sensors

MDPI

Article

Multi-Robot 2.5D Localization and Mapping Using a Monte Carlo Algorithm on a Multi-Level Surface

Vinicio Alejandro Rosas-Cervantes [1], Quoc-Dong Hoang [1,2], Soon-Geul Lee [1,2,*] and Jae-Hwan Choi [1]

[1] Mechanical Engineering Department, Kyung Hee University, Yongin 17104, Korea; viciorosas@hotmail.es (V.A.R.-C.); hoangquocdong.vimaru@gmail.com (Q.-D.H.); choisida89@gmail.com (J.-H.C.)

[2] Integrated Education Institute for Frontier Science and Technology (BK21 Four), Kyung Hee University, Yongin 17104, Korea

* Correspondence: sglee@khu.ac.kr

Abstract: Most indoor environments have wheelchair adaptations or ramps, providing an opportunity for mobile robots to navigate sloped areas avoiding steps. These indoor environments with integrated sloped areas are divided into different levels. The multi-level areas represent a challenge for mobile robot navigation due to the sudden change in reference sensors as visual, inertial, or laser scan instruments. Using multiple cooperative robots is advantageous for mapping and localization since they permit rapid exploration of the environment and provide higher redundancy than using a single robot. This study proposes a multi-robot localization using two robots (leader and follower) to perform a fast and robust environment exploration on multi-level areas. The leader robot is equipped with a 3D LIDAR for 2.5D mapping and a Kinect camera for RGB image acquisition. Using 3D LIDAR, the leader robot obtains information for particle localization, with particles sampled from the walls and obstacle tangents. We employ a convolutional neural network on the RGB images for multi-level area detection. Once the leader robot detects a multi-level area, it generates a path and sends a notification to the follower robot to go into the detected location. The follower robot utilizes a 2D LIDAR to explore the boundaries of the even areas and generate a 2D map using an extension of the iterative closest point. The 2D map is utilized as a re-localization resource in case of failure of the leader robot.

Keywords: multi-robot; localization; 2.5D mapping; Monte Carlo algorithm; multi-level surface

check for updates

Citation: Rosas-Cervantes, V.A.; Hoang, Q.-D.; Lee, S.-G.; Choi, J.-H. Multi-Robot 2.5D Localization and Mapping Using a Monte Carlo Algorithm on a Multi-Level Surface. *Sensors* **2021**, *21*, 4588. https://doi.org/10.3390/s21134588

Academic Editor: Gabriel Oliver-Codina

Received: 4 April 2021
Accepted: 28 June 2021
Published: 4 July 2021

1. Introduction

For mapping and localization on an uneven or multi-level surface, 3D LIDAR is needed. However, the price of a 3D LIDAR is very high, and the required computing power and resources are also high. A single small robot cannot process both 3D localization and mapping for real-time navigation. Since our previous work [1] studied mapping and localization on uneven surfaces, we decided to extend the experiment using a second robot. The mapping and exploration time is simplified, and the computational time is improved.

The present experiment intends to provide a starting point for a bigger multi-robot team, where the more robots, the faster the mapping task could be performed.

Multi-robot mapping and exploration encounter sparse features in large environments. Alternatives for 2.5D and 3D mappings include 2D–3D feature matching [2] and online point cloud segmentation. Multi-robot systems reduce the mapping time, improve map accuracy and robot localization [3–5]. The iterative closest point (ICP) and the random sample consensus (RANSAC) methods are solutions for point cloud registration, but those approaches have issues with the local minimum. Merging image-key points into the point cloud can also reduce the local minimum error [6]. LIDAR odometry and mapping (LOAM) [7] uses point features and edges-planes for scan registration but lacks loop closure in large-scale maps.

Further, scale mismatch and repeating patterns are common issues in robot mapping. A solution for those challenges involves fixing the coordinate transformations and minimizing the distance in input maps. Optimizing point registration [8,9] and using image registration [10] reduce scan misalignments. However, those methods lack an initial guess and local minima. For multi-robot simultaneous localization and mapping (SLAM), idleness optimizes patrol activities [11], and digital pheromones could provide event relevance [12]. The Hough transform uses rotation and translation to divide the mapping space [13]. Bayesian mapping [14] and open-source data [15] enhance SLAM. Mielle et al. [16] employed SLAM in extreme conditions. In an open space, topological representation and maximal common sub-graphs (MCS) allow for fast robot localization [17]. The vertices contained in metric space updates the robot paths. The Voronoi diagram [18,19], segmented regions matching [20], and environment metrics [21] aid map alignment.

The occupancy grid map plays a vital role in the context of multi-robot map representation. The said map upgrades each cell using the inverse sensor model. Multi-robot mapping and exploration have already been addressed in several research approaches [22,23]. Subsequently, the inverse sensor model evaluates the laser scan as a vector. Graph matching aligns the occupancy grid with floor maps [20]. Kaufman et al. [24] measure 2D distances using laser scan ray casting. The occupancy grid and a Monte Carlo algorithm calculate the 2D pose uncertainty [25]. Gaussian modeling has applications for the occupancy grid [26], mapping, and point cloud modeling [27]. A learning trial [28] and a Bayesian occupancy map also employ an inverse sensor model [29,30]. Despite the robustness of grid-based map representation, this method depends on the inverse sensor model and is sensitive to simulation errors.

In a volumetric representation of 3D space, map simplification is critical for robot mapping. Approaches such as the adaptive online method and voxel compression divide the map [31,32]. Octomap reduces the mapping framework [33,34], and a robot-centric grid enhances the map resolution [35]. Furthermore, communication enhancement is critical for multi-robot systems. Cieslewski et al. [36] propose a decentralized visual SLAM by encoding the environment information to reduce communication requirements. A coordinated multi-robot exploration under connectivity constraints is presented in [37], where each robot keeps connectivity with the teammates. Smith et al. [38] use distributed inference-based communication for a 3D space. The main problem in centralized and decentralized multi-robot exploration is map merging and robot localization. [4,13]. To establish proper communication, we utilize a decentralized system after considering the approaches in the recent years.

Deep learning and machine learning have become trends for robot mapping and localization in the last few years. Through trial and error, robots can adjust their behavior in a dynamic environment. In a multi-robot real-time obstacle avoidance [39], the authors propose a continuous domain detection for obstacle avoidance. An actor-critic component with specific training allows for obstacle avoidance on multi-robot systems [40]. Those techniques entail cooperative interactions. For object detection, methods as You only look once (YOLO) and SSD [41,42] use a single convolutional neural network (CNN) to detect the target position and properties. Approaches such as those in [43,44] use faster reinforce convolutional neural network (Faster R-CNN) for object detection. Our work also employs Faster R-CNN for object detection because of the speed detection accuracy under our experimental conditions.

Besides the limitations of mapping, localization, scan alignment, and map representation, multiple robot systems must adapt to new architectural environments. Nowadays, slopes commonly occur to facilitate transit from different areas in houses, warehouses, and buildings. Thus, having a robust localization in multi-level space is critical. Multi-level areas represent a challenging scenario wherein a mobile robot may lose balance, and 2D LIDAR information provides a vague reference for localization.

Multi-robot systems offer robust and fast field coverage [45]. There is limited work focused on autonomous navigation in sloped and unstructured interior areas, particularly in narrow slopes and crowded spaces. Input from multiple robots renders the system more fault-tolerant than its counterparts. The overlapping of multiple information compensates for sensor uncertainties. Nevertheless, multi-robot systems have limitations in exploring uneven spaces because of the sudden change of flat surfaces [46]. There is extensive work for single and multi-robot systems on 2D flat surfaces. However, robot localization is significantly affected when the robots need to explore areas provided with multiple levels. Thus, the major problem for a multi-robot system involves identifying and finding solutions for multi-robot localization. Multi-robot navigation should be capable of localizing the robot, distinguishing slopes from a staircase, and ascertaining a safe path.

This work proposes a robot framework for cooperative robot navigation in indoor environments with multi-level surfaces. We used two robots to validate the proposed method. The leader robot explores the uneven areas of the multi-level access using a Faster R-CNN to detect indoor ramps. By contrast, the follower robot examines the boundaries of the even areas using 2D LIDAR. We propose a novel 2.5D mapping approach to generate a 2.5D map while the robot is exploring an environment with multi-level terrain. Furthermore, we develop a 2D scan merging method to generate a map and obtain a resource to back up the robot localization. Furthermore, autonomous navigation in uneven and unstructured environments is helpful for mobile robots and provides meaningful information for the design of smart wheelchairs. The remainder of this paper is organized as follows. Section 2 addresses the ramp detection using Faster R-CNN, feature extraction, map merging, and localization. Section 3 provides the experimental results, and Section 4 presents the conclusions.

