JJMIE

Volume 5 , Number2 , April 2011
ISSN 1995-6665
Pages 177 - 184

Jordan Journal of Mechanical and Industrial Engineering

Exploration Algorithm Technique for Multi-Robot Collaboration

Mohammad Al-Khawaldah[a], Omar Badran[b] and Ibrahim Al-Adwan[a]

[a]Department of Mechatronics Engineering, Faculty of Engineering Technology, Al-Balqa` Applied University, Amman 11134- Jordan.

[b]Department of Mechanical Engineering, Faculty of Engineering Technology, Al-Balqa` Applied University, Amman 11134- Jordan.

Abstract

This paper focuses on wall-following exploration algorithm using two cooperating mobile robots. The aim is to decrease the exploration time and energy consumption. The new technique is a combination of wall-following exploration algorithm and frontier-based exploration algorithm. The proposed algorithm is divided into two stages: Firstly, one of the robots follows (detects) the entire of the environment walls. And secondly, they employ frontier-based algorithm to complete exploring the remaining unexplored areas in the environment. During these two stages, the robots sweep the line-of-sight between them in each step to maximize the exploration efficiency. Numbers of simulation experiments are presented. Moreover, testing with real robots will be introduced. In these experiments, the negatives and shortcomings of this exploration algorithm will be overcome.

Keywords: Robot, Exploration, Collaboration, Wall-following

1. Introduction

Exploration and mapping is an important issue in the robotics field. Its importance comes from the wide-range of beneficial applications such as path planning, search and rescue, accessing dangerous environments and cleaning. Many techniques were proposed to increase the exploration efficiency. Exploration efficiency depends on exploration time, energy consumption and exploration accuracy.

The two dimensional 2D environment is represented as occupancy grid map [1, 2]. Frontier-based exploration is proposed to minimize the overall exploration time by choosing appropriate target cell (frontier cells) for individual robots so that they explore different sections of the environment and the overlapping between them is minimized. The bidding function selects the cell with the maximum utility and the minimum cost with respect to each robot. A new different technique from those mentioned above was proposed by Ziparo *et al*. [3]. The goal is to reduce the size of the exploration area by using Radio Frequency Identification (RFID) tags as a coordination points between robots.

This paper focuses on testing the wall-following exploration algorithm through simulation and with real mobile robots. The aim is to assess the effectiveness of using the line-of-sight technique with grid-based maps to represent unknown environments in reality. Employing the line-of-sight technique to generate grid-based maps is innovative and it is the basis of the exploration algorithms proposed in this paper.

Corresponding Author. Engmohd77@yahoo.com

2. Wall-Following Exploration Algorithm Procedure

In this algorithm, the robots are directly guided to the environment walls to sweep cells as much as they can in each step. The new approach is an extension of the previous works [4-6]

The wall-following exploration algorithm can be briefly summarized as follows:

1. Call the two robots A &B. A is known as the “wall follower” and B as the “trouble shooter”. Both of them start at one of the environment corners or walls.

2. The wall follower starts following the walls. During each step of its movement it sweeps the line-of-sight to the other robot. It can also potentially correct its location estimate by using the trouble shooter as an intelligent land mark. It continues following the walls until the line-of-sight between the two robots is lost. Then it moves one step back to regain the line-of-sight, and then it stops.

3. The trouble shooter starts moving toward the wall follower to discover the cause of the line-of-sight obstruction which would be either an obstacle or a wall. During this movement the line-of-sight would be available.

4. When the trouble shooter reaches the cause of the line-of sight blockage, it starts following the walls in a clockwise direction if the cause of obstruction is on its right hand side. On the other hand, if the cause of the obstruction is on its left hand side, it starts following the walls in a counter-clockwise direction. During the

trouble shooter wall following, if it meets its partner (the cause of line-of-sight obstruction in this case should be a wall) then the procedure goes to point number 2. Wall following by robot B (the trouble shooter) continues until the line-of-sight is lost again.

5. The trouble shooter moves one step back and regains the line-of-sight again.
6. The procedure points from 2-5 is repeated until the wall follower completes detection of all the environment walls.
7. The remained unexplored area is explored by using frontier-cells and line-of-sight. The robots' position estimates can also be corrected during this stage but it slows down the exploration to make the robots take turns to move.

