HAL open science

MinPos : A Novel Frontier Allocation Algorithm for Multi-robot Exploration

Antoine Bautin, Olivier Simonin, François Charpillet

To cite this version:

Antoine Bautin, Olivier Simonin, François Charpillet. MinPos : A Novel Frontier Allocation Algorithm for Multi-robot Exploration. ICIRA - 5th International Conference on Intelligent Robotics and Applications - 2012, Oct 2012, Montréal, Canada. pp.496-508, 10.1007/978-3-642-33515-0_49 . hal-00757960

HAL Id: hal-00757960

https://inria.hal.science/hal-00757960

Submitted on 27 Nov 2012

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

MinPos : a Novel Frontier Allocation Algorithm for Multi-robot Exploration

Antoine Bautin, Olivier Simonin, and François Charpillet

INRIA Nancy Grand Est - LORIA Lab. - Université de Lorraine
firstname.lastname@inria.fr

Abstract. Exploring an unknown environment with multiple robots requires an efficient coordination method to minimize the total duration. A standard method to discover new areas is to assign frontiers (boundaries between unexplored and explored accessible areas) to robots. In this context, the frontier allocation method is paramount. This paper introduces a decentralized and computationally efficient frontier allocation method favoring a well balanced spatial distribution of robots in the environment. For this purpose, each robot evaluates its relative rank among the other robots in term of travel distance to each frontier. Accordingly, robots are allocated to the frontier for which it has the lowest rank. To evaluate this criteria, a wavefront propagation is computed from each frontier giving an interesting alternative to path planning from robot to frontiers. Comparisons with existing approaches in computerized simulation and on real robots demonstrated the validity and efficiency of our algorithm.

Keywords: frontier-based exploration; multi-robot exploration; coordinated multi-robot exploration; potential field; gradient methods; multi-robot task allocation; distributed robot systems;

1 Introduction

Exploration of unknown environments by autonomous mobile robots can be necessary in many real world situations, typically for areas where human access is difficult or hazardous. In exploration tasks, multiple robots systems are advantageous because, using an efficient coordination strategy, they are more accurate and/or faster to explore the environment than single robot systems [10]. Furthermore, multi-robot systems using a distributed algorithm are robust to failures, flexible and scalable [2]. Nevertheless collectively building a map with a team of robots also raises up some difficulties. Finding a good navigation strategy for each robot is not straightforward as the exploration policy of one robot strongly depends on the exploration policy of others. Another issue is the computational capabilities of the robots that are often limited when using multiple robots. Finally, in some cases, robots can negatively interact with each other, inducing blockage or avoidance maneuvers.

This work aims at providing an efficient method for assigning to each robot areas to visit within the environment in order to minimize the amount of time required to fully explore it. Frontiers are the boundaries between unexplored and explored empty/accessible areas. A robot "exploring a frontier" (moving towards a frontier) discovers new areas

to add to the map and new frontiers. When complete exploration is the objective, every frontiers should be explored. In this paper, we show that a good heuristic is for each robot to *go toward the frontier having fewer robots in its direction*.

The main contribution of this paper is a novel algorithm for the robot-frontier assignment problem. Each robot is assigned to the frontier for which it is in the best position i.e. the frontier where there is the fewest robots between the frontier and the robot to be assigned (in travel distance). Experimental results in simulation show that this algorithm gives significantly better results than a nearest frontier assignment and similar or better results to a greedy assignment while having a lower computation complexity. Cooperation is achieved by the robots sharing their location coordinates and the information gathered about the environment.

This paper is structered as follows. Section 2, next, defines the problem before introducing the notations used throughout the paper. Section 3 reviews the existing literature on the multi-robot exploration problem. Section 4 describes our exploration strategy. Finally, in section 5, we compare the performance of our algorithm with previously proposed algorithms by simulation in various environments, then we present first experimental results with real robots.