System Overview

The proposed experiment uses a base station computer as a manager for the data collection from the two robots. The wheel odometry of both robots constitutes the priority reference for the pose distribution using Monte Carlo localization. We measure the weight of the particles according to the input from each LIDAR. To ascertain the correspondences among the point clouds, we translate them to the given pose from the particle estimation.

The leader robot generates a global map from the collected LIDAR point clouds. The map can be updated using SLAM according to pose graph optimization and the LIDAR odometry. The follower robot uses the reference generated by the leader robot to identify multi-level access. Once the follower robot enters the new level, it explores all 2D boundaries on the floor. The diagram for the proposed multi-robot system is presented in Figure 1. The follower robot relies on the inertia measurement unit (IMU) to only proceed with the exploration on the X-Y plane. Unless an input of a new path comes from the leader robot, the follower robot will allow the reading in the Y-Z plane.

This work utilizes pose-graph optimization to update a map M. The contributions of this study are as follows: a multi-robot localization according to a Monte Carlo algorithm for multi-level areas, an extension of 2D ICP map merging, and a multi-robot exploration of multi-level areas using a deep CNN.

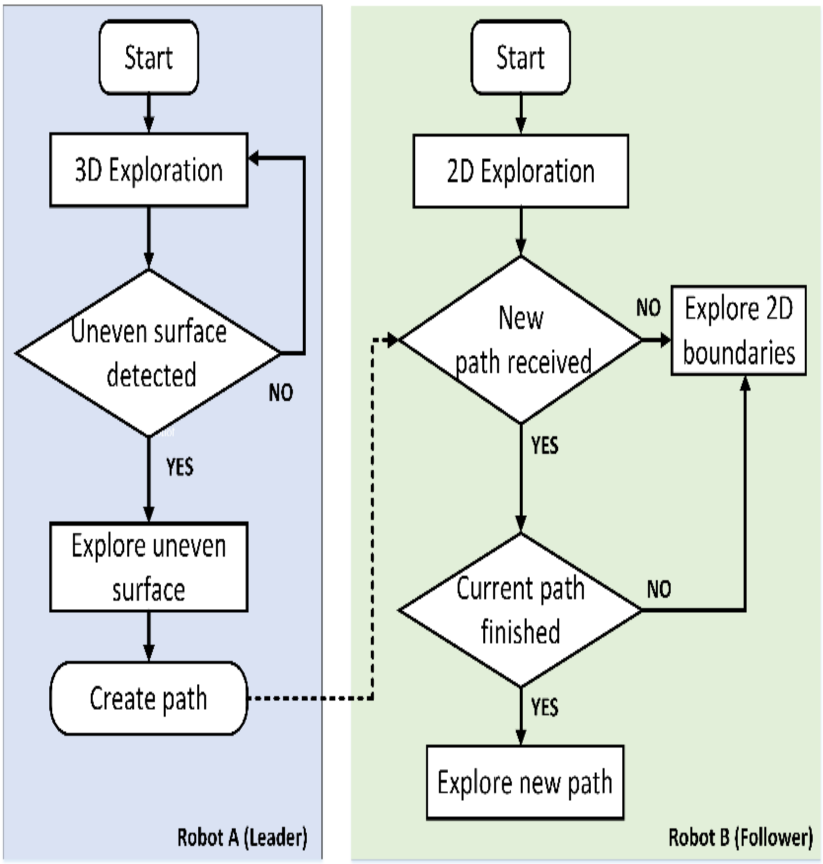


Figure 1. Task flow diagram for the multi robots.

2. Materials and Methods

2.1. Faster R-CNN for Indoor Ramp Detection

Our multi-robot system defines the location of multi-level areas. The leader robot uses a Faster R-CNN [47] for real-time object detection. The Faster R-CNN detector adds a region proposal network to generate region proposals directly in the network, thereby improving object detection. First, training images were collected using a Kinect camera and then resized to a resolution of 224 × 224. Next, each image was divided into grids and assigned a bounding box. A single CNN runs once on every image. The network consists of 15 convolutional layers followed by two fully connected layers. During training and testing, the Faster R-CNN checks the entire image. Figure 2 shows (a) the diagram for the Faster R-CNN training, (b) the 2D image capture by the Kinect camera, and (c) the detected ramp.

2.2. 3D Point Clouds and 2D Feature Extraction

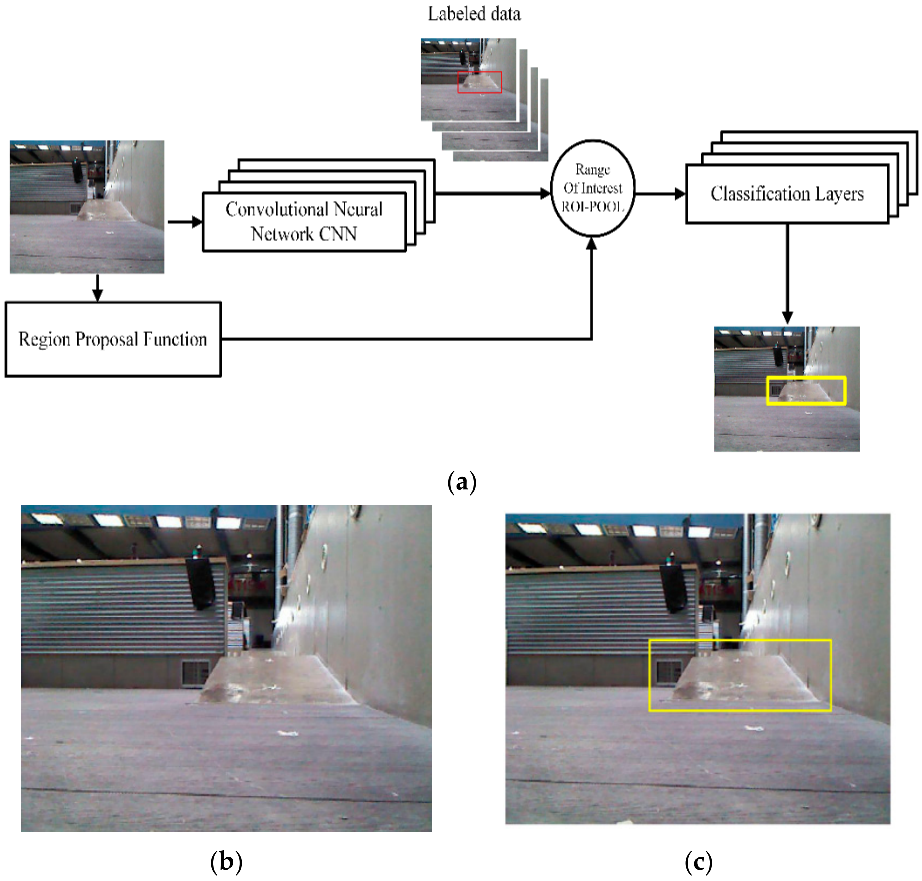
The leader robot performs three tasks: detecting uneven areas in the scenery, generating a 3D dense point cloud map, and extracting 2D features from the 3D point cloud.

Based on our previous research [1], the leader robot generates a 2D occupancy grid map OM_A using features obtained from a 3D point cloud for localization. The 2D features represent the main edges of the 3D point cloud. Figure 3a shows the projection of the 3D point cloud on a 2D plane. The range of interest (ROI) is set to detect the significant component of the 3D point cloud. To remove the invalid floor points, we employed the Random sampling consensus algorithm (RANSAC). The edge points are denoted as E_c^1 and arranged into polar coordinates. Then, the angle $\Delta\theta$ between the consecutive points

(r_n, r_{n+1}) of the polar coordinates can be obtained as Figure 3b shows. The distance between those two points is given by Equation (1).

$$D(r_n, r_{n+1}) = \sqrt{r_n^2 + r_{n+1} - 2r_n r_{n+1} \cos(\Delta\theta)}. \tag{1}$$

Given the 3D LIDAR resolution, the distance threshold is $D_{thd} = 0.02$ m to split the points. The ROI was divided into segments $Sg_k \,\epsilon\, \{p_1, p_2, \ldots, p_N\}$. $\{p_1, p_2, \ldots, p_N\}$ represent the set of points for every segment. The created segments allow us to generate and label features. The orientation for each feature line is $\Delta\theta$. Figure 3b illustrates how the points were split into segments. Using the described segments Sg_k and orientation $\Delta\theta$ we created a set of features $F_{L:A}$.



