

Frontier-based Exploration with Two Cooperating Mobile Robots

M. AL-KHAWALDAH, S. LIVATINO, D. LEE

Abstract: - In this paper, a frontier-based algorithm is used with two cooperating mobile robots to explore unknown environment. The aim is to decrease the exploration time. The proposed frontier-based exploration is based on a new bidding function in which we introduced a special parameter to decrease the overlap between the robots in addition to the utility and cost parameters. Tens of thousands experiments have been conducted, each experiment is executed with different weight values set, to see the relative importance of the weight parameters used in this technique. As a result of these experiments, the weight values can be chosen, according to the environment characteristics, to guarantee short exploration time. The proposed algorithm has been tested with a set of environments with different shapes and different numbers of obstacles. Finally, the results of our algorithm were compared with the results of one of known exploration algorithms available in the literature. The new technique led to promising results

Keywords— Cooperating robots, Exploration, Frontier, Multi-robot, Map building, Weight.

I. INTRODUCTION

The exploration and mapping of an unknown environment is a fundamental topic in mobile robots field. Generally, robots need the environment map to navigate effectively. Therefore, in many mobile robots applications, robots need to explore their environment themselves and arrange their sensory data to get an understandable representation for the environment. In addition robots need to localize themselves to accurate maps. Many techniques have solved the localization problem successfully [1-4]. The applications may include, search and rescue, hazardous material handling, and devastated area exploration after a disaster. There are continuous improvements to exploration techniques adopted by many researchers [5-20].

In [10] Vazquez, J. and C. Malcolm proposed a behavior-based exploration with multi-robot system. In this technique the exploration is guided in a distributed fashion to keep the local short-range communication in the mobile network. The exploration algorithm is implemented in an environment represented by means of a global probabilistic grid map. Frontier cells (frontier cell is any free cell for which at least one of its neighboring cells is unexplored) are evaluated according to the estimated cost and the utility of the information. The costs are computed depending on the target-

frontier-cell distance to the robot. And the utility depends on the size of the nearby unexplored area and is equal to the number of unexplored cells existing inside the circumference of the robot sensor range. The utility expected by a robot considering moving to a particular frontier is decreased if there are any robots near that destination. The cell of maximum difference [$Utility - Cost$] wins the bidding and the robot starts moving towards it. In this approach each individual robot executes the algorithm in a decentralized fashion.

In [22] Sheng *et al* provided a fully distributed bidding algorithm for the coordination that considers the limited communication range, and introduces a nearness measure in the bidding algorithm that keeps the robots together. A 2D occupancy grid is used to represent the environment to be explored. A group of robots start from initial positions which are close to each other, and the relative positions are known to all robots. Robots try to explore the area individually with the maximum exploration information and with minimum cost.

R. Roucha *et. al* in [23] proposed a grid-based probabilistic model of a 3D map, which stores for each cubic cell a coverage belief. And, because of the fact that accurate mapping mainly depends on localization, it is assumed that robots are externally localized through a global localization system to get rid of the problems of localization. The approach employs frontier-based exploration. At the beginning of the exploration process an initial map is given to the robot. After that it gets a new batch of measurements, updates the map, and shares useful information with other robots. Then it might receive information from other robots and the map is updated according to this information. After getting the new map, a new viewpoint for the sensor is chosen and the robot starts moving accordingly. Once the robot reaches the new target position, the process is repeated with a new batch of measurements provided by the sensor from its new pose.

In all of the above mentioned published works, nobody tried to tune the weight (i.e. to investigate the importance) for each of the “Utility” and the “Cost” used in the bidding functions of the exploration algorithms. Furthermore, in the papers mentioned above, the proposed techniques have not been tested with different environments and different obstacles

number.

In this paper a new exploration algorithm is proposed, in which the robots work in pairs and coordinate their actions. Our idea depends on selecting a frontier target cell to increase the efficiency of the exploration. The proposed technique is an extension of the algorithms described in [21-23]. The new technique tries to decrease the overlap between the robots as much as possible. To further improve the performance we also concentrated on finding the best combinations of weight parameters in the frontier-based algorithm used. The new exploration algorithms were tested with different environment sizes, different obstacle distributions and different obstacles numbers. Eventually, we compared the results of our exploration algorithm with the results of one of the known exploration algorithms in the literature.

