

## Research Article

# Mobile Robot Exploration by Using Environmental Boundary Information

**Theeraphol Wattanavekin, Taiki Ogata, Tatsunori Hara, and Jun Ota**

*Research Into Artifacts, Center for Engineering (RACE), The University of Tokyo, Kashiwa, Chiba 277-8568, Japan*

Correspondence should be addressed to Jun Ota; [ota@race.u-tokyo.ac.jp](mailto:ota@race.u-tokyo.ac.jp)

Received 4 June 2013; Accepted 19 June 2013

Academic Editors: O. Reinoso Garcia, A. Sabanovic, and C.-C. Tsai

Copyright © 2013 Theeraphol Wattanavekin et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

We present the method of exploration using environmental boundary information for an indoor map generation problem of a mobile robot. We introduce an exploration method by (i) integration of the exploration method with Reaction-Diffusion Equation on a Graph (RDEG) and connected components labeling and (ii) a replanning framework in updating exploration plan for the currently obtained sensor information. Our approach has been implemented in simulation environments and has been compared with two existing methods: frontier-based exploration method and zig-zag method. The results demonstrate the efficiency of our approach over others. Lastly, the approach was implemented and tested on an actual robot, demonstrating its efficiency in a real-world situation.

## 1. Introduction

Currently, an autonomous mobile robot needs to possess three important properties. They are the ability to efficiently explore an environment, to develop a map of the environment, and to automatically self-localize on the map. An exploration plan is the guideline for the robot to identify the location of obstacles, objects, and free spaces effectively by sensing the environment. Robot tasks, such as searching, require time and energy and, thus, an effective exploration plan is required for a robot to complete the task in a fast and energy-efficient manner. Without an efficient exploration plan, the robot might move inefficiently, wasting time and energy in unproductive searching.

Exploration is a basis for many other applications. One important application is mapping an unknown area to create a representation of the environment to be explored. Many successful robot systems rely on maps. The mapping problem is constantly interleaved with decision making involving the next move. A robot moving along the points to observe an environment area in a minimum amount of time is already an NP-hard problem [1–3].

In an actual environment, robots often encounter unanticipated obstacles that make it difficult to complete a task.

To deal with such situations, robots replan exploration paths to avoid obstacles while acquiring a wide and effective sensing range in an unexplored area. In addition, replanning happens several times, since unknown obstacles are frequently encountered in real environments. Therefore, a good exploration plan also includes a good replanning algorithm that must be simple, effective, and computationally efficient.

Here, what we need to consider is that, in many cases, we have the information of environmental boundary in advance, such as cleaning or patrolling tasks. For example, the trajectory using boundary information shown in Figure 1(a) would become shorter and better than that without using the information as shown in Figure 1(b).

From the previous discussion, we can recognize that utilizing the boundary information of environments and efficient replanning of the exploration path based on the obtained obstacle information are important issues for mobile robot exploration.

We classify the exploration into two categories: without environmental boundary information and with the one.

Some exploration plans in the context of mapping are usually constructed without using environmental boundary information such as references [4–7]. One of the famous techniques is frontier-based exploration, which was proposed

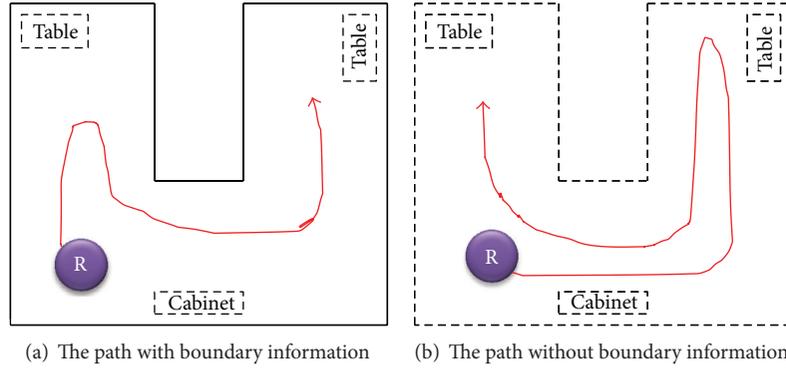


FIGURE 1: Comparison of exploration paths with and without prior knowledge of the boundary information.

by Yamauchi [4]. A lot of researches such as [8–12] extend the work. However, they hold the same idea which is created as an answer to the question, “given what you know about the world, where should you move to gain as much new information as possible?” and its concept is similar to a greedy algorithm by guiding the robot to the next goal which the robot can gain as much new information as possible. Their approach is based on the method of detecting frontiers, which is the border between open space and unexplored space. Although frontier-based exploration is advantageous because of its small computational cost, it has one important disadvantage. The solution to the exploration problem is often far from optimal in most cases. The algorithm itself is greedy because, instead of planning in advance all locations where the robot needs to acquire sensory information, it solely plans one step ahead by determining the next best sensing points that provides new maximum information about the environment. Consequently, the number of positions that need to be reached is usually larger than those in other strategies [13–15], and, in many occasions, the robot explores an area that has already been explored.

On the other hand, some researchers are focusing on exploration by using environmental boundary information. The fundamental one is zig-zag method [16], or some improving methods are also proposed like [17]. These approaches utilize the information of the boundaries of an environment, such as the shape of an office, which can be easily described to effectively generate complete path planning. However, a robot usually explores an environment containing unknown obstacles, and these approaches were not designed for that purpose. Therefore, when a robot encounters unknown obstacles, the predefined path cannot handle the situation, and a new path must be generated, which requires considerable computational cost. Another example is a work by Cao et al. [18]. The authors created regional path planning, which involves a sweeping operation. However, their method also applied the random obstacle avoidance strategy and aims for lawn mower’s general task. This means that their method has some problems with respect to the efficiency of generated path. Some other works by Fukazawa et al. [14] and Trevai et al. [15] also utilize the environmental boundary information for exploration. The algorithm is based on a Reaction-Diffusion Equation on a Graph (RDEG) [19]. The algorithm works by

distributing many observation points of the robot to explore the unknown area. In an unknown environment, a robot frequently encounters unanticipated obstacles that prevent use of a previously planned path. It is necessary to understand the obstacle area and be able to regenerate a new effective path. The authors assume that they already have the shape information of all obstacles in the database. Therefore, when the robot encounters an obstacle, it can immediately grasp the obstacle area and adjust the arrangement of observation points again using the shape information written in the database, which collects all information of all obstacles in the environment. In order to recognize the obstacle area, an artificial mark is utilized. QR codes [20] are attached to all the obstacles in the environment, and a QR reader machine is mounted on the robot to sense the QR codes and recognize the obstacles. However, this is not practical in real-world applications because it is rather troublesome for users to attach the artificial marks on all the obstacles. Also, replanning framework based on this situation alone cannot be used in the real world without prior information of obstacles in the database.

