


Article

Development of Autonomous Mobile Robot with 3DLidar Self-Localization Function Using Layout Map

Minoru Sasaki ^{1,*}, Yuki Tsuda ² and Kojiro Matsushita ²

¹ Intelligent Production Technology Research & Development Center for Aerospace, Tokai National Higher Education and Research System, Gifu University, 1-1 Yanagido, Gifu 501-1193, Japan

² Department of Mechanical Engineering, Faculty of Engineering, Gifu University, 1-1 Yanagido, Gifu 501-1193, Japan; tsuda.yuki.97@gmail.com (Y.T.); matsushita.kojiro.h7@f.gifu-u.ac.jp (K.M.)

* Correspondence: sasaki.minoru.n3@f.gifu-u.ac.jp; Tel.: +81-90-6462-0957

Abstract: In recent years, there has been growing interest in autonomous mobile robots equipped with Simultaneous Localization and Mapping (SLAM) technology as a solution to labour shortages in production and distribution settings. SLAM allows these robots to create maps of their environment using devices such as Lidar, radar, and sonar sensors, enabling them to navigate and track routes without prior knowledge of the environment. However, the manual operation of these robots for map construction can be labour-intensive. To address this issue, this research aims to develop a 3D SLAM autonomous mobile robot system that eliminates the need for manual map construction by utilizing existing layout maps. The system includes a PC for self-position estimation, 3DLidar, a camera for verification, a touch panel display, and the mobile robot itself. The proposed SLAM method extracts stable wall point cloud information from 3DLidar, matches it with the wall surface information in the layout map, and uses a particle filter to estimate the robot's position. The system also includes features such as route creation, tracking, and obstacle detection for autonomous movement. Experiments were conducted to compare the proposed system with conventional 3D SLAM methods. The results showed that the proposed system significantly reduced errors in self-positioning and enabled accurate autonomous movement on specified routes, even in the presence of slight differences in layout maps and obstacles. Ultimately, this research demonstrates the effectiveness of a system that can transport goods without the need for manual environment mapping, addressing labour shortages in such environments.

Keywords: autonomous mobile robot; SLAM; 3DLidar; layout map; breadth-first search; obstacle avoidance



Citation: Sasaki, M.; Tsuda, Y.; Matsushita, K. Development of Autonomous Mobile Robot with 3DLidar Self-Localization Function Using Layout Map. *Electronics* **2024**, *13*, 1082. <https://doi.org/10.3390/electronics13061082>

Academic Editors: Xing Wu and Quan Qian

Received: 26 January 2024

Revised: 24 February 2024

Accepted: 5 March 2024

Published: 14 March 2024



Copyright: © 2024 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

In recent years, the use of autonomous mobile robots (AMRs) has been increasing in factories and distribution warehouses to address labour shortages, reduce the burden on people when carrying heavy objects, and improve work efficiency. Examples include Amazon Robotics Kiva [1], which is used in Amazon distribution warehouses, and OMRON's automated guided vehicle [2], which is used mainly in factory automation and logistics.

Unmanned guided vehicles (UGVs) comprise two main categories: the guided type and the SLAM type. The guided type, traditionally employed across diverse sectors, relies on markers such as magnetic tape laid on the floor. Equipped with sensors, these vehicles detect the markers and autonomously navigate to their designated destinations. Conversely, the SLAM method utilizes laser-based environment measurement devices such as 2DLidar or 3DLidar to construct real-time maps of the surroundings. These maps facilitate self-position estimation and navigation along specified routes.

According to data from the Japan Industrial Vehicles Association, the total number of automatic guided vehicle system deliveries and exports in Japan reached 894 in 2020,

marking a continuous increase from 2013 to 2019 [3]. Notably, the proportion of the “magnetic type” decreased to 81.2%, while other types, including SLAM and mark recognition systems, increased to 4.0% from the previous year. This trend underscores the growing interest in unmanned guided vehicles, particularly for factory automation purposes.

In recent years, significant research and development efforts have focused on SLAM technology. This approach, utilizing laser range sensors for simultaneous self-position estimation and environment mapping, has garnered considerable attention [4,5]. These advancements hold promise for enhancing the efficiency and capabilities of unmanned guided vehicles across various industrial applications. SLAM technology has transcended its traditional industrial applications and found utility beyond factory settings [6,7]. For instance, iRobot’s Roomba 980 cleaning robot efficiently navigates rooms using SLAM technology [8].

As it stands, both magnetic and SLAM-based systems require modifications upon introduction or alterations to transportation route layouts. In the case of the magnetic type, embedded magnetic tape necessitates removal and reinstallation when adjustments are made. Similarly, SLAM systems require manual operation of the transport vehicle to comprehensively map the work environment and construct an environment map, a process essential for labour-saving measures. In the SLAM method, self-position is estimated by matching the surroundings using a laser range sensor with an environmental map, so if there are many disturbances, such as people or obstacles, it may not be able to match the environmental map and the location may not be specified [9,10].

A possible solution to the challenges posed by SLAM technology is visual SLAM [11–13]. Unlike traditional SLAM methods that rely on laser-based environment measurement devices, visual SLAM utilizes cameras to construct real-time maps and estimate the vehicle’s position [14,15]. This approach offers several advantages, chiefly, enhanced adaptability to dynamic environments. A disadvantage of visual SLAM is its limitation to 2D mapping. While visual SLAM excels in capturing rich environmental information, particularly in terms of texture and lighting, it struggles with accurately representing three-dimensional spaces. This limitation poses challenges in scenarios where precise depth perception and volumetric mapping are essential, such as navigating complex industrial environments or outdoor terrains with varied elevations. Despite advancements in visual SLAM algorithms and hardware, achieving accurate 3D mapping remains a significant hurdle [12,13]. The reliance on visual cues alone can lead to errors in depth estimation and spatial understanding, especially in environments with limited visual features or occlusions. Other explorations in the visual SLAM field have adopted machine learning approaches utilizing YOLO, convolutional neural networks, and semantic segmentation, among others [7,16–18]. In light of the limitations of visual SLAM, 3D SLAM emerges as a superior solution, albeit with certain drawbacks, particularly in terms of labour intensity. The current study proposes the construction of maps from 3D SLAM.

The review of the literature on UGVs encompasses a diverse array of studies and developments. L. Zhang et al. delved into visual SLAM technologies specifically tailored for autonomous driving vehicles, shedding light on advancements in this critical area [12]. H. Azpúrua et al. contributed to the literature by exploring state-of-the-art robotic exploration techniques, including both single and cooperative approaches with homogeneous and heterogeneous teams, offering insights into collaborative robotics systems [19].

Addressing the crucial localization problem, M. Guan and C. Wen proposed a novel method utilizing an extended Kalman filter in conjunction with particle filters to accurately estimate the robot’s state based on localized Ultra-Wideband (UWB) beacons [20]. X. Zhao and C. C. Cheah presented an innovative Building Information Model (BIM)-based indoor robot initialization system, employing object detection to autonomously initialize mobile robots in unknown environments [21]. These contributions offer valuable solutions to enhance UGV localization and initialization processes.

Additionally, advancements in material handling systems were highlighted by Harsh and S. S. Dhama, who developed a smart material handling system capable of autonomous

operation and seamless integration with other manufacturing subsystems, thus optimizing industrial processes [22]. Meanwhile, Ihtisham Ali et al. introduced a novel challenging dataset aimed at providing a comprehensive testing landscape for mobile robotics, autonomous driving research, and forestry operations, facilitating advancements in these domains [23].

Furthermore, Z. Zhou et al. proposed an object detection solution based on deep learning applied to 3D point clouds, enhancing the automation of Small- and Medium-sized Enterprise (SME) production processes through collaborative mobile robot manipulation [24]. M. Luperto et al. contributed a method for refining 2D grid maps by incorporating predictive layouts of rooms behind closed doors, improving mapping accuracy in indoor environments [25].

Moreover, G. Fracapane et al. conducted a comprehensive literature review on the planning and control of AMRs in intralogistics, elucidating the impact of technological advancements on decision-making processes in this domain [26]. A. Antonio et al. presented the methodology and outcomes of the MoPAD2 robot, an autonomous, cost-effective mobile platform equipped with a thermal scanning platform, showcasing practical applications in digitization efforts [27]. S. Park et al. introduced a novel vision-based global localization approach utilizing hybrid maps of objects and spatial layouts, contributing to the advancement of localization technologies [28]. Other studies working on localization have been conducted in numerous studies touching on different aspects robotics [29,30].

Both 2D SLAM and 3D SLAM rely on the fundamental process of generating a map of the environment they traverse and subsequently executing autonomous movements based on this map. Consequently, manual operation of the transport vehicle and thorough environmental mapping are essential prerequisites. This labour-intensive requirement underscores the necessity for labour-saving measures to streamline the process.

In our study, we employ a novel approach for self-position estimation by extracting data from the wall surface using point cloud information obtained from both the layout map and 3DLidar technology. This data fusion technique allows us to derive precise positional estimates, crucial for autonomous vehicle navigation. Subsequently, we implement our proposed self-localization method with the ultimate goal of achieving autonomous movement. By integrating advanced sensor data processing techniques with state-of-the-art localization algorithms, we aim to enable unmanned vehicles to navigate their environments autonomously and efficiently.

The proposed automatic guided vehicle system is designed to alleviate the manual labour associated with environmental measurement in existing systems. The system incorporates essential functionalities for autonomous movement, including the conversion of layout map data into point cloud maps for self-location estimation, route planning, route following, and obstacle detection and avoidance. By integrating these features, we ensure smooth and efficient navigation of guided vehicles within their operating environment. Additionally, we prioritize user-friendly operation by designing an intuitive interface that simplifies the process from loading the layout map to initiating autonomous movement. This user-centric approach enhances usability and facilitates seamless integration of our system into various industrial applications.

