

Fast Wall-Following Exploration with Two Cooperating Mobile Robots

Mohammad Al khawaldah
Salvatore Livatino
School of Engineering and Technology

University of Hertfordshire
m.m.d.al-khawaldah@herts.ac.uk
S.Livatino@herts.ac.uk

Abstract: - In this paper two cooperating robots are used to explore an unknown environment. The new technique aims to decrease the exploration time and the energy consumption. It combines the wall-following exploration algorithm and frontier-based exploration technique. In the proposed algorithm the robots detect the walls and scan different parts of the environment as quick as they can. Then they employ frontier-based algorithm to complete exploring the remained unexplored areas. During these two stages, the robots sweep the line-of-sight between them in each step to maximize the exploration efficiency.

I. INTRODUCTION

Exploration of an unknown environment is an important issue in mobile robot research field. It has many real-world applications, like path planning, and planetary exploration. Robots in multi-robot systems are capable to perform a single task faster than a single robot. And a group of cooperating robots can localize themselves more accurately, especially if they have different sensor capabilities. But employing multi-robot systems has some challenges. For instance, when robots operate in teams there is the risk of interference among them. Furthermore, as the number of robots increases, longer paths may become necessary to avoid collisions with other members [1, 2].

Large number of the published works in multi-robot exploration field is based on frontier cells e.g. [1, 3-6]. A frontier cell is any free cell (not occluded) that has at least one unexplored neighboring cells. Each robot chooses a frontier cell, for example the nearest one as in [4], to be its target and start travelling towards it. It is expected that it gains information about the unexplored area when it arrives. The way in which the robot chooses its target is an important task that controls the exploration process. However, none of the above mentioned published works has introduced the line-of-sight technique to increase the exploration efficiency. In addition, we thought that there is a possibility to find a better method to reduce the overlap between the robots.

The line-of-sight technique, in which the robots work in teams of two has been employed in some published works

e.g. [7-9]. In this technique, each robot depends on its partner to correct its position estimate. During the exploration one of the robots (an “intelligent landmark”) is fixed while the other one explores and localize itself depending on its fixed partner’s position. And after that they exchange their roles. A different Line-of-Sight procedure is employed by Rekleitis *et al.* [10]. They proposed an algorithm for the complete coverage of free space. The environment is divided into square cells, of different shapes and sizes, and a relatively complex procedure is used to explore each cell with a number of robots. None of the papers mentioned above [7-10] has employed the frontier-based exploration algorithm. In addition, they were not tested with different obstacles distributions or different obstacles numbers.

In this paper new exploration algorithm with two cooperating robots is proposed. The Algorithm is divided into two stages: Firstly, the robots follow (detect) the entire of the environment walls. During this stage the robots sweep the line-of-sight between them in each step. Secondly they employ the same frontier-based exploration algorithm introduced in [4], but with line-of-sight facility, to complete exploring the remained unexplored areas. We tested our new exploration algorithms with different obstacle distributions and with different obstacles numbers. Eventually, we compared the results of our new exploration algorithm with the results of one of the close exploration algorithms published in the literature. The results show that our technique has new advantages over the existing techniques [4].

II. WALL-FOLLOW ALGORITHM

This paper proposes a new algorithm for wall following to be employed when two cooperating robots are used to explore different environments. The algorithm is based on the principle that when the two robots are directed to frontier cells that keep them far away from each other (especially when each robot follows an environment wall), the exploration efficiency is larger (i.e. the energy consumed by the robots and the exploration time is less). The algorithm will function in cooperation with new and relatively complex

procedures employing the line-of-sight technique to increase the exploration efficiency. Our wall following algorithm directly guides the robots towards the environment walls to sweep (explore) as much cells as possible in each step. The approach we follow is an extension of the work in [7-10] with new improvements.