2 Problem statement

The following presents the notations used and states formally the problem addressed. The fleet of robots is assumed to be homogeneous (identical robots) and equipped with exteroceptive sensors allowing them to build a map. To limit the scope of the problem, robot localization and mapping are not presented in this paper. It can either be done using an external localization system or by having the robots work with a common reference frame and using one of the Simultaneous Localization and Mapping (SLAM) algorithms in the literature (e.g. [11], [8]). Communication is critical for cooperation, therefore we use a wireless communication network (WiFi) which covers the full area to be explored by the robots. Exactly one frontier is assigned per robot for 2 reasons. First, a robot exploring a frontier generally pushes it back and fits the best for exploring that frontier, therefore it is unavailable to explore another frontier. Second, robots should not be left standing still if no frontier is available to them because other robots exploring frontiers are likely to discover new ones and require support to explore them.

Notations

- $\mathcal{R}$ the set of robots, $\mathcal{R} : \{\mathcal{R}_1 ... \mathcal{R}_n\}$ with $n = |\mathcal{R}|$ the total number of robots
- $\mathcal{F}$ the set of frontiers, $\mathcal{F} : \{\mathcal{F}_1 ... \mathcal{F}_m\}$ with $m = |\mathcal{F}|$ the number of frontiers
- $\mathcal{C}$ a cost matrix with $\mathcal{C}_{ij}$ the cost associated with assigning robot $\mathcal{R}_i$ to frontier $\mathcal{F}_j$
- $\mathcal{A}$ an assignment matrix with $\alpha_{ij} \in [0, 1]$ computed as follows :

$$\alpha_{ij} = \begin{cases} 1 & \text{if robot } \mathcal{R}_i \text{ is assigned to } \mathcal{F}_j \\ 0 & \text{otherwise} \end{cases}$$

The main objective is to minimize the overall exploration time. As the problem is dynamic - robots discovering an unknown environment - it is difficult to evaluate the final exploration duration during the system execution. Indeed, when selecting among

different allocations, the impact of a given assignment on the global system performance is not known because no information about what is behind the frontiers is available. Consequently, optimization has to reevaluated at each time step or at least every time the robot observes the targeted frontier. The time for all assigned frontiers to be explored is determined by the maximum exploration time among all frontiers. The optimization criteria is therefore the minimization of the maximum cost :

$$C_{max}(\mathcal{A}) = \max_{\forall i} \sum_{j=1}^{m} \alpha_{ij}\, \mathcal{C}_{ij} \tag{1}$$

The challenge of the frontier allocation lies in the number of possible assignments being equal to the number of permutations i.e. sequences without repetition. Therefore, in the best and most common case when $n \leq m$, it is equal to $\frac{m!}{(m-n)!}$. When $n > m$ it becomes even greater. These figures make the search for the optimal assignment intractable for large teams of robots and an approximation is therefore necessary.

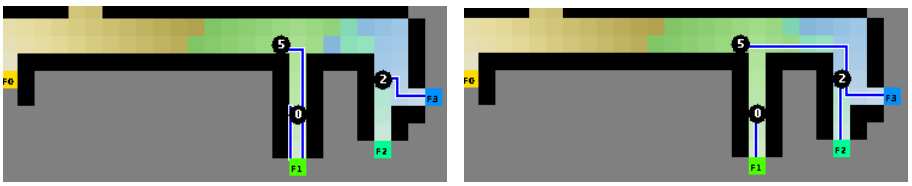
3 State of the art

In this section, we review previously proposed methods on multi-robot exploration strategies. These methods can be classified by their coordination method. First, implicit coordination that do not use communication for coordination. Then, the approaches which use communication to a central agent allocating an area to each robot or a decentralized decision making system.

3.1 Implicit coordination : nearest frontier

In 1997, Yamauchi introduced the very popular frontier-based exploration algorithm for single robot [12]. He first referred to frontiers as the border between known free space and an unexplored area. A robot moving towards a frontier will therefore sense new areas of the map. Repeating this operation increases the size of explored environment until the entire environment is explored: when no frontiers are left. In its multi-robot extension [13] (see algorithm 1), robots share the gathered information so that they build a similar map resulting in a similar list of frontiers. Each robot moves towards its nearest frontier, makes an observation and broadcasts its results. This cooperatively built map is enough to achieve some coordination: when a frontier is explored by one robot, the information acquired is shared with the other active robots and the frontier will not be explored again by another robot. Coordination is said to be implicit because it is achieved only by sharing information gathered on the environment. No communication is necessary to coordinate the robots. This method is asynchronous, distributed and robust to robot failures.

