

Metrics for Performance Benchmarking of Multi-robot Exploration

Zhi Yan, Luc Fabresse, Jannik Laval, and Noury Bouraqadi

firstname.lastname@mines-douai.fr

Ecole des Mines de Douai, 59508 Douai, France

<http://car.mines-douai.fr>

Abstract—Performance benchmarking has become an important topic within robotics. It is indeed, a critical way to compare different solutions under different conditions. In this paper, we focus on performance benchmarking of multi-robot systems which explore and map unknown terrains. We present a collection of metrics to objectively compare different algorithms that can be applied to collaborative multi-robot exploration. We also identify parameters that impact robotic fleet performances. By varying the parameters, we can identify strengths and limits of an algorithm. This work is also a first concrete step to address the general problem of objectively comparing different multi-robot coordination algorithms. We illustrate these contributions with realistic simulations of the frontier-based exploration strategy. The simulations were implemented in ROS, which enables to uncouple the control software from the drivers of the robot body. We can therefore use the same code on both simulation and real robots.

I. INTRODUCTION

Many robotic applications can benefit from using a fleet of multiple robots instead of relying on a single robot [1]. Indeed, having multiple robots means an increase of robustness through redundancy. Besides, multiple robots can perform tasks in parallel and thus speed up the execution, which ultimately can increase the benefits of applications such as search & rescue after earthquakes, fire searching inside buildings, mineral exploration, and mine clearance.

However, the use of multi-robot systems raises the coordination challenge [2]. To truly benefit from the potential parallelism of a robotic fleet, we must have strategies for organizing robots activity in a way that ensures highest performance. As an example, consider the exploration of an unknown environment [3] that is a common task in many applications. A coordination strategy should assign to each robot a set of areas to explore in a way that tend to minimize both the time required to build a complete map, as well as the total energy consumed by the robotic fleet [4]. Unfortunately, building optimal or near-optimal coordination strategies is not easy. This is why there is substantial effort put by the research community to address this instance of the problem such as multi-robot exploration [3], [5], [6], [7], [8], [9], [10].

The abundance of algorithms to collaborative multi-robot exploration is a curse when one needs to choose the most appropriate to use for some applications in a given environment, and with some particular set of robots. Authors evaluate their solutions with different robots or simulators, within different environments and conditions. Therefore, results presented in different papers often cannot be compared easily. Moreover,

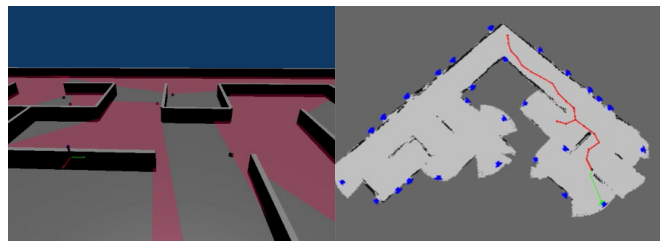


Fig. 1. MORSE 3D simulator (left) and collaborative generated map derived from one robot (right). In the left part, the red area represents a laser scanning. In the right part, the blue blocks indicate potential targets, the green block indicates the current target, the red line indicates the loop closure, and the green line represents the path planning from robot's current position to the target.

reproducibility of experiments, which is at the core of the scientific method, is almost impossible.

While a mathematical evaluation of algorithms - such as complexity analysis - is compelling, it is practically infeasible. This is because of the complexity of multi-robot systems and their environments. There are too many parameters that can dramatically impact performances. Examples of such parameters are: the number of robots, the available processing power or memory on each robot, and the fact that the fleet is homogeneous or heterogeneous.

In this paper, we adopt an alternative approach that is empirical evaluation. It consists in benchmarking algorithms that can be applied to multi-robot exploration, which belongs to the broader area of benchmarking multi-robot systems [11], [12]. To effectively compare different algorithms¹, we introduce five metrics that allow a quantitative performance evaluation. They allow measuring exploration time, cost, and efficiency, as well as completeness and quality of built maps.