II. EXPLORATION METHOD

All of our experiments were conducted using the simulation software – Netlogo [24] which is well known in the literature and employed in many published research works [25-26]. Netlog enables the computer-based investigation of the exploration process by a number of agents in an occupancy-grid-based environment. In Netlog, the environment is simulated as an m-by-n grid of square cells. Each cell has information about itself stored in variables. With Netlog, the same experiment can be repeated and results are stored in an Excel file for further analysis.

A. Experiment Assumptions

The experiments presented in this paper involve one team of two robots. The map is an m-by-n grid of square cells, each cell of which is allocated a code to represent its occupancy status.

Each robot is equipped with a 360° sensor, which can detect the occupancy status of all its eight neighbors. This process is known as “scanning”. Also, it can distinguish whether a neighboring cell is occupied by an obstacle or by its partner. Each robot knows exactly its own position and the position of its partner and they move between the centers of cells. Each robot requires an equal amount of time (a single step in Netlogo) to perform a 360° scan and move to a neighboring cell. Furthermore, robots can access a shared map of the environment which is updated in every step of the simulation. The communication between the robots is always on, and error free. Finally, the environment edges are treated as occupied cells.

B. Exploration Methodology

During the process of exploration, each cell of the map is assigned one of the states as shown in table 1. Table 1 also shows the color code used to identify each state.

Table 1 Cells states and color codes

Patch Code	Meaning	Patch Colour
F	“Fresh” No Idea Yet	Gray
S	“Free” by Scanning	Brown
O	“Occupied” by Scanning	Orange

At the beginning of each exploration, all of the cells in the environment are assigned as “F”. When a robot visits a cell, all of its free neighbors are assigned to be “S” by scanning. 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 (see Fig.1). The exploration process is completed when all the cells are explored (free or occupied) as shown in Fig.2.



Fig. 1: The environment with one robot after one step of exploration. An obstacle near the robot

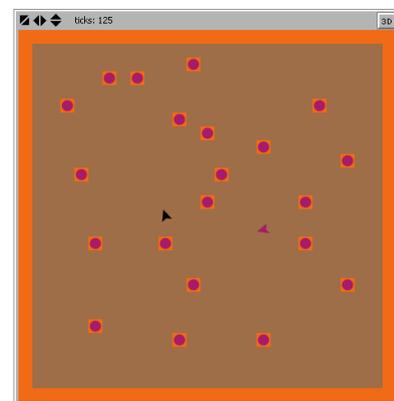


Fig.2: A completed map with two robots

3 Experimental Work “Frontier-based Algorithm”

A large number of published works in multi-robot exploration depends on the use of “frontier cells” e.g. [21-23, 27]. A frontier cell is any free cell for which at least one of its neighboring cells is unexplored. When a robot is directed to such a cell, it is expected that it will gain information about

the unexplored area when it arrives. Because a map may contain several unexplored areas, the challenge arises of how to plan the exploration mission by choosing the most appropriate frontier cells. In [21-23, 27] the utility of the target frontier cell is computed. The utility of a target cell is the number of unexplored cells which can be scanned from that target cell. Then the cost of reaching that target cell is computed. The cost is a function of the target cell distance. Finally, the target cell with the maximum $[Utility - Cost]$ value wins and the robot starts moving towards its target. This is the general algorithm that guides the exploration in the above mentioned published works.

Apart from the above mentioned two factors – cost and utility, the distance of the other robot from the cell is also considered when computing the bidding value for a frontier cell. This third factor introduces potential benefit of keeping the two robots apart. Each robot computes the bidding value B_i for each frontier cell in its shared map based on the bidding function represented in equation (1). The relative importance of each term in the formula is adjusted by a weight value. The most appropriate weight values have been studied and suggestions how they can be chosen are made later in this paper.

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

where

B_i : The bidding value for the frontier cell i (the target cell).

D_r : The distance from the robot to the target cell.

D_p : The distance from other robot (the partner) to the target cell.

N_u : The number of unknown neighbors for the target cell.

(weight neighbors), W_p (weight partner) and W_c (weight cost) are the weight factors for N_u , D_p and D_r , and respectively.