This pioneering work put in evidence the central issue of assigning frontiers to each robot because, using this approach, several robots can be assigned to the same frontier not taking advantage of their number to explore different areas. Figure 1(a) illustrates a situation where two robots are assigned to the same frontier, leaving unexplored areas without robots (lines show the assigned robot-frontier pairs).



(a) No consideration for other robots location. (b) Robot 5 and 2 go towards the same region.

Fig. 1. Resulting frontier assignment using the nearest frontier (a) and greedy (b) algorithms

Algorithm 1: Nearest frontier $O(m)$

Input: $\mathcal{R}_i, \mathcal{C}_i$ Cost Vector of $\mathcal{R}_i$
Output: α_{ij} assignment of robot $\mathcal{R}_i$ to frontier $\mathcal{F}_j$
begin
α_{ij} = 1 with $j = \underset{\forall \mathcal{F}j \in \mathcal{F}}{\operatorname{argmin}} \mathcal{C}_{ij}$
end

3.2 Coordination through communication : greedy assignment

Greedy algorithms are commonly used for task allocation and the frontier assignment problem is a task allocation problem. At each iteration of the allocation loop, the robot-frontier pair with lowest cost is selected, the robot is assigned to the frontier, and both are removed from their respective list. This is repeated until all robots have an assigned frontier. The standard way of applying this greedy algorithm is centralized and synchronous. A centralizer assigns a frontier to each robot, robots travel to their frontier, share the information collected on the environment and wait for a new assignment. The centralizer can wait for all robots to arrive at their frontier to assign the new set of frontiers (synchronous) or whenever a robot reaches its frontier (asynchronous). For a fair comparison between algorithms a decentralized and asynchronous version of the greedy allocation algorithm is given in Algorithm 2. Each robot computes the allocation of the other robots until it finds its assignment. Because robots have the same information about the environment including other robots locations, a robot can assume that the allocation computed for another robot will be the same as the one computed by that robot. An example of a greedy assignment is illustrated on figure 1(b). Due to the robots considering the assignment of other robots, robots are evenly assigned to frontier. However, note that frontier $\mathcal{F}_0$ is not assigned while robots $\mathcal{R}_2$ and $\mathcal{R}_5$ go towards the same zone. Most frontier allocation approaches are greedy based [3], [9] [14].

3.3 Utility-based vs frontier-based

Utility, proposed by Simmons et al. [9], refers to the difference or ratio between an estimated information gain and a cost. Evaluating it requires to estimate the expected area discovered from a target location reduced by the overlap in between robot sensed

Algorithm 2: Greedy (decentralized) Complexity $O(n^2m)$

Input: $\mathcal{R}_i, \mathcal{C}$ Cost Matrix
Output: α_{ij} assignment of robot $\mathcal{R}_i$ to frontier $\mathcal{F}_j$

while $\mathcal{R}_i$ *has no frontier assigned* **do**
  Find k, j = $\operatorname{argmin} \mathcal{C}_{kj} \ \forall \mathcal{R}_k \in \mathcal{R}, \ \forall \mathcal{F}_j \in \mathcal{F}$
  $\alpha_{kj} = 1$
  $\mathcal{R} = \mathcal{R} \setminus \mathcal{R}_k, \mathcal{F} = \mathcal{F} \setminus \mathcal{F}_j$
  IF $\mathcal{F} = \varnothing$ THEN $\mathcal{F} = \mathcal{F}_{init}$
end

areas and the cost of reaching that target. Targets are greedily assigned on utility to robots thus maximizing information gain and minimizing the total cost. Utility-based approaches improves the frontier based approaches (see Burgard et al. [4]) but induces an extra computation load. However, regarding frontier-based approaches, contiguous frontiers cells can be grouped or identified into a unique frontier (Franchi et al. [7]) thus making their performances close to utility-based approaches. In this case, utility and frontier based approaches avoid sending several robots toward nearby frontier cells.