The objective of this study is to develop an exploration method that relies on an environmental boundary and manages unknown obstacles. In addition, the quality of the solution along with its computational cost is considered. Finally, the proposed method must be sufficiently practical to be performed by mobile robots in an actual environment.

In this paper, we introduce an approach to these key problems in an actual mobile robot exploration. Our approach to the problem is based on exploration using RDEG with added obstacle detection and an effective replanning framework. Distributing observation points and efficient planning when encountering an unknown obstacle are used. As described in the related work section, the challenge in this research is presented below:

- (i) handling an unexpected unknown obstacle,
- (ii) constructing an efficient planning framework for an exploration trajectory,
- (iii) integration of exploration and localization.

As for the first challenge, introduction of simple obstacle avoidance when encountered with the obstacles results in defecting its previously planned path inefficiently. In order to

replan effectively, we propose the method that integrates the concept of RDEG and connected component analysis.

For the second challenging point, a new replanning methodology is proposed that includes efficient addition and removal of observation points and rapid path generation. The above two approaches are discussed in Section 3.

As for the third challenging point, we are going to integrate the exploration problem with a Simultaneous Localization and Mapping (SLAM) problem. To succeed in applying our proposed algorithm to an actual mobile robot, we applied one of the solutions to SLAM called Gmapping and integrated it with our proposed algorithm, which will be described in more detail in Section 4.

This paper extends the study of [21] by (a) newly including discussion part of the obtained results as for the scope of the proposed method and (b) including the SLAM problem and experimental results with discussion on applicability of the proposed algorithm to real worlds.

The paper is organized as follows. In Section 1, the importance of robot exploration has been discussed. In Section 2, an explanation of the exploration problem is presented along with its assumptions. In Section 3, we will introduce the method for detecting unknown obstacles and for replanning the path. In Section 4, the simulation and the experimental results of our approach are shown and discussed. In Section 5, the conclusion along with future work is described.

## 2. Problem Formulation and Overview of the Proposed Algorithm

In this section, the exploration problem in our research is formulated. First, the assumptions of the problem are given. Then, the proposed method is roughly described.

### 2.1. Problem Formulation

*2.1.1. Assumption.* Here, we explain the assumptions of the problem regarding both the environment (working area) and the robot used in our research.

#### Assumptions Regarding the Working Area

- (i) Working environment  $W$  is a finite two-dimensional space. Boundary of  $W$  is called environmental boundary.
- (ii) The area to which the robot has access is defined as  $W_s$ .
- (iii) There are nonmoving obstacles in  $W$  whose shapes and positions are either known or unknown.

#### Assumptions Regarding the Mobile Robot

- (i) A mobile robot can localize itself in  $W$ .
- (ii) The shape of a robot is approximated by a circle with radius  $R_{\text{robot}}$ .
- (iii) The sensing area of a robot is described as  $S_{\text{sen}}$ . The shape of the area is a circle with radius of  $r_s$ .

*2.1.2. Definition of Exploration Problem.* In our research, the exploration problem involves robot maneuvers within the work area. The robot can sense the obstacles within its sensing area  $S_{\text{sen}}$ . The shape of the sensing area is a circle whose center is the observation point of the robot. The exploration problem is defined as the problem of covering the working area  $W$  with the circles. Thus, the key of the exploration problem is a plan that can generate the shortest and effective trajectory that connects all the observation points.

The objective is to minimize the sum of the lengths of the exploration path. Because the robot is defined as a circle, the configuration space  $C$  and the free space  $C_{\text{free}}$  could be described as a finite two-dimensional space.

The observation point  $p_i$  ( $i = 1, 2, \dots, N$ ) must satisfy (1), and  $N$  is their total number:

$$\forall p_s \in W_s, \quad \exists i \in \{1, 2, \dots, N\}, \quad \text{s.t.}, \quad \|p_s - p_i\| \leq r_i. \quad (1)$$

The exploration path is the Hamiltonian path that connects  $p_i$  ( $i = 1, 2, \dots, N$ ). The robot's path  $p(t) : [0, 1] \rightarrow W_s$  will be the continuous projections that satisfy

$$\begin{aligned} p(0) &= p(1) = p_{\tau_1} \\ p\left(\frac{i-1}{N}\right) &= p_{\tau_i} \quad (i = 2, 3, \dots, N), \end{aligned} \quad (2)$$

where  $\tau_i$  is the order of traveling of all  $p_i$  ( $i = 1, 2, \dots, N$ ). The path starts from  $p(0)$  and then returns to the initial position at  $p(1)$ .

As the performance index, we set the path length from the initial point to the last point on which the robot explores all  $W_s$ . We intend to create a minimal path length  $L$  using our proposed method and compare that to other methods using this performance index.

An algorithm consisting of the following two steps is proposed: (a) generation of the initial exploration plan and (b) replanning of the exploration path when unknown obstacles are found.

*2.2. Overview of the Proposed Algorithm.* From the previous definition, the problem can be considered as a problem of distribution and connection of observation points in an unknown environment.

In step (a), the initial path is created to explore the entire accessible area  $W_s$ . In this step, the observation points are generated and distributed to cover the area  $W_s$ . The shortest path is then calculated to connect all the points by utilizing the Traveling Salesman Problem (TSP) solver. Here, we adopt the existing method with very small computational cost: RDEG [14, 15, 19] is utilized for distributing the observation points. LKH (Lin Kernighan Heuristic) [22] is adopted as a TSP solver. Please refer to references [14, 15, 21] for details of how to move observation points and how to adopt LKH to this problem.

After generating the initial exploration path, the robot starts to move on it. When the robot encounters an unknown obstacle, the path is regenerated with the replanning process in step (b). The robot then follows the replanned path. In this way, the exploration is conducted. Because it is important

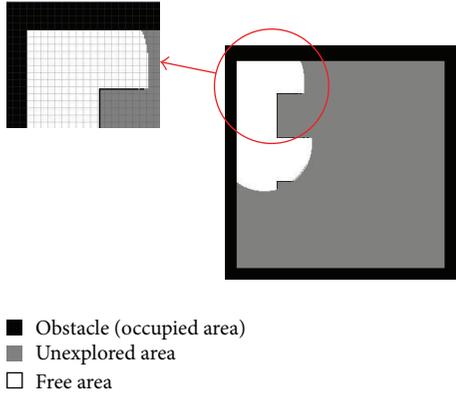


FIGURE 2: Map representation as grid map.

to derive the replanned path quickly, we introduce a new algorithm including both connected component analysis for detecting obstacle area and the fast replanning framework in consideration of known/unknown regions.

### 3. Obstacle Area Detection and Replanning Framework

**3.1. Obstacle Area Detection.** For a robot, detecting the obstacle area is very important because the shape of the environment is complicated. The heart of algorithm is to distribute and move all observation points only within the accessible area, while restricting the movement so as not to go into the obstacle area or not to go outside the given environment boundary.