2. Materials and Methods

The proposed automatic guided vehicle system aims to streamline the manual environmental measurement work inherent in existing systems by leveraging self-position estimation and autonomous movement based on a layout map. Following environmental measurement, the robot autonomously constructs an environmental map, mirroring the process of the existing system. Our approach focuses on leveraging the wall shape of the factory for self-position estimation. By utilizing point cloud information obtained from 3DLidar, the layout map, and odometry from wheels, we extract point cloud data specifically for wall surfaces. Subsequently, we employ Monte Carlo localization using a

particle filter, utilizing the degree of matching between the layout map and the converted point cloud information for estimation [4].

The methodological framework of the proposed automatic guided vehicle system is delineated into three primary components: sensors, map, and path planning, as described in Figure 1.

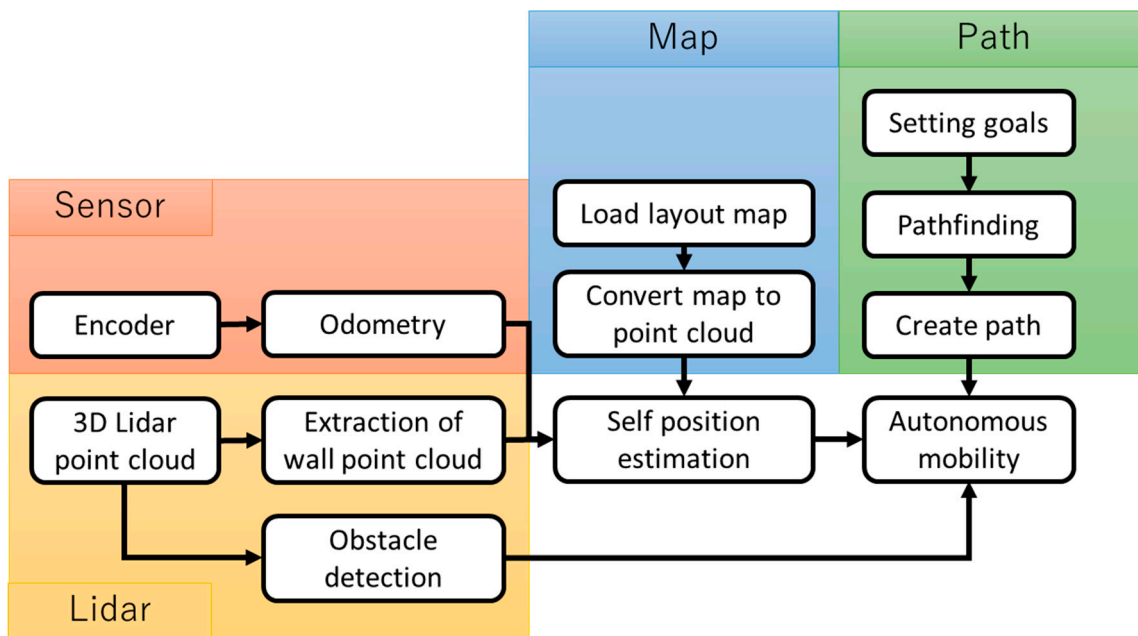


Figure 1. System configuration.

The sensor segment encompasses the integration of various sensors, notably the odometry and Lidar systems. The odometry system, incorporating a magnetic encoder, facilitates precise calculations of the vehicle's movements, transmitting data to the microcomputer via SPI communication. Concurrently, the Lidar system, comprising 3DLidar technology, captures detailed point cloud data of the environment, crucial for self-position estimation as well as obstacle detection. The self-position and obstacle information is passed on to the subsequent segments to achieve the overall goal.

In the map segment, the process of layout map extraction towards self-position estimation is developed. Utilizing point cloud information acquired from 3DLidar and the layout map, the methodology focuses on refining the self-position estimate and piecing together dynamic environmental detail. The extraction of point cloud data exclusively for wall surfaces enhances the precision of the self-position estimation. Additionally, the utilization of Monte Carlo localization, facilitated by a particle filter, enhances the accuracy of the self-position estimation based on the alignment between the layout map and converted point cloud data.

The final part of the methodology delineates the path planning process, encompassing target goal determination, pathfinding algorithms, and autonomous operation. Through comprehensive analysis and utilization of the constructed environment map, our system autonomously identifies target goals, employs pathfinding algorithms to determine optimal routes, and executes autonomous operations, ensuring efficient and obstacle-free navigation. Segments two and three are set out in sections by themselves.

2.1. The Hardware Unit

Figure 2 shows the setup of the automated guided vehicle system. In this case, the setup utilized a two-wheel mobile cart equipped with 3DLidar and a PC. An off-the-shelf bogie was used; it was electrified by replacing the front wheels with free-wheel casters and the rear wheels with geared DC motors, as shown in Figure 2. The configuration of the

motor drive is shown in Figure 3. A magnetic encoder was attached to the motor shaft, as seen in Figure 3b. A magnetic encoder operates by attaching a magnet to the shaft of a motor and measures the rotation angle without contact with a sensor attached to the frame. The acquired rotation angle is used for estimating the amount of movement of the guided vehicle.



Figure 2. Modified trolley cart that served as the automated vehicle in this study.

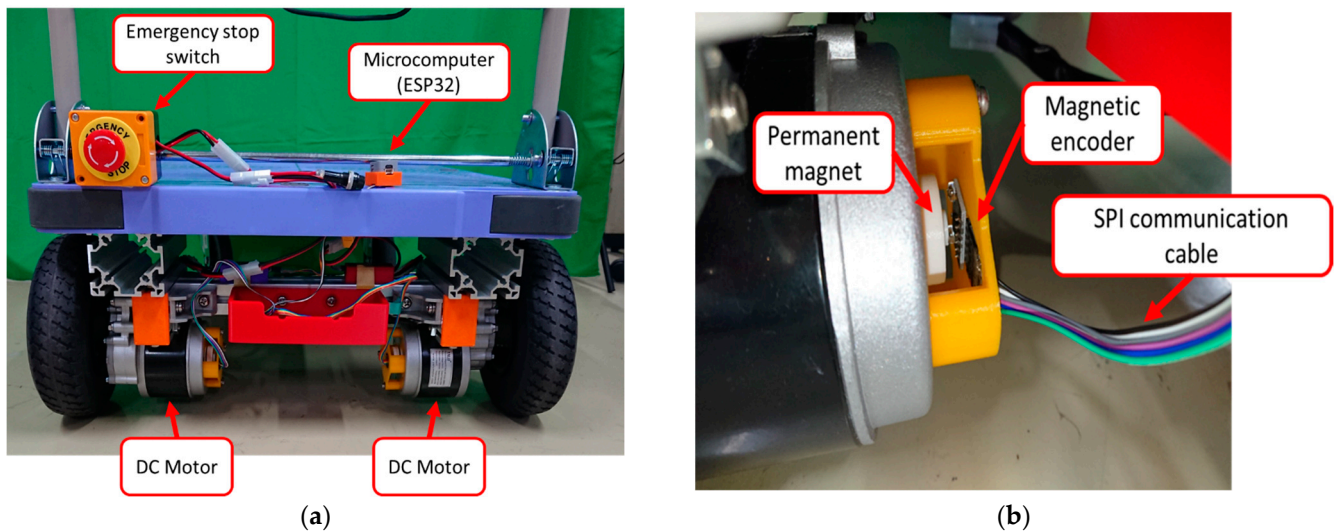


Figure 3. Motor drive setup (a). DC Motor setup. (b) Encoder configuration.

The motor receives the PWM signal and rotation direction Information from the microcomputer at the motor driver and rotates the motor. Commands to the microcomputer are controlled by serial communication from the PC. This makes it possible to control the trolley body by commands from the PC. In addition, the encoder attached to the motor sends the rotation angle to the microcomputer via SPI communication, and the microcomputer transmits it to the PC via serial communication. Details of each component are given in Table 1.

Table 1. Components specifications.

Components	Specifications	
DC motor	voltage	24 V
	Power	250 W
	Rotation speed	120 rpm
	Gear Ratio	40:1
Motor driver	Voltage	6.5~40 V
	Current	13 A
	PWM control frequency	100 kHz
	Control Voltage	1.8~5.5 V
Magnetic encoder AS5048A	Voltage	3.3 V
	Interface	SPI
	Resolution	14 [bit] (16,384)
	Sampling	Max 12 kHz
Microcomputer	Base	ESP32 PICO
	CPU	240 MHz Dual Core
	GPIO	6 [Pin]
	Voltage	5.0 V
	Current	500 mA
3DLidar (Velodyne VLP16)	Horizontal	360°
	Vertical	30°
	Range	0.52~100 [m]
	Rotation Rate	5~20 [Hz]
	Accuracy	±3 [cm]
PC	OS	Ubuntu 18.04
	CPU	Intel i9 9900K
	RAM	64 [GB]
	GPU	Nvidia RTX2080
	ROS	Melodic

2.2. Odometry

An odometry system was implemented using a magnetic encoder, which communicated with the microcomputer via SPI communication. This system calculated both the angle and number of rotations within the microcomputer and communicated this information to the computer via serial communication.

The relationship between odometry and the rotational speed of the left and right wheels is shown in Equation (1). Let θ_R be the rotation angle of the right wheel and θ_L be the rotation angle of the left wheel. Let v_R and v_L be the velocities of the left and right wheels in the external coordinate system, and let R be the radius of the wheel.

$$\begin{cases} v_R = R\dot{\theta}_R \\ v_L = R\dot{\theta}_L \end{cases} \quad (1)$$

If the robot's turning angular velocity is ω , the robot's speed is v , and the turning radius is r , then robot speed v can be stated as (2).

$$v = r\omega \quad (2)$$

where ω is the robot's turning angular velocity, v is the robot's speed, and r is the turning radius. If the distance between the wheels of the robot is d , Equation (3) is obtained.

$$\begin{cases} v_R = (r + d)\omega \\ v_L = (r - d)\omega \end{cases} \quad (3)$$

Based on the above equations, the robot's velocity v and turning angular velocity ω are obtained from the encoder angle attached to the motor, as shown by Equation (4).

$$\begin{cases} v = \frac{(v_R + v_L)}{2} \\ \omega = \frac{(v_R - v_L)}{2d} \end{cases} \quad (4)$$

The velocity and rotational acceleration obtained by the above calculations were used to update the estimated position during self-position estimation.