Each robot scans the eight neighboring cells around it and adds the new information (the scan information) to the shared map which is available to both robots. Then each robot computes the bidding function according to equation (1) for all of its frontier cells in the shared map. Each robot has its own bidding function values for the frontier cells in the shared map, computed from its own point of view. A cell can be a frontier cell to more than a robot. The same cell may yield different bidding value to different robots. Finally, each robot chooses among its frontier cells the one with the maximum bidding function value $\max\{B_i\}$ and starts moving towards it.

The two robots explore at the same time, and make their decisions on where to go at the next step based on the same shared map. Equation (1) shows that we are trying to guide each robot to a cell with a large number of unexplored neighboring cells, far away from its partner, and close to its current position.

We introduced the weights W_n , W_p and W_c , to investigate the relative importance of each of the three parameters N_u , D_p and D_r . W_n in our technique represents the weight of the utility expected from going to the frontier cell. In other words, it reflects the importance of the size of the region that will be explored when the robot visits that frontier cell. The weight W_p is introduced to reflect the importance of keeping the robots away from each other. Keeping the robots away from each other may, sometimes, make each robot explore on its own a different and relatively large portion of the environment completely. This appears to be better than making each robot explore small areas in different places in the environment, as there is a need to come back again to explore the unexplored spots between the explored areas. In addition, keeping the robots away from each other reduces the probability of overlap. W_c represents weight for the cost function. For distant frontier cells the $W_c D_r$ term reduces their bidding function value, because the value of this term is subtracted from the bidding function, and hence reduces their probability to win the bidding.

For a particular set of W_n , W_p and W_c in a given environment, a problem appears in which a robot starts to oscillate between two cells. The robot moves one step towards its target (called target number one) then it recalculates the bidding function and moves towards another cell (called target number two), and after that it calculates the bidding function again and moves towards the same previous target (target number one) and so on. In such a case, the robot oscillates between the two targets. This will not cause a dead loop as the oscillation will cease after approximately one to three steps. The oscillation disappears because the bidding function for a robot also depends on the movement of its partner or, more precisely, the distance between the frontier cell (the target) and its partner, which changes its position at each step independently. So, the winning target cell will change with the movement of the other robot. However, the oscillation can become a serious problem when both robots start to oscillate together. In such cases, the oscillation continues forever and becomes a dead loop. This problem has been solved by detecting the oscillation and then making one or both of the oscillating robots jump to a randomly chosen free neighboring cell.

The aim now is to find the appropriate combinations of the weight values that lead to relatively short exploration times. The idea is to execute many exploration experiments. Each experiment is conducted with different set of weight values (W_n , W_p and W_c). After that, the way in which these different weight sets affect the exploration time is investigated and analyzed. We started with an environment size of 25-by-25 cells. This environment size requires a relatively short but reasonable exploration time with which we can perform a large number of experiments. The results of these experiments will provide us with an idea about how the exploration time is affected with different weight values. In particular, we will focus on the weights that lead to

comparatively short exploration time (“good weights”).

Then the same procedure is repeated for the same environment but with different numbers of obstacles (0, 10, 20, 30 and 40 obstacles). In particular, we try to see whether the “good weights” are affected by the number of obstacles or not.

We also thought that it would be useful to test our algorithm with a different environment size to see how the good weight values vary with the size of the environment. An environment of 31-by-31 cells was chosen. The time required to explore this environment will be relatively longer than the ones of size 25-by-25 cell but still reasonable.

Finally the same procedure is repeated with a different environment shape (20-by-25 cells) in order to see if the “good weights” are affected by different environment shapes.

The effectiveness of each exploration is evaluated by counting the number of steps until the environment has been completely explored, and each cell has been identified as free or occupied.

To see the importance of the three weight values W_n , W_p and W_c , one thousand experiments for each of the above mentioned environments have been conducted. Positions of the obstacles are fixed in all of the one thousand experiments. Each experiment was conducted with a different set of weight values. For example, in the first experiment $W_n = 1$, $W_p = 1$, and $W_c = 1$, in the second experiment $W_n = 2$, $W_p = 1$, and $W_c = 1$, and in the thousandth experiment $W_n = 10$, $W_p = 10$, and $W_c = 10$.