3. Experimentation And Results

Webots® simulator is used for simulation and real world implementation. It allows testing the robot behavior in simulation before implementation in reality, in order to debug. This helps to fix problems that may appear, like code mistakes, during the simulation stage. After the robot's behavior is tested in simulation and all of the problems are solved then real exploration can be implemented.

To switch the control to real robots, there is an option in the Webots software to switch the control from virtual simulated robots to real robots. Once the control is switched to real robots, the real robots should behave the same as they behave in simulation.

In summary, the first step of real robots exploration with Webots® is to perform a simulation run then to switch the control to real robots.

3.1 Simulation

In the wall-following algorithm, the robots start at any point on an environment wall. The wall follower starts following the walls in a clockwise direction while the trouble-shooter watches it. The trouble shooter checks if it can see the wall follower in each step. If the trouble shooter can see the wall follower, all the cells between them are assigned as free. This method helps the robots to explore the free spaces in the environment quickly.

Three environments are tested with Webots simulator. These environments are shown in Figures 1 to 4. Figure 2 shows different exploration progressive snapshots for this environment.

The three environments in Figures (1, 3 and 4) are the same, but in Figures (3 & 4) the environment is partitioned (halved) in different ways to show how the generated maps are related to different environment sizes and shapes.

The results of these experiments are given below:
The environment shown in figure 1

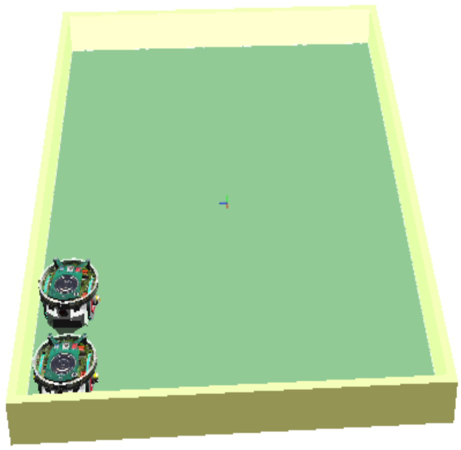
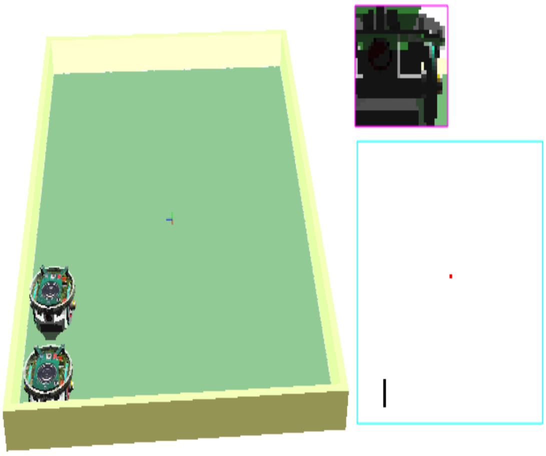


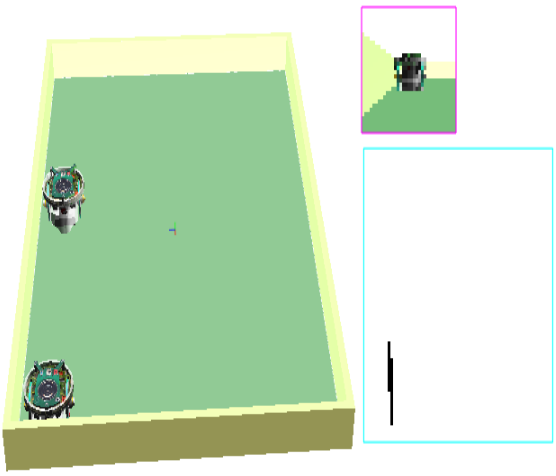
Figure 1: A rectangular environment to be explored by two Epuck robots with Wall-following exploration algorithm

The two Epuck robots start at the left bottom corner of the environment as shown in figure 1. The trouble shooter (the one very close to the corner) tries to keep looking at the wall follower while it is following the walls. If it can see the wall follower then there are no obstacles between the two robots and the area between them is assigned as free. In particular, the trouble shooter uses its camera to check if it can see the wall follower. The robots color is very dark compared to the environment (background) color. Therefore, when a picture is taken by the trouble shooter camera, this picture is checked for a dark area of higher intensity than the threshold. The threshold is chosen after an accurate calibration of the camera has been performed. Furthermore, when the trouble shooter finds a dark area in the picture, it adjusts its orientation to keep that dark area in the middle of the picture. This orientation adjustment is necessary to keep the wall follower in the field of view of the camera. As a result of the orientation adjustment, when the wall follower moves during the exploration, the trouble shooter keeps looking to it.