3.4 Centralized vs decentralized systems

In a centralized coordination scheme, one central agent decides which frontier each robot should explore [9]. Centralized coordination scheme represents a single point of failure and requires an additional computation and communication cost. In order to be decentralized, the utility based method was extended by Zlot et al. [14] by using a market based approach where robots bid on targets, thus negotiating their assignments. Existing bidding algorithms for frontier allocation usually work with the same principle as a greedy algorithm. Auctions systems can be centralized [9], one central agent receives bids from all robots and assigns frontiers to each robot, or decentralized as in [14] and [5], a robot discovering a frontier is auctioneer for this frontier. These approaches reduce the computation cost by making parallel computations of paths to frontiers and assignments but induce a communication overhead of $O(nm)$ because each robot sends a bid for each frontier.

4 Proposed approach

Our approach to the frontier allocation problem is based on the distribution of robots among the frontiers. For this purpose, we do not only take into account the distance to frontiers. We consider the notion of position or rank of a robot towards a frontier, by counting how many robots are closer to the frontier. Before detailing the frontier allocation algorithm, we present the scheme of robots' behavior and the assumptions on the environnement representation.

Each robot performs repeatedly the four following steps :

1. Frontiers identification and clustering
2. Computation of the distances to frontiers
3. Assignment to a frontier
4. Navigation towards the assigned frontiers for a fixed time period

The map representation used is an Occupancy Grid (OG) [6] which is a square tessellation of the environment into cells of the desired size, maintaining a probabilistic estimate of their occupancy state. This map is built from the robot perceptions and broadcasted maps of other robot. To reduce the computation cost, robots are assumed to have a circular shape and to be holonomic. The configuration space grid is then computed by enlarging obstacles (including other robots) by the size of the robot. This configuration space grid is then used for frontier identification and path planning.

Steps 1, 2 and 3 are detailed in the next sections.

4.1 Frontier identification and clustering

Each robot maintains a list of frontier cells updated with each information received or acquired. To reduce the distance computation step and avoid assigning robots to frontiers cells that will be observed with the same perception, contiguous frontiers cells are grouped. The size of groups of frontiers cells is thresholded with respect to the size of sensing areas.

4.2 Distances and path computation

To evaluate robots positions we use, as the other methods, a cost matrix. In order to compute the cost matrix we build a local minimum free artificial potential field from each group of frontier using the wavefront propagation algorithm (WPA) [1]. From each frontier cells in a group of frontiers, a monotonically increasing potential is built by iteratively propagating a wavefront through unoccupied cells in the Configuration grid. The result of the WPA gives for each cell in the environment the distance to the frontier using the shortest path. Building the cost matrix is then straightforward using the location information of each robot. In comparison, most approaches compute a path from each robot to each frontier (with a A* algorithm). Thus if a robot requires the cost for another robot, it needs to ask it to the other robot or to compute its path. The potential field grids is re-computed periodically, at least every time the robot is close to reaching the frontier, but ideally each time new information is significant enough to modify the configuration grid and could therefore affect the robot assignment.

4.3 Frontier Allocation

Frontier assignment is done in a decentralized way i.e. each robot autonomously decides which frontier it will explore next.

Our approach consists in assigning to a robot a frontier for which it is in best position, i.e. the frontier having the less robots closer than itself. Formally, we set $\mathcal{P}_{ij}$ the position of a robot $\mathcal{R}_i$ towards a frontier $\mathcal{F}_j$ as:

$$\sum_{\forall \mathcal{R}_k \in \mathcal{R},\ k \neq i,\ \mathcal{C}_{kj} < \mathcal{C}_{ij}} 1$$

which computes the cardinal of the set of robots closer to the considered frontier than the robot being assigned.

By reasoning on positions instead of distances, two close robots will be assigned on frontiers where they will be in first position whatever the distances, thus favoring a better spatial distribution of robots on frontiers. Figure 2 illustrates such an assignment. Robot $\mathcal{R}_5$ is assigned to frontier $\mathcal{F}_0$ instead of a closer frontier. Indeed robot $\mathcal{R}_5$ is in second position for frontiers $\mathcal{F}_1$, $\mathcal{F}_2$ and $\mathcal{F}_3$. This provides a well balanced direction assignment, which was not the case with the greedy approach (see fig 1(b)). The algorithm for assignment, called Minimum Position, is given in Algorithm 3.