2.3. Point Cloud from 3Dlidar for Wall Extraction

To estimate self-position from the layout map, the walls' layout map and the wall point cloud data extracted from the point cloud measured by 3Dlidar were employed. The self-position was determined based on the degree of matching between these data sets. An algorithm for extracting wall surfaces was developed to ensure stable measurements indoors. This algorithm utilized the fact that the laser from 3Dlidar often reached the wall through the top of equipment or gaps. Processing was performed within the 3Dlidar Velodyne VLP16's measurement cycle of 10 Hz. The 3D point cloud data measured indoors are shown in Figure 4.

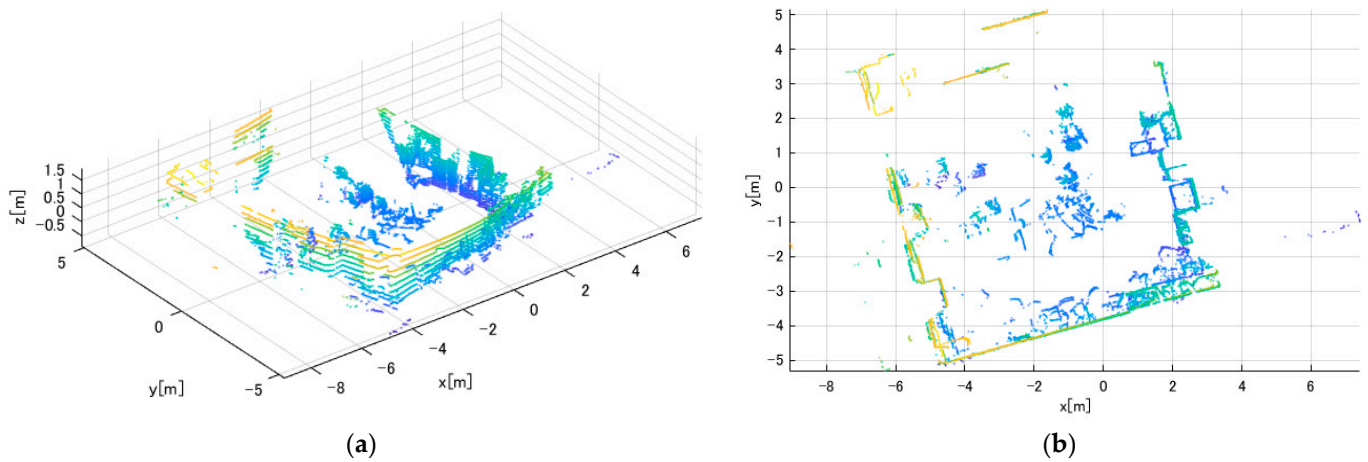


Figure 4. 3D point cloud from Lidar sensor. (a) Side view. (b) Top view.

The wall extraction algorithm is illustrated in Figure 5a. The algorithm defines a region of interest and separates the measurement point by the number of point clouds to create a blue fan-shaped area. The remaining point clouds are down-sampled using a voxel grid filter to reduce and equalize the number of point clouds. The processing results are shown in Figure 5b.

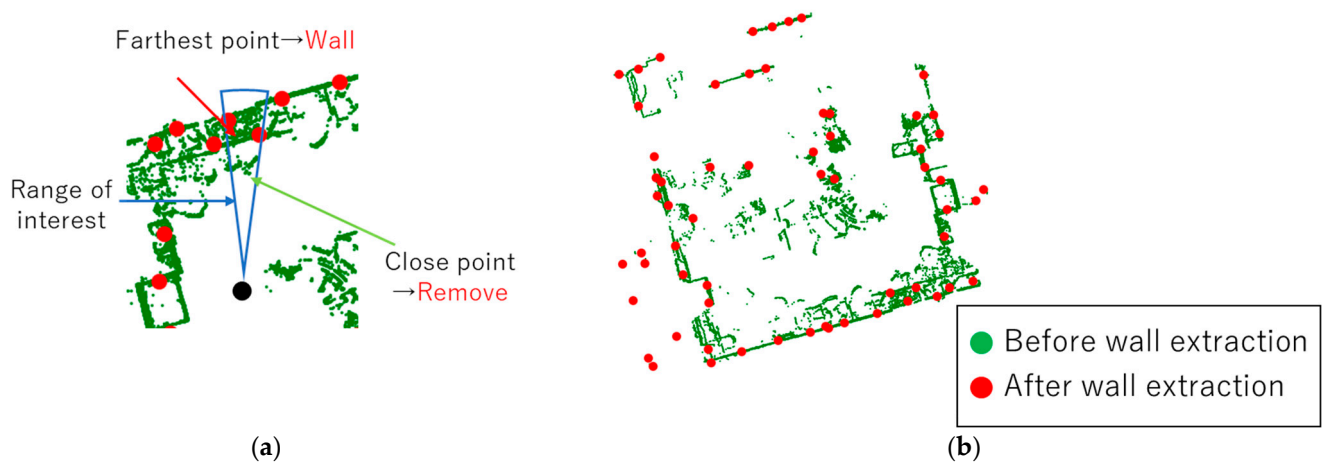


Figure 5. Wall extraction process. (a) Wall extraction region of interest. (b) Output of extraction.

The data derived from Section 2.2 (odometry) and Section 2.3 (3DLidar) is used to refine the proposed localization system derived from the layout maps. The following section elaborates on the generation of point clouds from aerial photographs and maps.

3. Localization Algorithm

The proposed self-localization utilizes a technique that determines the robot's position by amalgamating three crucial elements: the point cloud information derived from 3DLidar scans capturing wall surfaces, the odometry data obtained from the vehicle's encoder, and the point cloud map data originating from the layout map. The first two items have been described in the previous section. This section delves into the process of localization from maps and aerial photographs downloaded from satellites.

3.1. Convert Layout Map to Point Cloud

The conversion process involves transforming the layout map, typically represented as image format data, into point cloud map data suitable for self-location estimation. The layout map, depicting the building's structure with accuracy, serves as the foundation for this transformation. In our experimentation, layout maps of both the factory and the entire site were utilized. The process of converting a layout map into point cloud data involves several key steps:

1. Image preprocessing. Initially, the layout map undergoes preprocessing, which may include binarization to enhance contrast and facilitate the detection of structural elements.
2. Line detection. Techniques such as the Hough transform are employed to detect straight lines in the layout map, aiding in the identification of structural features such as walls and corridors.
3. Point cloud. Detected straight lines are converted into point cloud data, with each line segment represented as a series of points in three-dimensional space, capturing the spatial characteristics of the layout map.
4. Down-sampling techniques may be applied to reduce the density of points in the generated point cloud data, managing computational complexity while retaining essential structural features.
5. The resulting point cloud map data are integrated with the sensor data, particularly from the Lidar system, enhancing the accuracy of the self-location estimation and enabling precise navigation within the environment.

Each of these steps is briefly explained and illustrated by the following images. For illustrative purposes, we utilize the building layout of Gifu University.

The preprocessing stage eliminated unnecessary details, as shown in Figure 6b after the binarization processing. From this, straight line detection is performed using the

Hough transform. Since wall shapes in factories are often relatively long straight lines, the Hough transform removes short straight lines and curves other than walls. The result of straight-line detection by the Hough transform is shown in Figure 6c.

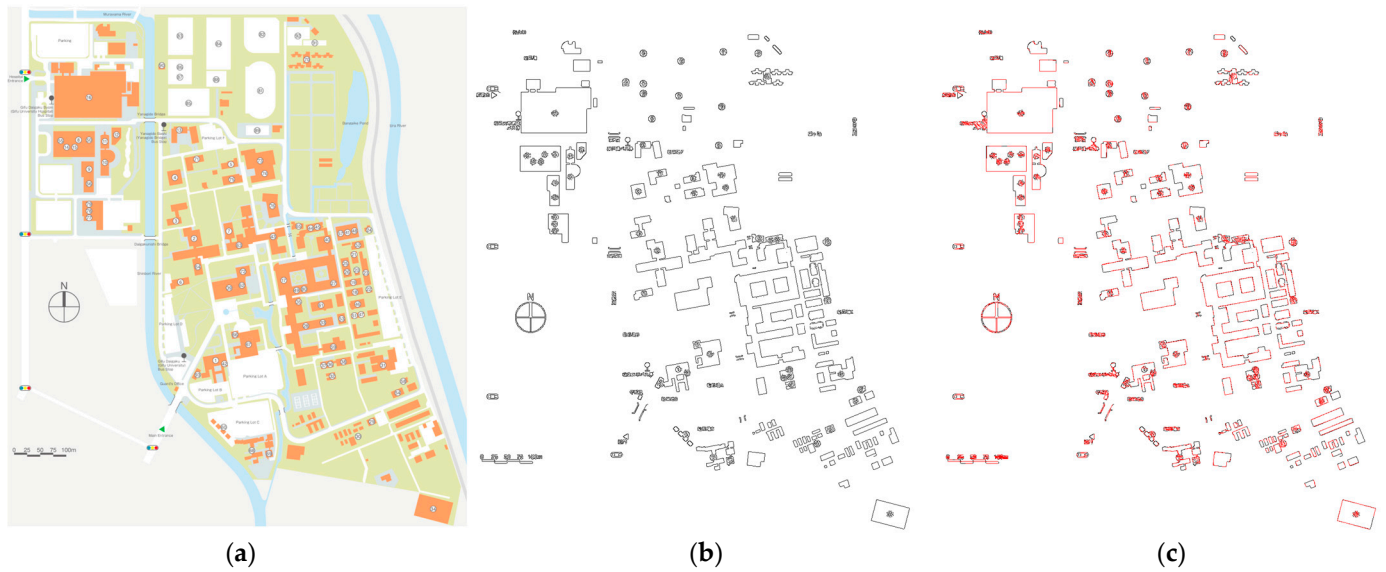


Figure 6. Process of converting layout map to point cloud. (a) Original layout map (Numbers represent different building/sections in the university). (b) Processed (binarized) map. (c) Line detection.