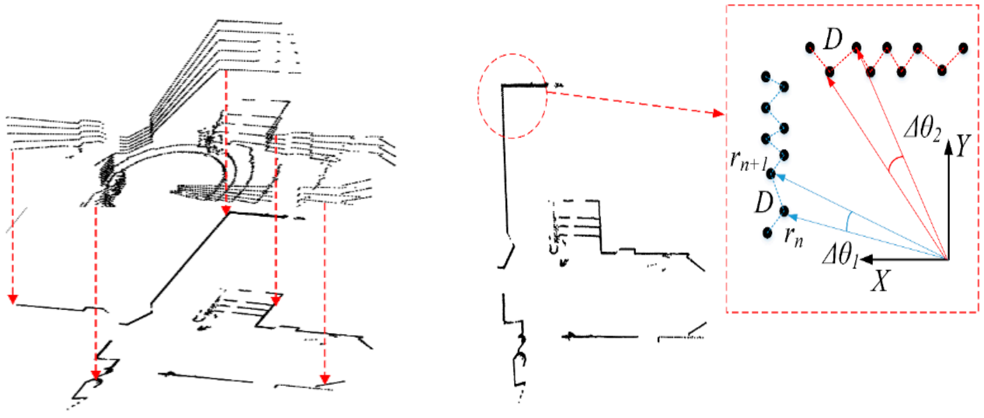
(**a**)

(**b**) (**c**)

Figure 2. Indoor ramp detection. (**a**) Diagram for Faster R-CNN training. (**b**) Original 2D image obtain by Kinect camera. (**c**) Indoor ramp detected.

2.3. 2D Mapping and Map Merging

The follower robot explores the flat area boundaries. The set of 2D LIDAR scans collected by the follower robot was divided according to the input received from the leader robot. The leader robot detects a ramp and the follower robot crosses to the ramp, thereby creating a new set of 2D LIDAR scans $s_{L:B}$. Then, using $s_{L:B}$, the follower robot filters each scan using the Voronoi diagram and Delaunay triangulation. Algorithm 1, Line 4 shows the filtering step. After filtering the scans, the follower robot creates an occupancy grid map OM_B.



(**a**) (**b**)

Figure 3. Point splitting method. (**a**) 3D point cloud projection onto the 2D plane. (**b**) Splitting and merging of the 2D edge features.

We propose an extended version of the ICP for scan merging and further occupancy grid map creation. For every robot pose, $\mathbf{M}^{i-1}$ is the reference scan and $\mathbf{S}_c^i$ is the current scan. Given the rotation matrix $\mathbf{R} = \mathbf{R}_\theta$ and the translation t, ICP computes the alignment error E between the two datasets and determines the proper rotation R and translation t that minimizes the outcome of Equation (2).

$$E(\mathbf{R}, \mathbf{t}) = \sum_{k=1}^{N_r} \sum_{j=1}^{N_c} w_{k,j} \|\mathbf{M}_k^{i-1} - (\mathbf{R}\mathbf{S}_{c,j}^i + \mathbf{t})\|^2, \tag{2}$$

where N_r and N_c are the number of the points in $\mathbf{M}^{i-1}$ and $\mathbf{S}_c^i$ respectively. $w_{k,j}$ is 1, if $\mathbf{M}_k^{i-1}$ is the closest point to $\mathbf{S}_{c,j}^i$ and is 0 otherwise. ICP rotates the scanned data $\mathbf{S}_{c,j}^i$ by θ and translates using t to obtain the best alignment to the reference data $\mathbf{M}^{i-1}$. We started the mapping and localization matching $\begin{bmatrix} 0 & 0 & 0 \end{bmatrix}^T$ with the coordinate frame. Using ICP scan matching, we obtained a pose correction vector $\begin{bmatrix} x_{sm}^i & y_{sm}^i & \theta_{sm}^i \end{bmatrix}^T$. The pose correction vector derives the homogeneous coordinate transformation matrix H. By employing the 2D geometric transformation, H was expanded to a 3 × 3 matrix as shown in Equation (3).

$$\mathbf{H} = \begin{bmatrix} \cos\theta_{sm}^i & -\sin\theta_{sm}^i & x_{sm}^i \\ \sin\theta_{sm}^i & \cos\theta_{sm}^i & y_{sm}^i \\ 0 & 0 & 1 \end{bmatrix} \tag{3}$$

Then the pose of the robot can be updated by Equation (4).

$$\begin{bmatrix} x_{est}^i \\ y_{est}^i \\ \theta_{est}^i \end{bmatrix} = \begin{bmatrix} x_{est}^{i-1} \\ y_{est}^{i-1} \\ \theta_{est}^{i-1} \end{bmatrix} + \begin{bmatrix} dx_{i-1}^i \\ dy_{i-1}^i \\ d\theta_{i-1}^i \end{bmatrix} + \begin{bmatrix} x_{sm}^i \\ y_{sm}^i \\ \theta_{sm}^i \end{bmatrix} \tag{4}$$

where $\begin{bmatrix} x_{est}^{i-1} & y_{est}^{i-1} & \theta_{est}^{i-1} \end{bmatrix}^T$ is the estimated pose of the leader robot t $(i-1)-th$ sampling time, and $\begin{bmatrix} dx_{i-1}^i & dy_{i-1}^i & d\theta_{i-1}^i \end{bmatrix}^T$ is the difference between the $i-th$ and the $(i-1)-th$ pose of the robot as estimated through the odometer. Scan merging combines the reference data set $\mathbf{M}^{i-1}$ and the current data set $\mathbf{S}_c^i$ into a new data set $\mathbf{M}^i$. Then, we determined the outlier points and validated the workspace scans. Using a sparse point map, we avoid point duplication. The map of the environment was incrementally built according to Equation (5).

$$\mathbf{M}^i = \mathbf{M}^{i-1} \cup \left\{ (x_q, y_q) \in \mathbf{S}^i{}_c \middle| \exists (x_p, y_p) \in \mathbf{M}^{i-1} : \|\ (x_q, y_q) - (x_p, y_p)\ \| < d_{th} \right\} \tag{5}$$

where (x_q, y_q) is a data point of the current scan $\mathbf{S}_c^i$, (x_p, y_p) is a data point in $\mathbf{M}^{i-1}$, and d_{th} is the threshold value for the scan merging. The sensor range determines the proper value d_{th} to achieve optimal scan alignment. The initial state $\mathbf{S}_c^i$ is one "1" and d_{th} is the threshold circle radius.

The merging was executed from left to right, and the threshold circle moves as the data line increases. The new reference $\mathbf{M}^{i-1}$ was obtained from the data line "5." The larger the value d_{th}, the smaller the number of points on the map. Figure 4 shows the reference scan $\mathbf{M}^{i-1}$ (red triangles) and the current scan $\mathbf{S}_c^i$ (blue circles). Figure 4a presents a circle moving from the previous P_{i+1} line "1" to the new position p_i line "2." Points that do not belong to p_i and p_{i+1} are the new scanned points. Figure 4b shows lines "1" and "4" as a duplicated data set and lines "2" and "3" as new scanned points. Our method merges LIDAR scans using a circle threshold that omits adjacent points. If the Euclidian distances between,p_i, p_{i-1} and p_{i+1} are shorter than the threshold d_{th}, then the points are invalid. Algorithm 1, Line 6 includes the described scan merging. Figure 5a–c shows the different stages of the map merging using the segmented data set.

