Multi-Robot Exploration and Mapping with a rotating 3D Scanner ⋆

Mohammad Al-khawaldah * **Andreas Nüchter** **

* *Faculty of Engineering Technology-Albalqa Applied University, Jordan* mohammad.alkhawaldah@gmail.com

** *Jacobs University Bremen gGmbH, Automation Group, Campus Ring 1, D-28759 Bremen, Germany*
a.nuechter@jacobs-university.de

Abstract: This paper investigates the field of exploration and map-building with multiple cooperating mobile robots. New and efficient exploration and mapping technique is proposed by employing laser scanners. The paper also aims to extend existing exploration and mapping techniques of single robot to multi-robot to increase the exploration efficiency (i.e. to reduce the environment exploration time required). The goal of the proposed method is to have multiple mobile robots exploring a given unknown environment as fast as possible, while coordinating their actions and sharing their local maps in certain time instances. In the suggested technique, each robot is equipped with a laser scanner that is continuously rotating to scan the environment, and is employing a frontier-based exploration algorithm which is important to guide the robots during the exploration. A new factor is introduced to enhance the performance of the frontier-based exploration. This factor aims at spreading robots in the environment to reduce overlap.

Keywords: Laserscanner, Multi-robot, Exploration, Simulation, Frontier.

1. INTRODUCTION

The exploration and map building of an unknown environment is a very important topic in mobile robot research because of its wide range of applications such as reconnaissance (Albers and Henzinger (2000)), planetary exploration (Al-khawaldah et al. (2010); Burgard et al. (2005)) search and rescue (Cao et al. (1997)), military actions, hazardous material handling, cleaning, mowing and harvesting. Due to such important applications, the field of exploration is intensively studied and new techniques are developed continuously.

Systems employing multi-robots have several advantages over single robot systems. Firstly, cooperating robots can accomplish a single task quicker than a single robot. Also, redundancy introduced by multiple robots makes the system more fault-tolerant than those with a single robot. Finally, information overlapping in multi-robot systems helps to compensate sensor uncertainties. For example, a team of robots localizes themselves more precisely, especially when they have different sensor capabilities. On the other hand, when robots operate in teams there is the risk of possible interference between them. For example, if the robots use the same type of sensors, the overall performance is expected to be degraded because of cross-talk between the sensors. In addition, as the number of robots increases, longer detours become necessary to avoid collisions with other members of the team as Cao et al. (1997) and Schneider-Fontan and Mataric (1998) report. To perform tasks in unknown environments, robots should be able to gather information and understand their surroundings. Some environments are hostile and not accessible, and it is therefore necessary to use robots in order to avoid risking human lives. In some applications, like planetary exploration, map-building is the main aim. While in some other cases (e.g. navigation and planning) generating a map of the environment is required for other goals. There are cases in which it is desired to minimize repeated coverage to accelerate the mission, while in cases of dynamic environments repeated coverage may be desirable. To effectively explore an unknown environment, it is important for an exploration system to be reliable and robust (Burgard et al. (2005); Cao et al. (1997)).

This paper addresses the problem of finding a good exploration strategy for multiple mobile robots equipped with continuously rotating 3D scanner. Figure 1 shows the mobile robot Irma3D with its rotating laser scanner, a RIEGL VZ-400 (see Digor et al. (2010)) which continuously rotates around the vertical axis and is therefore capable of acquiring 3D scans while in motion. In this paper we develop a simulation-based evaluation testbed that allows us to quickly evaluate different multi-robot exploration strategies while considering kinematic motion constraints.

