as the electric wheelchair is not an omnidirectional control model. Thus, the wheelchair can move along an optimal route from the starting point to the desired position of the user.

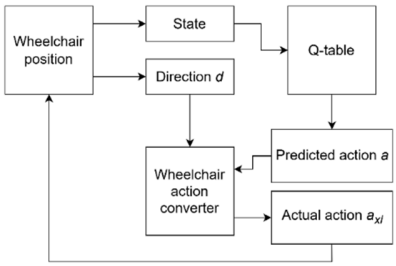


Fig. 13. Control based on the Q-learning method and the safe 2D grid map.

The electric wheelchair is set to move at a speed of 3 km/h to match the system's processing performance. The experimental environment is shown in Figure 14, where the wheelchair is placed at the green cell position in Figure 14(c) to reach the destination at the red cell position. The Q-learning method is applied to suggest the wheelchair's movement route to the predefined destination, as shown in Figure 14(c), with blue arrows representing the simulated path on the 2D grid map. With the simulated path, the system provides a sequence of actual actions for the wheelchair, including forward, backward, left-forward, and right-forward commands to follow the simulated route, as shown in Table II. Figure 14(a) and Figure 14(b) illustrate the wheelchair moving from the starting point to the destination in the real environment. In addition, Figure 14(c) shows the simulated path with the starting and destination points of the wheelchair on the safe 2D grid map, and Figure 14(d) depicts the actual path of the wheelchair retraced as it moves from the starting point to the destination on the safe 2D grid map. In the map, the red path represents the actual path of the wheelchair, while the blue path is the reference path. The result shows that the electric wheelchair can autonomously reach the desired destination based on the safe 2D grid map with minimal deviation without user control. This highlights the potential of using the safe 2D grid map for the autonomous control of electric wheelchairs in real-world applications.

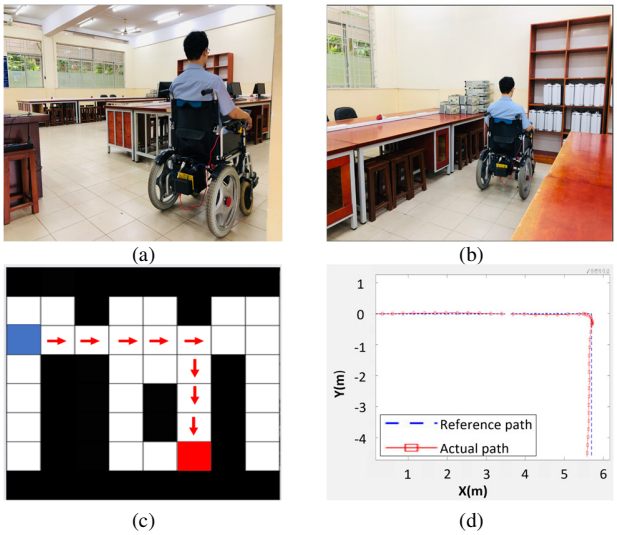


Fig. 14. Wheelchair path and verification results: (a) Starting point, (b) Destination point, (c) Starting and destination points on the map, (d) Actual and reference paths.

TABLE II. ACTUAL ACTIONS OF THE WHEELCHAIR CONVERTED FROM SIMULATED ACTIONS

State of wheelchair	Current direction (d)	Predicted action (a)	New direction (d')	Actual action (a_{xl})
(2,0) to (2,1)	Down	Right	Right	Left–Forward
(2,1) to (2,2)	Right	Right	Right	Forward
(2,2) to (2,3)	Right	Right	Right	Forward
(2,3) to (2,4)	Right	Right	Right	Forward
(2,4) to (2,5)	Right	Right	Right	Forward
(2,5) to (3,5)	Right	Down	Down	Right–Forward
(3,5) to (4,5)	Down	Down	Down	Forward
(4,5) to (5,5)	Down	Down	Down	Forward
(5,5) to (6,5)	Down	Down	Down	Forward

Two-dimensional laser scanners are widely used in robotic mapping due to their simplicity, but they struggle with detecting obstacles of varying heights, such as tables and doorways, and often suffer from noise caused by reflective surfaces. To address these limitations, several studies have proposed enhanced approaches. In particular, in [10], 3D laser scans were used to generate more accurate 2D occupancy grid maps by preserving full obstacle geometry and introducing a novel filtering method to reduce reflection noise based on indoor structural analysis, resulting in more robust and reliable navigation. Similarly, in [23], 2D grid maps were developed using a semi-direct LCSD-SLAM method with a monocular camera, applying the A* algorithm for path planning and demonstrating effective map construction and navigation in robotic applications. In [24], an improved Real-Time Localization System (RTLS) was based on ORB RGB-D SLAM combined with a 2D occupancy grid map, integrated with the ROS framework to allow planning, navigation, and interaction. This system achieves high positioning accuracy, with landmark errors ranging from 39 to 186 mm, and distance errors from 18 to 235 mm, significantly outperforming traditional vSLAM-based maps.