Next, we discuss the conversion process from lines to the point cloud. A line plot is divided into points at regular intervals (d [m]) between the line start points $X1, Y1$ and the line end points $X2, Y2$, as shown in Figure 7. By applying this operation to all straight lines detected by the Hough transform, the layout map was converted into point cloud data.

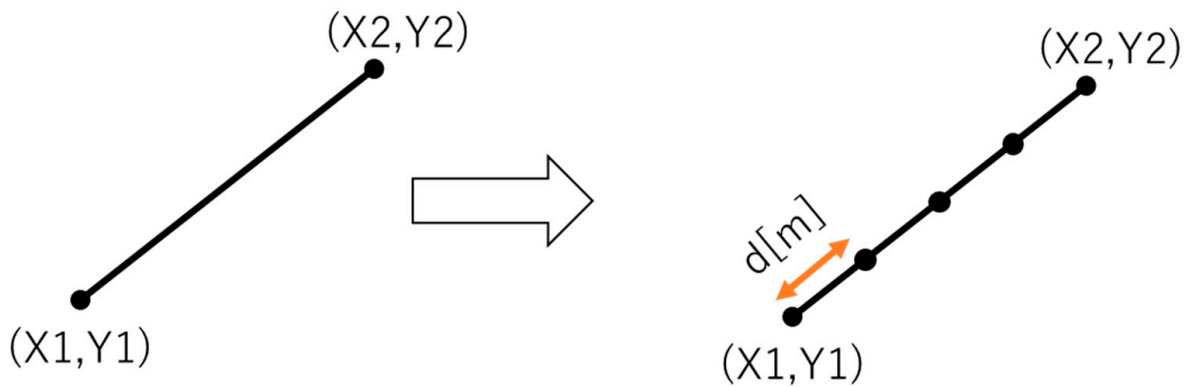


Figure 7. Conversion from a straight line to a point cloud.

A voxel grid filter was used for down-sampling. A voxel grid filter is a method commonly used to reduce point groups. It partitions the space on a grid and removes other point groups to obtain one point group per grid. As a result, the point cloud density becomes uniform, and, at the same time, the number of point clouds can be greatly reduced. The mechanism is shown in Figure 8.

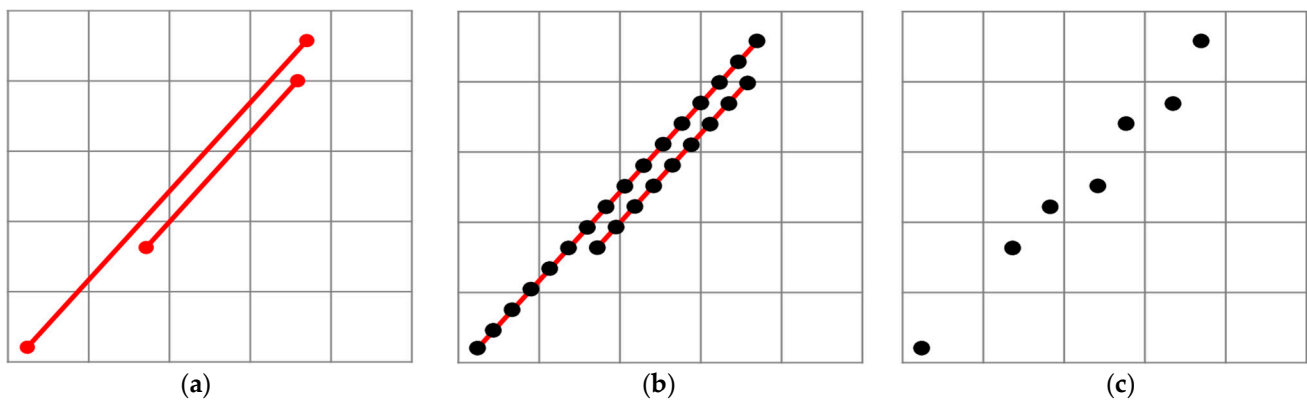


Figure 8. Down-sampling process. (a) After line detection. (b) Convert to point cloud. (c) After down sampling.

From the figure, it is expected that multiple lines will be detected by the Hough transform algorithm. In Figure 8b, the algorithm generated more than 30 point clouds. By using down-sampling, a 75% reduction in points is achieved, and the challenge of overlapping straight lines is overcome.

The final point cloud map data created by these processes are shown in Figure 9. The resultant point cloud map is then integrated with other segments to produce a refined self-localization estimation.

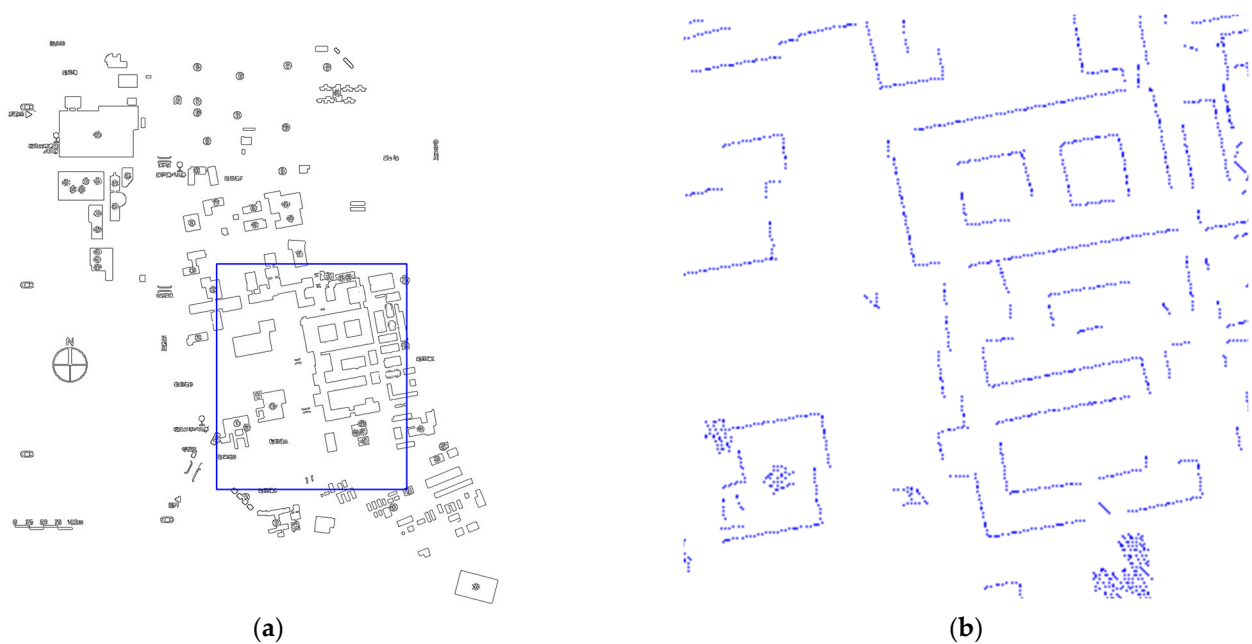


Figure 9. Layout map after conversion. (a) Original map. (b) Zoomed out region.

3.2. Layout Map Scale Determination from Aerial Photographs

In the process of converting the layout map to point cloud map data, it is crucial to scale it accurately to align with the data obtained from 3DLidar measurements. Since acquiring distance information directly from image data is challenging, a separate method for determining the scale of the layout map is required. This study assumes that the aspect ratio of the layout map corresponds to that of the actual structure. The scale of the layout map was calculated based on the scale of the aerial photograph. Additionally, an interface was developed to facilitate this operation.

3.2.1. Superimposition of Aerial Photography with Layout Map

The interface for determining the scale of the layout map comprises three main functionalities: acquiring and saving aerial photographs, adjusting the position of layout maps and aerial photographs, and saving parameters. Figure 10 shows a sample of the downloaded image of Gifu University.



Figure 10. Downloaded aerial photograph of Gifu University [31].

The downloaded image is superimposed on the point cloud generated previously. The result of the alignment adjustment is depicted in Figure 11.

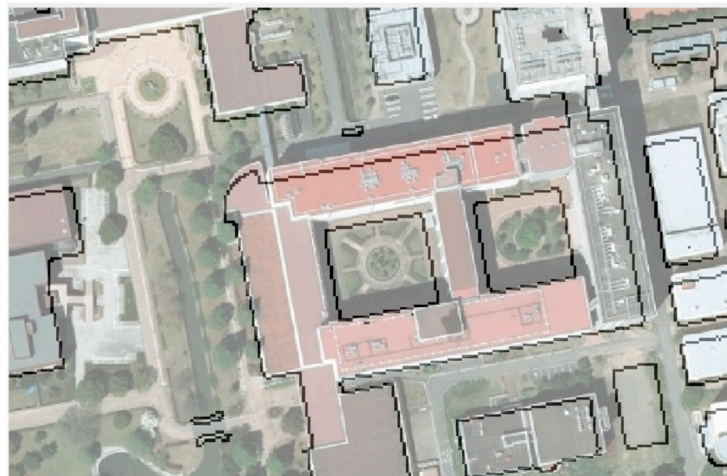


Figure 11. Results after superimposition of aerial photograph with layout map.

Since the layout map and the shape of the building in the aerial photograph match, it can be said that it is possible to superimpose the two images.