2. RELATED WORK

Exploration of unknown environments with team of mobile robots has received considerable importance recently. A seminal solution for this problem was introduced by Yamauchi (1997) who devised a technique to build maps for unknown terrains with a team of mobile robots. He proposed the concept of frontier cells (frontiers) which are

⋆ This work was partially supported by DFG under the GZ number NU230/7-1

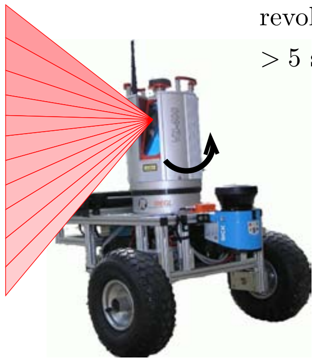

Fig. 1. The mobile robot Irma3D with its sensor: RIEGL VZ400, SICK LMS100, xsens gyro and wheel encoders. The VZ400 needs at least 6 seconds for one revolution.

the borders between known and unknown areas in a grid map. His technique is still widely used to select potential target locations for the robots during exploration. Burgard et al. (2000) suggested a useful extension of Yamauchi's technique in which each robot is directed to a frontier cell. The idea is to specify how to assign frontier cells to the individual robots. The goal is to avoid several of robots moving to the same location. The technique considers the cost of reaching a frontier cell and the utility of that cell. For each robot, the cost of a cell is a function of the distance between the robot and that cell. The utility of a frontier cell is a function of the number of robots that are moving to that cell.

In the further research of Grabowski et al. (2000) an exploration algorithm for a team of mobile robots is proposed that exchange mapping and sensor information. In this system, one robot plays the role of team leader that integrates the information gathered by the other individual robots. This team leader controls the movement of other robots to unknown areas.

In the above mentioned published works the proposed exploration algorithms do not account for a continuously rotating laser scanners to increase the exploration efficiency. In addition, we think that there should be more efficient way to further reduce the overlap between the team members. Finally, none of these algorithms has studied the weight parameters used in its bidding function.

3. LASER SCANNER AND ROBOT MODEL

In our experimentations, we continue to use the same robot and scanner models used in the work of Digor et al. (2010). Moreover, the same idea of continuously rotating laser scanner is used. The main difference from their work is that the work presented in this paper uses these ideas with multiple robots. Another novelty is introduced in our algorithms to reduce overlap among robots to decrease the exploration time. Furthermore, we have investigated the effect of using the utility factor on the exploration time.

A basic simulation framework is constructed to simulate the constantly rotating scanner and the mobile robot. Scanning is the central part of exploration missions. Our simulator is 2D Netlogo (Wilensky (2000)). We simulate 72 scans per second. Therefore, a full 360° scan takes 5 seconds, which corresponds to our used hardware the Riegl VZ400, which originates from geodetic surveying. The Riegl VZ400 scanner is a 3D scanner that produces high-precise 3D point clouds. Faster scanning is not supported by the hardware, while the rotation speed can be reduced to yield high-density range values. Typical coarse indoor scans yield 300.000 points, while 22.500.000 points are obtained when the scan time is adjusted to 3 minutes. For the initial study in this paper, we restrict the exploration to the horizontal beam, thus we produce 2D maps that represents a slice through the environment. By adjusting the length of a beam, we can easily calculate a far-most point (in our coordinate system) for each scan line. At every time step, the exploration algorithm has to mark all grid cells starting with the current robot position and ending either with far-most point or at the closest encountered obstacle.

Irma3D is a differential drive robot that can rotate on the spot. In principle the robot is cabable to execute motions that compensate the rotation of the scanner. We simulate the kinematics of its differential drive similarly to the work of Digor et al. (2010).

4. EXPLORATION STRATEGIES

The majority of related published works employ the frontier-based algorithm for the motion strategy, e.g., Fox et al. (2005); Grabowski et al. (2000); Rocha et al. (2005); Burgard et al. (2000); Zipparo et al. (2007); Thrun (2001); Al-khawaldah et al. (2010); Yamauchi (1997). In the here proposed technique, each robot chooses one of the frontier cells to be its next target. The wining frontier cell is chosen based on to the following three factors:

(1) The distance of the robot to the frontier cell.
(2) The distance of target cells of the other robots' to the frontier cell.
(3) The size of the environment that is expected to be explored when the robot gets to the frontier cell, i.e., the information gain.