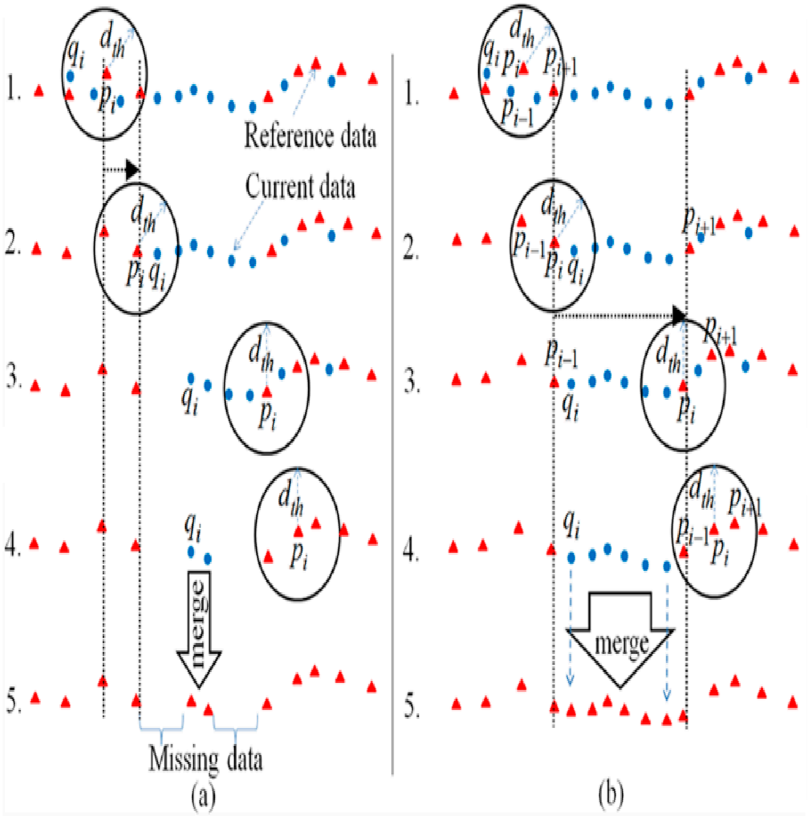


Figure 4. Scan merging. (**a**) Conventional merging rule. (**b**) Modified scan merging rule.

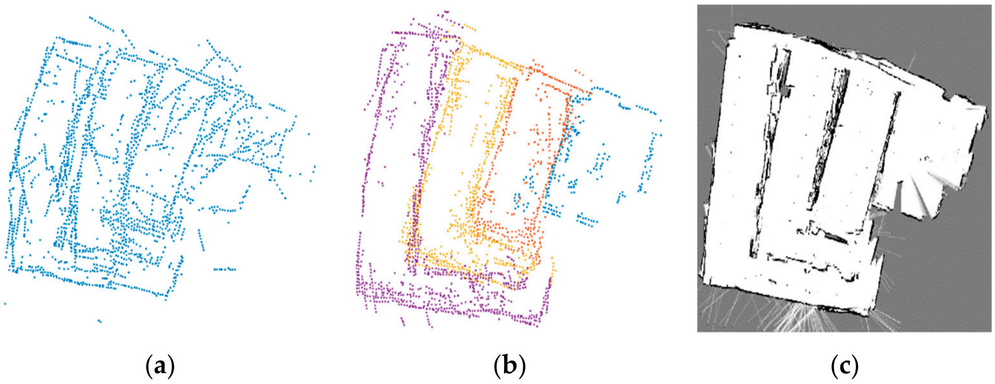

(**a**) (**b**) (**c**)

Figure 5. 2D LIDAR segmentation and filtering. (**a**) Aligned 2D LIDAR scans without filtering and segmentation (**b**) Segmented and filtered LIDAR scans. (**c**) Generated occupancy grid map.

Algorithm 1 Follower Robot Occupancy Grid Map

```
Input: the follower robot sets local point − clouds : $C_{L:B}$
Output: Occupancy grid map : $OM_B$
for $i = 1$ to $Size(C_{L:B})$ do
    $C_{filter\_L:B} = \text{Filter}\left(C_{L:B(i)}\right)$
    if Leader robot(uneven = 1) then
        $s_{L:B(j)} = C_{filter\_L:B(i:i+1)}$
        $S_{L:B(j)} = \text{Merge}\left(s_{L:B(j)}\right)$
        j = j + 1
    end if
    $OM_B = Occupancy\ Map(S_{L:B})$
end for
```

2.4. Communication

To establish multi-robot communication, we used an ad hoc on-demand distance vector (AODV) and a WLAN access point. The AODV is a package of the robot operative system (ROS) and has a unicast and multicast transition. The ROS allows us to create nodes for multi-robot communication. The AODV uses an automatic request datalink to achieve reliable data transfer. The robots do not communicate with each other directly. Instead, the AODV transmits the data using a raw socket, thereby avoiding kernel space. Once each robot has received the data, the master computer publishes the AODV package. With the mentioned protocol and ROS node-publication method, we establish flexible communication among the robots. Figure 6a shows the multi-robot communication diagram. The master station receives a ROS message containing the occupancy grid created by each robot. The master computer then merges the multi-robot trajectories into a single-occupancy grid map. Figure 6b presents the communication flowchart.

2.5. Multi-Robot Localization

We propose a multi-robot localization using a Monte Carlo algorithm from our previous study [1]. The leader robot gives the starting point in the initial pose. Each robot uses a sensor reading for particle estimation. We used an iterative process; each robot moves, senses, and re-samples to determine its pose. We can execute a single robot localization when multi-robots have mutual poses. This work assumes that the initial robot pose is known, but each robot does not have global positioning. The leader robot extracts 2D features from the 3D point cloud projection and generates an occupancy grind map $\boldsymbol{OM_A}$. The follower robot then uses 2D LIDAR scans to localize itself in the occupancy grid map $\boldsymbol{OM_B}$.

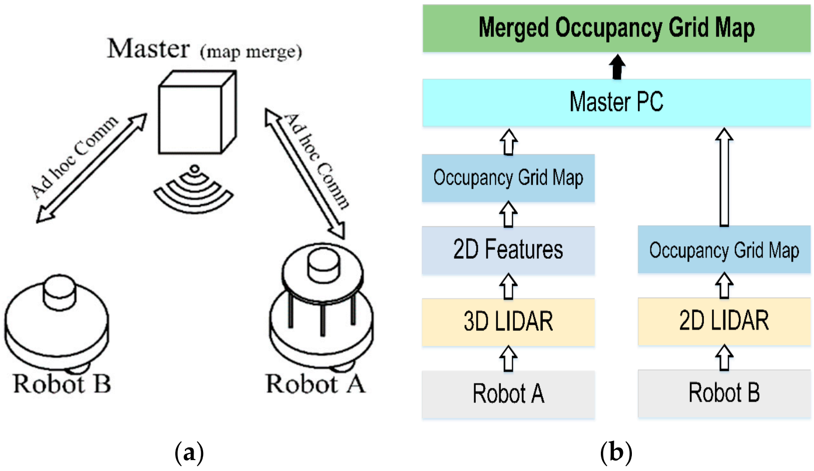


Figure 6. Multi-robot communication. (**a**) Robot A (leader) and Robot B (follower) communication with the master computer. (**b**) Communication flowchart.

The leader robot provides the initial position for localization using the Monte Carlo algorithm. Then, the follower robot proceeds with the localization in the occupancy grid map $\boldsymbol{OM_B}$ using the features $F_{L:A}$ described in the Section 2.2 as shown in Algorithm 2, Line 5. To obtain the multi-robot localization, we merge the follower robot trajectory onto the leader robot occupancy-grid-map $\boldsymbol{OM_A}$ as shown in Algorithm 2, Line 6. Lastly, we ascertain each robot's global position in an occupancy grid map. Figure 7 reveals the occupancy map created by each robot: (a) the map for the leader robot, (b) the follower robot map, and (c) the combination of both in one map.