3.2.2. Calculation of Actual Distance from Aerial Photographs

Aerial photographs, provided by sources such as the Geospatial Information Authority of Japan, Google Maps, and Yahoo Maps, represent maps through arrangements of 256×256 pixel image data called tile images. These tile images, with different zoom levels, are lined up during scaling.

Calculation of Distance on the Tile Image: In the calculation, tile coordinates are converted to latitude and longitude, and the distance is determined using Equations (5) and (6) for latitude (φ) and longitude (λ).

$$\lambda = 180 \left(\frac{x}{2^{z-1}} - 1 \right) \quad (5)$$

$$\varphi = \frac{180}{\pi} \left(\sin^{-1} \left(\tanh \left(-\frac{\pi}{2^{z-1}} y + \tanh^{-1} \left(\sin \left(\frac{\pi}{180} L \right) \right) \right) \right) \right) \quad (6)$$

where $L = 85.05112878$.

To calculate the distance between two points on the Earth's surface based on their latitude and longitude coordinates, we use Equation (7).

$$d = r \cdot \cos^{-1}(\sin\phi_1 \cdot \sin\phi_2 + \cos\phi_1 \cdot \cos\phi_2 \cdot \cos(\lambda_2 - \lambda_1)) \quad (7)$$

where:

d is the distance between the two points.

r is the radius of the Earth (approximately 6378.13 km).

ϕ_1 and ϕ_2 are the latitudes of the two points.

λ_1 and λ_2 are the longitudes of the two points.

As an example, let us consider the tile coordinates that include a specific point, calculated from the latitude and longitude of the courtyard of the Gifu University engineering building, with coordinates $\phi = 35.464478$, $\phi = 35.464478$ and $\lambda = 136.739235$, $\lambda = 136.739235$. We can use these coordinates to determine the distance between the upper-right and lower-right corners of the image. Using the provided latitude ($\phi = 35.464478$) and longitude ($\lambda = 136.739235$), we can calculate the tile coordinates including the specified point as follows:

For x ,

$$x = 2^{z-1} \left(\frac{\lambda}{180} + 1 \right) = 2^{18-1} \left(\frac{35.464478}{180} + 1 \right) \approx 230642.4$$

For y ,

$$y = \frac{2^{z-1}}{\pi} \left(-\tanh^{-1} \left(\sin \left(\frac{\pi}{180} \varphi \right) \right) + \tanh^{-1} \left(\sin \left(\frac{\pi}{180} L \right) \right) \right) \\ = 41721.51 \dots \times (-0.6627613 \dots + 3.141592) \approx 103420.595$$

Therefore, the tile coordinates including the specified point are approximately $(x, y) = (230,642, 103,420)$. Finally, the distance is obtained from Equation (7) as:

$$d = 0.1245 \text{ [km]} \quad (8)$$

From the above, the length of one side of the tile image near the university at zoom level 18 was found to be about 124.5 [m]. Since this number varies depending on the latitude, it is calculated each time.

3.2.3. Scale Parameter Calculation

The target is to calculate the actual distance [m] of 1 [pixel] of the layout map from the result of superimposing the aerial photograph and the layout map. Here, we verify the calculation method and whether the distance is actually calculated correctly.

In Figure 11, the magnification of the layout map is $Z_m = 3.3$, the magnification of the aerial photograph is $Z_p = 1.67$, the length of one side of the tile image is $L_t = 124.5$ [m], and the resolution of the tile image is $R_t = 256$ [pixel]. The length U of 1 [pixel] of the layout map image is expressed by Equation (9). Note that the display at 100 [%] is 1.0.

$$U = \frac{L_t}{R_t \times Z_p} \times Z_m = \frac{124.5}{256 \times 1.67} \times 3.3 \\ = 0.9610 \text{ [m/pixel]} \quad (9)$$

The calculation result shows that the length of 1 [pixel] of the layout map is about 0.96 [m]. Next, we verify whether the calculated value is correct. In this case, verification was performed based on the length of the building shown in Figure 12. The length of the building on the layout map was 72.4 [pixel].

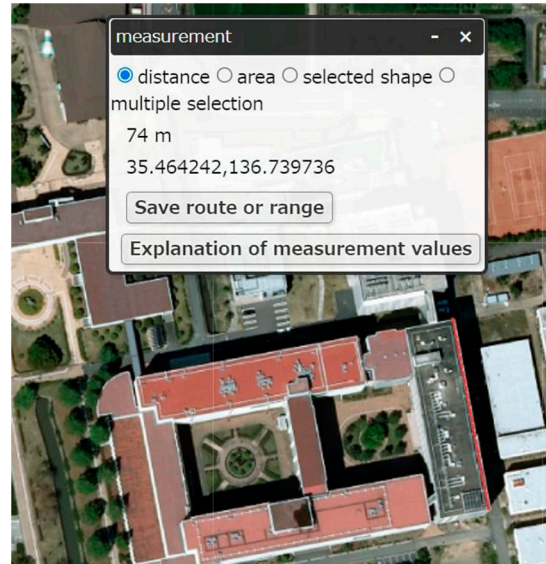


Figure 12. Building length comparison [32].

From this result, as a result of calculating using the interface, the length of the factory is $0.9610 \text{ [m/pixel]} \times 72.4 \text{ [pixel]} = 69.6 \text{ [m]}$. As shown in Figure 12, the length measured from the map of the Geospatial Information Authority of Japan is 74 [m], so there is an error of 5 [m]. Although the accuracy of the layout map data used this time was low due to its low resolution, it was confirmed that the scale of the layout map could be calculated within an error of 1 [%] for the actual factory layout map.

3.3. Route Management

This section describes how to create route data. The route data are in the form of connecting points within lines, as shown in Figure 13 below.

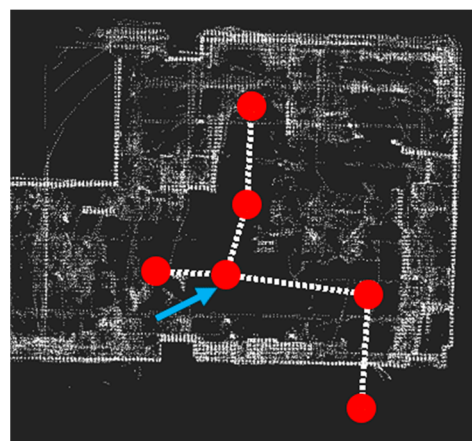


Figure 13. Sample path data. The red dots represent decision points on the trajectory.

The data consist of two CSV files (Tables 2 and 3) which describe the relationship between points and “path.csv”, which describes the position of each point. Sample CSV data are shown in Table 2.

Table 2. Sample CSV “tree” data.

ID	EDA
0	1
1	0 2 3
2	1 3
3	1 4
4	3
5	2

Table 3. Sample Data of “path.csv”.

ID	x	y
0	0	0
1	1.8	−0.1
2	5.46	−0.61
3	2.34	1.7
4	2.46	4.22
5	5.28	−3.52

From the CSV file, it is the case that the point with ID = 1 is connected to (x, y) = (1.8, −0.1) from “path.csv” and ID = 0, 2, 3 from “tree.csv”. Based on this information, route data are displayed as shown in Figure 14.

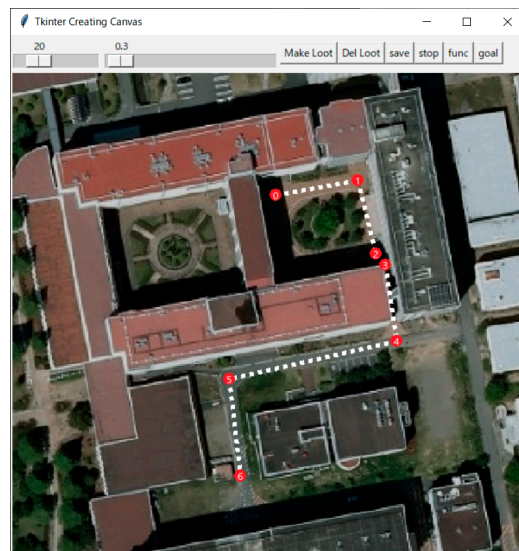


Figure 14. Interface for creating route data. The route is created by adding line segments as indicated. The numbers in red circles indicate the search order of vertices and branchpoints.

3.3.1. Route Search

The breadth-first search (BFS) algorithm is utilized to search for routes within the system, guaranteeing the discovery of the shortest path from the initial point to the target destination. This algorithm systematically explores all neighbour nodes at the present depth level before moving on to nodes at the next depth level. It operates by traversing the graph or tree level by level, starting from the initial node and moving outward. BFS ensures optimality by prioritizing nodes closest to the starting point, making it suitable for route planning in autonomous vehicles.

3.3.2. Creation of Route Tracking Data

The process of generating route tracking data for autonomous movement involves several steps. Initially, the position coordinates (x, y) of each ID are extracted sequentially from the "path.csv" file based on the route obtained from the route search. Subsequently, additional points are interpolated at specified intervals between each pair of consecutive points along the route. This interpolation increases the density of points along the route, providing a more detailed representation of the path from the starting point to the destination. The resulting position information for all points is then saved in a CSV file format as described in Figure 15. This transformation into an equidistant point sequence streamlines the calculation process and facilitates the implementation of route-following and obstacle-avoidance algorithms, which will be elaborated upon later.

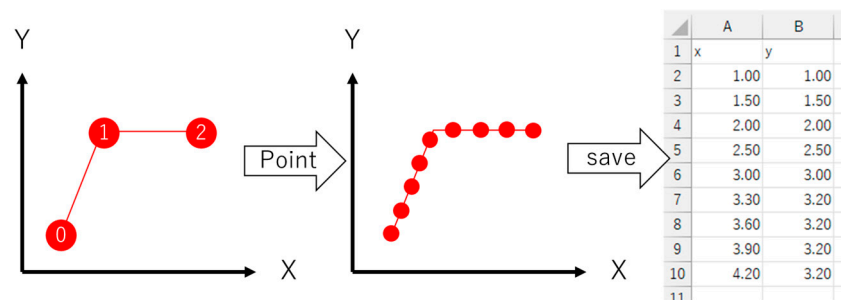


Figure 15. Path data creation. The numbers in red circles indicate the search order of vertices and branchpoints.