We are interested only in the sets of W_n , W_p and W_c values that lead to short exploration time. We focus on the sets of W_n , W_p and W_c that generate the shortest 10 percent of exploration times. For example, after trying one thousand different sets of W_n , W_p and W_c values, if the minimum exploration time obtained is 100 and the maximum (worst case) is 400, then the exploration time ranges from 100 to 400. We concentrate on lowest 10 percent of the range, that means we only accept the sets of W_n , W_p and W_c that can lead to an exploration time of not more than 130. To study the relative values of W_n , W_p and W_c and their effects on the exploration time we put all the data from the one thousand experiments into an Excel sheet. Each row contains W_n , W_p and W_c values in addition to the corresponding steps number for that experiment. For two of the environments mentioned above (environments of ten and twenty obstacles with 25-by-25 cells), eight thousand experiments were repeated with W_n , W_p and W_c varying from 1 to 20. No better result was obtained.

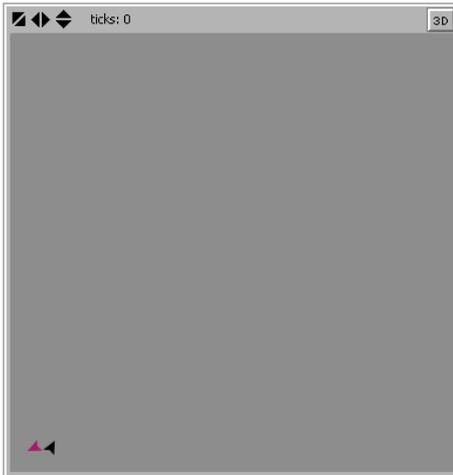
This indicates that no need for the weight values to go beyond ten.

It should be noted that in (1) W_n , W_p and W_c represent relative not absolute importance. For example, choosing $W_n = 1$, $W_p = 2$ and $W_c = 3$ is same as choosing $W_n = 2$, $W_p = 4$ and $W_c = 6$, and they lead to the same set of movements of robots and hence the same total number of steps (exploration time). Considering this, W_c is fixed at 1 in all our next experiments while W_p & W_n are varied from 0 to 10 in steps of 0.1 independently. This means that for each of the above mentioned environments ten thousand experiments have been conducted again. Figs. (3-7) show the results of these experiments. The black squares show the combinations of W_p & W_n that have resulted in the best 10 percent of the exploration time.

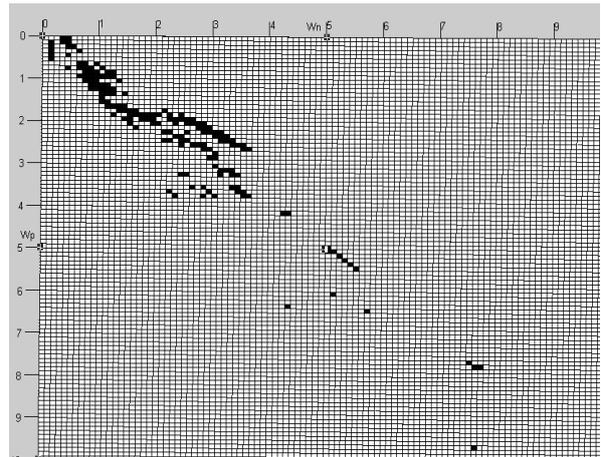
Fig. 3 shows the exploration of an 25-by-25 environment with no obstacle and its results. The black squares are concentrated between $W_p = 0 - 4$ & $W_n = 0 - 4$, meaning that high values of W_p & W_n leads to long exploration time. The same can be observed in Figs 4-7, where neither of W_p & W_n should ever go beyond 6 or 7. This observation confirms the necessity of introducing the cost of the robot moving to the target frontier cell (i.e. the term associated with W_c) into (1).

In Figs. 3-7, the black squares scatter around the line of slope equal to 1. This indicates that the terms associated with W_p & W_n are equally important and should both be considered when calculating the bidding function.

Fig. 7b shows the best combinations of weight values for 15 obstacles but with an environment area equal to one and half times that of the previous environments (31-by-31 cells). This bigger environment has been introduced to see the effect of the size of the environment on the weight values. Again, the black squares scatter around the line of slope 1. Approximately the same results have been obtained with the rectangle-shaped (20-by-25 cells) environment. Based on this result, it can be concluded that changing the environment size does not change the best weight values.