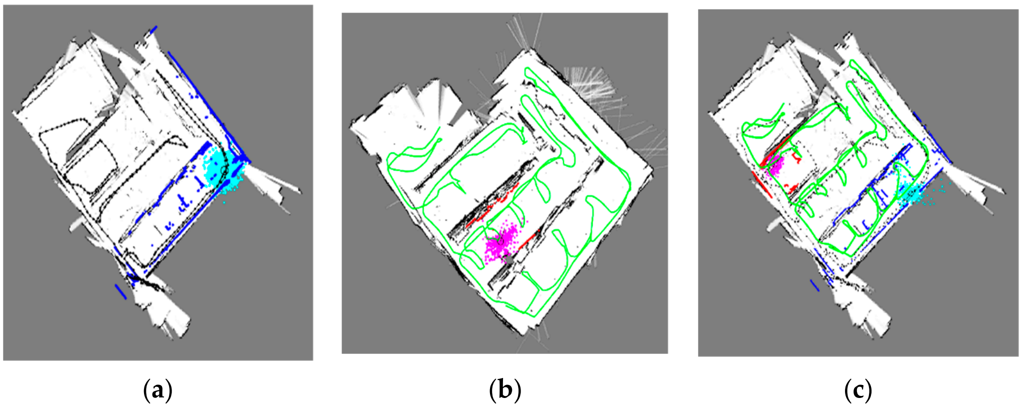


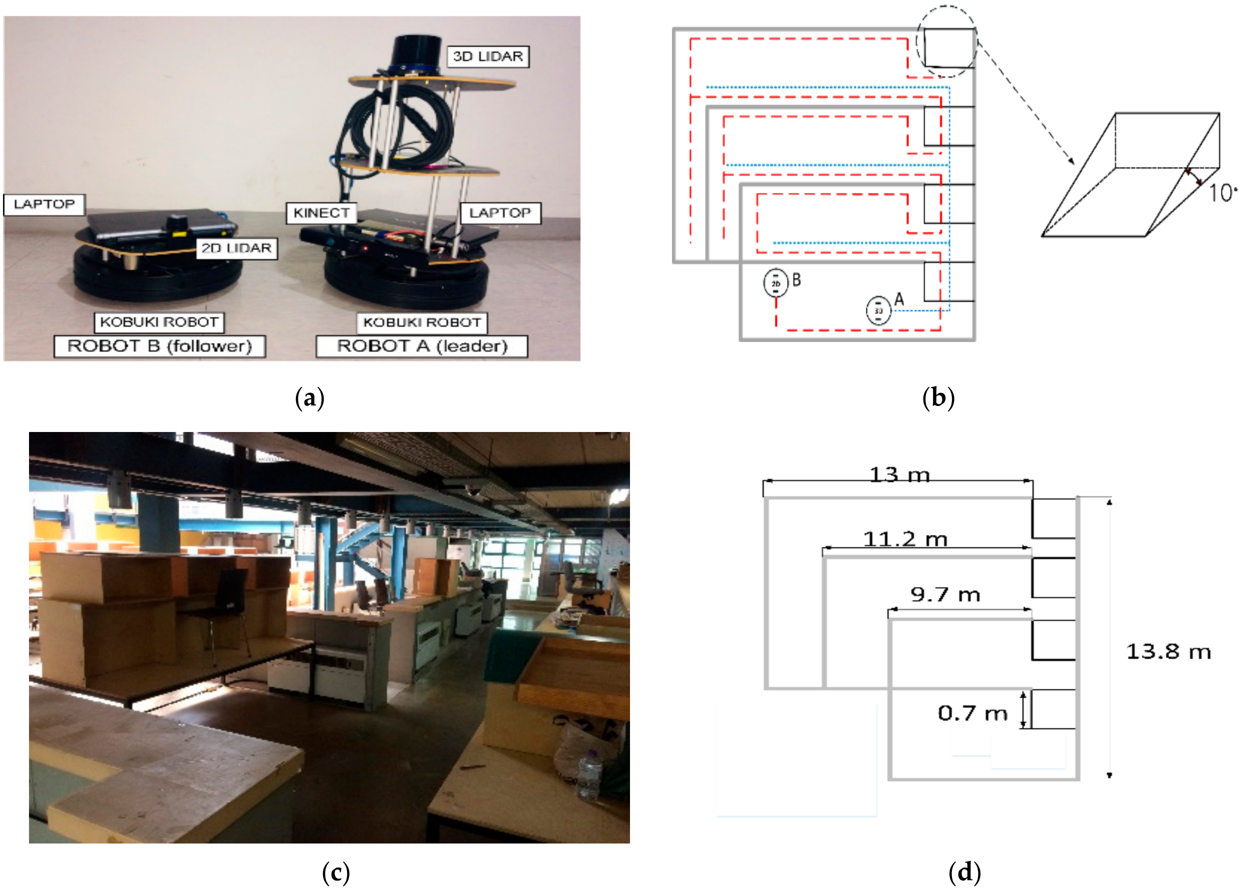
Figure 7. Multi-Robot occupancy maps and trajectories. (**a**) Map of the leader robot, (**b**) Map of the follower robot. (**c**) Map containing leader and follower robots.

Algorithm 2 Multi-Robot Localization

```
1 : Input: Set local clouds $(C_{L:A})$ in the follower robot occupancy grid map $(OM_B)$
2 : Output: Global robot position : $R_{G:A},\ R_{G:B}$
3 :  for $i = 1$ to $N = Size(C_{L:A})$ do
4 :            $F_{L:A} = Extract\ 2D\ Features(C_{L:A})$
5 :          $OM_A = Occupancy\ Map(F_{L:A})$
6 :    $OM_A = merge(OM_A,\ OM_B)$
7 :    $AMCL\left[F_{L:A(i)}, C_{L:B(i)}, OM_A\right] \rightarrow R_{G:A(i)}, R_{G:B(i)}$
8:    end for
```

3. Results

Our multi-robot system has two robots: Robot A (leader) and Robot B (follower). The leader robot is equipped with a 3D LIDAR scan, a Kinect camera, a Kobuki robot platform, and a laptop running Linux Ubuntu 14.04. The follower robot is equipped with a 2D LIDAR scan, a Kobuki platform, and a laptop running Linux Ubuntu 14.04. The master station uses Matlab with the robotics Tool Box 1.4 with an Intel i7 processor. Figure 8a depicts the two robots used for the experiment, and Figure 8b–d shows the experiment location. For the experiment, we use a university location provided with four ramps with 10-degree slopes. The ramps allow us to test the multi-robot performance in an uneven space. The leader robot collects the features using the process described in Section 2.2. The follower robot uses raw information from a 2D LIDAR scan.



(**a**) (**b**)

(**c**) (**d**)

Figure 8. Experiment setting: (**a**) Two mobile robots and their parts, (**b**) the experiment location and corresponding robot paths, (**c**) Image taken from a corner of the experiment location, (**d**) experiment location diagram.

The experiment coordinates systems are as follows: the world coordinate (x, y, z), leader robot (x_A, y_A, z_A), and the follower robot (x_B, y_B, z_B). The robot coordinates are always parallel to each robot's velocity. Axes z_A and z_B are perpendicular to the soil. Both robots move with a linear velocity of 0.05 m/s.

The 2D and 3D LIDAR sensor axes are concurrent with the follower and leader robots, respectively. The speed of both robots V_A and V_B are considered as non-slip speeds. The scanning frequency for the leader robot is every 0.5 m, and that of the follower robot is 0.1 m. The leader robot sampling period is $\Delta t = 1$ s, and that of the follower robot is $\Delta t = 0.5$ s. The follower robot has a shorter sampling period because 2D LIDAR scans are lighter than their 3D LIDAR counterparts. Both robots have enough time for data acquisition from LIDAR and odometry sensors employing the described sampling time. The proposed multi-robot system uses 3D point cloud information to identify features and recreate a

2.5D dense map. By contrast, the 2D LIDAR scans allow for fast localization on the even space. The ground truth for the robot localization was obtained using the odometer and the Inertia measurement unit (IMU) integrated into the robot. The information collected from these sensors was fused using the Extended Kalman Filter (EKF) within a ROS node.