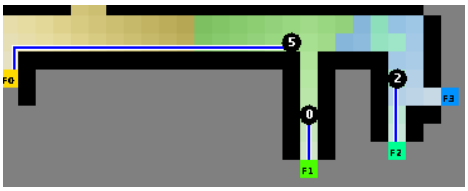


Fig. 2. Resulting frontier assignment using the MinPos algo. : directions taken are well balanced.

Algorithm 3: MinPos — Complexity $O(nm)$

Input: $\mathcal{R}_i, \mathcal{C}$ cost matrix
Output: α_{ij} assignment of robot $\mathcal{R}_i$

foreach $\mathcal{F}_j \in \mathcal{F}$ **do**
$$\mathcal{P}_{ij} = \sum_{\forall \mathcal{R}_k \in \mathcal{R},\ k \neq i,\ \mathcal{C}_{kj} < \mathcal{C}_{ij}} 1$$
end
$\alpha_{ij} = 1$ with $j = \underset{\forall \mathcal{F}j \in \mathcal{F}}{\operatorname{argmin}}\ \mathcal{P}_{ij}$
In case of equality choose the minimum cost among min $\mathcal{P}_{ij}$

Wavefront propagation stopping To limit the computation load, the propagation of a wavefront is stopped when it encounters the location of the robot computing its allocation. At this point all robots closer to frontiers have been reached by the wavefront. The robots not encountered are further to the frontier.

Parallel computation of wavefronts Further reduction of the computation load can be obtained by computing wavefront propagation in parallel. Wavefronts are started simultaneously and propagate at the same speed. When a wavefront encounters a robot it is paused. Then two cases are considered :

- If the encountered robot is the robot computing its assignment then the process is finished. The robot is assigned to the frontier corresponding to this wavefront.
- If all wavefronts are paused on other robots, they are restarted sequentially from the lowest to the highest potential value in order to re-synchronize them. The whole process is repeated.

This process allows to synchronize the wavefronts propagation first on the number of encountered robots and second on the distance.

Figure 3 illustrates the result of a parallel wavefront computation.

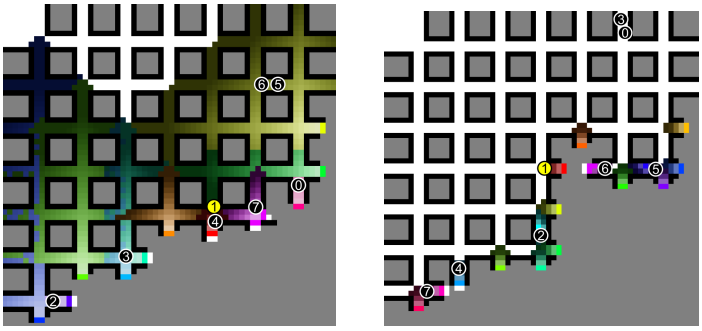


Fig. 3. Resulting potential values from the wavefront propagation algorithm when stopped on the robot computing its assignated frontier (left) and using the parallel wavefront computation (right). Robot computing its allocation is in yellow. Cell color is white when no potential value has been computed in this cell, otherwise it has the color of the frontier associated with minimum potential values computed in that cell.

4.4 Path or trajectory planning

The wavefront propagation algorithm allows to navigate to the frontier by descending the negative gradient. A different planner can be used to account for non-holonomic or dynamic constraints but a feasible trajectory may not exist.

Coordination in between robots is achieved only by cooperatively building the map and by sharing robots locations. This feature also applies to the decentralized versions of the greedy algorithms but in comparison the MinPos algorithm using the cost matrix is less complex. Furthermore, using the parallel wavefront computation, the cost matrix is not computed, only one complete path is computed: the one from the robot to its assigned frontier. In the next section, we show that this computation cost reduction does not compromise performances and can even improve them compared to state-of-the-art multi-robot exploration algorithms.