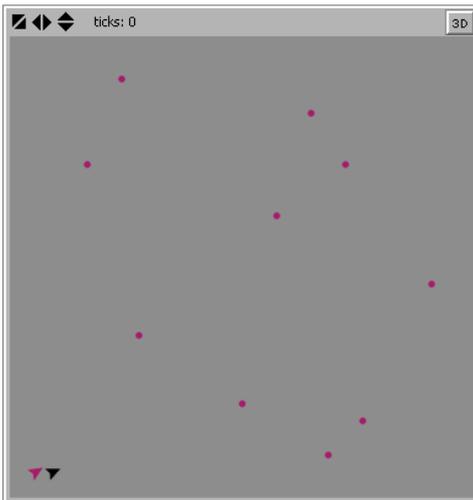


a

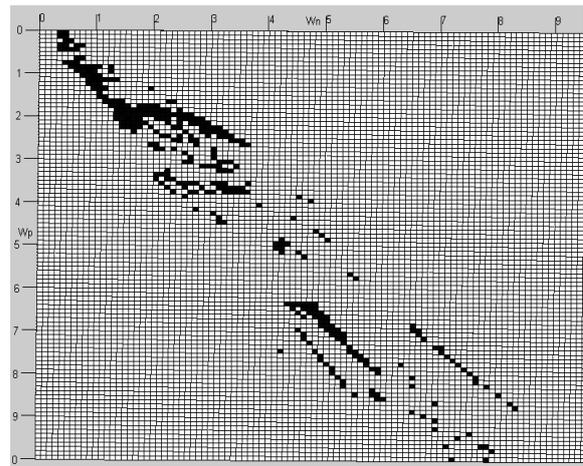


b

Fig.3: Exploration environment (25-by-25 cells) with no obstacles: a is the environment and b is the weight values. The black squares are the weights of the minimum exploration time experiments in this environment when $W_c = 1$.



a



b

Fig.4: Exploration environment (25-by-25 cells) with 10 obstacles: a is the environment and b is the weight values. The black squares are the weights of the minimum exploration time experiments in this environment when $W_c = 1$.

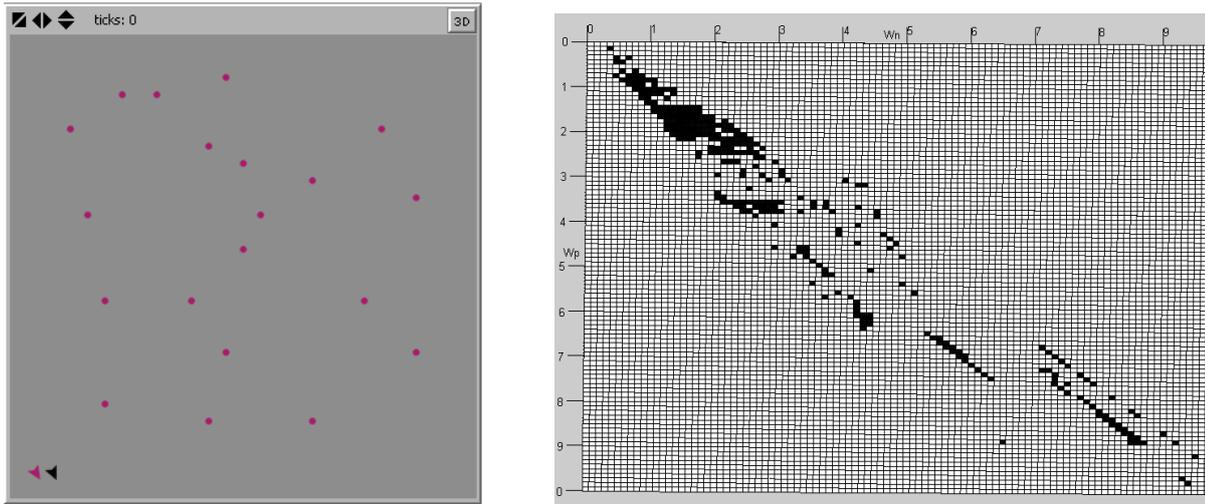


Fig.5: Exploration environment (25-by-25 cells) with 20 obstacles: a is the environment and b is the weight values. The black squares are the weights of the minimum exploration time experiments in this environment when $W_c = 1$.