The following subsections give a detailed explanation of the exploration strategies presented in this paper.

4.1 Stop-scan-replanning-go

This algorithm proceeds as follows:

(1) Each Robot scans 360° in five time steps (72 degree in each step) before starting to move. The new data are then published to other robots.

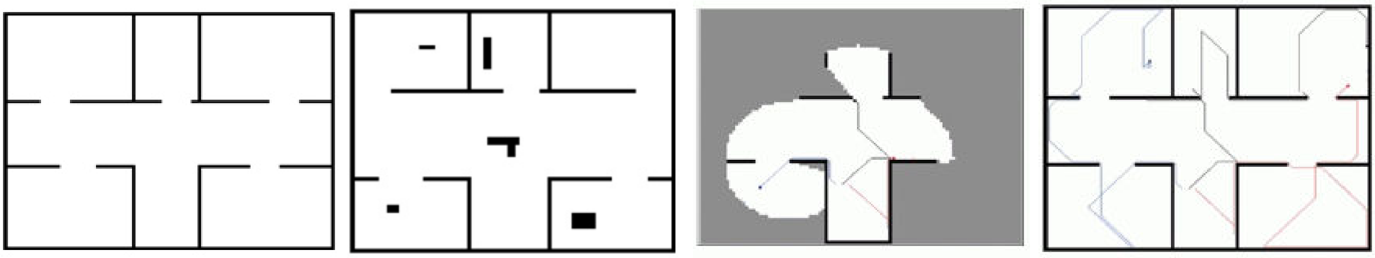

Fig. 2. From left to right: (1) The environment used for testing the exploration algorithms without obstacles and (2) with obstacles. (3) Simulation snapshot for the exploration algorithms with three robots. (4) Simulation snapshot for the exploration with three robots taken just after finishing the exploration.

(2) The robot performs the frontier selection procedure according to the bidding function in equation (1). Robots are encouraged to spread in the environment to reduce overlap.

$$B_i = W_n N_u + W_P D_p - W_c D_r \quad (1)$$

Where,

- B_i is the bidding value for the frontier cell i,
- N_u is the unexplored area that is expected to be explored when the robot gets to the frontier cell. This parameter represents the utility of a frontier cell. It decreases as the number of explored cells close to the frontier cell increases. In our experiments, it is calculated by subtracting the number of explored cells within a circle of diameter of n cells centered on the frontier cell from the whole circle area, the result is the frontier cell utility N_u (here n=40).
- D_p is the distance between the frontier cell and the closest target cell of other robots, (This parameter helps to spread the robots in the environment. In particular, the robot is encouraged to go to areas that other robots are not travelling to. It would not be beneficial to direct a robot to explore an area close to a target cell of other robot. It would be more efficient to make only one robot explore that part of the environment.)
- D_r is the distance of the robot to the frontier cell,
- W_n is the weight neighbours,
- W_p is the weight partner, and
- W_c is the weight costs of the weight factors for N_u, D_p, and D_r, respectively.

The frontier cell with maximum bidding value wins the bidding. Once the winner target cell is assigned, the coordinates of this target cell is published to other robots. Some robots might receive this coordinates while travelling or standing on another target cell.

(3) The robot starts moving to its goal (winner frontier cell), while doing so, it performs scanning (72° in each time-step and in each time-step it travels one cell).

(4) When it reaches its goal target, robot scans complete 360° degree (in five time steps). Finally, the new information the robot collected during its journey and during standing on the goal frontier cell is broadcasted to the other robots. In particular, each robot publishes information about the scanned cells in this step. This information includes the coordinates and the results of the scanning (zero if the cell is free and one if the cell is occupied) of each of the scanned cells in this step. (The new information is only available to other robots after the robot sends these data to its partners, this takes place only after robot finishes its complete 360° scan on its target cell)

(5) After broadcasting its new information to other partners, robot starts new bid.