**3.1.1. Connected Components Labeling.** Connected components labeling [23] is an algorithm that is often used in computer vision to detect connected regions in binary digital images. It is a basic step to segment an image into objects, regions, or blobs. All the pixels within a blob of spatially connected pixels will be assigned the same label. The algorithm can be used to establish boundaries of objects, to count the number of blobs in an image or to understand the desired region like the obstacle area detection in our exploration problem.

**3.1.2. Connected Components Labeling in Exploration Problem.** In our work, when a robot explores an environment, the map of the environment will be generated as a set of grid cells as shown in Figure 2. Each cell has the potential to represent three values. In other words, it can be free, occupied, or of an unknown value.

Figure 3 illustrates all steps in detecting the obstacle area. Figure 3(a) shows current situation. Applying the method in Section 3.1.1 to our problem, 0 is defined as the unknown value when computing connected components labeling in the map, whereas the occupied value is assigned as 1 and free grid cell is defined as 2. Connected components labeling will then be calculated for the cell with the value 1 as reported earlier as shown in Figure 3(b). The rest cells are labeled as 2 as shown in Figure 3(c). Lastly, all labels which are included to be

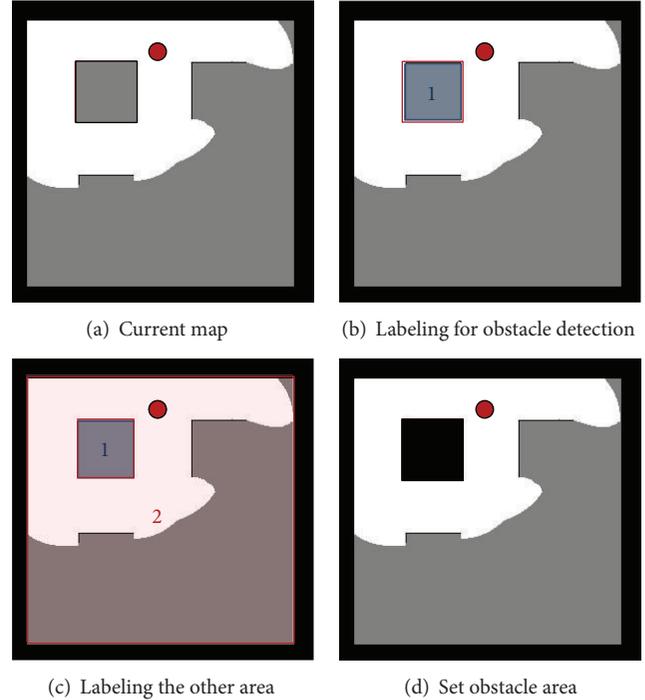


FIGURE 3: Connected components labeling in our exploration problem.

within the obstacle area will be set to be occupied as shown in Figure 3(d) and used in the exploration path replanning step.

When implementing in an actual robot, there are always errors from positioning, sensing, and so forth which cause the robot not to sense the boundaries of obstacles clearly as shown in Figure 4(a). However, obstacle boundary should be measured precisely in order to apply connected components labeling, which means we cannot apply the connected components labeling algorithm to noisy data directly. To solve this problem, we first make dilation procedure with  $n$  times, and next make erosion procedure with  $n$  times before applying the connected components labeling. Note that increasing  $n$  values helps solving the problem from errors. However, using too large  $n$  values could lead to a wrong detection.

**3.2. Replanning Algorithm.** After distributing observation points and generating an exploration path, the robot starts to move along the generated path while updating the map. In the real world environment, the robot often encounters unexpected obstacles and should replan a new path. In order to create the path, the robot needs to have the revised map and replan the path. Earlier, we described the process for detecting the obstacle area. Now, we describe how to replan a new path for the robot.

When an obstacle is encountered, problems can occur and affect the previous planned path. The newly found obstacle could block the sensing area, as shown in Figure 5. Moreover, there may be some unexplored area in the sensing area that is blocked by the obstacle. Figure 5(a) illustrates the observation area containing the unidentified obstacle. When

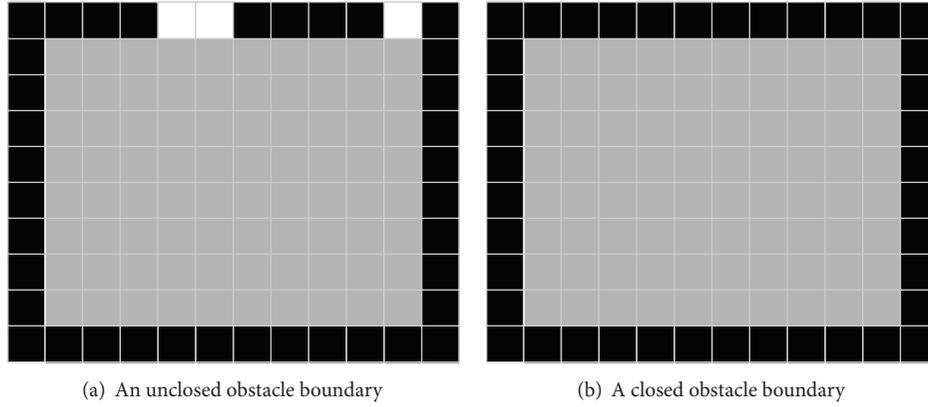


FIGURE 4: Examples of complete and incomplete obstacle boundaries.

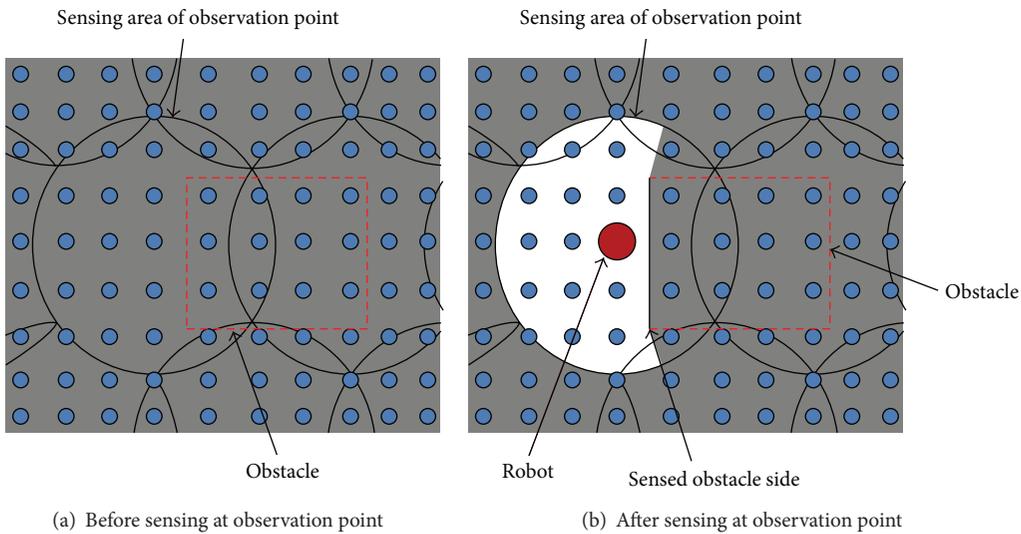


FIGURE 5: Robot's sensing area is blocked by an unexpected obstacle.

the robot visits and senses the environment at this observation point, a part of the sensing area will be blocked by the obstacle, as seen in Figure 5(b). Small circles mean candidate of observation points. Therefore, a replanning algorithm is necessary to explore the uncovered area affected by this newly found obstacle.