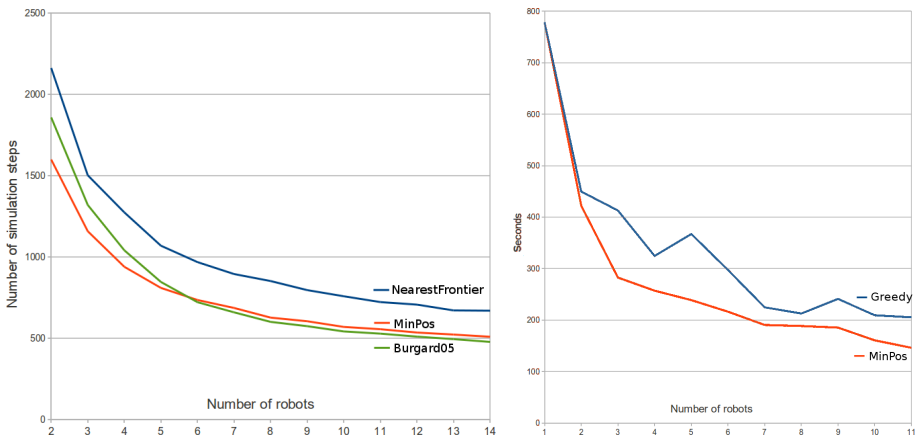
5 Experiments and analysis

5.1 Simulation : Frontier assignment evaluation

Evaluation of the proposed method was carried out in a simulator specifically developed. Environment and time are discrete, the robot's size is set to the dimension of a cell. Robots know with certainty their location and can sense their neighborhood perfectly at 360° around them with a parameterized range. The simulated environments are buildings, randomly generated mazes and a regular grid (illustrated in figure 4). Exploration times are measured when robots have discovered every part of the environment.

Fig. 4. Example of the environments used for the experiments: an hospital section, a regular grid, and a stable.

Figure 5(a) compares the exploration times given in simulation steps, while varying the number of robots of different methods. The methods compared are nearest frontier, the Burgard et al. [4] greedy-based algorithm and MinPos on the hospital section environment. Results shown are an average of 60 runs of each algorithm with a given robot count. We observe that the Burgard et al. and Min Position algorithms are more efficient improving by 13% on average the number exploration steps required to fully explore the environment. On all the environments except the stable, when the number of robots is small (from 1 to 5 or 6 depending on the environment), the MinPos algorithm performs better. On the other hand, our implementation of the Burgard et al. algorithm has close but slightly better performance when the number of robots is large. Figure 5(b) compares a greedy allocation of frontiers with MinPos allocation on a more realistic simulator, exploration times are in seconds. On the environment named stable the MinPos algorithm is 18% more efficient on average, and this improvement remains true regardless the number of robots. Results on the mean number of steps and standard deviation on all the test environments show that the MinPos algorithm has less variability. This has been confirmed by an analysis of variance test run on the obtained data.



(a) (b)

Fig. 5. Exploration results for simulations in the hospital (a) and stable (b) environment.

5.2 Analysis

As a tentative explanation of the results obtained we analyze, in this section, different assignments on static crossroad situations.

During system execution, an adequate dynamical behavior emerges that tends to separate grouped robots. For example, as shown in Figure 6(a), two robots following each other to reach the same frontier will separate as soon as the first one moves away from an unassigned frontier, when re-computing its assignment frequently and as soon as it reached the frontier otherwise. Indeed, the second robot is then in first position towards this frontier and therefore assigned to it. With the nearest frontier algorithm both robots would have continued exploring the area behind frontier $\mathcal{F}_1$, with a greedy algorithm robots would have initially been assigned to the different frontiers.