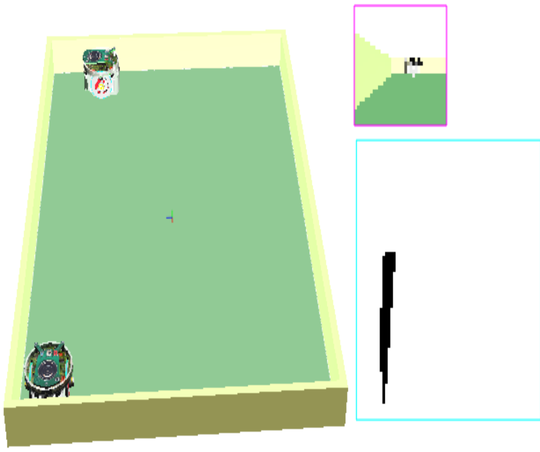
Figure 2 below shows the the process of exploring the environment shown in Figure 1. Figures 2a to f show progressive snapshots of the exploration experiment. In each one of these snapshots, the picture taken for the wall-follower at the corresponding snapshot is shown. The trouble shooter takes a picture for the wall follower in each time step (156ms) and it adjusts its orientation to keep the darkest area in its picture (i.e.the wall-follower) in the center of its field of view. In addition, in each one of the snapshots the map constructed thus far is shown. The black area in the map indicates free spaces.



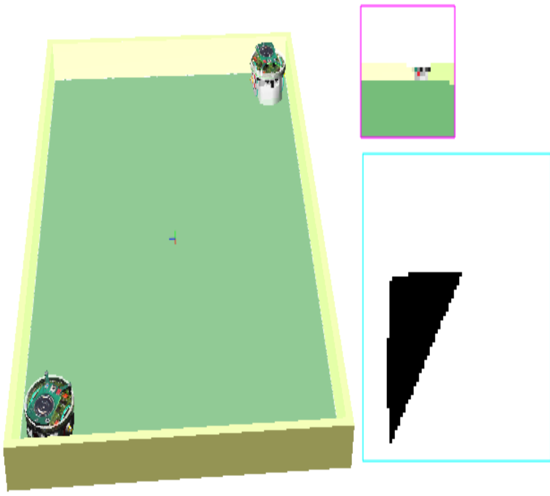
(a)



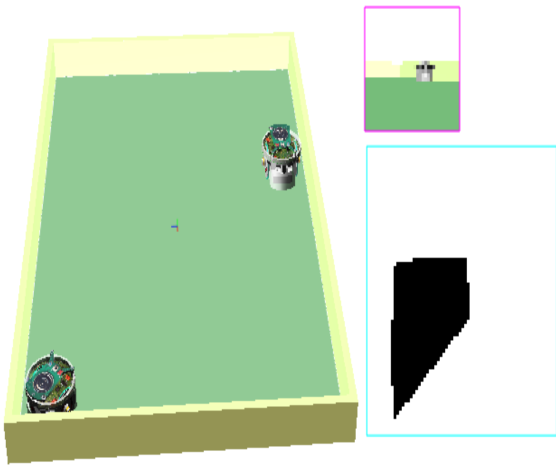
(b)



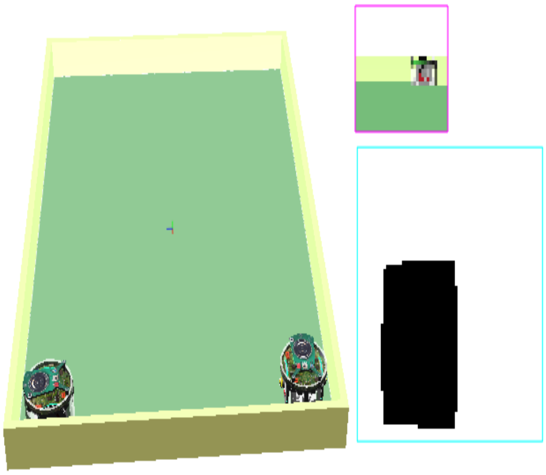
(c)



