Flocking-Based Multi-Robot Exploration

Noury Bouraqadi and Arnaud Doniec

Dépt. Informatique & Automatique
Ecole des Mines de Douai
France
{bouraqadi,doniec}@ensm-douai.fr

Abstract

Exploration of an unknown environment is one of the major applications of Multi-Robot Systems. Many works have proposed multi-robot coordination algorithms to accomplish exploration missions based on multi-agent systems techniques. Some of these works particularly focus on multi-robot exploration under communication constraint. In this paper, we propose a solution based on flocking rules in order to explore an open area while keeping robots close enough to avoid disconnections. We show that the multi-robot coordination resulting from this approach is effective and exhibits interesting properties compared to other works in the literature.

Keywords

Multi-Robot Systems ; Coordination ; Exploration ; Flocking ; Network Connectivity Maintenance.

1 Introduction & State of the Art

Multi-robot systems (MRS) consist in a set of autonomous mobile robots which collaborate to perform a mission. This collaboration is allowed by communication abilities which usually rely on radio communication technologies. For example, Mobile Ad Hoc Networks (MANET) are frequently employed to support explicit communications in MRS: each robot becomes a node of the network and is able to send, receive and relay data to other robots. In order to have an accurate and efficient collaboration between robots, each node (robot) of the network has to be reachable at each instant. This implies that, in addition to their collaborative task, the robots have to perform an extra task which consists in maintaining the network connectivity.

One of the major applications of multi-robot systems is the exploration of an unknown environment. It consists, for a fleet of robots, to navigate while incrementally constructing a map of the environment. In this application, the robots have to collaborate:

- to spread out on the ground in order to speed up the exploration and use less energy,
- to keep in touch with each other in order to exchange partial maps and share areas that have not been explored yet.

Therefore, a good collaboration scheme for multi-robot exploration has to conciliate these two antagonist constraints.

The multi-robot exploration issue has been addressed in the literature using different approaches and has been originally initiated by two works: [6] and [11].

In the Yamauchi's approach [11], robots try to build and use a global map for the exploration. The postulate of the Yamauchi's work is the following: to speed up the exploration, robots have to gain new information about the environment. Therefore, they have to move towards the boundary between open space and unexplored area. In practice, Yamauchi describes some robots trying to get closer to the boundary but without real cooperation with the rest of the fleet. Nevertheless, this article states the concept of *frontier-based exploration* which has inspired many other works including ours.

Main improvements of the Yamauchi's approach have consisted in introducing coordination between robots. This has firstly been performed in a centralized way. In [8], the authors propose to assign a target destination to each robot in a way that maximizes the expected information gain over time. As the computation of optimal solution is intractable in practice, they proposed a heuristic based on bids construction. Each robot estimates the utility and the cost to travel towards various locations. A central server receives all the bids and assigns a location target to each robot taking into account possible overlaps in the coverage of the ground.

More recently, Rooker and Birk [5] also proposed a centralised coordination ensuring that, during the exploration, no robot will loose the connection with the rest of the fleet. To achieve this goal, a central entity collects the current positions of all the robots and generates a set of future configurations for the fleet (i.e. the next possible positions of the robots). Due to the high number of available combinations, all configurations can not be considered but only a limited number of them. Among this number of generated configurations, the central entity chooses the best one according to an utility function. This function gives penality when the evaluated future position is occupied by an obstacle or when it puts the robot out of the communication range of all other robots.

Clearly, centralized coordinations dedicated to frontier-based exploration seem difficult to employ in real applications. They are not fault tolerant since the entire system will fail whether the central entity fails, or in case of disconnections. Moreover, when considering large scale MRS, the use of a central entity which concentrates all data of the system accounts for a bottleneck in term of decision computing and communication.

To avoid these drawbacks, some works have investigated multi-robot coordination in a distributed way. In [10], the authors propose an exploration algorithm based on selection of different behaviors: avoiding obstacles, maintaining network connectivity, exploring around the frontier. This selection takes into account the current network condition which is known by each robot thanks to periodically exchanged messages. To achieve the connectivity of the network, each robot analyses the topology of the network and makes distinction between a simple articulation and a bridge. A simple articulation is a link whose disconnection does not imply the loss of connectivity in the network. A bridge is a link whose disconnection creates two unconnected sub-networks.

Many other works use multi-agent bidding algorithms to achieve multi-robot coordination. An example of such an approach can be found in [7]. The authors introduce a bid calculation allowing robots to find their best target locations. The bid calculation is based on a ponderated sum of three elements: the potential information gain of the targeted location, its distance from the current position of the robot and a nearness measure intended to characterize the ability of maintaining communication links with other robots. To perform coordination, robots periodically broadcast their best bids to all other robots within the same subnetwork. At the end of a constant time, the robot who provides the best bid is declared as winner and is allowed to move towards the target.