4.2 Scan-replanning-go

This algorithm is similar to the previous one but the robot does not stop to perform a 360° scan when it reaches its target cell. Alternatively, it instantly computes the new target cell (through the bidding function computed by equation (1) and starts travelling toward it. The new information is only available to other robots after the robot sends these data to its partners, this takes place only after the robot reaches its target cell.

4.3 Stop-scan-plan-go

In this strategy, robot stops on its frontier target cell and stay there until it performs a complete 360 scan. Then, it computes the bidding value for each of the frontier cell. The frontier cell with maximum bidding value wins and the robot starts moving toward it. While in motion, robot does not perform scanning. This is the main difference between this strategy and the stop-scan-replanning-go mentioned in section 4.1. For the strategies in section 4.1 (stop-scan-replanning-go) and in 4.2 (scan-replanning-go) the laser scanner is continuously rotating.

5. SIMULATION EXPERIMENTATION

The experimentations started with the well-known approach, stop-scan-plan-go method, which is an extension of art gallery problem (see O'Rourke (1987) for details). This third approach does not employ the continuous rotating scanner while the robot in motion. Alternatively, the scanner rotates (scans) only when the robot reaches its target cell. We introduced this approach here just for comparison purposes. The comparison will show the effectiveness of the continuous rotating scanner approach.

Figure 2 (1) and (2) shows the environment that used for testing our algorithms. Each one of the three algorithms: stop-scan-replanning-go and scan-replanning-go in addition to the classical stop-scan-plan-go, are tested as follows (Figure 2 (3) and (4) show simulation snapshots during and after the completion of the exploration respectively).

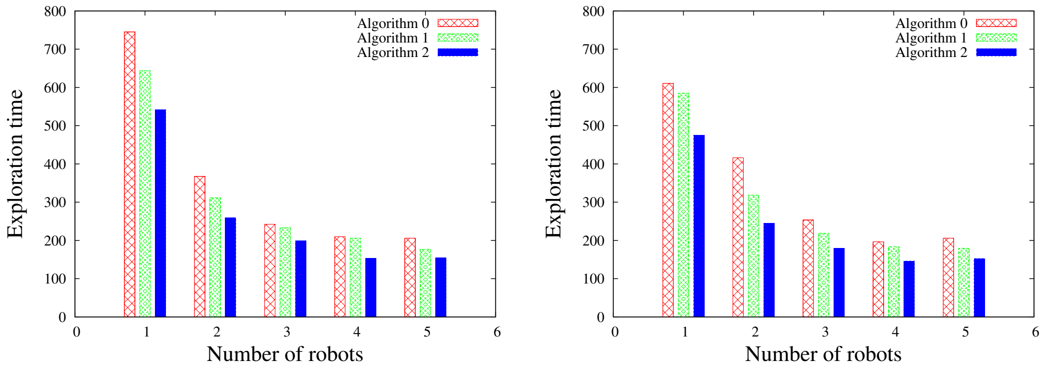

Fig. 3. Exploration time (time steps) versus number of robots when utility weight is set to zero (left) and when utility weight is set to 0.2 (right).