(d)



(e)

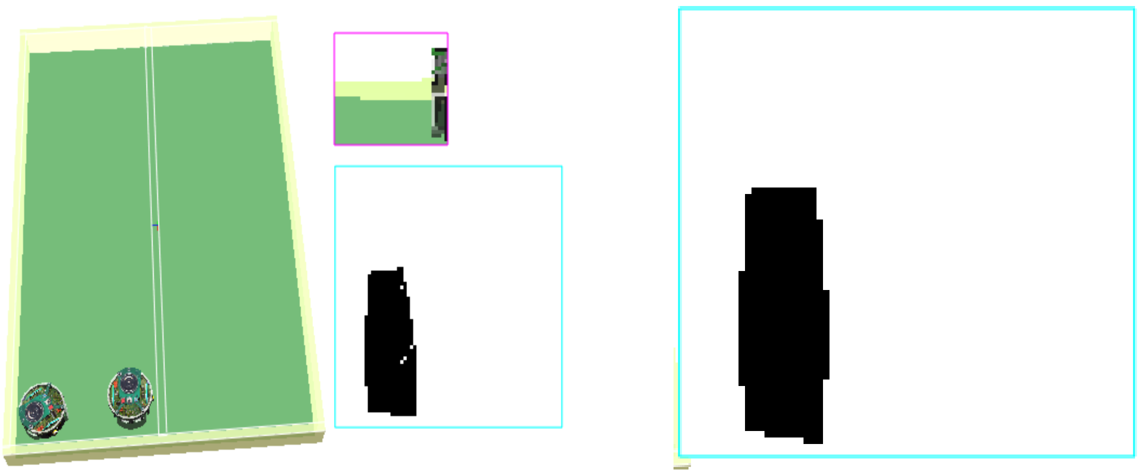


(f)

Figure 2: Wall-following exploration progressive snapshots with camera image and the constructed map thus far

Looking at the exploration snapshots in Figures 2a to f, it is clear that the trouble shooter adjusts its orientation to keep the wall follower in the centre of its field of view. The trouble shooter has to be fast enough to adjust its orientation to achieve that.

Figure 3 a shows a snapshot of the exploration when the wall follower is about to finish following the environment walls at a speed of 300cm/min. While Figure 3 b shows the generated map when robot’s speed is 200cm/min



(a)

(b)

Figure 3: (a) Wall-following exploration experiment with camera image and constructed map thus far (robot speed is 300cm/minute), (b) Map generated at speed of 200cm/minute

The map shown in Figure 3 a shows that the robots missed few cells that have not been swept with line-of-sight. It seems that this is due to the fact that robots need to sweep the line-of-sight between them more frequently to reduce the probability of missing some cells. The other solution is to reduce the robot speed. Therefore, robots will continuously sweep the line-of-sight between them in shorter distances. Figure 3 b shows a map generated at 200cm/minute for the same environment. It is clear that the problem has been solved and there are no unexplored cells in the map.

Figure 4 shows a snapshot of the exploration when the wall follower is about to finish following the environment walls.

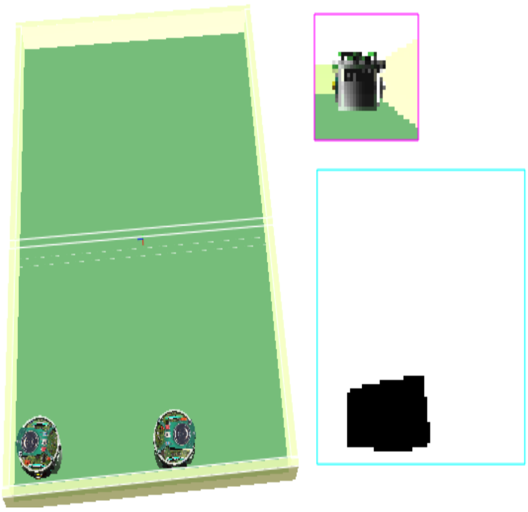


Figure 4: Wall-following exploration experiment with camera image and the constructed map thus far

The system is checked with different environments to check its ability to cope with different environments configurations. After the successful Webots® simulation runs of wall-following exploration, the system is now ready for testing with real robots.