Compared to the greedy assignment of frontiers the MinPos algorithm does not only consider distances to frontiers but also directions to frontiers. This feature has the advantage of separating grouped robots in different directions. For example, in figure 6(b), the MinPos assignment will send robot $\mathcal{R}_1$ to frontier $\mathcal{F}_1$ which is further away than $\mathcal{F}_2$ and $\mathcal{F}_3$ but in a different direction whereas the greedy assignment will send $\mathcal{R}_3$ to $\mathcal{F}_2$ and $\mathcal{R}_1$ to $\mathcal{F}_3$. The MinPos assignment will lead to a shorter exploration time of all frontiers because once $\mathcal{R}_3$ has explored $\mathcal{F}_2$ it will then explore $\mathcal{F}_3$ to complete the exploration ($\mathcal{F}_2$-$\mathcal{F}_3$ are closer to each other than $\mathcal{F}_2$-$\mathcal{F}_1$).

A limit of the MinPos frontier assignment appears when, from a robot point of view, there is a direction with multiple frontiers and multiple robots in front. The robot will consider the robots in front even though the number of frontiers is larger than the number of robots, leaving unassigned frontiers. Figure 6(c) illustrates such a situation where $\mathcal{R}_3$ "sees" frontier $\mathcal{F}_1$ with 2 robots in front and will go towards $\mathcal{F}_4$ where it is in 2nd position. In these situations, a greedy assignment performs better. Nevertheless results show that the impact on performance are low.

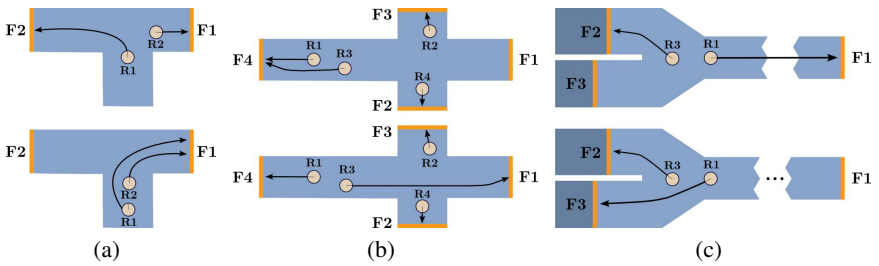


Fig. 6. (a) : two robots separating on a T intersection using MinPos; (b)(c) : comparisons with greedy assignment, frontier assignment using MinPos (top) and greedy (bottom)

5.3 Experiments with robots

The MinPos algorithm was also tested with up to four robots. The robots used are MiniRex robot built in the LISA lab at the university of Angers France. For exploration, they are equipped with a Hokuyo utm-30lx laser range sensor and three ultrasound range sensors. The environment to explore, shown in Figure 7, is approximately

$35m^2$ and features glass walls. Figure 7 shows the results of the exploration with 3 robots. The trajectories of each robot demonstrate the validity and efficiency of the proposed approach, indeed each robot explored a different part of the environment. The comparison of MinPos with nearest frontier with the MiniRex robots showed that the exploration is faster with MinPos and generates less conflict in between robots.

Videos of an exploration of the arena can be seen at `www.loria.fr/~bautin/outputfile.avi` and `www.loria.fr/~bautin/outputfile2.avi`.

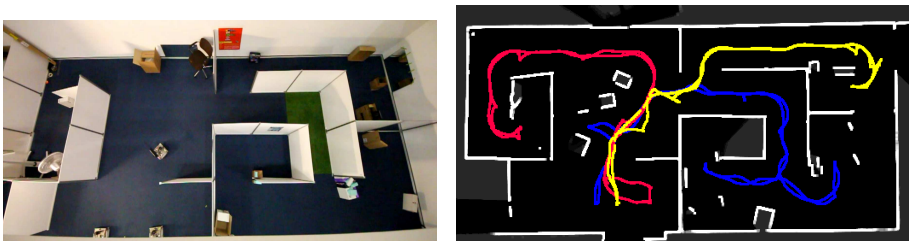


Fig. 7. Photo of the environment (left), map and trajectories resulting from an exploration (right)

6 Conclusion

We addressed the problem of exploring an unknown environment with multiple autonomous robots. A novel algorithm is proposed to assign frontiers that should be explored by each robots. This algorithm addresses the limitation of previous approaches in that it achieves a good coordination with a low complexity algorithm not requiring coordination with explicit communication.