This process restarts for all the remaining robots.

Considering the multi-agent literature, speech acts or bidding algorithms are not the only means to achieve coordination; and many other techniques could be most probably applied with success to solve the multi-robot exploration issue. In this article, we propose an original way to formalize and solve this issue based on flocking. Although the idea of flocking was already applied to multi-robots coordination [2], to our best knowledge, this is the first study about applying flocking to the problem of exploration.

The reminder of this paper is organized as following. Section 2 provides a short summary of flocking. Next, we expose in section 3 our proposal and explain how we integrate flocking and exploration. Section 4 is dedicated to the validation of our proposal. We present our simulation setup and analyze results we obtained. Last, section 5 concludes the paper and draws some future works.

2 Flocking Basics

Our proposal takes its inspiration from the work on flocking agents [3]. As introduced by Craig Reynolds, a flock is a group of autonomous agents named *boids* that follow a path in an organized manner. This organization is emerging from decisions taken locally by each boid based on three simple steering rules (see figure 1):

- "**Separation:** *steer to avoid crowding local flockmates*
- **Alignment:** *steer towards the average heading of local flockmates*
- **Cohesion:** *steer to move toward the average position of local flockmates*[1]" [4]

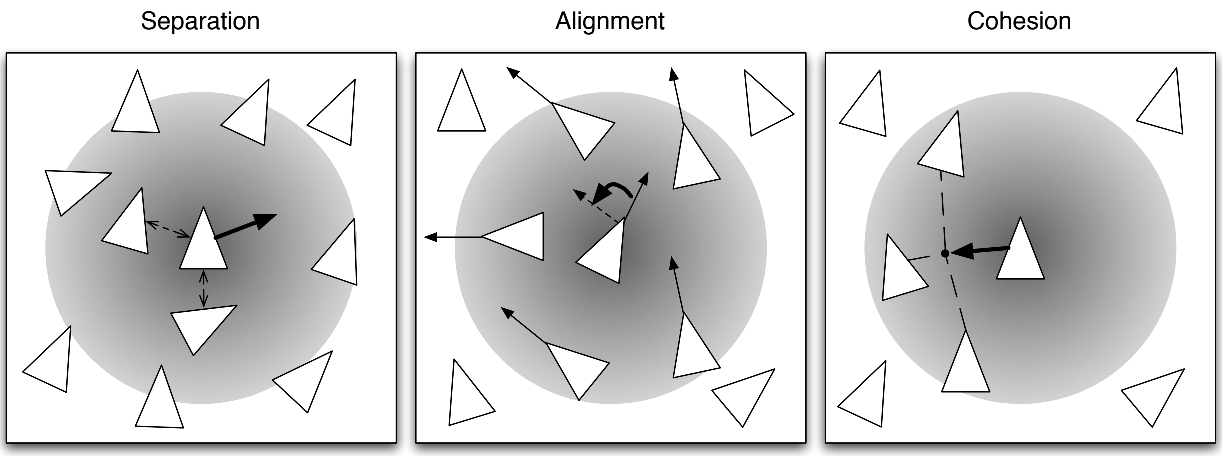

Figure 1: Flocking Relies on 3 Rules for Steering Individual Boids

Each one of these rules participate to computing the velocity vector of a boid. That is, each rule produces a velocity vector based on the velocity vectors of neighboring boids. The boid's actual velocity vector can simply be obtained by a weighted sum of the velocity vectors produced by each steering rule.

[1] `http://www.red3d.com/cwr/boids/`

In order to achieve obstacle avoidance, an extra rule need to be introduced. This rule should be of a higher priority as compared to the previous rules in order to avoid being cancelled out and effectively prevent collisions. In other words, when an obstacle is detected, the boid's velocity vector need to be obtained from the obstacle avoidance rule. Other rules should be temporarily dismissed.

3 Adapting Flocking to Multi-Robot Exploration

3.1 Properties of our proposal

The first property of our solution is that it relies on a *multi-robot system (MRS)*. The challenge of multi-robot exploration is to maximize the exploration speed while minimizing the amount of computational resources needed by each robot. Maximizing the exploration speed can be achieved by minimizing exploration redundancies and maximizing the number of newly visited areas. That is :

- a single robot should minimize visits to areas already explored (by itself or by other robots)
- two robots should not explore the same area at the same time.

Communication is a prerequisite to address these requirements. Robots should be able to share information about explored areas in order to coordinate. Robots network connectivity can be continuous or intermittent.