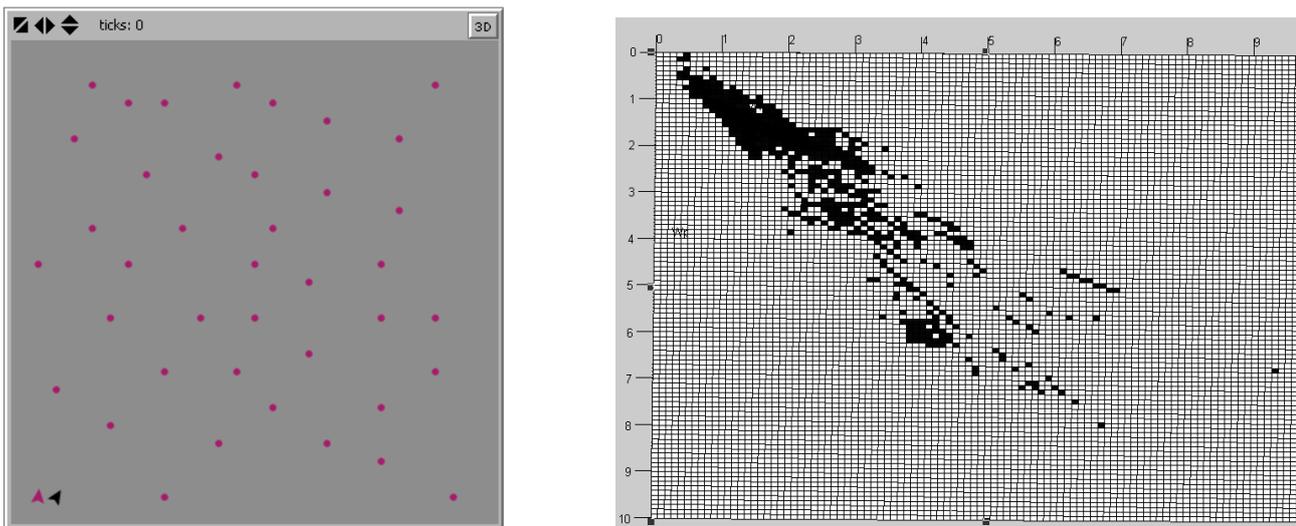


Fig.6: Exploration environment (25-by-25 cells) with 40 obstacles: a is the environment and b is the weight values. The black squares are the weights of the minimum exploration time experiments in this environment when $W_c = 1$.

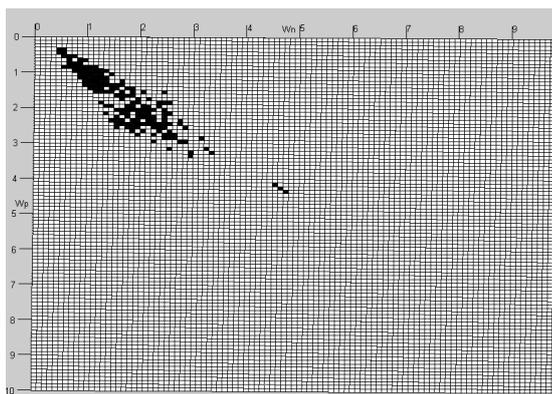


Fig.7: Exploration environment (31-by-31 cells) with 15 obstacles. The black squares are the weights of the minimum exploration time experiments in this environment when $W_c = 1$

Based on these results, we have come to the conclusion that changing the obstacles number, environment size and environment shape has a very slight influence on the best weight values. It is recommended to choose weight values (W_n and W_p) such that they lie on or close to a line of slope equal 1. And based on the figures obtained, we recommend the same value for both W_n and W_p and to be any value between 0.5 and 3.0 and for sure with $W_c = 1$. Furthermore, we tested some weight combinations that lie very close to the line of slope equal 1 but not in the area of the black squares. That means they are not in the best ten percent exploration times. It was noticed that these weight combinations lead to exploration time that is very close to the best ten percent ones.

We can suggest many weight combinations to guarantee a short exploration time (i.e. within or very close to the best 10 percent). For example, we can suggest the weight values to be as follows $W_p = 2$, $W_n = 2$ and $W_c = 1$. This combination leads to an exploration time around 125 steps in all of the above mentioned environments of size 25-by-25 cells. This exploration time is one of the best ones in these experiments.