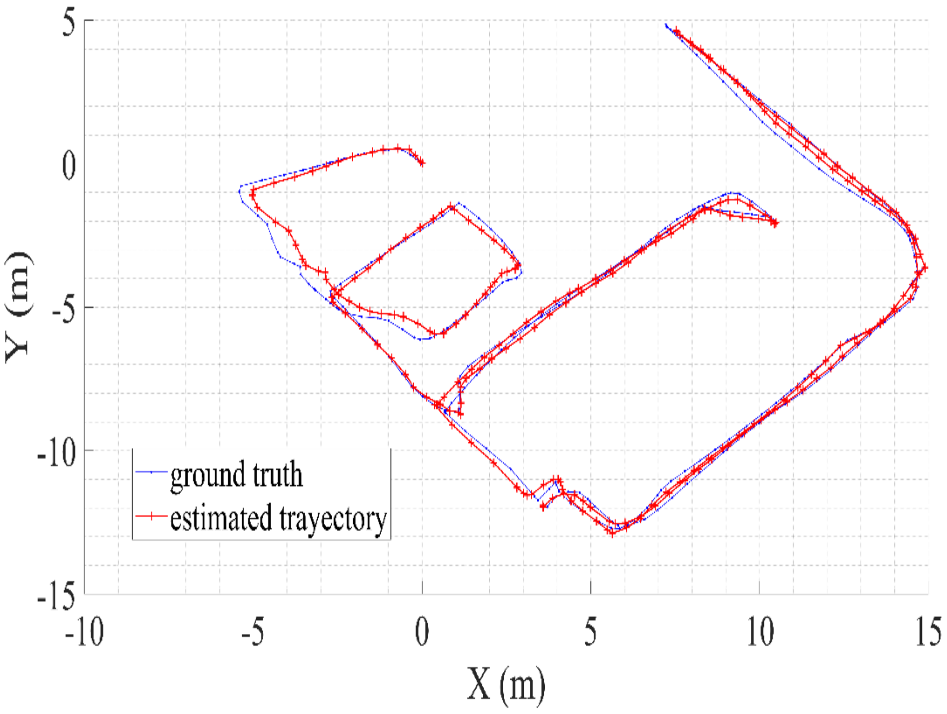
We calculated the root-mean-square error (RMSE) in meters for the scan registration. Table 1 lists the RSME errors for both robots. Figure 9 shows the multi-robot estimated and ground-truth trajectories. Figure 9a,b are the trajectories of the leader and follower robots, respectively. The follower robot has a larger trajectory than that of the leader robot. As the leader robot uses a 3D LIDAR for exploration, his trajectory is shorter than that of the follower robot. Figure 9c shows the combined trajectory for both robots. Table 1 indicates that the leader and follower robots' scanning errors are approximately 0.2 m in the X and Y axes, and this outcome acceptable for robot mapping and exploration.

Table 1. Root mean square error for scan registration.

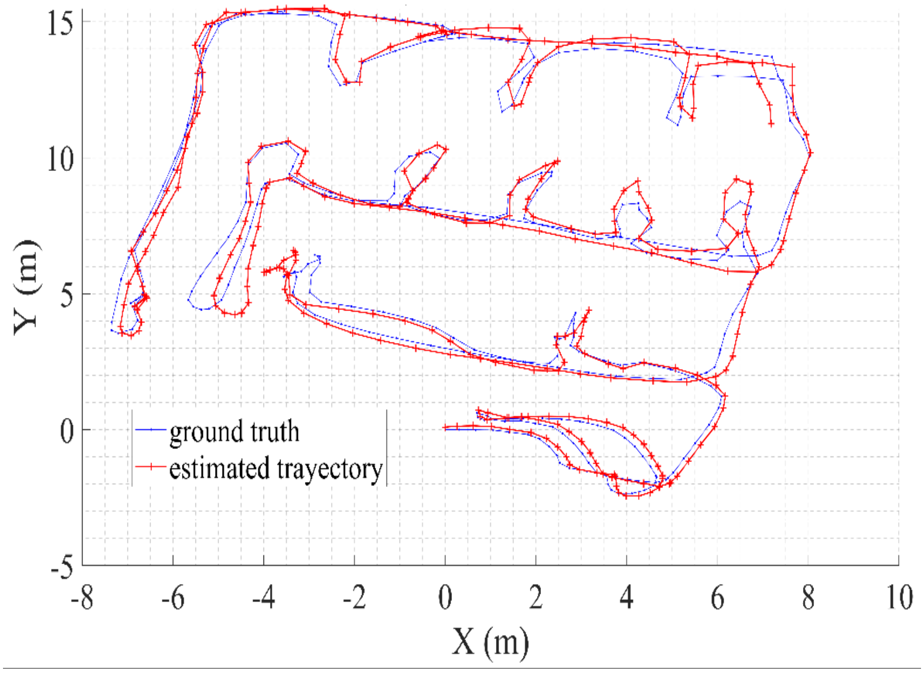
		Leader Robot	**Follower Robot**
Error (m)	X-Axis	0.26	0.22
	Y Axis	0.19	0.24
	Z Axis	0.31	0.86

The leader robot collects 3D point clouds for every pose and creates a 2.5D dense map. We used a box grid filter (BGF) with a voxel size of 0.1 m to down-sample each 3D point cloud. Once the robots completed the exploration, each robot generates a map. The leader robot generates a 2.5D dense map including all the collected 3D LIDAR point clouds. The 2.5D dense map created by the leader robot includes the floor points. The floor points are only removed for the generation of the occupancy grid map, as is described in Section 2.2. The follower robot recreates the 2D map using all the collected 2D LIDAR scans, the 2D LIDAR map was merged and filter using the process described in Section 2.3.

Filtering the 3D point clouds enable quick registration, thereby maintaining accurate results. Figure 10a depicts the 2.5D map generated by the leader robot. Figure 10b shows the 2D map generated by the follower robot following the process described in Section 2.3. We compare our results with those of two state-of-art methods: generalized GICP and LOAM, to validate our method. As the proposed method was designed for multiple robots, we test the performance of each robot. To validate the trajectory obtained, we measure (in meters) the mean square error (RMSE) between the ground truth and the estimated trajectory. First, we employed two search algorithms: kd-tree and knearest-neighbor; which are the optimal ways to find the distance between two neighboring points. Second, once the closest neighbor pair was found, RMSE calculated the distance between two neighboring points. Tables 2 and 3 show the localization error for each robot. LOAM and GICP are dependent on features extracted for even surfaces. As an uneven surface induces a sudden point cloud rotation, our method provides an accurate response for multi-robot localization on multi-level spaces.

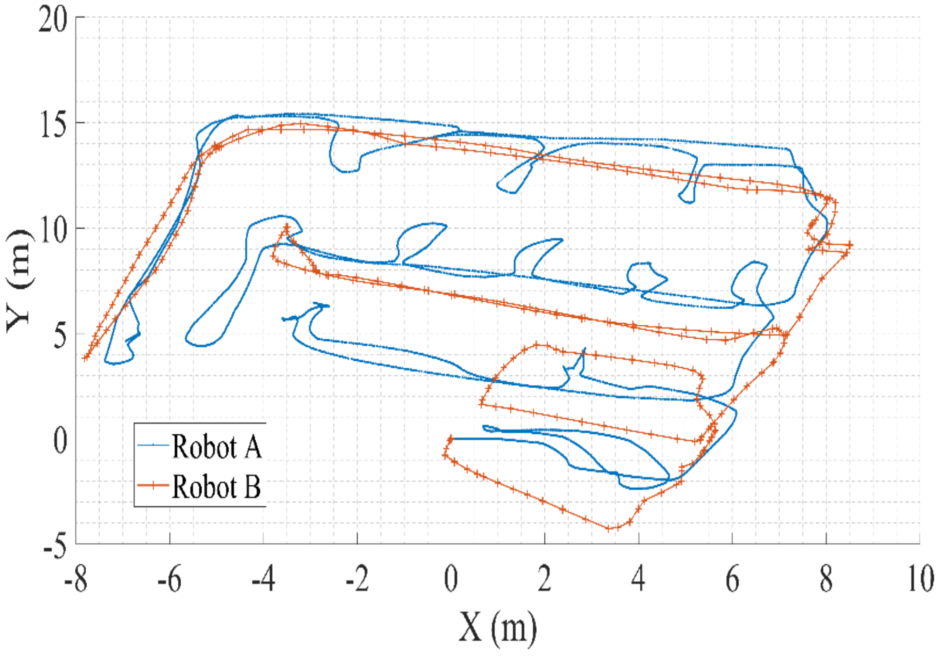


(**a**)



(**b**)

Figure 9. *Cont.*



(**c**)

Figure 9. Multi-robot estimated trajectories and ground truth. (**a**) Leader robot (Robot A) trajectory. (**b**) Follower Robot (Robot B) trajectory. (**c**) Multi-robot trajectories.