The second property of our solution concern the maintenance of the *robot network connectivity*. Robots maintain the network connectivity throughout the flocking rules for cohesion and alignment. Therefore, robots can continuously share information they collect and thus have up to date about explored areas. Still redundancies are minimized thanks to the flocking separation rule.

The third and last property of our proposal is that *robots plan their motion to explore new areas*. Robot head towards the frontier between explored regions and unexplored ones, based on the map built collectively. We reuse here the idea of frontier-based exploration [11].

3.2 Robots Steering Rules

Every robot is driven using the same set of rules provided in the ordered list below starting with the rule of highest priority:

($R1$) **Collision avoidance:** Avoid collisions with obstacles or other robots.

($R2$) **Separation:** Turn away from robots within twice the sensors range.

($R3$) **Alignment:** Turn towards the average heading of robots within the wireless network range.

($R4$) **Cohesion:** Turn towards the average position of robots within the wireless network range.

($R5$) **Exploration:** Turn towards the nearest frontier.

Rule $R1$ is of the highest priority to ensure the robot safety and recovery from disconnections. In order to minimize exploration redundancy, rule $R2$ makes the robot turn away from robots that are too close. The minimum distance between two robots is the sum of their respective sensors range. Making the assumption that robots are homogeneous, their sensors have the same range. Thus, the minimum distance to avoid overlapping is twice the sensors range.

By keeping the robot heading close its neighbors headings, rule $R3$ avoids network disconnections. Thus, the robot is prevented from diverging to positions beyond its neighbors wireless network range. Rule $R4$ also avoids network disconnections by ensuring that the robot will keep close to most of its neighbors. We make here the assumption that the communication range is wider than the range of sensors used for exploration. We also assume that the wireless antenna is omnidirectional. Therefore, the orientation of a robot relative to its neighbors does not impact the communication. Only the distance between robots matters.

Last, relying on frontier-based exploration [11], rule $R5$ makes the robot visit new areas. Following Yamauchi's proposal, the frontier is discretized into a collection of cells. Thus, rule $R5$ translates into heading towards the nearest frontier cell.

Rules $R3$, $R4$ and $R5$ are of equal priority. The three are applied when there is no collision, and neighboring robots are far enough. A weighted combination of their respective influence produces the robot heading change.

Note that in order to perform rules $R2$, $R3$, $R4$ and $R5$, every robot requires the distance between itself and its neighbors. It also needs to know the heading of its neighbors. This is why, besides collected information about explored areas (e.g. obstacles), every robot has to broadcast its current position and heading repeatedly. The frequency of these broadcasts depends on the robots speed and the dynamics of the environment in case of a changing environment.

4 Simulation and Validation

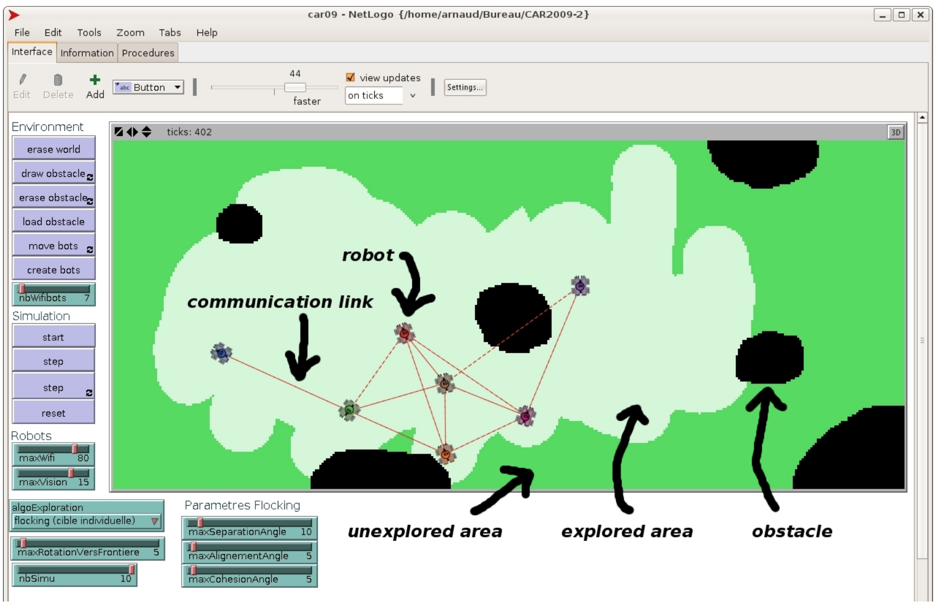

Figure 2: Simulation of Flocking-Based Exploration with Netlogo

To evaluate the performance and the relevance of our approach, we have implemented the exploration algorithm in NetLogo. NetLogo is a multi-agent programmable modeling environment [9] which allows to prototype quickly system of situated agents evolving in a two dimensions world. The discretization of the world can be parametrized by users: the 2D environment can be either simulated as a grid or simulated as a continuous metric space.