The proposed exploration algorithm can be summarized as follows:

- 1- Call robots A & B. A is the wall follower and B is the trouble shooter. Both of them start at one of the environment corners or walls.
- 2- The wall follower starts following the walls in a clockwise direction, and the trouble shooter starts following the walls in a counter-clockwise direction. During their movement they sweep the line-of-sight between them in each step. They continue following the walls until the line-of-sight between them is lost. And then they **both** move one back step to get the line-of-sight back again.
- 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 is available.
- 4- When the trouble shooter reaches the cause of the line-of-sight blockage cause, 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. As soon as the trouble shooter reaches the obstruction the wall follower continues following the walls again (i.e. both of the two robots are moving now) and the line-of-sight between the two robots is swept. When the line-of-sight between them is lost, they go on back step and get it again.
- 5- During the trouble shooter movement toward the obstruction cause, if it meets its partner (the obstruction in this case should be a wall and not an obstacle inside the environment) then it will stop at that point while the wall follower starts its wall following again until the line-of-sight is lost, and it then moves one step back to get the line-of-sight again.
- 6- The procedure 3 – 4 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.

III. EXPERIMENTAL WORK

Our experiments were conducted using Netlogo software [11]. It is a very powerful simulation that allows us to simulate an occupancy-grid-based environment exploration process with different numbers of agents (robots). It is a very flexible tool in which the environment is simulated as an m-by-n grid of cells and each cell has information stored in variables. This tool allows us to repeat an experiment many times and store the results in an Excel file for further

processing. In the literature there are many published works based on Netlogo e.g. [12-14]. We started with a set of assumptions which can be relaxed later.

A. Key Assumptions

Concerning robots, it is assumed that:

1. Each robot is equipped with a 360° sensor that can detect the occupancy of all its eight neighbors. Also, it can distinguish between a robot and an obstacle when the other robot is in a neighbouring cell. This process is known as “scanning”
2. Each robot can see a ray from its partner, if there are no obstacles or other robots on the line between them (Line-of-sight technique).
3. Each robot knows exactly its own position and the position of its partner.
4. Robots move between the centers of cells.
5. All robots require equal time (a single tick in Netlogo) to perform a 360° scan, try to make line-of-sight with its partner, and move to any neighboring cell.
6. Each robot can access a shared map which is continuously updated.
7. The communication between two partners is perfect.
8. Robots deal with environment edges as occupied cells.

B. Exploration Methodology

Each cell is allocated a state as shown in Table 1 below. Fig.1 shows two robots when there is a LOS between them. The red dots indicate obstacle positions. Fig.2 below shows two robots when there is NLOS between them. Fig.3 shows a completed map.

Table 1 Cells states and codes

Patch Code	Meaning	Patch Displayed Color
F	“Fresh” No Idea Yet	Gray
S	“Free” by Scanning	Brown
LOS	“Free” by Line-of-Sight Only	Red
S+LOS	“Free” by Scanning and Line-of- Sight	Yellow
NLOS	“Potentially Occupied” by No-Line-of-Sight	Blue
O	“Occupied” by Scanning	Orange

The exploration process runs as follows:

1. When a robot visits a cell, then all of its free neighbors are assigned to be “S” by scanning
2. If a robot scans a LOS cell, it is assigned to be “S+LOS”.
3. If two robots in a team can see each other, then;
 - 3.1 All the F and NLOS cells on the line-of -sight between the two robots are assigned to be “LOS” cells.
 - 3.2 All the S cells on the line-of-sight between the two robots are assigned to be “S+LOS” cells.
4. If two robots in a team cannot see each other, then all the F cells between them are assigned to be “NLOS” cells

5. If a robot scans a cell and discovers that there is an obstacle in that cell, this cell is assigned to be an "O" cell until the end of the exploration.
6. The exploration process is stopped when all the cells are explored (free or occupied).