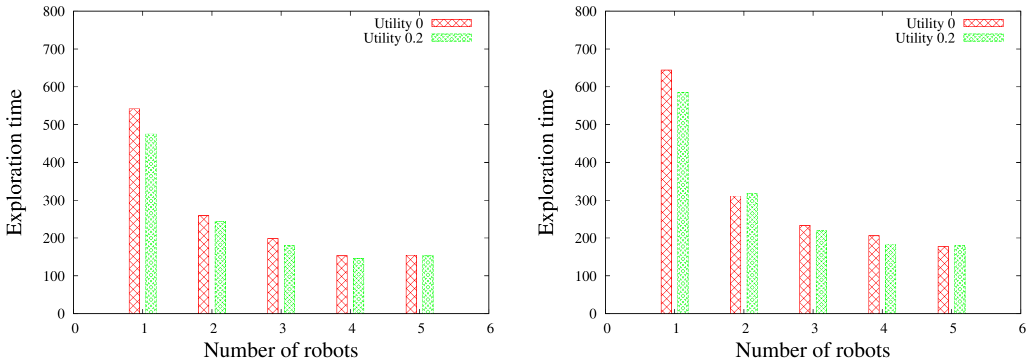

Fig. 4. Exploration time (time steps) versus number of robots for algorithm 1 (left) and algorithm 2 (right). Red bars represent the results when utility weight = 0, green bars represent the results when utility weight = 0.2.

5.1 Rotating speed = 72° per second

The rotating speed of the laser scanner is initially set to 72° per second. The exploration experiments were run as follows:

(1) With the weight of the utility switched to zero, each algorithm is tested with different numbers of robots (1 to 5) then the experiment is repeated five times and the average time to complete the exploration is recorded. For instance, stop-scan-replanning-go algorithm was tested with one robot, then this experiment was repeated five times, finally the average time to complete the exploration is recorded. Then it is tested with two robots and repeated five times, and as before, the average time is recorded. This procedure is repeated until the number of robots is five. Same procedure is repeated for the other algorithms. The results are shown in Figure 3 (left).

(2) Same procedure as in (1) was repeated with the weight of the utility switched to 0.2. The results are show in Figure 3 (right).

Figure 3 (left) shows the results of the exploration runs when utility is ignored (utility weight switched to zero). Algorithm0 stands for the classical art gallery algorithm (stop-scan-plan-go), Algorithm1 stands for the proposed stop-scan-replanning-go algorithm and finally Algorithm2 stands for the proposed scan-replanning-go algorithm.

It is clear that the exploration time for the two proposed algorithms stop-scan-replanning-go and scan-replanning-go is less than the exploration time of classical stop-scan-plan-go. It is also clear that scan-replanning-go is faster than stop-scan-replanning-go. This appears to be due to the fact that performing complete scan for 360° in the frontier cell is time consuming and not important.

Figure 3 (right) shows the results of the exploration runs when utility is not ignored (utility weight switched to 0.2). As before, the exploration time for the two proposed algorithms stop-scan-replanning-go and scan-replanning-go is less than the exploration time of classical stop-scan-plan-go. It is also clear that scan-replanning-go is faster than stop-scan-replanning-go for the same reason mentioned above.

Figure 4 focus on the effectiveness of involving the utility factor in the exploration algorithms. Figure 4 (left) shows the effect of involving the utility factor for Algorithm1 (stop-scan-replanning-go). It is clear that including this parameter improves the performance by reducing the exploration time. Figure 4 (right) shows the effect of involving the utility parameter for Algorithm2 (scan-replanning-go). As in algorithm1, involving this parameter improves the performance by reducing the exploration time.

5.2 Rotating speed = 18° per second

The rotating speed of the laser scanner is now set to 18° per second to investigate environment digitalization in a higher resolution. A number of exploration experiments were run as follows:

(1) With the weight of the utility switched to zero, the proposed algorithms were tested with different numbers of robots, again 1 to 5, then each experiment is repeated five times and the average time to complete the exploration is recorded. The results are shown in Figure 5.
(2) Same experiments mentioned above were repeated in the same environment but with some obstacles added to the environment as shown in shown in Figure 2. The results are shown in Figure 5.

Figure 5 (1) shows the time that the proposed algorithms require to explore the environment that has no obstacles. While Figure 5 (2) shows the time that the proposed algorithms require to explore the environment that has number of obstacles. Figure 5 (1) shows that the exploration time for Algorithm1 is much more than the exploration time for Algorithm2. As before, this appears to be due to the fact that performing complete scan for 360° in the frontier cell is time consuming and not important.