Since there are some uncovered grid cells that occur as a result of a newly found obstacle, we must decide whether or not the robot can recognize the shape of the obstacle completely. When it is known, a new working area  $W_s$  will be redefined again as an area that excludes the part that is overlapped with this obstacle area and it will be changed to an inaccessible area as shown in Figure 3(d). In contrast, if it is not known,  $W_s$  remains the same. However, some observation points can have their sensing area blocked, and thus some grid cells are not covered. In both cases, addition, removal, and rearrangement of observation points are necessary in order to adjust themselves to cover all the area of the updated map and in order to replan a new efficient exploration path.

(i) *Moving of Observation Point*: the original path is cut at the moved observation point.

(ii) *Addition of Observation Point*: the original path is cut at the neighbor of the added observation point.

(iii) *Removal of Observation Point*: the original path is cut at the removed observation point.

Figures 6(b), 7(b), and 8(b) show how we eliminate the original path in the three cases. Next, the shortest path that connects to the noneliminated part of the original path is calculated. This kind of problem can be interpreted as a form of the Rural Chinese Postman Problem (RPP) [24]. This problem can be transformed into TSP problem [14, 25]. We can then solve the problem as the TSP using Lin-Kernighan Heuristics. Figure 9 is a flowchart of all steps in the algorithm.

## 4. Simulation and Experimental Results

4.1. *Simulation*. We compare our proposal with two other methods: (1) frontier-based exploration [4] and (2) the zig-zag method [17]. The frontier-based exploration is implemented by setting next target point as the point that has maximum information gain among all the observation points.

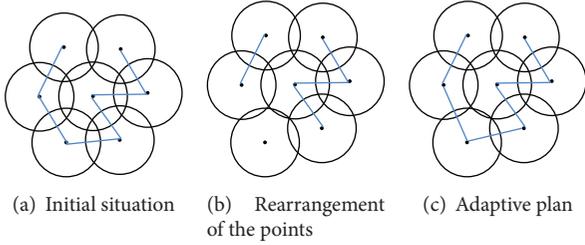


FIGURE 6: Adaptive plan by moving observation points.

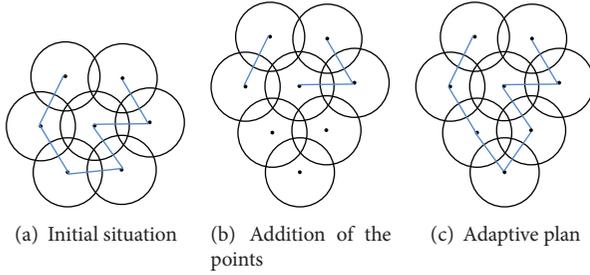


FIGURE 7: Adaptive plan by adding observation points.

For the zig-zag method, the predefined path is generated as a zig-zag shape in a given defined boundary; if the robot encounters an obstacle, simple obstacle avoidance is applied.

We used a robot with a sensing radius of 3.5 m and experimented with 3 simple maps with a size of  $20 \times 20 \text{ m}^2$  depicted in Figures 10(a), 10(b), and 10(c) and a practical map depicted in Figure 10(d). To verify the influence of the proposed method on the efficiency of exploring a boundary environment with unknown obstacles, maps with different boundary environments and unknown obstacles were explored. Figure 11 shows the results as average path lengths of each method for each experiment map. Here, please be careful that we can assume that the path lengths correspond to the search time when omnidirectional mobile robot systems are considered. On average, the proposed method is better than the frontier-based exploration method with 82% of path length and is better than the zig-zag method with 89% of path length. In Map 1, which is the simplest environment without an unknown obstacle, the zig-zag method proved to be better than other two methods. This is because in an environment without an unknown obstacle, the zig-zag method creates the simple back-and-forth motions efficiently. On the other hand, the proposed method uses the distribution of observation points which have some overlapping regions, with longer path length than that of the zig-zag method. The frontier-based exploration results in the worst path length, since the method is greedy and creates a greater number of future goal positions that the robot needs to reach. However, our proposed method has the best result and shortest path lengths of all other maps. Our method also obtained the best result for practical Map 4, which is a typical working environment with one unknown obstacle. The result path of Map 4 for each method is shown in Figure 12.

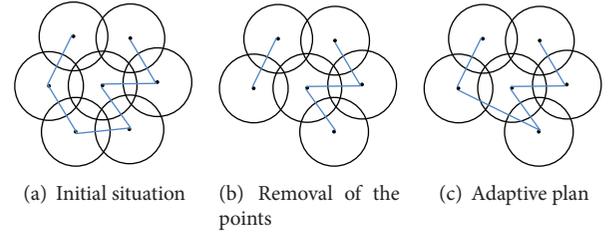


FIGURE 8: Adaptive plan by removing observation points.

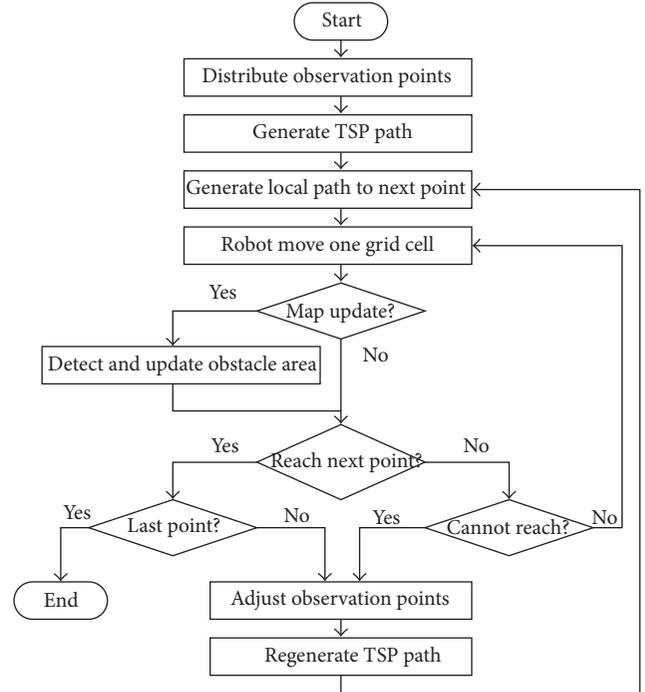


FIGURE 9: Flowchart of the proposed method.