3.2 Real Robots Exploration

Number of exploration experiments with real robots are presented. Following are some environments and their generated maps. As before, the black area in the maps represents the free space in the environment.

Figire 5a shows a sketch of the real-environment bounderies, and Figure 5b shows the generated map of the environment.

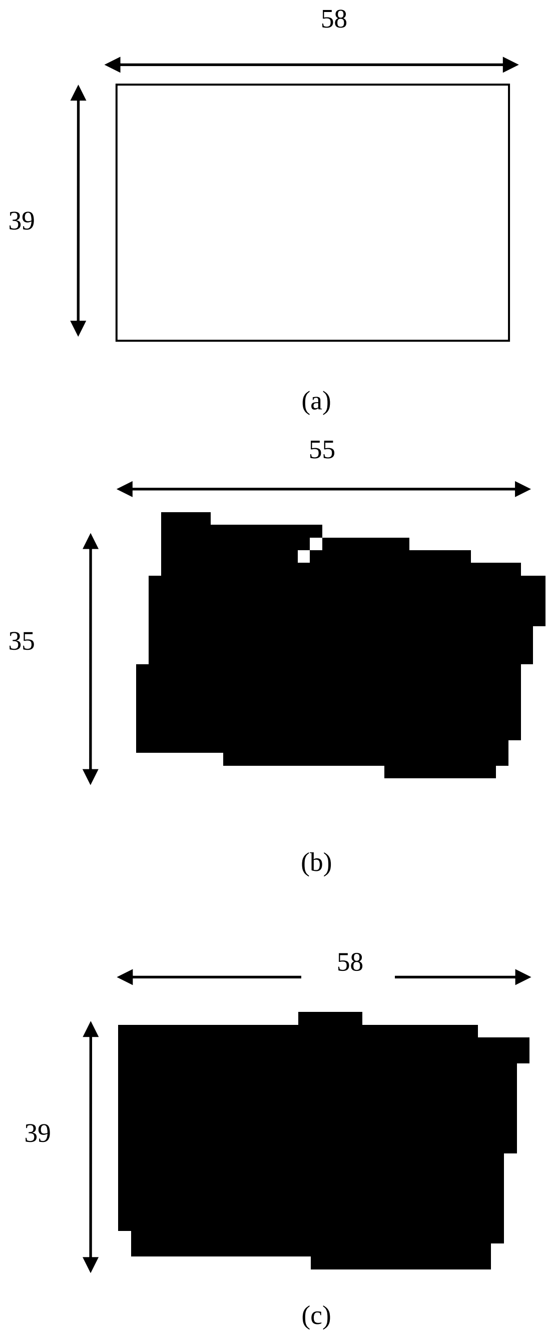
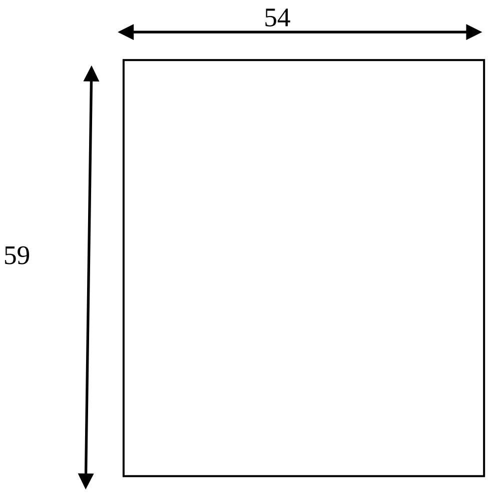


Figure 5: Exploration with real robots. Environment of 37-by-58 cm. (a) is the environment to be explored, (b) is the envirnment map generated at speed of 300 cm/min, (c) is the environment map generated at speed of 200 cm/minute. The dimensions shown are in centimeters.

In the map shown in Figure 5b, it is clear that the robots missed two cells that have not been swept with line-of-sight. As before, this appears to be due to the fact that robots needs to sweep the line-of-sight between them more frequently in order to reduce the probability of missing some cells. The other solution is to reduce the speed. Figure 5c shows a new map for the same environment generated at a speed of 200cm/min.

Figire 6a shows a sketch of the real-environment bounderies, and Figure 6b shows the generated map of the environment.