III. RESULTS COMPARISONS

In this section, the algorithm proposed in this paper is tested across a range of environments. All of the environments are 25-by-25 cells. The results are shown in Table 2.

In Table 2 we also compare our work with the exploration algorithm in [21]. The algorithm in [21] is much related to ours but its bidding function is slightly different. In this algorithm the robots choose their next frontier target cell according to the following equation:

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

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 operate within their communication range, we just focus on the exploration algorithms. In practice, when the entire area to be explored is larger than the communication range, we can divide the area into smaller blocks and explore them block by block. So the robots can share their maps in each step. Therefore, the nearness measure (λ_i) in (2) is ignored, by setting w_3 to zero, when we compare the results of our technique (which is based on (1)) with this technique based on (2). w_1 and w_2 are set to 1 as Sheng suggests in his simulation in [2].

Table 2 shows comparisons between the experiments' results for both of these exploration techniques. The weights used in these experiments are $W_p = 2$, $W_n = 2$ and $W_c = 1$ as suggested earlier. Fig.8 shows how the exploration time varies with number of scattered obstacles.

Table 2 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 2006	Frontier-Bsd
No Obstacles	158.7	128
One Obstacle	155.4	127
Two Obstacles	155.7	126.2
Five Obstacles	155.2	124.6
10 Obstacles	153.3	122.6
20 Obstacles	163.3	123.3
30 Obstacles	158.8	122.9
40 Obstacles	155.6	123.1

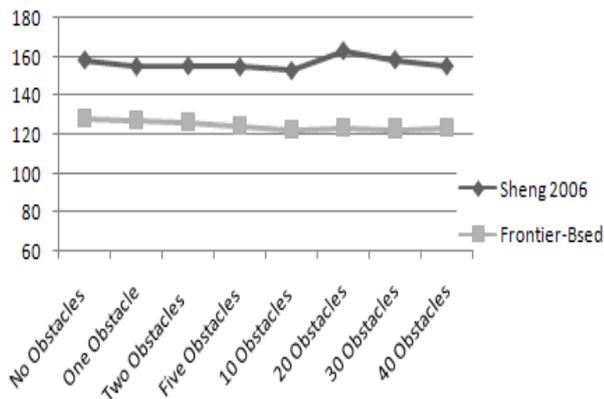


Fig.8 Exploration time (steps) vs. number of scattered obstacles

The following points can be observed from the results in Table 2 and Fig.8:

- Our proposed frontier-based exploration algorithm, with its recommended weight parameters, is always better than the Sheng et. al algorithm in terms of the exploration time.
- The exploration time with frontier-based algorithm is approximately fixed if the environment size is fixed whatever the number of obstacles is.
- Changing the scattered obstacles distribution does not affect the exploration time.

5 Conclusions

This paper proposes an exploration algorithm in which a couple of robots are used to perform the exploration task. The algorithm is an extension of the frontier-based algorithms known in the literature. The proposed technique tries to reduce the overlap as much as possible between the robots. Tens of thousands experiments have been conducted to tune the weight parameters used in this algorithm. This algorithm with the suggested (recommended) weight values have been tested with environments with 0, 1, 2, 5, 10, 20, 30 and 40 obstacles and with different environment shapes and sizes. Results with all of these environments have confirmed that our algorithm effectively reduced the exploration time.

References:

- [1] Cumani, A. and A. Guiducci, *Fast point features for accurate visual odometry*. Proc. 12th WSEAS CSCC Multiconference, 2008.
- [2] Lorence, A.G., M.P.G. Gaffare, and J.A.S. de los Rıos. *Mobile robot global localization using just a visual landmark*. 2006: World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA.
- [3] Cumani, A. and A. Guiducci. *Mobile robot localisation with stereo vision*. 2005: World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA.
- [4] ChJvez, A. and H. Raposo. *Robot path planning using SIFT and sonar sensor fusion*. 2007: World Scientific and Engineering Academy and Society (WSEAS) Stevens Point, Wisconsin, USA.
- [5] Kian Hsiang, L., et al. *Adaptive Sampling for Multi-Robot Wide-Area Exploration*. in *IEEE International Conference on Robotics and Automation*. 2007.
- [6] Giannetti, L. and P. Valigi. *Collaboration among Members of a Team: a Heuristic Strategy for Multi-Robot Exploration*. in *14th Mediterranean Conference on Control and Automation*. 2006.
- [7] Xin, M., et al. *Collaborative Control for Internet-based Multi-robot Exploration*. in *IEEE International Conference on Information Acquisition*. 2006.
- [8] Rekleitis, I., et al. *Collaborative exploration for map construction*. in *Proceedings IEEE International Symposium on Computational Intelligence in Robotics and Automation*. 2001.
- [9] Wurm, K.M., C. Stachniss, and W. Burgard. *Coordinated multi-robot exploration using a segmentation of the environment*. in *IEEE/RSJ International Conference on Intelligent Robots and Systems. IROS 2008*.
- [10] Vazquez, J. and C. Malcolm. *Distributed multirobot exploration maintaining a mobile network*. in *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference*. 2004.
- [11] Senthilkumar, K.S. and K.K. Bharadwaj. *An Efficient Global Optimization Approach to Multi Robot Path Exploration Problem Using Hybrid Genetic Algorithm*. in *ICIAFS. 4th International Conference on Information and Automation for Sustainability*. 2008.
- [12] Xin, M., Z. Qin, and L. Yibin. *Genetic Algorithm-based Multi-robot Cooperative Exploration*. in *IEEE International Conference on Control and Automation. ICCA 2007*.
- [13] Visser, A. and B.A. Slamet. *Including communication success in the estimation of information gain for multi-robot exploration*. in *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*. 2008.
- [14] Gifford, C.M., et al. *Low-cost multi-robot exploration and mapping*. in *IEEE International Conference on Technologies for Practical Robot Applications, TePRA 2008*.
- [15] Xin, M., et al. *Multi-agent-based Auctions for Multi-*

robot Exploration. in *The Sixth World Congress on Intelligent Control and Automation. WCICA 2006*.

with teams of mobile robots, *Lecture Notes In Computer Science*, 2002, p. 52-70.

16. Hao, W., T. Guohui, and H. Bin. *Multi-robot collaboration exploration based on immune network model*. in *Advanced Intelligent Mechatronics, 2008. AIM 2008. IEEE/ASME International Conference on*. 2008.
17. Guangming, X., et al. *Multi-robot Exploration Based on Market Approach and Immune Optimizing Strategy*. in *Autonomic and Autonomous Systems, 2007. ICAS07. Third International Conference on*. 2007.
- [18] Poernomo, A.K. and Y. Huang Shell. *New Cost Function for Multi-Robot Exploration*. in *Control, Automation, Robotics and Vision, 2006. ICARCV '06. 9th International Conference on*. 2006.
- [19] Zhao, J., X. Su, and J. Yan. *A Novel Strategy for Distributed Multi-robot Coordination in Area Exploration*. in *International Conference on Measuring Technology and Mechatronics Automation. ICMTMA 2009*.
- [20] Stachniss, C., O.M. Mozos, and W. Burgard. *Speeding-up multi-robot exploration by considering semantic place information*. in *Proceedings IEEE International Conference on Robotics and Automation, ICRA* . 2006.
- [21] W. Sheng, Q. Yang, J. Tan, & N. Xi Distributed multi-robot coordination in area exploration, *Robotics and Autonomous Systems*, 54(12),2006: p. 945-955,.
- [22] R. Rocha, J. Dias, and A. Carvalho, Cooperative multi-robot systems: A study of vision-based 3-D mapping using information theor, *Robotics and Autonomous Systems*, 53(3-4), 2005, p. 282-311,.
- [23] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, & B. Stewart, Distributed multi-robot exploration and mapping, *Proceedings-IEEE*, 94(7),2006, p. 1325
- [24] U. Wilensky, NetLogo. <http://ccl.northwestern.edu/netlogo/>. Center for Connected Learning and Computer-Based Modeling, Northwestern University. Evanston, IL, 1999.
- [25] 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.
- [26] R. Menezes, and H. Bullen, A study of terrain coverage models, In *ACM symposium on applied computing*, 2008.
- [27] W. Burgard, M. Moors, and F. Schneider, Collaborative exploration of enknown Environments