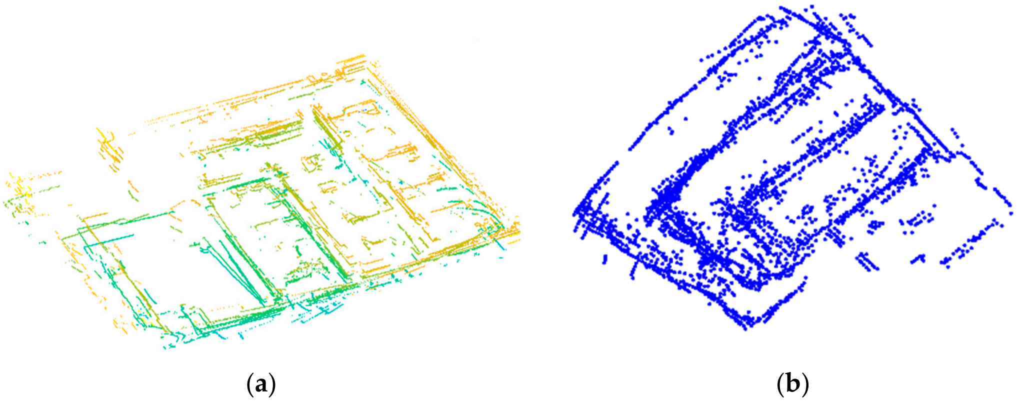


Figure 10. Maps generated by the multi-robot system. (**a**) 2.5D map generated by Robot A (Leader). (**b**) 2D map generated by Robot B (Follower).

Table 2. Leader robot localization errors for the proposed method, the generalized iterative closest point (GICP) technique, and the LIDAR and odometry mapping (LOAM) approach. The lowest values in the axis X, Y, Z, and consumed time are denoted in bold.

Leader Robot	Error (m)			Time (min)
	X	Y	Z	
Our Method	0.26	0.32	0.27	14.64
GICP	**0.18**	0.17	**0.17**	11.90
LOAM	0.25	**0.12**	0.19	**11.67**

Table 3. Follower robot localization errors for the proposed method, the generalized iterative closest point (GICP) technique, and the LIDAR and odometry mapping (LOAM) approach. The lowest values in the axis X, Y, and consumed time are denoted in bold.

Follower Robot	**Error (m)**		**Time (min)**
	X	**Y**	
Our Method	**0.02**	**0.05**	**12.10**
GICP	0.20	0.27	18.14
LOAM	0.20	0.27	18.14

This section may be divided by subheadings. It should provide a concise and precise description of the experimental results, their interpretation, as well as the experimental conclusions that can be drawn. Table 2 shows the error for the leader robot, for which our method error in axis Y is approximately 0.15 m that that of the GICP and LOAM. Table 3 shows the error for the follower robot. Our method errors in axes X and Y are approximately 0.2 m lower than those of the GICP and LOAM, and the processing time is also lower because we enhance the filtering and map merging for 2D scans. Table 4 shows the general errors in the multi-robot system, for which our method reduces the error in axes X and Y. The processing time of our technique is likewise lower than that of the GICP and LOAM. As our proposed method includes using a CNN, the 3D–2D exploration is versatile and allows for robot mapping within an optimal result. To provide a quantitative and qualitative comparison of the three methods, we only used MATLAB and ROS for programming. It was clarified in the results section.

Table 4. Total errors and times for a multi-robot system using our proposed method versus the general iterative closest Point (GICP) technique and the LIDAR odometry and Mapping (LOAM) approach. The lowest values in the axis X, Y, and consumed time are denoted in bold.

Multi-Robot	**Error (m)**		**Time (min)**
	X	**Y**	
Our Method	**0.28**	**0.37**	**26.74**
GICP	0.38	0.44	30.04
LOAM	0.45	0.39	29.81

4. Conclusions

This work allows the mapping and exploration of multi-level surfaces for multi-robots. Our mapping approach simplifies the global representation and localization using a merged occupancy grid map. The experiment results show a robust response in an environment integrated with multi-level surfaces. Our proposed 2D LIDAR scan merging method reduces the error localization around the x and y axes. The created Faster R-CNN has a robust response for detecting ramps in indoor environments. In future work, we shall extend larger indoor and outdoor mapping scenery.

Author Contributions: Conceptualization, V.A.R.-C. and S.-G.L.; methodology, V.A.R.-C.; software, J.-H.C.; validation, V.A.R.-C., Q.-D.H. and J.-H.C.; formal analysis, Q.-D.H.; investigation, V.A.R.-C.; resources, J.-H.C.; data curation, Q.-D.H.; writing—original draft preparation, V.A.R.-C.; writing—review and editing, S.-G.L. visualization, Q.-D.H. supervision, S.-G.L.; project administration, S.-G.L.; funding acquisition, S.-G.L. and J.-H.C. All authors have read and agreed to the published version of the manuscript.

Funding: This research was funded by the National Research Foundation of Korea, funded by the Ministry of Education [2019R1A2C2010195] and by the Ministry of Trade, Industry and Energy under the Robot Industrial Core Technology Development Project program [K_G012000921401] supervised by the KEIT. The APC was funded by [2019R1A2C2010195].

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Not applicable.

Acknowledgments: This research was also supported by the BK21 FOUR (the Integrated Education Institute for Frontier Science and Technology) funded by the Ministry of Education (MOE) of Korea and National Research Foundation (NRF) of Korea.

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

References

1. Rosas-Cervantes, V.; Lee, S.-G. 3D Localization of a Mobile Robot by Using Monte Carlo Algorithm and 2D Features of 3D Point Cloud. *Int. J. Control. Autom. Syst.* **2020**, *18*, 2955–2965. [CrossRef]
2. Sakai, T.; Koide, K.; Miura, J.; Oishi, S. Large-scale 3D outdoor mapping and on-line localization using 3D-2D matching. In Proceedings of the 2017 IEEE/SICE International Symposium on System Integration (SII), Taipei, Taiwan, 11–14 December 2017.
3. Aragüés, R.; Cortes, J.; Sagues, C. Distributed consensus algorithms for merging feature-based maps with limited communication. *Robot. Auton. Syst.* **2011**, *59*, 163–180. [CrossRef]
4. Burgard, W.; Moors, M.; Fox, D.; Simmons, R.; Thrun, S. Collaborative multi-robot exploration in Proceedings 2000 ICRA. Millennium Conference. In Proceedings of the IEEE International Conference on Robotics and Automation. Symposia Proceedings (Cat. No.00CH37065), San Francisco, CA, USA, 24–28 April 2000; Volume 1, pp. 476–481.
5. Zhou, X.S.; Roumeliotis, S.I. Multi-robot SLAM with Unknown Initial Correspondence: The Robot Rendezvous Case. In Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, 9–15 October 2006; pp. 1785–1792.
6. Muñoz-Salinas, R.; Medina-Carnicer, R. UcoSLAM: Simultaneous localization and mapping by fusion of keypoints and squared planar markers. *Pattern Recognit.* **2020**, *101*, 107193. [CrossRef]
7. Zhang, J.; Singh, S. LOAM: Lidar Odometry and Mapping in Real-Time. In Proceedings of the Robotics: Science and Systems (RSS '14), Berkeley, CA, USA, 12–16 July 2014; pp. 109–111. [CrossRef]
8. Gold, S.; Lu, C.-P.; Rangarajan, A.; Pappu, S.; Mjolsness, E. New Algorithms for 2D and 3D Point Matching: Pose Estimation and Correspondence. *Pattern Recognit.* **1994**, *31*, 1019–1031. [CrossRef]
9. Tsin, Y.; Kanade, T. A Correlation-Based Approach to Robust Point Set Registration. In *Computer Vision—ECCV 2004*; Springer: Berlin/Heidelberg, Germany, 2004; pp. 558–569.
10. Baker, S.; Matthews, I. Lucas-Kanade 20 Years On: A Unifying Framework. *Int. J. Comput. Vis.* **2004**, *56*, 221–255. [CrossRef]
11. Zhang, Y.; Xiao, Y. Notice of Retraction: A patrolling scheme in wireless sensor and robot networks. In Proceedings of the 2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), Shanghai, China, 10–15 April 2011; pp. 513–518. [CrossRef]
12. Yan, C.; Zhang, T. Multi-robot patrol: A distributed algorithm based on expected idleness. *Int. J. Adv. Robot. Syst.* **2016**, *13*. [CrossRef]
13. Carpin, S. Fast and accurate map merging for multi-robot systems. *Auton. Robot.* **2008**, *25*, 305–316. [CrossRef]
14. Georgiou, C.; Anderson, S.; Dodd, T. Constructing informative Bayesian map priors: A multi-objective optimisation approach applied to indoor occupancy grid mapping. *Int. J. Robot. Res.* **2017**, *36*, 274–291. [CrossRef]
15. Vysotska, O.; Stachniss, C. Improving SLAM by Exploiting Building Information from Publicly Available Maps and Localization Priors. *PFG—J. Photogramm. Remote. Sens. Geoinf. Sci.* **2017**, *85*, 53–65. [CrossRef]
16. Mielle, M.; Magnusson, M.; Andreasson, H.; Lilienthal, A. SLAM auto-complete: Completing a robot map using an emergency map. In Proceedings of the 2017 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Shanghai, China, 11–13 October 2017; pp. 35–40. [CrossRef]
17. Huang, W.H.; Beevers, K.R. Topological Map Merging. *Int. J. Robot. Res.* **2005**, *24*, 601–613. [CrossRef]
18. Mielle, M.; Magnusson, M.; Lilienthal, A.J. Using sketch-maps for robot navigation: Interpretation and matching. In Proceedings of the 2016 IEEE International Symposium on Safety, Security and Rescue Robotics (SSRR), Lausanne, Switzerland, 23–27 October 2016; pp. 252–257.
19. Pippin, C.; Christensen, H.; Weiss, L. Performance based task assignment in multi-robot patrolling. In Proceedings of the 28th Annual ACM Symposium on Applied Computing, Coimbra, Portugal, 18–22 March 2013; pp. 70–76. [CrossRef]
20. Kakuma, D.; Tsuichihara, S.; Ricardez, G.A.G.; Takamatsu, J.; Ogasawara, T. Alignment of Occupancy Grid and Floor Maps Using Graph Matching. In Proceedings of the 2017 IEEE 11th International Conference on Semantic Computing (ICSC), San Diego, CA, USA, 30 Janaury–1 February 2017; pp. 57–60.
21. Krajník, T.; Fentanes, J.P.; Hanheide, M.; Duckett, T. Persistent localization and life-long mapping in changing environments using the Frequency Map Enhancement. In Proceedings of the 2016 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Daejeon, Korea, 9–14 October 2016; pp. 4558–4563.
22. Pfingsthorn, M.; Birk, A. Efficiently communicating map updates with the pose graph. In Proceedings of the 2008 IEEE/RSJ International Conference on Intelligent Robots and Systems, Nice, France, 22–26 September 2008; pp. 2519–2524.