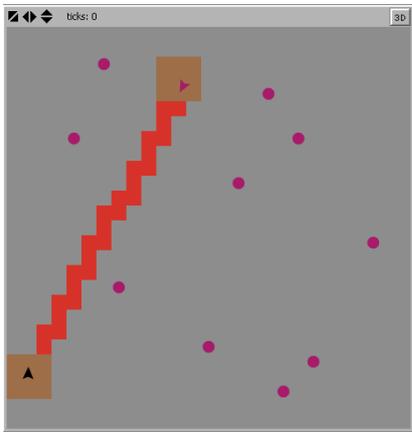


Fig.1: One tick exploration of two robots with LOS available between them

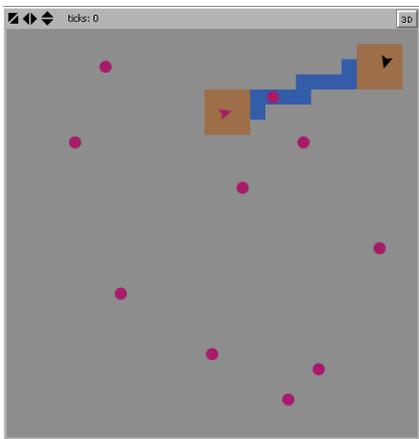


Fig.2: One tick exploration of two robots with NLOS between them

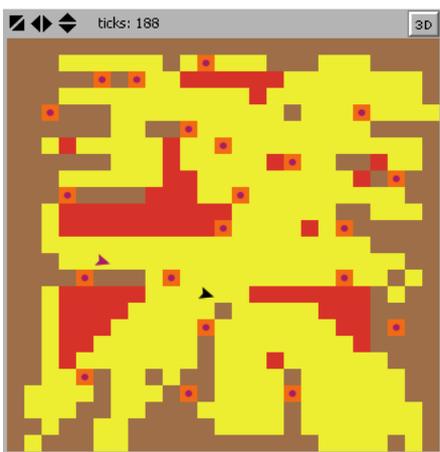


Fig.3A completed map

Following is an exploration example based on our wall-follow algorithm detailed in section 2. The environment to be explored has one block of obstacles. In this example the robots' trajectory lines are shown and colored with the

same corresponding robot colour. The exploration stages are shown in Fig.4.

The two robots started at the lower left corner of the environment as shown in Fig.4A. They both start the exploration by following the walls at the same time. The wall follower (the pink one) starts following the walls in a clockwise direction, while the trouble shooter (the black one) follows the walls in a counter clockwise direction. They go on following the walls until the line-of-sight between them is broken. After that they both go back one step and get the line-of-sight again as shown in Fig.4B. Now the trouble shooter starts moving towards the wall follower to discover the cause of the line-of-sight blockage (the block of six obstacles in the middle of the environment). As soon as it finds the cause for the obstruction as shown in Fig.4C, the two robots start following the walls at the same time. The trouble shooter follows the walls of the block in a clockwise direction and the wall follower follows the walls of the environment in a clockwise direction as well, and the line-of-sight is swept in each step. When the line-of-sight between them is lost they both go back one step and get the line-of-sight again as shown in Fig.4D. Now the trouble shooter stays in its place and the wall follower continues following the walls and sweeping the line-of-sight until all of the environment walls are detected as shown in Fig.4E. The small remaining unexplored island shown in Fig.4E is explored by frontier-based algorithm presented in [4] but with sweeping the line-of-sight in each step, as shown in Fig.4F.

It should be noted that in our Fast wall-follow algorithm the exploration time varies not only with the number of obstacles but also with the distribution of the obstacles themselves. And to investigate the effect of varying the obstacle distribution (positions of the obstacles) on the exploration time, a number of experiments have been repeated with the same number of obstacles but with different obstacle positions. Each experiment is repeated ten times except the experiments of the blocks of obstacles they are repeated just five times because they take long setup time. Table 2 below shows the results of these experiments.

It is clear from Table 2 that changing the positions of the obstacles does not dramatically affect the exploration time. For instance, if all of the obstacles are close to each other, the trouble shooter will not need to travel long distances to find the causes of the line-of-sight obstruction. On the other hand, if the obstacles are far away from each other, the worst case would be if each corner has one or more obstacles near to it, then the trouble shooter will spend plenty of time detecting the causes for the line-of-sight obstructions. As a result, the exploration time would increase.