Traditional 3D mapping and navigation systems for mobile robots often rely on expensive, complex sensors, such as terrestrial laser scanners, high-end LiDAR, or RGB-D multi-sensor setups. Although accurate, these systems pose challenges in portability, power consumption, and real-time integration, especially for assistive devices such as smart wheelchairs. For example, LiDAR-based methods [25] and RGB-D SLAM systems [26] require powerful processors and bulky hardware. In contrast, this study proposes a practical and low-cost solution using the built-in LiDAR camera system of the iPhone 12 Pro to create accurate indoor 3D maps with less than 2% error. Unlike typical smartphone-based applications focused on AR or visualization, this method generates 2D occupancy grid maps for autonomous navigation, capturing obstacle dimensions both horizontally and vertically to enhance safety in constrained spaces. This approach reduces hardware requirements while maintaining mapping quality, making it highly suitable for real-world assistive robotics.

IV. CONCLUSION

A high-accuracy 3D environmental map was created using the LiDAR camera system on the iPhone 12 Pro, demonstrating greater flexibility and convenience compared to traditional 3D mapping techniques using earlier sensors. A safe 2D grid map

for a smart wheelchair was successfully constructed from a 3D environmental map, quickly and efficiently. The experimental results on the wheelchair based on the Q-learning method showed that the built 2D grid map has a high practical applicability. The safe 2D grid map derived from the 3D environmental map is of good quality and can be applied for navigation, enabling the wheelchair to move to the user's desired location. However, the system only operates safely in an unchanging environment with static obstacles during the creation of the 3D map. Therefore, the system will continue to be improved by adding sensors to the wheelchair to detect unwanted obstacles during movement, also allowing it to reposition and find new paths to avoid these obstacles.

ACKNOWLEDGMENT

This work was supported by Ho Chi Minh City University of Technology and Education (HCMUTE) under Grant No. T2024-122.

REFERENCES

[1] A. Waga, A. Ba-ichou, S. Benhlima, A. Bekri, and J. Abdouni, "Efficient autonomous navigation for mobile robots using machine learning," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 3, Sep. 2024, Art. no. 3061, https://doi.org/10.11591/ijai.v13.i3.pp3061-3071.

[2] G. Singh, D. Goyal, and V. Kumar, "Mobile robot localization using visual odometry in indoor environments with TurtleBot4," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 14, no. 1, Feb. 2025, Art. no. 760, https://doi.org/10.11591/ijai.v14.i1.pp760-768.

[3] A. A. Kusumo, B. Sandi Marta, B. S. Bayu Dewantara, and D. Pramadihanto, "2D Mapping and Localization using Laser Range Finder for Omnidirectional Mobile Robot," in *2019 International Electronics Symposium (IES)*, Surabaya, Indonesia, Sep. 2019, pp. 126–131, https://doi.org/10.1109/ELECSYM.2019.8901645.

[4] H. Wicaksono *et al.*, "The integrated smart system to assist elderly at home," *IAES International Journal of Artificial Intelligence (IJ-AI)*, vol. 13, no. 4, Dec. 2024, Art. no. 4988, https://doi.org/10.11591/ijai.v13.i4.pp4988-4997.

[5] J. Liu, X. Chen, J. Xiao, S. Lin, Z. Zheng, and H. Lu, "Hybrid Map-Based Path Planning for Robot Navigation in Unstructured Environments," in *2023 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Detroit, MI, USA, Oct. 2023, pp. 2216–2223, https://doi.org/10.1109/IROS55552.2023.10341666.

[6] P. Sankalprajan, T. Sharma, H. D. Perur, and P. Sekhar Pagala, "Comparative analysis of ROS based 2D and 3D SLAM algorithms for Autonomous Ground Vehicles," in *2020 International Conference for Emerging Technology (INCET)*, Belgaum, India, Jun. 2020, pp. 1–6, https://doi.org/10.1109/INCET49848.2020.9154101.

[7] Q. Li and H. Zhu, "Performance evaluation of 2D LiDAR SLAM algorithms in simulated orchard environments," *Computers and Electronics in Agriculture*, vol. 221, Jun. 2024, Art. no. 108994, https://doi.org/10.1016/j.compag.2024.108994.

[8] G. Grisetti, C. Stachniss, and W. Burgard, "Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, Feb. 2007, https://doi.org/10.1109/TRO.2006.889486.

[9] K. Konolige, G. Grisetti, R. Kümmerle, W. Burgard, B. Limketkai, and R. Vincent, "Efficient Sparse Pose Adjustment for 2D mapping," in *2010 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Taipei, Taiwan, Oct. 2010, pp. 22–29, https://doi.org/10.1109/IROS.2010.5649043.

[10] A. Mora, R. Barber, and L. Moreno, "Leveraging 3-D Data for Whole Object Shape and Reflection Aware 2-D Map Building," *IEEE Sensors Journal*, vol. 24, no. 14, pp. 21941–21948, Jul. 2024, https://doi.org/10.1109/JSEN.2023.3321936.