Moreover, to give a quantitative comparison between the proposed method and the previous method, we use the value of sparsity [26], which represents the complexity of each environment. It can be calculated by the following equation:

$$S = 2d_w - 2 \max w_{o_j}, \quad (3)$$

where  $S$  is the sparsity of an environment,  $w_{o_j}$  is the width of obstacle  $o_j$ , and  $d_w$  is the average of the minimum distances to a wall from the border of the Voronoi region for line segments regarding each wall as a line segment. We assume that  $2d_w$  represents the average width between walls and  $\max w_{o_j}$  means the maximum width of all the obstacles. Figure 13 shows the relationship between the average path length divided by vacant area in the environments map and sparsity. From the result, the proposed method overcomes other method when sparsity becomes smaller value. A smaller value of sparsity indicates a more complex environment. This means that there might be many obstacles or that the path in the environment is quite narrow and difficult for the robot to move freely on.

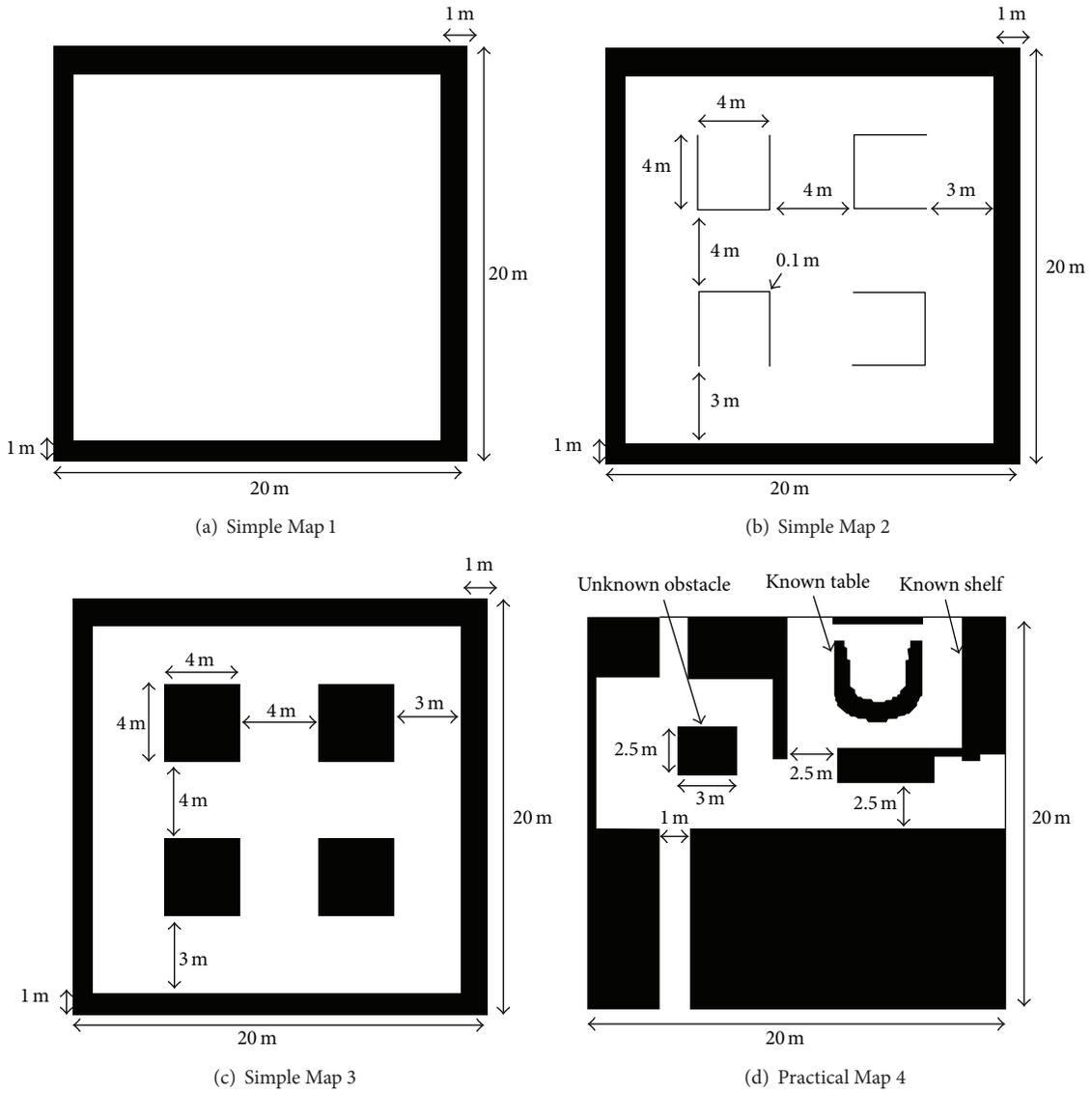


FIGURE 10: Maps for evaluation.

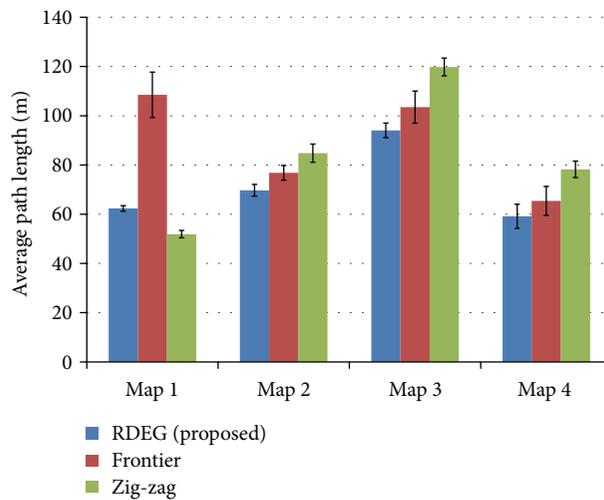


FIGURE 11: Comparison graph of our proposed method, frontier-based exploration, and the zig-zag.