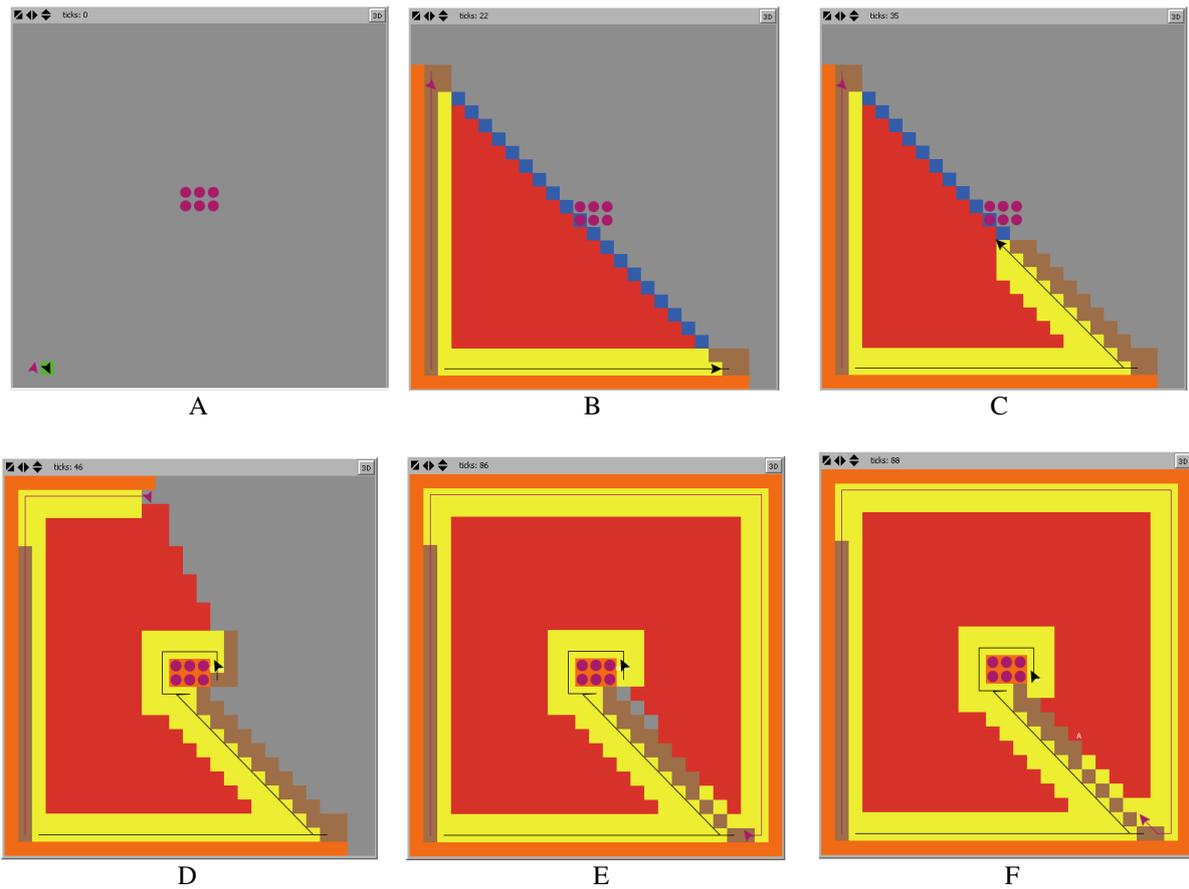


Fig.4: One blocks of obstacles. Environment exploration stages with Fast Wall-Follow algorithm

Table 2 Number of steps for repeated experiments with same number of obstacles and different obstacle positions with wall-follow algorithm

Experiment No.	One Obstacle	Two Obstacles	Five Obstacles	Ten Obstacles	One Block (six obstacles)	Two Blocks(six obstacles each)
1	81	94	111	118	96	108
2	89	107	135	145	84	102
3	88	80	102	158	83	94
4	63	122	125	158	84	101
5	89	81	114	150	92	96
6	85	102	134	143	-----	-----
7	62	99	121	137	-----	-----
8	83	109	150	152	-----	-----
9	89	95	122	154	-----	-----
10	96	92	101	146	-----	-----
Average	82.5	98.1	121.5	146.1	87.8	100.2
S. Deviation	11.9	12.7	15.3	11.9	5.8	5.5

IV. RESULTS COMPARISONS

In this section, the algorithm presented in this paper is compared across a range of environments. All of the environments are 25-by-25 (625 cells). The exploration time required to explore an environment with this size is reasonable and therefore we can try our algorithm with a large number of runs. The results of the experiments are shown below in Table 3.