The proposed algorithm is based on the concept of position of a robot toward a frontier, defined by the number of robots closer to the frontier than the robot evaluating its position. To cooperate, the robots share their location and local maps. Each robot then decides autonomously which frontier it will explore next.

Performance measures in simulation demonstrated that our approach is significantly more efficient in total exploration time than the nearest frontier assignment. Our algorithm performs better than a utility greedy assignment of frontier when the number of robots is small and has similar performances when the number of robots grows. Furthermore, the MinPos algorithm has a lower complexity than greedy algorithms and does not require to compute a path from each robot to every frontier beforehand. Indeed, distances to frontiers are computed using a wavefront propagation started from frontiers independently of the number of robots.

We have successfully used this approach during the french robotics challenge Carotte[1] that has been won by deploying five robots in a $120m^2$ unkown 'appartement-like' environment. As future work, we are considering the use of a hybrid metric/topological map representation. We aim to further reduce bandwidth requirements by sharing only topological information in between robots exploring different parts of the environment. This should allow a more distributed and therefore scalable approach.

[1] **Acknowledgement:** This work has been supported by the French National Research Agency (ANR) and Defence Procurement Agency (DGA) in the Cartomatic project of ANR Carotte

Bibliography

[1] J. Barraquand, B. Langlois, and J.-C. Latombe. Numerical potential field techniques for robot path planning. In *Advanced Robotics, 1991. 'Robots in Unstructured Environments', 91 ICAR., Fifth International Conference on*, pages 1012 –1017 vol.2, jun 1991.

[2] D.J. Bennet and C.R. McInnes. Distributed control of multi-robot systems using bifurcating potential fields. *Robot. Auton. Syst.*, 58:256–264, March 2010. ISSN 0921-8890.

[3] W. Burgard, M. Moors, and F. Schneider. Collaborative Exploration of Unknown Environments with Teams of Mobile Robots. In *Revised Papers from the International Seminar on Advances in Plan-Based Control of Robotic Agents,*, pages 52–70, London, UK, 2002. Springer-Verlag. ISBN 3-540-00168-9.

[4] W. Burgard, M. Moors, C. Stachniss, and F.E. Schneider. Coordinated multi-robot exploration. *Robotics, IEEE Transactions on*, 21(3):376 – 386, june 2005. ISSN 1552-3098.

[5] L. Chaimowicz, M.F.M. Campos, and V. Kumar. Dynamic role assignment for cooperative robots. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 1, pages 293 – 298 vol.1, 2002.

[6] A. Elfes. Using occupancy grids for mobile robot perception and navigation. *Computer*, 22(6):46–57, Jun 1989. ISSN 0018-9162. doi: 10.1109/2.30720.

[7] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli. The sensor-based random graph method for cooperative robot exploration. *IEEE/ASME Trans. on Mechatronics*, 14(2):163–175, 04/2009 2009. Winner of the IEEE RAS ICYA Best Paper Award 2010.

[8] A. Howard. Multi-robot Simultaneous Localization and Mapping using Particle Filters. *The International Journal of Robotics Research*, 25(12):1243–1256, 2006.

[9] R. Simmons, D. Apfelbaum, W. Burgard, M. Fox, D. an Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proceedings of the AAAI National Conference on Artificial Intelligence*, Austin, TX, 2000. AAAI.

[10] C. Stachniss. *Exploration and Mapping with Mobile Robots*. PhD thesis, University of Freiburg, Department of Computer Science, April 2006.

[11] S. Thrun and Y. Liu. Multi-robot slam with sparse extended information filers. In *International Journal of Robotics Research*, pages 254–266. SAGE, 2005.

[12] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Computational Intelligence in Robotics and Automation, 1997. CIRA'97., Proceedings., 1997 IEEE International Symposium on*, pages 146 –151, jul 1997.

[13] B. Yamauchi. Frontier-based exploration using multiple robots. In *AGENTS '98: Proceedings of the second international conference on Autonomous agents*, pages 47–53, New York, NY, USA, 1998. ACM. ISBN 0-89791-983-1.

[14] R. Zlot, A. Stentz, M.B. Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, volume 3, pages 3016 –3023, 2002.