Figure 5 (3) compares between the exploration times of Algorithm1 when the environment has no obstacles and when the environment has number of obstacles. Similarly, Figure 5 (4) compares between the exploration times of Algorithm2 when the environment has no obstacles and when the environment has number of obstacles. It is clear that for both algorithms, the time required to explore an environment with obstacles is slightly more than the time required to explore the environment with obstacles. This appears to be due to the fact that the obstacles obstruct the laser rays preventing them from scanning more areas.

5.3 Rotating speed = 7.66° per second

The exploration experiments were repeated with a rotating speed of 7.66° per second to test mapping with even higher resolutions. Figure 6 gives the results, similar to the previous subsection.

6. FURTHER RESULTS AND CONCLUSION

To see the effect of the two proposed algorithms on the robot trajectories for exploration of an environment without obstacles and with some obstacles, see Figure 7. It can be seen that Algorithm 2 leads to more *nervous* trajectories than Algorithm 1. It is clear that this nervousness increases when the environment has some obstacles. As claimed by Fekete et al. (2006) the search strategy and "'How to look around the corner"' are crucial. Future work will concentrate of this aspect.

This paper makes advantage of a constantly rotating laser scanner. We developed and tested in simulation two new exploration strategies which are based on the frontier approach combined with an extension of food fill algorithm. One of the algorithms involves stopping at frontier points to take full 360° scans of the environment, and the other one implied constant movement until the entire map is covered. From the results of our experiments the following conclusions could be drawn:

(1) Employing continuously rotating scanners for multi-robot systems improves the exploration efficiency by reducing the exploration time. The comparison with the classical exploration methods shows the obtained effectiveness.
(2) As in single robot exploration, scan-replanning-go algorithm is faster than stop-scan-replanning-go (i.e., full 360° scans in the frontier cells seems to be time consuming).
(3) In single or multi-robot exploration, utility factor is better to be included in these algorithms. However, the effect of this factor is clearer for one or small number of robots, this appears to be due to the fact that when several robots explore an environment, it is expected to have more overlap. In this case, more weight is proposed to be given to the factor that keeps robot away from each other, especially when the environment size is large.
(4) More robots lead to less exploration time. But after certain number of robots, exploration time seems to be the same. This is due to the fact that overlap is directly proportional to the number of robots, especially when they start from adjacent positions.

REFERENCES

Al-khawaldah, M., Livatino, S., and Lee, D. (2010). Reduced Overlap Frontier-based Exploration with Two Cooperating Mobile Robots. In *Proc. of the IEEE Int. Symp. on Ind. El. (ISIE '10)*, Italy.

Albers, S. and Henzinger, M. (2000). Exploring unknown environments. *SIAM Journal on Computing*, 29, 1164–1188.

Burgard, W., Moors, M., Fox, D., Simmons, R., and Thrun, S. (2000). Collaborative multi-robot exploration. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, San Francisco, CA, USA.

Burgard, W., Moors, M., Stachniss, C., and Schneider, F.E. (2005). Coordinated multi-robot exploration. *IEEE TRO*, 21(3), 376–386.

Cao, Y., Fukunaga, F., and .-Kahng, A.B. (1997). Cooperative mobile robotics: antecedents and directions. *Journal Autonomous Robots*, 4(1), 7–27.

Digor, E., Birk, A., and Nüchter, A. (2010). Exploration strategies for a robot with a continously rotating 3d scanner. In *Proc. SIMPAR '10*, Darmstadt, Germany.

Fekete, S.P., Klein, R., and Nüchter, A. (2006). Online searching with an autonomous robot. *Computational Geometry: Theory and Applications (CGTA)*, 34(2), 102–115.

Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., and Stewart, B. (2005). Distributed multirobot exploration and mapping. *Proceedings of the IEEE*, 94(7), 15–25.