In Table 3 we also compare our work with the work in [4] in which the robots choose their next frontier target cell according to the equation:

$$g_i = w_1 I_i - w_2 D_i + w_3 \lambda_i \dots\dots\dots(2)$$

where:

I_i : The information gain for the frontier cell i (the number of unexplored cells within the robot sensor range but, at the same time, not in the range of other robots or target cells for other robots)

D_i : the shortest travelling distance to the frontier cell i .

λ_i : is the nearness measure.

w_1 , w_2 , and w_3 are the weights for these three parameters and respectively.

The nearness measure is included in this equation to keep the robots close to each other to guarantee the communication amongst them. But in our simulation it is assumed that the robots are operating within their communication range, we just focus on the exploration algorithms. And the robots can share their maps in each step. Therefore, the nearness measure (λ_i) in eq. (2) is ignored, by setting w_3 to zero, when we compare the results of our technique (which is based on the equation (1)) with this technique based on equation (2). w_1 and w_2 are set to 1 as recommended in [4].

Table 3 shows comparisons between the experiments' results for the both exploration techniques presented in this paper. In addition to the elapsed number of steps taken by the exploration, the table also shows "Estimated Motor Energy" (E.M.E), which is the sum of the number of steps taken by each robot. This value is intended to give an approximation of the amount of energy expended during the exploration, and is higher than the elapsed time when both robots move simultaneously. Figs.5 and 6 show how the estimated motor energy and the exploration time vary with the number of scattered obstacles according to Table 3.

The following points can be observed from the results in Table 3:

1. Our proposed wall-follow algorithm is better than the exploration algorithm in [4] in terms of robot motor energy consumption (see Fig.5).
2. The proposed exploration algorithm consumes less exploration time than in environments with no obstacles or with few obstacles (less than ten) as shown in Fig.6.

Table 3 Comparisons between the experiments' results for the exploration techniques presented in this paper. All results are averages across 10 experiments for each technique in each environment.

	Sheng et al 2006		Fast Wall-Follow	
	Steps	E.M.E	Steps	E.M.E
No Obstacles	158.7	317.4	47	94
One Obstacle	155.4	310.8	82.5	122.1
Two Obstacles	155.7	311.4	98.1	143.8
Five Obstacles	155.2	310.4	121.5	193.7
10 Obstacles	153.3	306.6	146.1	252.1
One Block of Obstacles	157.2	314.4	87.8	127.4
Two Blocks of Obstacles	159.4	318.8	100.2	147.6