We also introduce parameters that impact on exploration performance of a multi-robot system. These parameters aim at easing reproducibility of experiments. They favor the definition of standard environments and reference experiment setups that can be shared by the community, and hence ease the comparison of results obtained by different researchers.

Last, we illustrate our metrics and parameters using a set of results of 3D simulations with ROS-based robots (see Figure 1). ROS (Robotic Operating System) [13] is a meta-operating system, something between an operating system and middleware. It is nowadays acknowledged as a standard

¹In fact, it is algorithms' implementations that are compared.

software platform and is used by numerous institutions. Our simulations target Yamauchi’s strategy for multi-robot exploration based on the concept of frontiers [5] with two map merging algorithms. We show that our metrics can sensitively reflect the impact of different algorithms on the system’s performance. Moreover, to show the influence of parameters, we measured performances of different fleet sizes, in different terrains.

The remainder of the paper is organized as follows: Section II gives an overview of related work; Section III presents our performance metrics for measuring benchmark processes; Section IV discusses the parameters of an exploration system for a fleet of mobile robots; Section V describes the benchmark simulations and the results; the paper is concluded in Section VI.

II. RELATED WORK

There are a few comprehensive and real-world available studies about performance metrics, parameters and benchmarking for multi-robot exploration.

Couceiro *et al.* [14] presented several simulation experiments conducted to benchmark five algorithms for multi-robot exploration and mapping. They used two performance metrics: 1) the exploration ratio of the environment over time, which is calculated as the collective explored map at time divided by the ground truth map; 2) the area under the curve (AUC) that is obtained by calculating the average of 500 iterations of the exploration ratio. In their experiments, the time is represented as the number of exploration iterations. The experiments were conducted using the MRSim Simulator, which is a non-realistic simulator based on the Matlab.

Frank *et al.* [15] compared four different frontier selection strategies for multi-robot exploration and analyzed the performance in terms of amount of iterations needed to explore the entire environment and amount of explored area per time step. The frontier is the set of regions on the border between open space and unexplored space, which is defined by Yamauchi [5]. The simulations were realized in Matlab, and the experiments were performed under ideal conditions: the localization issue was neglected and only convex, obstacle-free environments were regarded.

Amigoni [16] experimentally compared four exploration strategies in order to assess their strengths and weaknesses. The experiments were conducted on a homemade abstract simulator with only one robot. Two metrics were considered to compare the performances of the exploration strategies: the number of laser scanning operations needed to complete the exploration and the total distance travelled by the robot during the exploration.

Balaguer *et al.* [17] presented a methodology for evaluating maps produced by multi-robot systems in the context of RoboCup Rescue competition. They assessed map quality from four criterias including metric quality, skeleton quality, attribution, and utility. They also presented a benchmarking methodology for a simulation testbed. The experiments were performed in simulation with the USARSim simulator and also with two real robots.

Scraper *et al.* [18] focused on the development of standard test methods and techniques for quantitatively evaluating the performance of mobile robotic mapping systems against user-defined requirements. They defined the map quality as the ratio between the number of valid feature points found in the robot built map and the number of feature points in the ground truth map. However, this metric does not assess if the features have the same shape.

Lass *et al.* [19] surveyed several evaluation metrics for multi-agent systems. They classified the metrics along two axes: performance and data types. Performance metrics quantify the resource consumption of the system, such as bandwidth usage, energy consumption, and task duration. The data types include nominal, ordinal, interval, and ratio. This work can be used as a reference, due to multi-robot systems can be regarded as a particular case of multi-agent systems.

We therefore present in this paper a collection of metrics which are independent of the software and hardware, such as the overall exploration time, the total distance traveled, the exploration ratio, and the map quality, while giving a detailed definition. Metrics like the number of laser scanning operations and the simulation step are thus rejected, since different robots and simulators may have different capabilities. In addition, we use a realistic simulator to make the benchmarking results more meaningful for the community.