Grabowski, R., Navarro-Serment, L., Paredis, C., and Khosla, P. (2000). Heterogeneous teams of modular robots for mapping and exploration. *Journal Autonous Robots*, 8(3), 293–308.

O'Rourke, J. (1987). *Art Gallery Theorems and Algorithms*. Oxford University Press, New York, USA.

Rocha, R., Dias, J., and Carvalho, A. (2005). Cooperative multi-robot systems: A study of vision-based 3-d mapping using information theory. *J. Robotics and Autonomous Systems*, 53(3-4).

Schneider-Fontan, M. and Mataric, M. (1998). Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation (TRO)*, 14(5), 815–822.

Thrun, S. (2001). A probabilistic online mapping algorithm for teams of mobile robots. *Int. J. Robotics Research*, 20(5), 335–363.

Wilensky, U. (2000). Netlogo, `http://ccl.northwestern.edu/netlogo/`, center for connected learning and computer-based modeling, northwestern university. evanston, il.

Yamauchi, B. (1997). A frontier-based approach for autonomous exploration. In *Proceedings of the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA '97)*, 146–151.

Zipparo, V., Kleiner, A., Nebel, B., and Nardi, D. (2007). Rfid-based exploration for large robot teams. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA '07)*, 4606–4613. Rome, Italy.

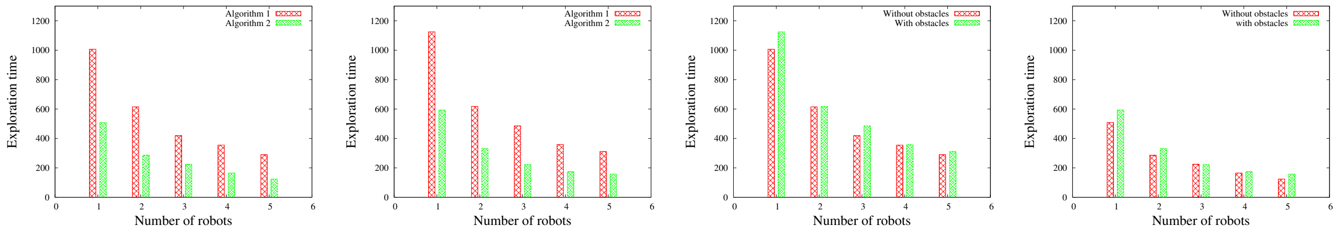

Fig. 5. From left to right: (1) Exploration time (time steps) versus number of robots for the environment without obstacles. (2) Exploration time (time steps) versus number of robots for the environment with obstacles. (3) Exploration time (time steps) versus number of robots for Algorithm 1. Red bars represent the results when the environment has no obstacles, green bars represent the results when the environment has number of obstacles. (4) Exploration time (time steps) versus Number of Robots for Algorithm 2.

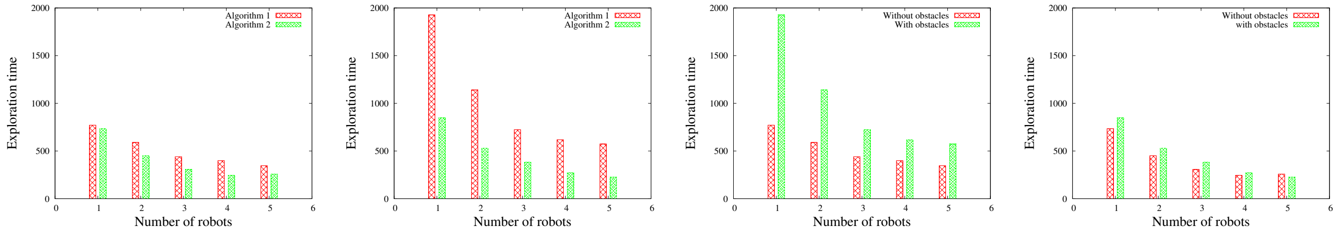

Fig. 6. From left to right: (1) Exploration time (time steps) versus number of robots for the environment without obstacles. (2) Exploration time (time steps) versus number of robots for the environment with obstacles. (3) Exploration time (time steps) versus number of robots for algorithm 1. Red bars represent the results when the environment has no obstacles, green bars represent the results when the environment has number of obstacles. (4) Exploration time (time steps) versus Number of Robots for Algorithm 2.