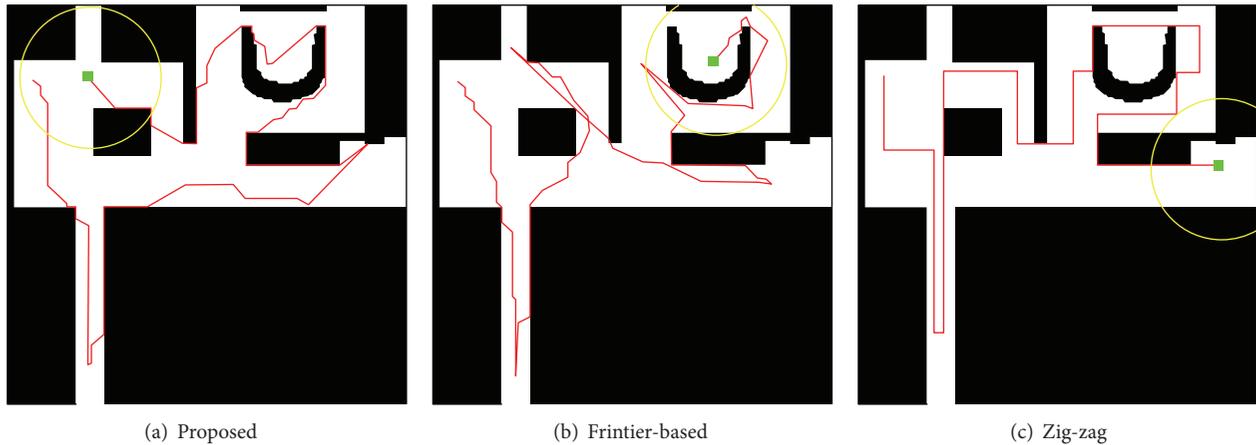


FIGURE 12: Generated path of the three methods on Map 4.

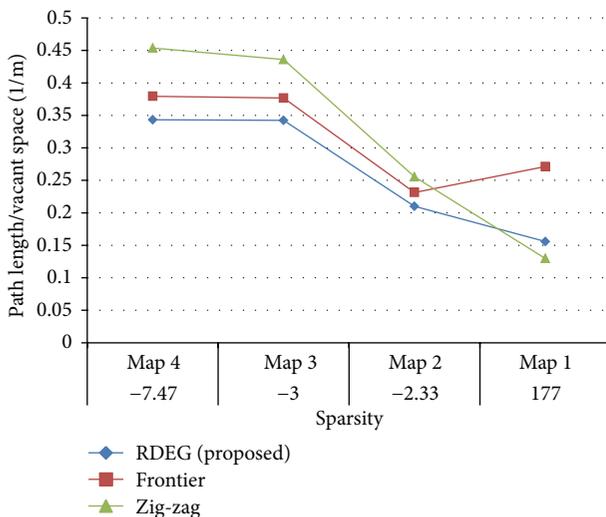


FIGURE 13: Comparison of the three methods.

In conclusion, our proposed method can be used to obtain the best result in almost all maps because it uses the boundary information, while the frontier-based exploration does not. Furthermore, it has a good replanning algorithm and can adapt to an unanticipated obstacle, whereas the zig-zag method cannot. Thus, these different maps demonstrate the superiority of our proposal over other methods for managing a boundary environment with unknown obstacles.

**4.2. Experiments with a Real Robot.** Real robot experiments are conducted to verify that the proposed method can work with the existence of localization error of a mobile robot. The environment is a square room with one box in the center of the floor, as shown in Figure 15(a). Realizing an exploration task in a real-world situation is quite different from doing so in a simulated world. Many issues must be considered, such as positioning errors and sensing errors. Hardware construction is also important. In the following section, we will discuss these issues and how to deal with them.

**4.2.1. Mobile Robot Specifications.** Our experiments were conducted using the mobile robot named Pioneer P3-DX from Adept Mobile Robotics Inc. It has two differential driven wheels and can reach speeds of 1.2 meters per second. Moreover, we attached a laser range finder, that is, UBG-04LX-F01 model from the Hokuyo Company. The notebook computer has a CPU Core i5-520 M (2.4 GHz), and its operating system is Ubuntu Linux (Maverick version). Furthermore, the range of the sensor is less than 360 degrees, which is different from that in the simulation. In order to realize the proposed method, the actual robot needs to rotate around each observation point to gain all information around it.

**4.2.2. Integration of Exploration and SLAM.** To localize the robot itself efficiently, we need an accurate map, and, to succeed in creating a map, we need an accurate robot position, which is called SLAM [27].

Among the many map representation methods, occupancy grid maps are well suited for unstructured environments, as they make no assumption about the structure of the robot surroundings and account for both the sensor and the robot motion uncertainties. Moreover, efficient SLAM methods exist for incrementally constructing the map of the environment as the robot moves. We, therefore, adopt Gmapping SLAM [28] as our mapping and localization algorithm. A map is maintained as a matrix of probabilities of occupancy of grid cells. The map has the information that the robot has gathered during its exploration and is used as the input in the calculation of RDEG and path planning modules of the algorithm.

The Gmapping SLAM uses Rao-Blackwellized particle filters as an effective means with grid maps to solve the SLAM problem. The Gmapping SLAM computes a highly accurate proposal distribution based on the observation likelihood of the most recent sensor information, the odometry, and a scan matching process. Moreover, it applied a selective resampling strategy based on the effective sample size which reduces the number of unnecessary resampling actions in the particle filter. The effectiveness of using Gmapping SLAM can be shown in our real experiment.

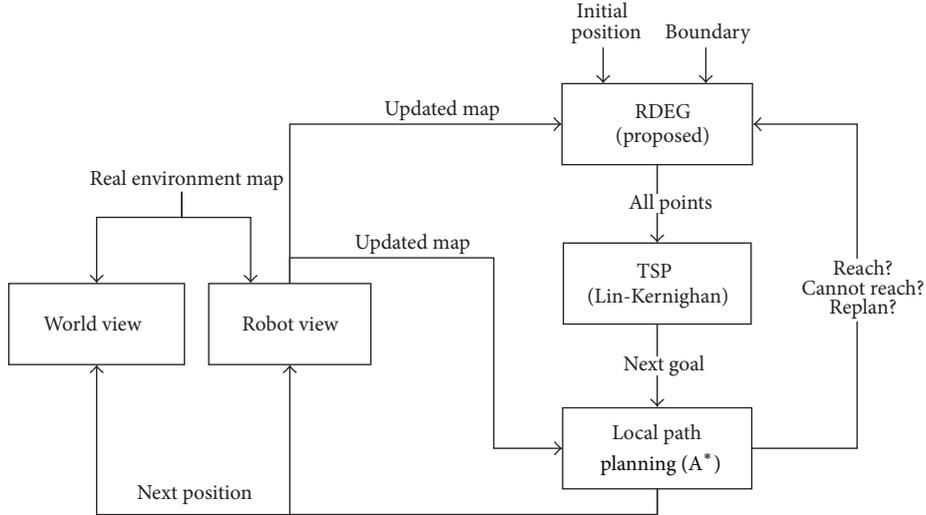


FIGURE 14: Algorithm structure in actual robot system.