III. METRICS

Performance measurement is a cornerstone of the rigorous analysis and quantitative comparison. It is especially necessary for robotic exploration nowadays because some of the real world applications have a tight connection with human life, such as search & rescue after earthquakes and fire searching inside buildings. A quantitative comparison of several exploration strategies enables to choose the most efficient one in order to locate victims more quickly.

To our knowledge, there is no accepted standard for quantitatively measuring the performance of multi-robot exploration against user-defined requirements. This motivates us to work towards the development of standardized performance metrics. Our selected metrics have the advantage of being applicable to a wide range of exploration problems with robotic fleets of different features that operate in different types of environments. They are also experimented in the existing literature, which are more likely to be recognized as standard.

On the other hand, the performance metrics are difficult to define because the requirements on which the exploration system is based can be changed according to the user’s needs. In our opinion, firstly, the metrics should be practical and constructed to encourage exploration improvement and secondly, the metrics should meet the requirements of high-efficiency-low-cost exploration and high-accuracy map building, with are of general interest in the community.

In the rest of this section, we introduce five performance metrics to quantify the exploration results, which includes *exploration time*, *exploration cost*, *exploration efficiency*,

map completeness, and *map quality*. These metrics can be used to experimentally assess and compare the performance of different algorithms in both simulated and real world.

A. Exploration Time

One of the goals regarding the optimization of multi-robot exploration is to minimize the overall exploration time [3], [4], [5], [6]. The challenge to achieve this goal is to make each robot move to a different direction in order to maximize the discovered area and minimize at the same time that an area is visited by more than one robot.

We define the exploration time metric as the total time required to complete an exploration mission for a robot fleet. In our definition, timing begins when at least one robot of the fleet starts the exploration, and ends when at least one has explored a target percentage (*e.g.* 99%) of the whole terrain. The time is measured in wall-clock time, showing us how many days, hours, minutes, and seconds that the fleet had spent on the exploration task.

B. Exploration Cost

The definition of the exploration cost highly depends on user's requirements. It could be the energy consumed by the computational resources (*e.g.* CPU, RAM, and network bandwidth), or the price of the robots, or their handling and maintenance costs.

However, the energy consumption is the only one to be directly impacted by exploration strategies. Furthermore, the energy consumed to perform computation can be neglected to that consumed by the motors for the robot's movements. The distance traveled by the robot is a good approximation of the energy cost of the motors. It is simple to measure, which has been widely used in the existing literature [3], [4], [6], [7], and especially for solving the problem of task allocation in multi-robot systems.

Thus, we define the exploration cost metric as the sum of the distances traveled by each robot of the fleet:

$$\text{explorationCost}(n) = \sum_{i=1}^n d_i \quad (1)$$

where n is the number of robots in the fleet, and d_i is the distance traveled by robot i .

C. Exploration Efficiency

Efficiency can be defined as the ratio between the input to run a task and the output gained from the task, from the economic point of view. In the context of multi-robot exploration, the exploration cost can be considered as the input while the explored area can be considered as the output. The exploration efficiency is therefore directly proportional to the amount of information retrieved from the environment, and is inversely proportional to the costs incurred by the robot fleet [7]:

$$\text{explorationEfficiency}(n) = \frac{M}{\text{explorationCost}(n)} \quad (2)$$

where n is the number of robots in the fleet, and M is the total explored area in square meters.

For example, if the value of the exploration efficiency is 1.6, meaning that each time all robots from the fleet move by 1m, they discover on average 1.6m² of the terrain. Inspired from the benefit-cost ratio analysis in economics, users can consider that an algorithm is worthy to use if it has a value greater than or equal to 1.