3.3.3. Path-Following Algorithm

The pursuit algorithm is employed for path tracking. This algorithm computes the angular velocity required to steer the robot from its current position towards a designated point of focus located ahead of it. A flowchart of the pursuit algorithm is shown in Figure 16.

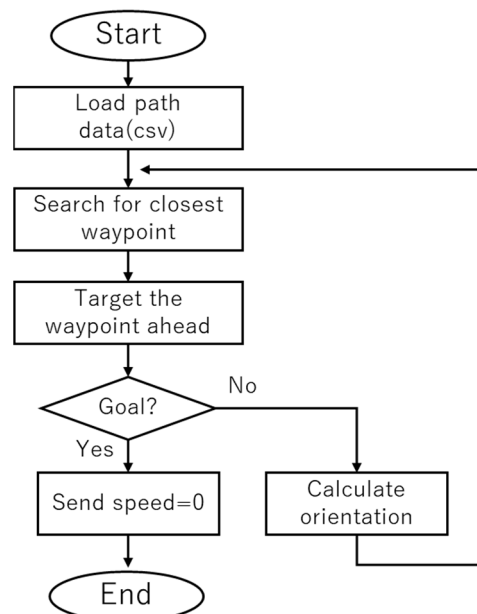


Figure 16. Flowchart of the path tracking algorithm.

The target is to follow a waypoint that is a specified distance ahead of the searched waypoint. Next, we calculate the velocities and angular velocities that can reach this target. By repeating this process, the robot always aims at a virtual target that exists in front of it, so autonomous movement along the route is possible. If there is no route data for the

designated distance, it judges that it has reached the destination and stops autonomous movement.

3.4. Obstacle Avoidance Algorithm

Obstacle avoidance encompasses two main components: obstacle detection and route adjustment in the presence of obstacles. Illustrated below is a scenario where a straight path is obstructed by an obstacle on the left side. The route data comprises multiple waypoints, each surrounded by a detection area. This detection area, depicted as a cylinder around the waypoint, identifies obstacles. When point clouds are detected within this area, excluding those representing the ground, the path is marked as impassable, as depicted in Figure 17.

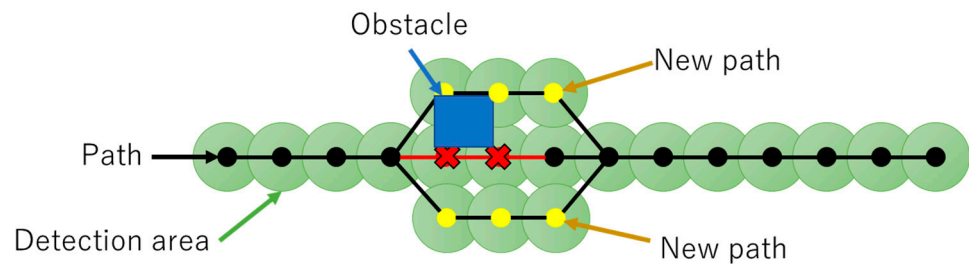


Figure 17. Image of obstacle avoidance. The red lines indicate a blocked path while the yellow dots represent the new path to be investigated.

The obstacle avoidance process involves identifying impassable waypoints marked in red and generating alternative paths on either side of the obstructed route. Utilizing an obstacle detection algorithm, the system evaluates the viability of these alternate paths. If one side is clear, it becomes the chosen avoidance route, communicated to the route-following algorithm for execution. In cases where obstacles block both sides, indicating an impassable barrier, the vehicle halts until a clear path is identified. This adaptive strategy enables safe navigation around obstacles, accommodating dynamic environments.

4. Verification and Discussion

In this section, the focus lies on the verification of the created layout map intended for the self-location estimation, alongside the validation of system performance. The proposed system was validated against the existing 3D-SLAM Autoware version 1.14.0. Autoware is open-source software designed for autonomous driving systems, operating on Linux ROS [33]. Normal Distribution Transform (NDT) Matching, utilized within Autoware, is a conventional 3D-SLAM method based on point cloud scan matching. It involves representing point clouds using normal distributions within defined regions. This representation enables efficient and accurate registration of overlapping point clouds, facilitating precise determination of a robot or vehicle's position within its environment. NDT matching is particularly advantageous in applications requiring high levels of localization accuracy, such as autonomous navigation systems. Further details can be obtained in a paper reported in [34].

The experiment involved the utilization of data acquired by manually relocating a mobile trolley and subsequently measuring its position to verify self-position estimation. This data collection process encompassed various measurement locations, both within the confines of the factory premises and in outdoor spaces between different factory buildings. In addition, we verified the proposed self-localization method through ROS simulation utilizing the measured data.

4.1. Self-Position Estimation

The experimental results are presented below, where the point cloud map derived from the layout map is depicted as white dots, providing a spatial representation of the environment. The trajectory of the proposed self-localization system, represented by

red lines, demonstrates the path estimated by our method based on the gathered data. Conversely, the trajectory generated by the conventional Autoware NDT Matching method, shown in blue lines, serves as a benchmark for comparison.

From Figure 18a,b, for indoor and outdoor settings, it becomes apparent that the trajectories of both the proposed self-localization system and the conventional Autoware NDT Matching method closely align. This alignment signifies the successful self-position estimation achieved by our method, as it concurs with the trajectory derived from the established Autoware method. Such congruence underscores the accuracy and reliability of our proposed self-localization approach.

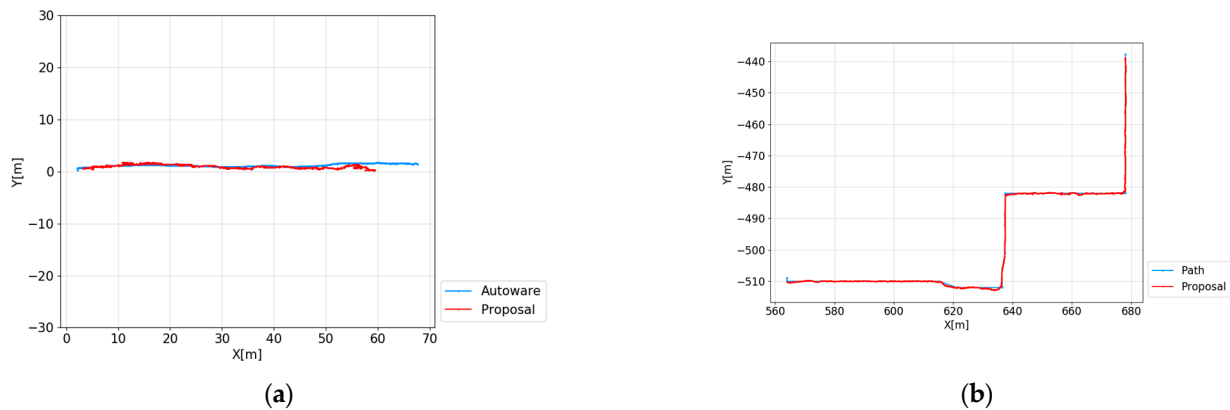


Figure 18. Comparison of self-position estimation (a) inside the factory and (b) outside the factory.

Within the factory premises, as seen in Figure 18a, our system exhibited a comparable performance, with an average horizontal position accuracy of 1.3 m and a std 1.5 m. Moreover, the angle error averaged at 1.1 degrees, with a maximum error of 3.5 degrees, highlighting the consistent and precise self-localization capabilities of our method even in complex indoor settings as shown in Figure 19.

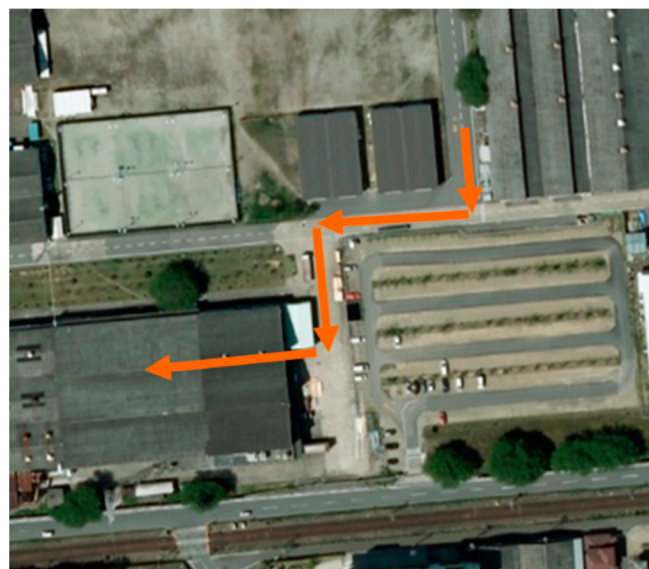


Figure 19. Verification route. The arrows represent the driving route used for verification.

Figure 18b, which shows mapping between factories, has a position accuracy averaging at 1.3 m and a std of 2.2 m. This underscores the reliability of our system in outdoor environments. Additionally, the angle error averaged at 1.8 degrees, showing the robustness of our method in maintaining orientation accuracy in varying conditions.

The achieved accuracies are typical for what is reported in the literature. Researchers in [35] reported the absolute error from visual SLAM ranging between 0.002–4 m, depending on the Lidar equipment used, the correction algorithm in place, as well as the environment under consideration (outdoor/indoor). Another paper by [36] utilized similar devices (Velodyne) and programmatic 3D reconstruction in an outdoor environment. The results showed a poor localization before the engagement of the loop closure algorithm. In essence, loop closure requires that the robot circle back to the start before the final map can be generated.