In many works on multi-robot exploration, simulations are done with a grid like environment where cells can be empty, occupied by a robot or an obstacle. For our evaluation, we have chosen to simulate a continuous environment of 1083×477 pixels and robots with a size equal to 30 pixels (figure 2). Communication links between robots are represented by dot lines.

In our simulations, we have considered an environment with obstacles at random positions. The same environment was reused for all simulations in order to exclude its impact on the multi-robot systems performance. At the beginning of each simulation, the robots was initially in line with a random heading. The distance between two neighboring robots was less than the communication range in order to ensure the network connectivity on simulation startup. For each simulation, we measured the average duration of the exploration over 30 runs. The number of robots was changed at each simulation in order to observe the impact of the flock size on the performance.

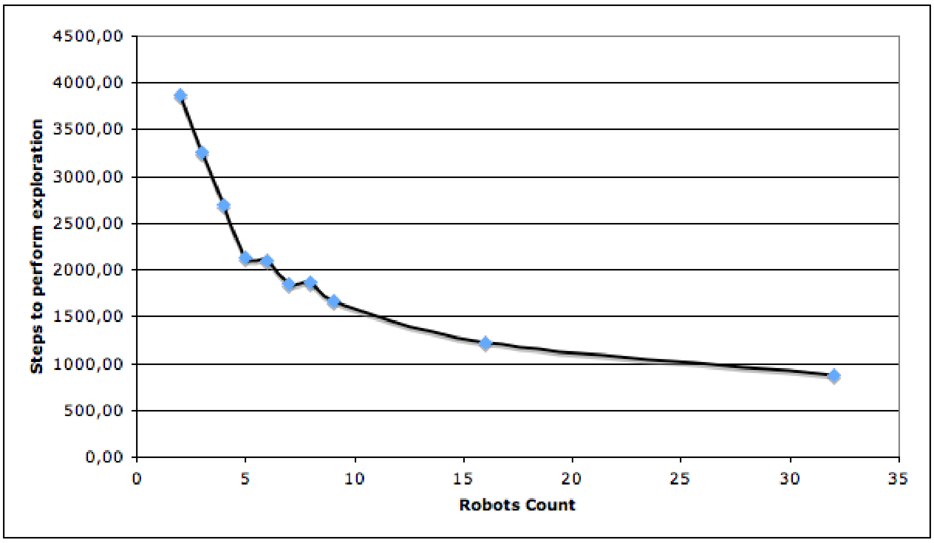

Figure 3: Variation of the Exploration Duration

Figure 3 shows the variation of the exploration duration when the number of robots change. We observe that larger flocks finish exploration faster. The diminution is steady even with a large number of robots. In our simulations we experimented with up to 32 robots and noticed a significant decrease of the amount of time required for an exploration. However, the downside of flocking is an important amount of messages processed by each robot. Indeed, because each robot need to keep track of its neighbors positions and heading, every robot broadcasts on every decision step the information of its location and heading. In the worse case where every robot is within the range of every other robot of the flock, each robot has to process $n - 1$ messages at each decision step, where n is the total count off the robots within the flock. Nevertheless, because of the separation rule ($R2$), this situation does not occur in practice only for small flocks (see figure 4).

It worth noting that flocking does not totally forbid disconnections. According to the robots positions relative to each other and to obstacles, the flock may divide. We found that this separa-