D. Map Completeness

Map building is a task tightly coupled with the exploration. The completeness of robot-built map is a major problem researchers are concerned about [14], [15], [18]. This metric requires a prior knowledge about the terrain to be explored and mapped. We define the map completeness as the ratio between the amount of explored area M and the total area of ground truth map P :

$$\text{mapCompleteness} = \frac{M}{P} \quad (3)$$

E. Map Quality

Build up an entirely accurate map by autonomous robots is still an open problem. Reasons for the errors of a map could be accuracy of sensors or algorithms used for SLAM. To identify these errors, we need a ground truth map of the terrain. Since the occupancy grid map is widely used to represent the unknown, occupied and free space in the exploration problem, we first define the map error as the number of cells in the explored grid map that have a different value from the corresponding cell in the ground truth map.

Results calculated using this definition are also affected by the resolution of the map. A high resolution requirement creates a large cardinal of cell. By using the same sensors and algorithms, the error is likely to be more important in a high resolution map than in a low one. A good exploration performance must display a tradeoff between the map error and its resolution.

Unlike the map quality metric defined in [14], [18] which mainly focuses on the completeness of the built map (*i.e.* the percent of area mapped), we are more concerned about its topological accuracy [17]. We then define the map quality as the overlap of the explored map and the ground truth map as a ratio to the total area of the ground truth map P :

$$\text{mapQuality} = \frac{M - A(\text{mapError})}{P} \quad (4)$$

where M is the total explored area in square meters, and $A(\text{mapError})$ is the area occupied by the error cells.

IV. BENCHMARK PARAMETERS

To evaluate an algorithm for collaborative exploration or to compare many of them, one has to choose the environment to explore and the robots to use. This is a specification of the benchmark. Different parts of a such specification may vary and have an impact on exploration performance. Indeed, changing one of these parameters may significantly affect one or more metrics. We list below, parameters that we found relevant grouped into three families: *Robot*, *Fleet*, and *Environment*.

Our goal is to provide the community with basis to define a reference database of benchmark setups. Each setup refers to a different configuration of parameters. This idea already adopted in other areas (*e.g.* databases of images for object or facial recognition), has already been partially addressed in the RoboCup Rescue competition for example with different arena².

A. Robot

- Locomotion properties. It covers characteristics of the robot such as the motion model (holonomic or not).
- Computing capabilities: CPU, RAM, Clock frequency. When choosing an exploration algorithm, one has to take into consideration resources available for computing. Simple algorithms running on constraint devices might have better performance than sophisticated complex algorithms.
- Sensor precision, frequency, and range. Sensor characteristics impact localization and map construction, and hence may impact map quality.

B. Fleet

- Number of robots. Intuitively one might think that more robot can lead to faster exploration. But, this actually depends on the coordination strategy.
- Fleet homogeneity. The use of heterogeneous fleet such as ones mixing arial robots with terrestrial ones may leverage exploration performance.
- Robots initial positions. Depending on the environment and obstacles, exploration performance may be significantly impacted by robot positions when starting up the exploration [4].
- Communication bandwidth. Some algorithms require robot exchange large amounts of data. Their performance might significantly drop when using robots with network interfaces that offer a limited bandwidth.
- Communication range. Collaborative multi-robot exploration often requires communication which is usually achieved through some wireless networks. Depending on the used wireless network, communication range can vary. This range impacts coordination and thus exploration performance. Indeed, in large terrains or depending on the obstacle densities and materials, wireless transmissions may be slowed down. Robots might even get disconnected and be unable to communicate or cooperate. However, this issue can be mitigated by taking into account network connectivity in path planning [10], [20].

C. Environment

- Terrain size. Usually, exploring a large terrain requires more time than a smaller one. However, this can sometimes be mitigated by increasing the number of robots.
- Obstacles density and shapes. In an environment with many obstacles, there is less space to explore. On the other hand, navigation may be more complicated,

especially with concave obstacles where deadlocks can occur or when multiple robots are located in the same area [21].

- Landforms. The exploration of a large single area takes probably less time than an environment that is decomposed into a number of open areas, but connected with narrow corridors. In the latter, it is likely that robots might obstruct one another.
- Dynamicity. If the environment is changing (*e.g.* building collapses) or if they are other mobile entities (*e.g.* human rescuers or other robots), exploration time and associated costs can vary for different test runs.