(a)

50

(b)

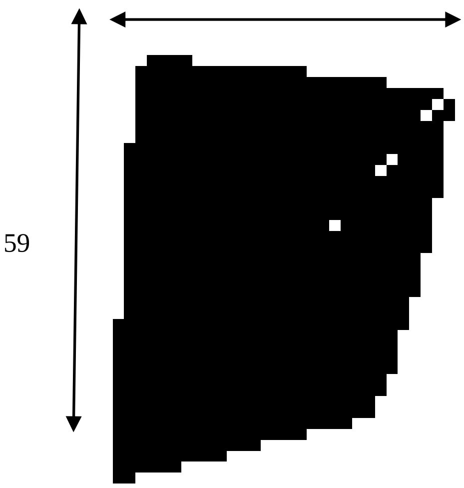
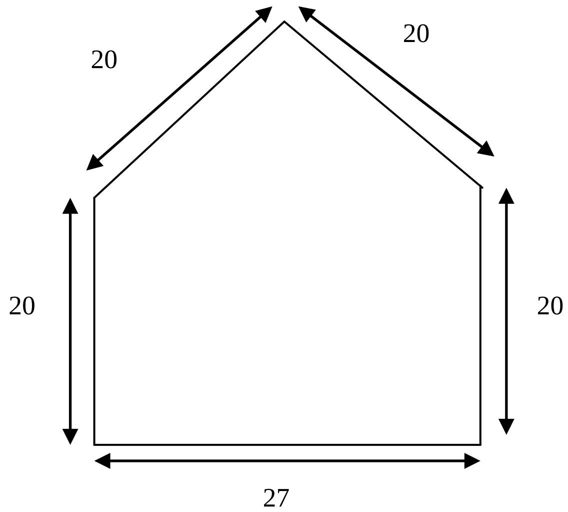
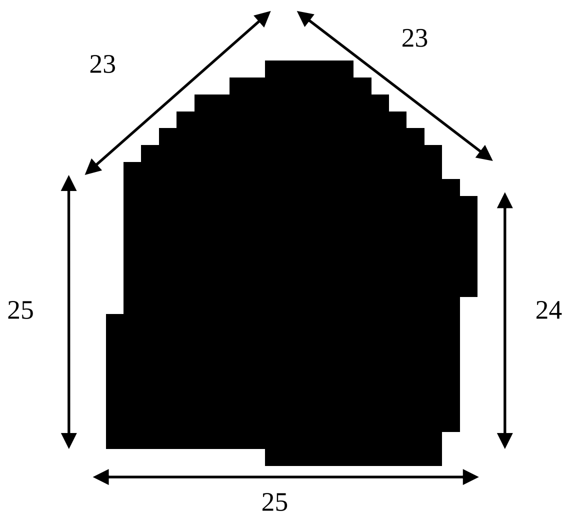


Figure 6: Exploration with real robots. Environment of 59-by-50 cm. (a) is the environment to be explored, (b) is the envirnment map generated at speed of 300 cm/min. The dimensions shown are in centimeters.

Figire 7a shows a sketch of the real-environment bounderies, and Figure 7b shows the generated map of the environment.



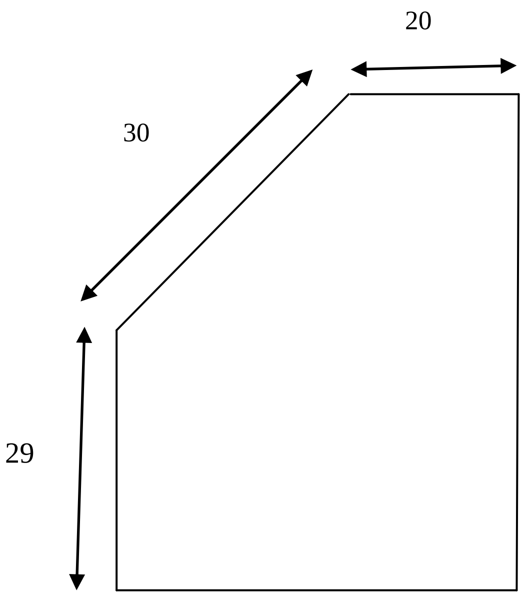
(a)