**4.2.3. Algorithm Structure in Mobile Robot System.** The core of our mobile robot system is based on the Robot Operating System (ROS) [29]. ROS is an open-source software framework for a robot software development from Willow Garage and was designed to provide an operating system-like functionality. We choose it as our based framework for implementing our proposed algorithm. We implement our algorithm on ROS's node and design all algorithm structure based on the ROS's principle. We utilize the ROS's message-passing structure and connect the created nodes in order to send important information, such as the created map, the robot's position, sensor data, and an observation point's position. All the structure and connections are shown in Figure 14. Integration of exploration and SLAM is realized as follows: SLAM is calculated and map information is revised every time the robot gets sensory information. Map information is utilized for exploration.

**4.2.4. Experimental Results.** We conduct our experiment in a  $5 \times 5 \text{ m}^2$  room as shown in Figure 15(a). This room is not truly square because it has a small square pillar at one corner. Moreover, we add a box as an additional obstacle (beside the pillar at the corner) at the center of the room. The sensing radius is limited to merely 3 m. Note that, in this experiment, the distribution of observation points is done by utilizing the information of initial map. The purpose of this experiment is to show the efficiency of the proposed algorithm in a real situation with errors such as sensing errors, odometry errors, and actuation errors.

The process of experiments is shown below. First, the robot started from the position shown in Figure 15(b). We distribute all observation points within a boundary that we defined. In Figure 15(c), the robot moved to the first observation point while doing SLAM to localize and map the room using the sensor and odometry data. Next, it walked to the second point as shown in Figure 15(d). When it reached the point shown in Figure 15(e), it sensed an obstacle (the square pillar at the corner) which was blocking the sensing area.

This made the original arrangement of observation points impossible to cover the whole room. Therefore, the observation points were rearranged again, which resulted in the addition of a new observation point as shown in Figure 15(f). Subsequently, it continued walking to this new added point but found it to be a dead-end, as shown in Figure 15(g). The algorithm itself detected this obstacle and marked it as an obstacle area. It then deleted this goal and went for the next one as shown in Figure 15(h). The robot reached the next observation point as in Figure 15(i). In the end, the robot continued to move and finished its work at the last position as shown in Figures 15(j) and 15(k).

We conducted several experiments in the same environment and evaluated errors or differences of the created maps compared to the true map of the environment. The error can be calculated as the difference area between the true map area and the created map divided by the true map area. The results show that the error from the created map is around 7.3%, which is enough accuracy for robot's navigation.

## 5. Conclusion and Future Work

We have introduced our approach in exploration problem of bounded region by using RDEG with connected component analysis and replanning algorithm. In both simulation and real experiment the rearrangement of observation points using connected component analysis has been proved to be effective.

Moreover, the replanning algorithm has also been proved effective in dealing with an unexpected obstacle by using the obstacle detection. From all the simulation results, it is found that the proposed method shortens the total average path length to 82% of its value for the frontier-based exploration approach and to 89% for the zig-zag method. The results also show that our method can increase the efficiency of the boundary exploration in mapping problem by utilizing the boundary information of work space and the reasonable

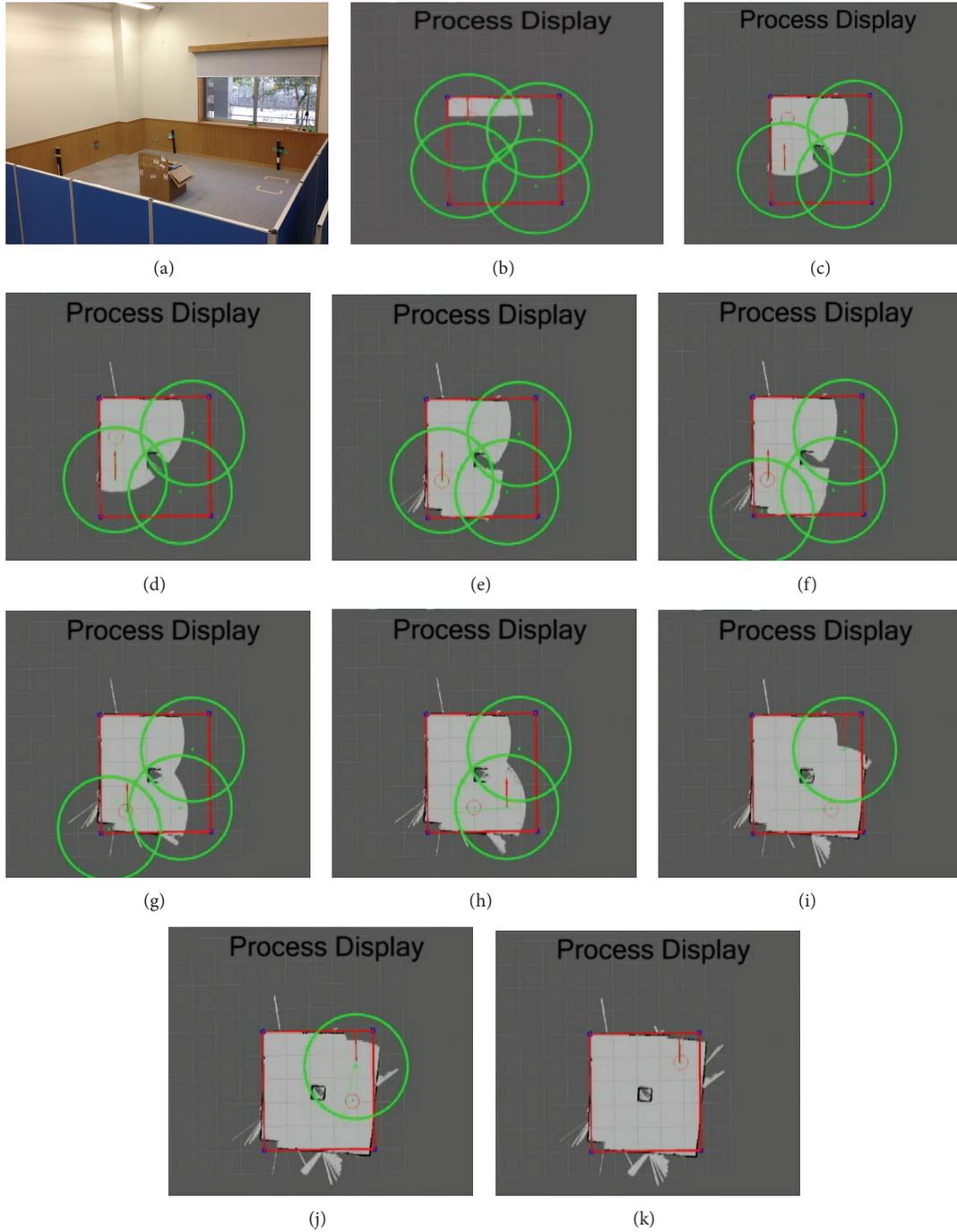


FIGURE 15: Obtained map with a real robot experiment.

replanning algorithm when encountering an unexpected obstacle. The results show that our proposed method can work very well in a complex environment where some unexpected obstacles exist.