V. SIMULATIONS

To illustrate our benchmark metrics, we conducted a series of simulations using the Yamauchi's frontier-based multi-robot exploration strategy [5]. In this decentralized strategy, each robot decides autonomously where to go based on its exploration map. Map exchange is the only cooperative task. Once a robot updates its map, it selects the nearest frontier and moves towards it. As a benchmarking example, we assess the impact of two map merging algorithms on the exploration performance.

We experimented with different robotic fleet sizes ranging from 1 to 30 robots, and four terrains varying from simple to complex shown in Figure 2. These terrains, inspired by the RoboCup Rescue competition, have the same size but different explorable areas:

- 1) The *loop* terrain has a low obstacle density and a simple obstacle shape, in which there is no road fork. This terrain represents a beltway.
- 2) The *cross* terrain contains five road forks but the obstacle density is still low. This terrain corresponds to a crossroad.
- 3) The *zigzag* terrain has no road fork but more obstacles. Moreover, it has a long solution path for the robot. This terrain inspired by the square-grid street network like in Barcelona.
- 4) The *maze* terrain is the most complex one which contains a lot of obstacles and dead ends. This terrain can be viewed as a sample of the whole city.

A. Testbed

To facilitate an analytical comparison of different algorithms in different conditions, we have constructed a testbed for data collection. Figure 3 depicts the architecture of our testbed. It consists in:

- MORSE robotics simulator [22] provides a realistic physics engine enabling 3D simulation. It is deployed on a workstation with 8 processors, 8GB RAM, a GeForce GTX 770 graphics card and a 1000Mbps Ethernet adapter.
- ROS *de facto* standard middleware is used to build the robot control software. Its modularity enables to uncouple the control software from the drivers of the robot body. This allows us to use exactly the same control software on both simulation and real robots.

²http://wiki.robocup.org/wiki/Robot_League#Field_Description

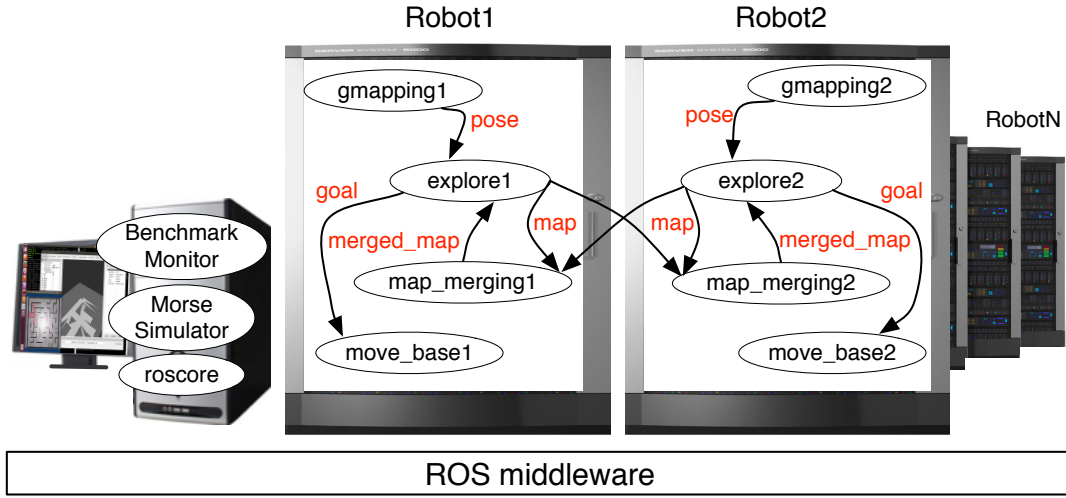


Fig. 3. Our testbed architecture.