A study by [37] presented a SLAM for 3DLidar-based online mapping. The study reported its generated output in map entropy, so it is hard to compare the performance. Notably, the authors utilized an iterative process to home in on a final map.

In general, studies about mobile (online) map generators are not easily compared since the environment is dynamic in nature. As such, one of the schemes of comparison is a visual inspection of the resultant cloud points [37]. Arguably, our mapping system achieved comparable or superior performance from what is currently reported.

4.2. Autonomous Movement

For the autonomous movement verification, the path used in the prior experiment is duplicated. The path began from the part transfer position in the factory, passing through the automatic door at the factory entrance and going to another factory entrance. The route for the autonomous movement verification is shown in the trajectory map in Figure 19. The use of this technique guaranteed consistency in comparison analysis with earlier experimental results.

The route was chosen based on several reasons, including its portrayal of real-world events found in industrial settings and its usefulness for assessing the autonomous movement system's performance under various conditions. It also enabled an assessment of the system's capacity to navigate through various environmental elements and barriers often seen in production environments.

In Figure 20, the route planned for autonomous movement is the blue line and the position determined by the suggested self-localization approach is the red line. Remarkably, the trajectory showed effective independent navigation from the starting point to the target with no notable deviations. This observation demonstrates the self-localization method's efficiency in properly directing the autonomous mobility system along a predefined course.

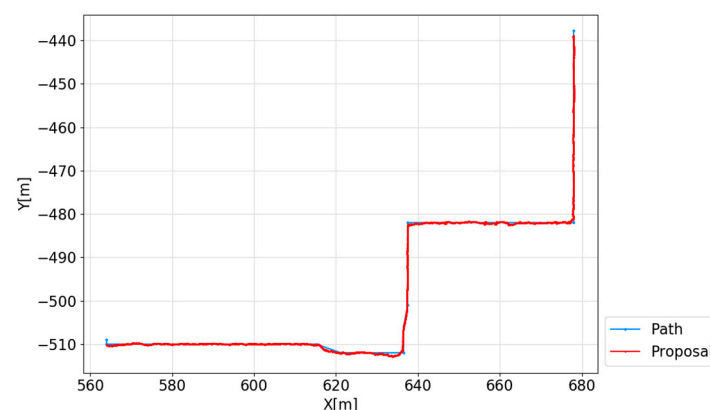


Figure 20. Trajectory of autonomous movement.

Furthermore, the alignment of the trajectory with the defined route, as shown in Figure 20, demonstrated the system's capacity to move independently along the desired path. This alignment validates the self-localization method's precision and dependability in enabling accurate and consistent autonomous mobility in complicated industrial contexts.

The graph illustrating the distance of the running position (self-position) from the specified route during autonomous movement, as seen in Figure 21, provides valuable

insights into the system's performance. Despite minor deviations, autonomous movement was achieved with an average distance of 0.12 m from the specified route, with a maximum deviation of 0.85 m. These results indicate a high degree of accuracy and precision in the system's navigation capabilities, particularly in maintaining proximity to the intended path during operation.

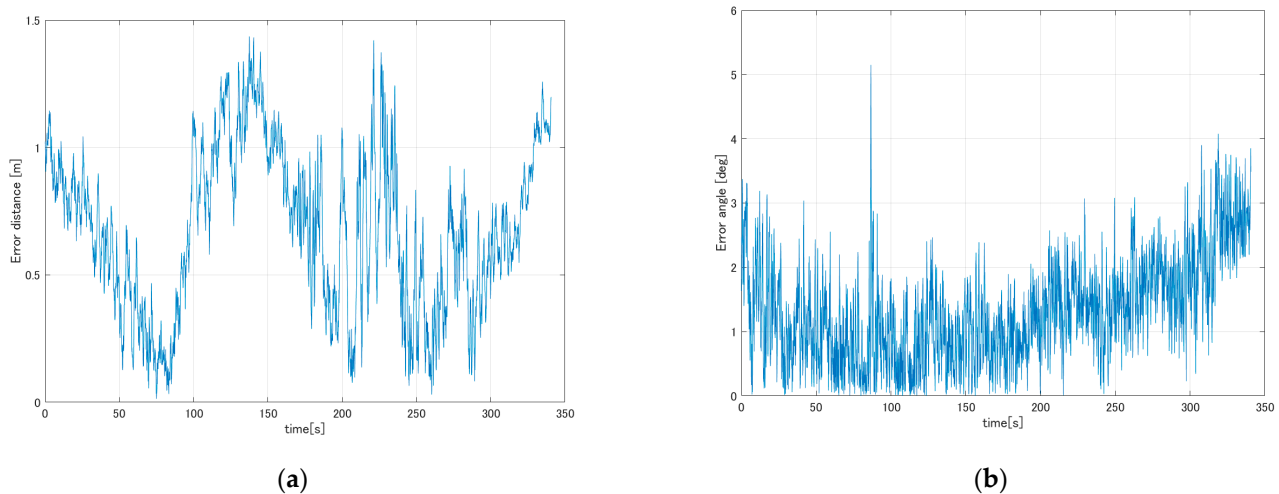


Figure 21. Comparison of estimated positions. (a) Distance error. (b) Angle error.

The effective execution of the suggested self-localization approach using the layout map allowed confirmation of the viability of autonomous mobility. The technology exhibited the capacity to move independently along the defined path with an impressive average variation of only 0.12 m, as seen in Figure 21a. This is a tremendous feat, especially given the complexity and dynamic nature of industrial contexts.

The accuracy of self-position estimation during autonomous movement significantly improved, with the average error decreasing from 1.3 m in the original trial to 0.68 m, as shown in Figure 21b. This improvement, which is roughly half the prior inaccuracy, indicates a significant refinement in the system's capacity to precisely predict its position in real time.

The reduction in self-position estimate error not only improves the dependability of autonomous movement but also leads to better safety and efficiency in industrial processes. With more precise localization, the system can better respond to environmental changes and unexpected barriers, lowering the chance of crashes and increasing navigation efficiency.

Overall, these findings represent a substantial step forward in the development of autonomous navigation systems, opening a path for their broad use in a variety of industrial applications. Moving forward, further development and optimisation of the self-localization approach has the potential to achieve even higher levels of precision and dependability, eventually improving the performance and usefulness of autonomous movement systems in real-world settings.

During the autonomous movement verification, different barriers such as lifts and sensor-type automated doors were encountered throughout the defined route. To guarantee safe navigation, an obstacle detection and avoidance feature was included, essentially halting autonomous travel until impediments were cleared. Specifically, when meeting a lift door, the algorithm properly identified it as an impediment and momentarily paused travel. As a result, even as the door was opening, the system determined that the guided vehicle's height clearance was insufficient, necessitating another brief stop in travel. After completely opening the lift door and verifying the absence of obstacles, autonomous movement began, allowing the car to enter the lift. This complete functionality validation illustrates the system's capacity to respond to a variety of scenario difficulties while maintaining safe and efficient autonomous mobility.

Despite possible differences between the present layout map and recent modifications to building structures, such as freshly erected prefabricated buildings, the system demonstrated robust self-position estimates. This stability was due to the system's capacity to reliably determine self-position from the surrounding environment, even when the layout map did not perfectly coincide with the real surroundings. Thus, the verification highlights the system's durability and flexibility, confirming its suitability for autonomous mobility in real-world circumstances.

A potential avenue for future research to enhance the path planning and map generation strategy proposed in this paper could involve focusing specifically on refining the loop closing mechanism, as adopted by other methods of online map generation from SLAM in the literature [10,37,38]. By developing advanced algorithms that leverage sensor data to detect loop closure opportunities more accurately and efficiently, researchers could improve the system's ability to recognize previously visited locations and ensure global consistency of the map and robot trajectory. Additionally, integrating techniques such as feature-based mapping and graph-based SLAM approaches could further enhance the robustness and reliability of the loop closing mechanism. Moreover, exploring adaptive loop closure thresholds and real-time loop correction mechanisms could offer flexibility and adaptability to varying environmental conditions, ultimately leading to more precise and efficient navigation for autonomous mobile robots.

5. Conclusions

The proposed 3D SLAM autonomous mobile robot system effectively addressed labour shortages in production and distribution settings by utilizing existing layout maps, eliminating the need for manual map generation. By integrating advanced self-localization methods and a novel SLAM approach leveraging 3DLidar data and layout map information, the system achieved precise self-positioning and navigation along random paths with remarkable accuracy.

The experimental validation revealed notable reductions in self-positioning errors, with a mean deviation of 0.12 m from the intended path, showcasing the system's precision and accuracy in autonomous navigation. Additionally, consistent autonomous movement was observed even in scenarios with slight variations in layout maps and obstacles, underscoring the system's reliability in real-world conditions.

Featuring functionalities such as route planning, tracking, and obstacle detection, the system demonstrated agile and adaptable navigation in dynamic environments. Experimental validation confirmed significant reductions in self-positioning errors and consistent autonomous movement, highlighting its reliability and precision in real-world scenarios.

Future efforts will focus on enhancing the system's robustness through extensive validation across diverse production environments and further improving its capabilities for proactive navigation and obstacle avoidance.

In summary, this study's findings lay the groundwork for the development of sophisticated autonomous mobile systems capable of navigating intricate environments efficiently and precisely, thereby addressing current labour challenges in production and distribution operations.

Author Contributions: Conceptualization, M.S. and K.M.; Methodology, M.S.; software, Y.T.; validation, M.S., Y.T. and K.M.; formal analysis, Y.T.; investigation, Y.T.; Resources, Y.T.; data curation, Y.T.; writing—original draft preparation, M.S. and Y.T.; writing—review and editing, M.S.; visualization, Y.T.; supervision, M.S.; project administration, M.S.; funding acquisition, M.S. and K.M. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by Grants-in-aid for the Promotion of Regional Industry–University–Government Collaboration from the Cabinet Office, Japan.