In the real experiments with an actual robot, we utilize Gmapping SLAM, which is based on the occupancy grid map for help with localizing and mapping the environment. This has shown a good result from the viewpoint of the generated trajectory of the robot and the created map. Our proposed

algorithm can work very well in such a real environment and is resistant to many errors, such as positioning errors, sensing errors, and boundary size errors.

Future directions of this research are experiments on more complicated environments, more quantitative evaluation of the proposed algorithm in comparison to other existing algorithms than the original frontier-based method, and integration of this algorithm with transportation tasks and extension to multirobot systems.

## References

- [1] E. L. Lawler, J. K. Lenstra, A. H. G. Rinooy Kan, and D. B. Shmoys, *The Traveling Salesman Problem—A Guided Tour of Combinatorial Optimization*, Wiley-Interscience, New York, NY, USA, 1985.
- [2] J. K. Lenstra and A. H. G. R. Kan, “On general routing problems,” *Networks*, vol. 6, pp. 273–280, 1976.
- [3] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP Completeness*, W.H. Freeman, New York, NY, USA, 1979.
- [4] B. Yamauchi, “Decentralized coordination for multirobot exploration,” *Robotics and Autonomous Systems*, vol. 29, no. 2, pp. 111–118, 1998.
- [5] H. Choset, *Sensor based motion planning: the hierarchical generalized Voronoi Graph [Ph.D. thesis]*, California Institute of Technology, Pasadena, Calif, USA, 1996.
- [6] R. Simmons, D. Apfelbaum, W. Burgard et al., “Coordination for multirobot exploration and mapping,” in *Proceedings of the 17th National Conference on Artificial Intelligence (AAAI '00)*, pp. 852–858, September 2000.
- [7] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun, “Collaborative multi-robot exploration,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, pp. 476–481, April 2000.
- [8] A. Mobarhani, S. Nazari, A. H. Tamjidi, and H. D. Taghirad, “Histogram based frontier exploration,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS '11)*, pp. 1128–1133, September 2011.
- [9] B. Tovar, L. Muñoz-Gómez, R. Murrieta-Cid, M. Alencastre-Miranda, R. Monroy, and S. Hutchinson, “Planning exploration strategies for simultaneous localization and mapping,” *Robotics and Autonomous Systems*, vol. 54, no. 4, pp. 314–331, 2006.
- [10] L. Freda and G. Oriolo, “Frontier-based probabilistic strategies for sensor-based exploration,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '05)*, pp. 3881–3887, Barcelona, Spain, April 2005.
- [11] A. Mobarhani, S. Nazari, A. H. Tamjidi, and H. D. Taghirad, “Histogram based frontier exploration,” in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems: Celebrating 50 Years of Robotics (IROS '11)*, pp. 1128–1133, San Francisco, Calif, USA, September 2011.
- [12] R. A. Prieto, J. M. Cuadra-Troncoso, J. R. Álvarez-Sánchez, and I. N. Navarro-Santosjuanes, “Reactive navigation and online SLAM in autonomous frontierbased exploration,” in *Natural and Artificial Computation in Engineering and Medical Applications*, vol. 7931 of *Lecture Notes in Computer Science*, pp. 45–55, Springer, Berlin, Germany, 2013.
- [13] W. Burgard, M. Moors, C. Stachniss, and F. E. Schneider, “Coordinated multi-robot exploration,” *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376–386, 2005.
- [14] Y. Fukazawa, T. Chomchana, J. Ota, H. Yuasa, T. Arai, and H. Asama, “Region exploration path planning for a mobile robot expressing working environment by grid points,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '03)*, pp. 2448–2454, Taipei, Taiwan, September 2003.
- [15] C. Trevai, K. Ichikawa, Y. Fukazawa et al., “Real-time cooperative exploration by reaction-diffusion equation on a graph,” in *Proceedings of the International Symposium on Distributed Autonomous Robotic Systems (DARS '02)*, pp. 383–392, 2002.
- [16] J.-C. Latombe, *Robot Motion Planning*, Kluwer Academic, New York, NY, USA, 1991.
- [17] H. Choset, E. Acar, A. A. Rizzi, and J. Luntz, “Exact cellular decompositions in terms of critical points of Morse functions,” in *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '00)*, pp. 2270–2277, April 2000.
- [18] Z. L. Cao, H. Huang, and E. L. Hall, “Region filling operations with random obstacle avoidance for mobile robots,” *Journal of Robotic Systems*, vol. 5, no. 2, pp. 87–102, 1988.
- [19] H. Yuasa and M. Ito, “Internal observation systems and a theory of reaction-diffusion equation on a graph,” in *Proceedings of the 1998 IEEE International Conference on Systems, Man, and Cybernetics*, pp. 3669–3673, October 1998.
- [20] QR Code.com, <http://www.qrcode.com/en/index.html>.
- [21] T. Wattanavekin and J. Ota, “Exploration in a boundary environment with unknown obstacles using Reaction-Diffusion equation on a graph,” in *Proceedings of the IEEE International Conference on Robotics and Biomimetics*, pp. 2623–2628, Tokyo, Japan, 2011.
- [22] S. Lin and B. W. Kernighan, “An effective heuristic algorithm for the traveling-salesman problem,” *Operations Research*, vol. 21, no. 2, pp. 498–516, 1973.
- [23] A. Rosenfeld and J. L. Pfaltz, “Sequential operations in digital processing,” *Journal of the ACM*, vol. 13, pp. 471–494, 1966.
- [24] H. A. Eiselt, M. Gendreau, and G. Laporte, “Arc routing problems, part II: the rural postman problem,” *Operations Research*, vol. 43, no. 3, pp. 399–414, 1995.
- [25] M. J. Todd, G. L. Lemhauser, and A. H. G. R. Kan, “The traveling salesman problem,” in *Handbooks in Operations Research and Management Science*, vol. 7, pp. 527–562, Elsevier Science, Amsterdam, The Netherlands, 1989.
- [26] N. Fujii, R. Inoue, Y. Takebe, and J. Ota, “Multiple robot rearrangement planning using a territorial approach and an extended project scheduling problem solver,” *Advanced Robotics*, vol. 24, no. 1-2, pp. 103–122, 2010.
- [27] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*, The MIT press, Cambridge, Mass, USA, 2005.
- [28] G. Grisetti, C. Stachniss, and W. Burgard, “Improved techniques for grid mapping with Rao-Blackwellized particle filters,” *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, 2007.
- [29] M. Quigley, B. Gerkey, K. Conley et al., “ROS: an open-source robot operating system,” in *Proceedings of the OpenSource Software Workshop of International Conference on Robotics and Automation (ICRA '09)*, 2009.