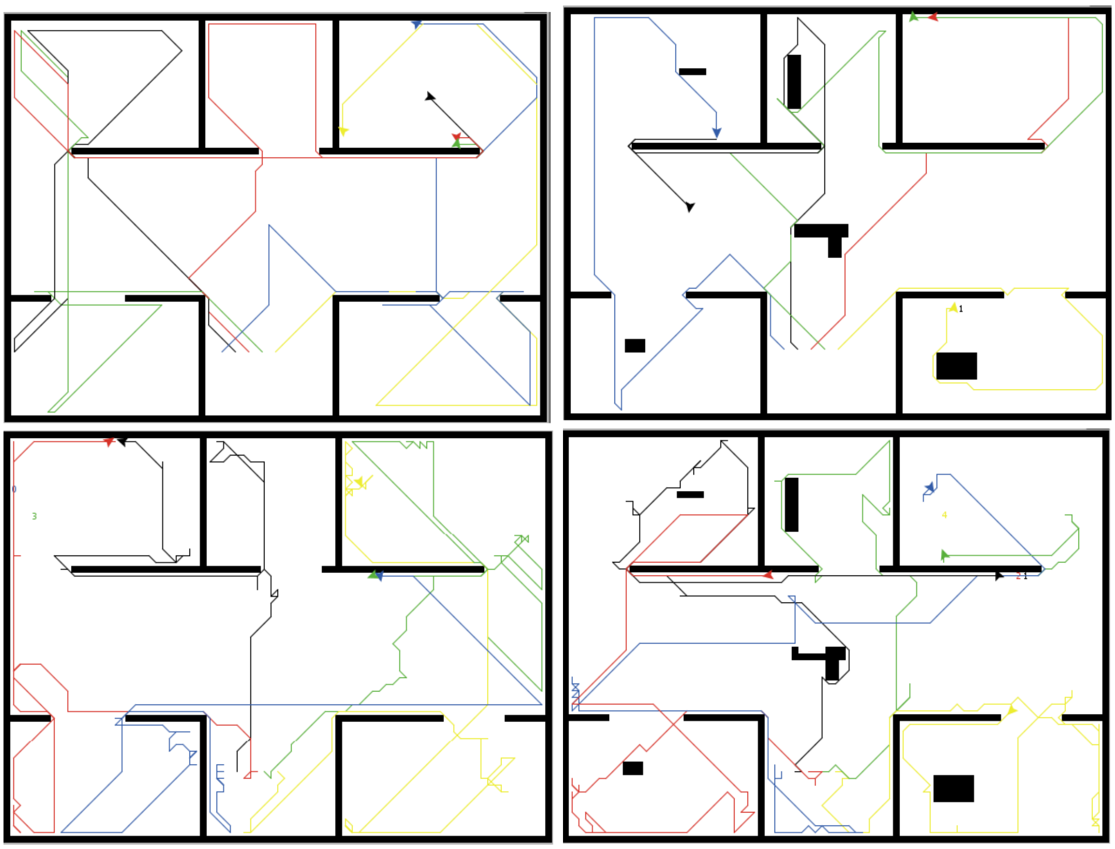

Fig. 7. The trajectories of five robots for the exploration algorithms after exploring the environment shown in Figure 2, with Algorithm 1 without obstacles (top left), with Algorithm 1 with some obstacles (top right), with Algorithm 2 without obstacles (bottom left), and with Algorithm 2 with some obstacles (bottom right).