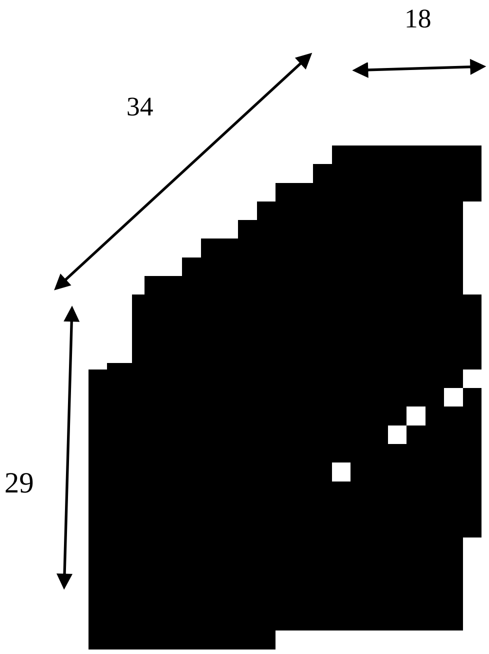
(b)

Figure 7: Exploration with real robots. (a) is the environment to be explored, (b) is the generated map. The dimensions shown are in centimeters.

Figure 8a shows a sketch of the real-environment bounderies, and Figure 8b shows the generated map for the environment.



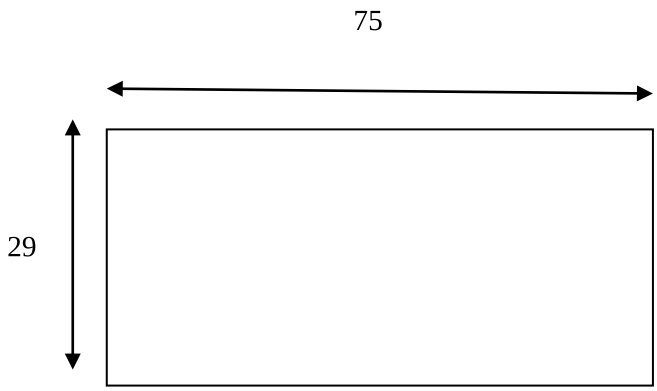
(a)



(b)

Figure 8: Exploration with real robots. (a) is the environment to be explored, (b) is the generated map. The shown dimentions are in centimeters

Figire 9a shows a sketch of the real-environment bounderies, and Figure 9b shows the generated map for the environment.



(a)

(b)

Figure 9: Exploration with real robots. (a) is the environment to be explored, (b) is the generated map. The dimensions shown are in centimeters.

The difference between the real environmennts and their maps is due to the localization problem. Dring the expl oration, robots are not localized precisely. They depend on odometry to localize themselves. Therefore, the errors in position estimates is accumulated during the exploration.

4. Conclusions

In this research the exploration experiments with real robots have been successfully implemented to verify the robot behavior using the line-of-sight technique with grid-based maps. The results were very promising. The proposed wall-following algorithm has been successfully implemented in reality to produce representative maps for the real environment. Wall-following exploration algorithm dependency on accurate robots positions is less than other exploration algorithms.

References

[1] Burgard, W., Moors, M., Stachniss, C. and Schneider, F.E. “Coordinated multi-robot exploration”, IEEE Trans. Robot. Autom. (2005), 21(3), pp. 376-386

[2] Sheng, W., Yang, Q. and Tan, J, Xi,. “Distributed multi-robot coordination in area exploration”, Robot. Auto. Syst. (2006), 54(12), 945-955

[3] Ziparo, V.A., Kleiner, A., Nebel, B. and Nardi, D. “RFID-based exploration for large robot teams”, In: Proc. of the IEEE Int. Conf. on Robotics and Automation (2007), 4606-4613. IEEE, Roma.

[4] Rekleitis, I., V. Lee-Shue, A.P. New and H. Choset “Limited Communication, Multi-Robot Team Based Coverage”, IEEE International Conference On Robotics And Automation. (2004)

[5] Rekleitis, I., G. Dudek and E. Milios “Multi-robot collaboration for robust exploration”, Annals of Mathematics and Artificial Intelligence, (2001), 31(1): p. 7-40.

[6] Rekleitis, I.M., G. Dudek and E.E. Milios “On multiagent exploration”, Visual Interface (1998), p. 455-461.