[11] S. Thrun, "Learning Occupancy Grid Maps with Forward Sensor Models," *Autonomous Robots*, vol. 15, no. 2, pp. 111–127, Sep. 2003, https://doi.org/10.1023/A:1025584807625.

[12] J. Li, H. Li, and H. Zeng, "SEGM: A Novel Semantic Evidential Grid Map by Fusing Multiple Sensors," in *Pattern Recognition and Computer Vision*, 2020, pp. 155–166, https://doi.org/10.1007/978-3-030-60633-6_13.

[13] L. Xi and M. Shino, "Shared Control Design Methodologies of an Electric Wheelchair for Individuals with Severe Disabilities using Reinforcement Learning," *Journal of Advanced Simulation in Science and Engineering*, vol. 7, no. 2, pp. 300–319, 2020, https://doi.org/10.15748/jasse.7.300.

[14] D. J. Lee, Y. M. Yun, Y. S. Hwang, and J. M. Lee, "2D grid map building using ICP algorithm and line extraction," in *2014 14th International Conference on Control, Automation and Systems (ICCAS 2014)*, Gyeonggi-do, South Korea, Oct. 2014, pp. 852–855, https://doi.org/10.1109/ICCAS.2014.6987899.

[15] S. Vorapatratorn, A. Suchato, and P. Punyabukkana, "Fast obstacle detection system for the blind using depth image and machine learning," *Engineering and Applied Science Research*, vol. 48, no. 5, pp. 593–603, Jul. 2021.

[16] A. N. Sazaly, M. F. Mohd Ariff, and A. F. Razali, "3D Indoor Crime Scene Reconstruction from Micro UAV Photogrammetry Technique," *Engineering, Technology & Applied Science Research*, vol. 13, no. 6, pp. 12020–12025, Dec. 2023, https://doi.org/10.48084/etasr.6260.

[17] S. I. Sukri, M. F. Mohd Ariff, A. F. Razali, K. Zainuddin, and A. R. Yusof, "Pixel Binning Effects of Smartphone Camera on Three-Dimensional (3D) Model Reconstructed Crime Scene," *Engineering, Technology & Applied Science Research*, vol. 14, no. 5, pp. 17344–17349, Oct. 2024, https://doi.org/10.48084/etasr.8309.

[18] L. Zhou, G. Wu, Y. Zuo, X. Chen, and H. Hu, "A Comprehensive Review of Vision-Based 3D Reconstruction Methods," *Sensors*, vol. 24, no. 7, Apr. 2024, Art. no. 2314, https://doi.org/10.3390/s24072314.

[19] G. Luetzenburg, A. Kroon, and A. A. Bjørk, "Evaluation of the Apple iPhone 12 Pro LiDAR for an Application in Geosciences," *Scientific Reports*, vol. 11, no. 1, Nov. 2021, Art. no. 22221, https://doi.org/10.1038/s41598-021-01763-9.

[20] I. Permozer and T. Orehovacki, "Utilizing Apple's ARKit 2.0 for Augmented Reality Application Development," in *2019 42nd International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)*, Opatija, Croatia, May 2019, pp. 1629–1634, https://doi.org/10.23919/MIPRO.2019.8756928.

[21] X. Li, S. Du, G. Li, and H. Li, "Integrate Point-Cloud Segmentation with 3D LiDAR Scan-Matching for Mobile Robot Localization and Mapping," *Sensors*, vol. 20, no. 1, Dec. 2019, Art. no. 237, https://doi.org/10.3390/s20010237.

[22] T. Weyrich, M. Pauly, R. Keiser, S. Heinzle, S. Scandella, and M. Gross, "Post-processing of Scanned 3D Surface Data," in *SPBG'04 Symposium on Point - Based Graphics 2004*, 2004, https://doi.org/10.2312/SPBG/SPBG04/085-094.

[23] Y. Zhou, B. Li, D. Wang, and J. Mu, "2D Grid map for navigation based on LCSD-SLAM," in *2021 11th International Conference on Information Science and Technology (ICIST)*, Chengdu, China, May 2021, pp. 499–504, https://doi.org/10.1109/ICIST52614.2021.9440650.

[24] L. Xu, C. Feng, V. R. Kamat, and C. C. Menassa, "An Occupancy Grid Mapping enhanced visual SLAM for real-time locating applications in indoor GPS-denied environments," *Automation in Construction*, vol. 104, pp. 230–245, Aug. 2019, https://doi.org/10.1016/j.autcon.2019.04.011.

[25] G. Lu, H. Yang, J. Li, Z. Kuang, and R. Yang, "A Lightweight Real-Time 3D LiDAR SLAM for Autonomous Vehicles in Large-Scale Urban Environment," *IEEE Access*, vol. 11, pp. 12594–12606, 2023, https://doi.org/10.1109/ACCESS.2023.3241800.

[26] S. Zhang, L. Zheng, and W. Tao, "Survey and Evaluation of RGB-D SLAM," *IEEE Access*, vol. 9, pp. 21367–21387, 2021, https://doi.org/10.1109/ACCESS.2021.3053188.