23. Dissanayake, M.W.M.G.; Newman, P.; Clark, S.; Durrant-Whyte, H.F.; Csorba, M. A solution to the simultaneous localization and map building (SLAM) problem. *IEEE Trans. Robot. Autom.* **2001**, *17*, 229–241. [CrossRef]
24. Kaufman, E.; Takami, K.; Lee, T.; Ai, Z. Autonomous Exploration with Exact Inverse Sensor Models. *J. Intell. Robot. Syst.* **2018**, *92*, 435–452. [CrossRef]
25. Joubert, D.; Brink, W.; Herbst, B. Pose Uncertainty in Occupancy Grids through Monte Carlo Integration. *J. Intell. Robot. Syst.* **2015**, *77*, 5–16. [CrossRef]
26. Jadidi, M.G.; Miro, J.V.; Dissanayake, G. Mutual information-based exploration on continuous occupancy maps. In Proceedings of the 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Hamburg, Germany, 28 September–2 October 2015; pp. 6086–6092.
27. Eckart, B.; Kim, K.; Troccoli, A.; Kelly, A.; Kautz, J. Accelerated Generative Models for 3D Point Cloud Data. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 5497–5505.
28. Souza, A.; Maia, R.; Gonçalves, L. 3D Probabilistic Occupancy Grid to Robotic Mapping with Stereo Vision. In *Current Advancements in Stereo Vision*; IntechOpen: London, UK, 2012; pp. 181–195.
29. Kaufman, E.; Lee, T.; Ai, Z. Autonomous exploration by expected information gain from probabilistic occupancy grid mapping. In Proceedings of the 2016 IEEE International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR), San Francisco, CA, USA, 13–16 December 2016; pp. 246–251.
30. Kaufman, E.; Takami, K.; Ai, Z.; Lee, T. Autonomous Quadrotor 3D Mapping and Exploration Using Exact Occupancy Probabilities. In Proceedings of the 2018 Second IEEE International Conference on Robotic Computing (IRC), Laguna Hills, CA, USA, 31 Janaury–2 February 2018; pp. 49–55.
31. Einhorn, E.; Schröter, C.; Gross, H. Finding the adequate resolution for grid mapping—Cell sizes locally adapting on-the-fly. In Proceedings of the 2011 IEEE International Conference on Robotics and Automation, Shanghai, China, 9–13 May 2011; pp. 1843–1848.
32. Khan, S.; Wollherr, D.; Buss, M. Adaptive rectangular cuboids for 3D mapping. In Proceedings of the 2015 IEEE International Conference on Robotics and Automation (ICRA), Seattle, WA, USA, 26–30 May 2015; pp. 2132–2139.
33. Zhu, C.; Ding, R.; Lin, M.; Wu, Y. A 3D Frontier-Based Exploration Tool for MAVs. In Proceedings of the 2015 IEEE 27th International Conference on Tools with Artificial Intelligence (ICTAI), Vietri sul Mare, Italy, 9–11 November 2015; pp. 348–352.
34. Hornung, A.; Wurm, K.M.; Bennewitz, M.; Stachniss, C.; Burgard, W. OctoMap: An efficient probabilistic 3D mapping framework based on octrees. *Auton. Robot.* **2013**, *34*, 189–206. [CrossRef]
35. Droeschel, D.; Schwarz, M.; Behnke, S. Continuous mapping and localization for autonomous navigation in rough terrain using a 3D laser scanner. *Robot. Auton. Syst.* **2017**, *88*, 104–115. [CrossRef]
36. Cieslewski, T.; Choudhary, S.; Scaramuzza, D. Data-Efficient Decentralized Visual SLAM. In Proceedings of the 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, Australia, 21–25 May 2018; pp. 2466–2473.
37. Pal, A.; Tiwari, R.; Shukla, A. Multi-Robot Exploration in Wireless Environments. *Cogn. Comput.* **2012**, *4*, 526–542. [CrossRef]
38. Smith, A.J.; Hollinger, G.A. Distributed inference-based multi-robot exploration. *Auton. Robot.* **2018**, *42*, 1651–1668. [CrossRef]
39. Fan, T.; Long, P.; Liu, W.; Pan, J. Distributed multi-robot collision avoidance via deep reinforcement learning for navigation in complex scenarios. *Int. J. Robot. Res.* **2020**, *39*, 856–892. [CrossRef]
40. Lowe, R.; Wu, Y.; Tamar, A.; Harb, J.; Abbeel, P.; Mordatch, I. Multi-agent actor-critic for mixed cooperative-competitive environments. In Proceedings of the 31st International Conference on Neural Information Processing Systems, Long Beach, CA, USA, 4–9 December 2017.
41. Redmon, J.; Divvala, S.; Girshick, R.; Farhadi, A. You Only Look Once: Unified, Real-Time Object Detection. In Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 779–788.
42. Liu, W.; Anguelov, D.; Erhan, D.; Szegedy, C.; Reed, S.; Fu, C.Y.; Berg, A.C. SSD: Single Shot MultiBox Detector. In *Computer Vision—ECCV 2016*; Springer: Cham, Switzerland, 2016; pp. 21–37.
43. Bargoti, S.; Underwood, J. Deep fruit detection in orchards. In Proceedings of the IEEE International Conference on Robotics and Automation, Singapore, 29 May–3 June 2017; pp. 3626–3633.
44. Sa, I.; Ge, Z.; Dayoub, F.; Upcroft, B.; Perez, T.; McCool, C. DeepFruits: A Fruit Detection System Using Deep Neural Networks. *Sensors* **2016**, *16*, 1222. [CrossRef]
45. Fox, D.; Ko, J.; Konolige, K.; Limketkai, B.; Schulz, D.; Stewart, B. Distributed Multirobot Exploration and Mapping. *Proc. IEEE* **2006**, *94*, 1325–1339. [CrossRef]
46. Pütz, S.; Wiemann, T.; Sprickerhof, J.; Hertzberg, J. 3D Navigation Mesh Generation for Path Planning in Uneven Terrain. *IFAC-PapersOnLine* **2016**, *49*, 212–217. [CrossRef]
47. Girshick, R. Fast R-CNN. In Proceedings of the 2015 IEEE International Conference on Computer Vision (ICCV), Santiago, Chile, 7–13 December 2015; pp. 1440–1448.