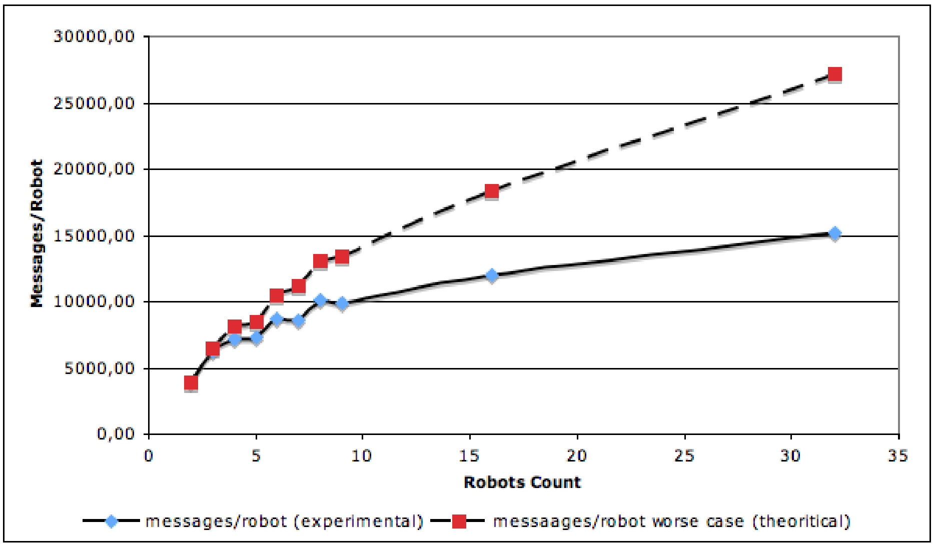

Figure 4: Variation of the Number of Processed Messages per Robot

tion occurs more often with larger flocks than with smaller ones.

5 Conclusion and Future Works

This paper proposes the original idea of mixing flocking with frontier-based exploration strategy in order to achieve multi-robot cooperative exploration while maintaining the network connectivity. Flocking alignment and cohesion rules allow robots to stay close to each other which contributes to keep the robots' network connected. Still, thanks to the flocking separation rule, robots keep away for each other resulting in a reduction of exploration redundancies. We introduced frontier-based exploration as an extra rule that contributes to computing robots' headings. Therefore, it encourages robots to visit new regions.

Simulations we performed using NetLogo exhibit interesting results. The exploration time decreases when the number of robots grows. Yet there is still room for improving the proposed solution. A first future work is related to disconnections that may still occur depending on the obstacle shapes, robots relative positions or failures. We believe that there is no perfect solution regarding connectivity maintenance. Instead, it is more realistic to study approaches for smoothly recovering from disconnections. Besides, we can improve our proposal by taking into account neighboring robots positions while avoiding obstacles in order to maximize connectivity maintenance. For instance, when facing an obstacle, a robot should choose the direction where to turn (i.e. left or right) in such a way to keep close to other robots.

A second important direction for future work consists in taking into account the impact of the environment and actual sensors and actuators on the robot behavior. The environment may for example reduce the wireless range. The sensors are subject to disturbance and noise, while actuators can be of limited precision. In order to evaluate the validity and robustness of our proposal regarding such imperfections we plan on the one hand to perform simulation on a more realistic simulator, namely Player/Stage [1]. On the other hand, we will make experiments on a fleet of four wheeled Wifibot robots with onboard computing resources equivalent to a smartphone, and Wifi communication capabilities. Our first experiments with Wifibots shows that one of the major challenges concerns collecting accurate localization and heading data.

References

[1] B. P. Gerkey, R. T. Vaughan, and A. Howard. The player/stage project : Tools for multi-robot and distributed sensor systems. In *Proceedings of ICAR'03*, 2003.

[2] Magnus Lindhé, Petter Ögren, and Karl Henrik Johansson. Flocking with obstacle avoidance: A new distributed coordination algorithm based on voronoi partitions. In *Proceedings of ICRA 2005*, Barcelona, Spain, 2005.

[3] Craig Reynolds. Flocks, herds and schools: A distributed behavioral model. In Maureen C. Stone, editor, *Computer Graphics (proceeding of SIGGRAPH '87)*, volume 21, pages 25–34, July 1987.

[4] Craig W. Reynolds. Steering behaviors for autonomous characters. In *Proceedings of Game Developers Conference*, pages 763–782, San Jose, California, 1999.

[5] M. N. Rooker and A. Birk. Multi-robot exploration under the constraints of wireless networking. *Control Engineering Practice*, 15(4):435–445, 2007.

[6] H. Shatkay and L. P. Kaelbling. Learning topo-logical maps with weak local odometric information. In *Proceeding of the International Joint Conference on Artificial Intelligence (IJCAI)*, pages 920–929, 1997.

[7] W. Sheng, Q. Yang, J. Tan, and N. Xi. Distributed multi-robot coordination in area exploration. *Robotics and Autonomous Systems*, 54:945–955, 2006.

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

[9] S. Tisue and U. Wilensky. Netlogo: Design and implementation of a multi-agent modeling environment. In *Proceedings of the Agent 2004 Conference*, 2004.

[10] J. Vazquez and C. Malcolm. Distributed multirobot exploration maintaining a mobile network. In *Proceedings of the 2nd International IEEE Conference on Intelligent Systems*, volume 3, pages 113–118, 2004.

[11] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceeding of the second International Conference on Autonomous Agents (Agent'98)*, 1998.