Data Availability Statement: All data underlying the results are available as part of the article and no additional source data are required.

Acknowledgments: The authors would like to thank Joseph Muguro for useful discussions and carefully proofreading the manuscript.

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

References

1. Kiva. Amazon Robotics Kiva, Robostar Robot Database. 2018. Available online: <https://robotstart.info/robot-database/kiva> (accessed on 3 February 2022).
2. OMRON. OMRON Mobile Robot. 2018. Available online: <https://www.fa.omron.co.jp/product/robotics/lineup/mobile/feature/> (accessed on 3 February 2022).
3. JIVA, Japan Industrial Vehicles Association (JIVA). We Will Announce the Delivery Results of Automatic Guided Vehicle Systems in 2020. 2019. Available online: <http://www.jiva.or.jp/pdf/AGV-Stat-2020.pdf> (accessed on 3 February 2022).
4. Grisetti, G.; Stachniss, C.; Burgard, W. Improved Techniques for Grid Mapping With Rao-Blackwellized Particle Filters. *IEEE Trans. Robot.* **2007**, *23*, 34–46. [[CrossRef](#)]
5. Farina, M.; Shaker, W.K.; Ali, A.M.; Hussein, S.A.; Dalang, F.S.; Bassey, J.O. Automated guided vehicles with a mounted serial manipulator: A systematic literature review. *Heliyon* **2023**, *9*, e15950. [[CrossRef](#)]
6. Racinkis, P.; Arents, J.; Greitans, M. Constructing Maps for Autonomous Robotics: An Introductory Conceptual Overview. *Electronics* **2023**, *12*, 2925. [[CrossRef](#)]
7. Yu, Y.; Zhu, K.; Yu, W. YG-SLAM: GPU-Accelerated RGBD-SLAM Using YOLOv5 in a Dynamic Environment. *Electronics* **2023**, *12*, 4377. [[CrossRef](#)]
8. Roomba, iRobot Official HP Roomba 900 Series. 2020. Available online: <https://www.irobot-jp.com/product/900series/> (accessed on 3 February 2022).
9. Takita, Y.; Ohkawa, S.; Date, H. Recognition Method Applied to Smart Dump 9 Using Multi-Beam 3D LiDAR for the Tsukuba Challenge. *J. Robot. Mechatron.* **2016**, *28*, 451–460. [[CrossRef](#)]
10. Mihálik, M.; Malobický, B.; Peniak, P.; Vestenický, P. The New Method of Active SLAM for Mapping Using LiDAR. *Electronics* **2022**, *11*, 1082. [[CrossRef](#)]
11. Duan, C.; Junginger, S.; Huang, J.; Jin, K.; Thurow, K. Deep Learning for Visual SLAM in Transportation Robotics: A review. *Transp. Saf. Environ.* **2019**, *1*, 177–184. [[CrossRef](#)]
12. Cheng, J.; Zhang, L.; Chen, Q.; Hu, X.; Cai, J. A review of visual SLAM methods for autonomous driving vehicles. *Eng. Appl. Artif. Intell.* **2022**, *114*, 104992. [[CrossRef](#)]
13. Sasaki, M.; Kunii, E.; Uda, T.; Matsushita, K.; Muguro, J.K.; bin Suhaimi, M.S.A.; Njeri, W. Construction of an Environmental Map including Road Surface Classification Based on a Coaxial Two-Wheeled Robot. *J. Sustain. Res. Eng.* **2020**, *5*, 159–169.
14. Liu, L.; Guo, J.; Zhang, R. YKP-SLAM: A Visual SLAM Based on Static Probability Update Strategy for Dynamic Environments. *Electronics* **2022**, *11*, 2872. [[CrossRef](#)]
15. Shan, D.; Su, J.; Wang, X.; Liu, Y.; Zhou, T.; Wu, Z. VID-SLAM: Robust Pose Estimation with RGBD-Inertial Input for Indoor Robotic Localization. *Electronics* **2024**, *13*, 318. [[CrossRef](#)]
16. Ai, Y.; Sun, Q.; Xi, Z.; Li, N.; Dong, J.; Wang, X. Stereo SLAM in Dynamic Environments Using Semantic Segmentation. *Electronics* **2023**, *12*, 3112. [[CrossRef](#)]
17. Song, Z.; Su, W.; Chen, H.; Feng, M.; Peng, J.; Zhang, A. VSLAM Optimization Method in Dynamic Scenes Based on YOLO-Fastest. *Electronics* **2023**, *12*, 3538. [[CrossRef](#)]
18. Wang, X.; Zhang, X. MCBM-SLAM: An Improved Mask-Region-Convolutional Neural Network-Based Simultaneous Localization and Mapping System for Dynamic Environments. *Electronics* **2023**, *12*, 3596. [[CrossRef](#)]
19. Azpúrua, H.; Saboia, M.; Freitas, G.M.; Clark, L.; Agha-mohammadi, A.A.; Pessin, G.; Campos, M.F.; Macharet, D.G. A Survey on the autonomous exploration of confined subterranean spaces: Perspectives from real-word and industrial robotic deployments. *Rob. Auton. Syst.* **2023**, *160*, 104304. [[CrossRef](#)]
20. Guan, M.; Wen, C. Autonomous exploration using UWB and LiDAR. *J. Autom. Intell.* **2023**, *2*, 51–60. [[CrossRef](#)]
21. Zhao, X.; Cheah, C.C. BIM-based indoor mobile robot initialization for construction automation using object detection. *Autom. Constr.* **2023**, *146*, 104647. [[CrossRef](#)]
22. Harsh, A.; Dhama, S.S. Development of mobile smart material-handling system. *Mater. Today Proc.* **2023**. [[CrossRef](#)]
23. Ali, I.; Durmush, A.; Suominen, O.; Yli-Hietanen, J.; Peltonen, S.; Collin, J.; Gotchev, A. FinnForest dataset: A forest landscape for visual SLAM. *Rob. Auton. Syst.* **2020**, *132*, 103610. [[CrossRef](#)]
24. Zhou, Z.; Li, L.; Fürsterling, A.; Durocher, H.J.; Mouridsen, J.; Zhang, X. Learning-based object detection and localization for a mobile robot manipulator in SME production. *Robot. Comput. Integr. Manuf.* **2022**, *73*, 102229. [[CrossRef](#)]
25. Luperto, M.; Amadelli, F.; Di Bernardino, M.; Amigoni, F. Mapping beyond what you can see: Predicting the layout of rooms behind closed doors. *Rob. Auton. Syst.* **2023**, *159*, 104282. [[CrossRef](#)]
26. Fragapane, G.; de Koster, R.; Sgarbossa, F.; Strandhagen, J.O. Planning and control of autonomous mobile robots for intralogistics: Literature review and research agenda. *Eur. J. Oper. Res.* **2021**, *294*, 405–426. [[CrossRef](#)]
27. Adán, A.; López-Rey, A.; Ramón, A. Robot for thermal monitoring of buildings. *Autom. Constr.* **2023**, *154*, 105009. [[CrossRef](#)]

28. Park, S.; Kim, S.; Park, M.; Park, S.-K. Vision-based global localization for mobile robots with hybrid maps of objects and spatial layouts. *Inf. Sci.* **2009**, *179*, 4174–4198. [[CrossRef](#)]
29. Ma, T.; Bai, N.; Shi, W.; Wu, X.; Wang, L.; Wu, T.; Zhao, C. Research on the Application of Visual SLAM in Embedded GPU. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 1–17. [[CrossRef](#)]
30. Mao, Y.; Zhu, Y.; Tang, Z.; Chen, Z. A Novel Airspace Planning Algorithm for Cooperative Target Localization. *Electronics* **2022**, *11*, 2950. [[CrossRef](#)]
31. GIAJ, Geospatial Information Authority of Japan (GIAJ), ‘About Geospatial Information Authority of Japan Tiles,’ Ministry of Land, Infrastructure, Transport and Tourism. Available online: <https://maps.gsi.go.jp/development/siyou.html> (accessed on 3 February 2022).
32. TrialNote, Convert Coordinates (World Coordinates, Pixel Coordinates, Tile Coordinates, Latitude/Longitude). Available online: <https://www.trail-note.net/tech/coordinate/> (accessed on 3 February 2022).
33. utoware, Autonomous Driving Software: Autoware. Parallel and Distributed Systems Lab. Available online: <https://www.pdsl.jp/fot/autoware> (accessed on 3 February 2022).
34. Biber, P.; Strasser, W. The normal distributions transform: A new approach to laser scan matching. In Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS 2003) (Cat. No.03CH37453), Las Vegas, NV, USA, 27–31 October 2003; Volume 3, pp. 2743–2748. [[CrossRef](#)]
35. Handa, A.; Whelan, T.; McDonald, J.; Davison, A.J. A benchmark for RGB-D visual odometry, 3D reconstruction and SLAM. In Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China, 31 May–7 June 2014; pp. 1524–1531. [[CrossRef](#)]
36. Pang, C.; Tan, Y.; Li, S.; Li, Y.; Ji, B.; Song, R. Low-cost and High-accuracy LIDAR SLAM for Large Outdoor Scenarios. In Proceedings of the 2019 IEEE International Conference on Real-Time Computing and Robotics (RCAR), Irkutsk, Russia, 4–9 August 2019; pp. 868–873. [[CrossRef](#)]
37. Droschel, D.; Behnke, S. Efficient Continuous-Time SLAM for 3D Lidar-Based Online Mapping. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, Australia, 21–25 May 2018; pp. 5000–5007. [[CrossRef](#)]
38. Junaedy, A.; Masuta, H.; Sawai, K.; Motoyoshi, T.; Takagi, N. Real-Time 3D Map Building in a Mobile Robot System with Low-Bandwidth Communication. *Robotics* **2023**, *12*, 157. [[CrossRef](#)]

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.