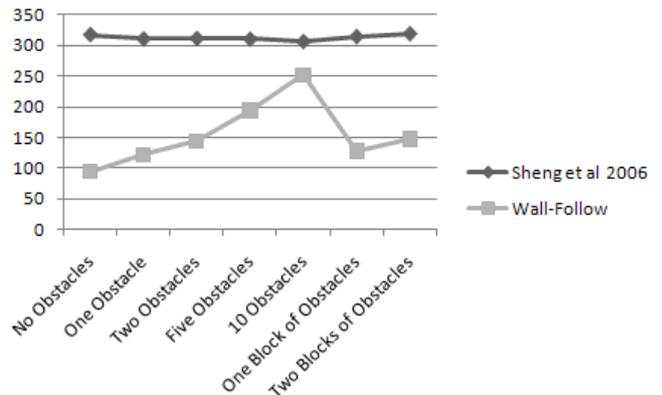


Fig.5 Estimated motor energy vs. Number of scattered obstacles

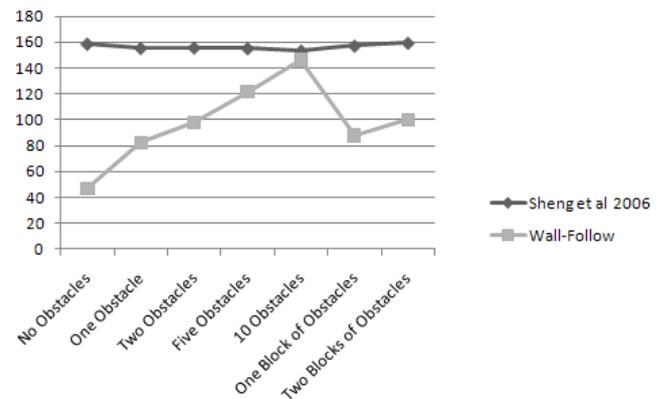


Fig.6 Exploration time (steps) vs. Number of scattered obstacles

V. CONCLUSIONS

In this paper a new exploration algorithm is proposed to explore an environment with two cooperating mobile robots. In this algorithm, the robots follow the entire of the environment walls and sweep the free spaces in the environment as quick as possible by the line-of-sight technique. Both robots may follow the walls at the same time when appropriate to save time. After that the robots switch to frontier-based with line-of sight exploration algorithm to explore the remaining unexplored part of the environment (if any). The results showed that the new proposed algorithm is very efficient to decrease the exploration time and the energy consumption by the robots.

REFERENCES

- [1] W. Burgard et al., "Coordinated multi-robot exploration" *IEEE Transactions on Robotics*, 2005. 21(3): p. 376-386.
- [2] M. Alkhalaf, and D. Lee, "Cooperative Robots Exploration with Line-of-Sight Technique", *International Conference on Information and Communication Systems*. To appear (Dec.2009)
- [3] D. Fox et al., Distributed Multirobot Exploration and Mapping. *PROCEEDINGS-IEEE*, 2006. 94(7): p. 1325.
- [4] W. Sheng et al., "Distributed multi-robot coordination in area exploration" *Robotics and Autonomous Systems*, 2006. 54(12): p. 945-955.
- [5] Rocha, R., J. Dias, and A. Carvalho, "Cooperative multi-robot systems: A study of vision-based 3-D mapping using information theory" *Robotics and Autonomous Systems*, 2005. 53(3-4): p. 282-311.
- [6] W. Burgard, M. Moors, and F. Schneider, Collaborative Exploration of Unknown Environments with Teams of Mobile Robots. "*Lecture Notes In Computer Science*", 2002: p. 52-70.
- [7] I. Rekleitis, G. Dudek, and E. Milios, "Multi-robot collaboration for robust exploration", *Annals of Mathematics and Artificial Intelligence*, 2001. 31(1): p. 7-40.
- [8] I.M. Rekleitis, G. Dudek, and E.E. Milios. "Multi-Robot Exploration of an Unknown Environment, Efficiently Reducing the Odometry Error", *International Joint Conference On Artificial Intelligence*. Lawrence Erlbaum Associates Ltd. 1997.
- [9] I.M. Rekleitis, G. Dudek, and E.E. Milios, "On multiagent exploration", *Visual Interface*, 1998: p. 455-461.
- [10] I. Rekleitis et al. Limited Communication, "Multi-Robot Team Based Coverage", *IEEE International Conference On Robotics And Automation* 2004.
- [11] NetLogo. Wilensky, U., . 1999. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL.
- [12] S. Magg, and R.e.t. Boekhorst, "Pattern Formation in Homogeneous and Heterogeneous Swarms: Differences Between Versatile and Specialized Agents", *IEEE Symposium on Artificial Life*. 2007.
- [13] M. Niaziand and A. Hussain, "Agent-Based Tools for Modeling and Simulation of Self-Organization in Peer-to-Peer, Ad Hoc, and Other Complex Networks. *IEEE Communications Magazine*. 2009.
- [14] R. Menezes, and H. Bullen "A study of terrain coverage models", *ACM symposium on Applied computing*, 2008.