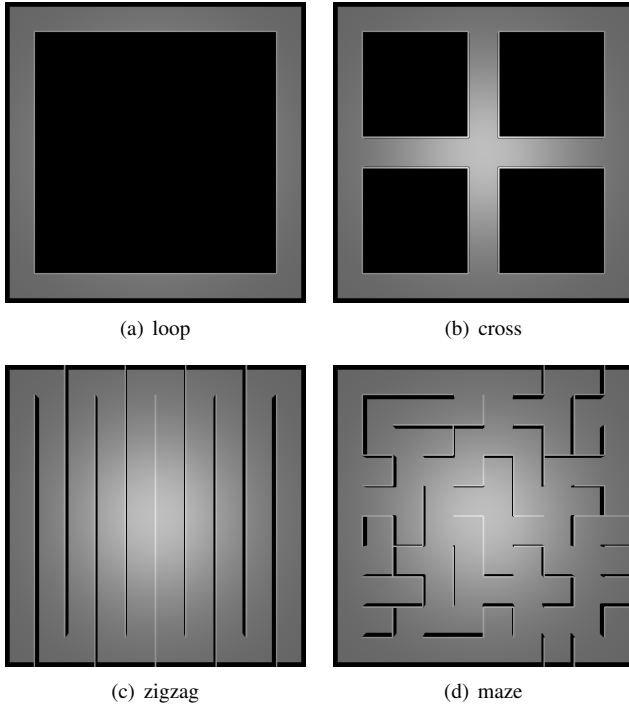


Fig. 2. Terrains created for performance benchmarking.

- A computer cluster is used to provide a high performance distributed computing to meet the computation requirements for realistically simulating large-scale robots. Our cluster consists of 70 computing nodes, in which each computing node contains multiple processors varying from 8 to 12, and RAM varying from 16GB to 48GB.

Based on our testbed, we define each robot controller as a graph of ROS packages. Most important ones are represented in Figure 3 and described below.

- *gmapping*: This package performs a laser-based SLAM

[23]. We use it to extract robots pose, that is feeded to the *explore* package.

- *explore*: The original package performs Yamauchi's single robot frontier-based exploration. Our adaptation introduces support for multi-robot exploration, by making the node use maps provided by the *map_merging* package.
- *map_merging*: This package merges multiple exploration maps by knowing the relative initial robot positions. The resulting map is sent to the *explore* package to make the robot choose the next target location.
- *move_base*: This package implements the Dijkstra's algorithm for global path planning. It supports Trajectory Rollout and the Dynamic Window for local collision avoidance.

We released our developed packages on our website at <http://car.mines-douai.fr> as *open source* with the intention to provide the community a replicable system, in order to speed up result comparisons. We also validated our ROS-based multi-robot system with a fleet containing two robuLAB-10 robots for indoor exploration. A video of the implementation is also available on our website.

B. Map merging algorithms

We are interested in the impact on the exploration performance of two map merging algorithms respectively used in [5] and [6]. The first one [5] is a greedy algorithm that simply focuses on unknown space. The second one [6] is a probabilistic algorithm in which the robot builds the merged map by calculating the probability that each cell in the explored grid map is occupied. Both of these algorithms are run in real time during exploration, and require the knowledge of the initials positions of the robots.

C. Fixed parameters

Table I summarizes parameters for which we have chosen fixed values in our experiments. We can see that we used a homogeneous team of simulated Pioneer 3-DX robots with 2

TABLE I

THE FIXED PARAMETERS IN THE BENCHMARK EXPERIMENT.

Robot	
Computing capability	2 CPUs, 2GB RAM
Maximum speed	1.2m/s, 5.24rad/s
Odometry noise	0.022m, 0.02rad
Laser rangefinder	SICK LMS500
Fleet	
Homogeneity	homogeneous (Pioneer 3-DX)
Robot initial positions	top left to bottom left corner, every 2m
Communication network	gigabit wired Ethernet
Communication range	200m
Environment	
Terrain size	80m \times 80m
Obstacle height	2m
Corridor width	8m

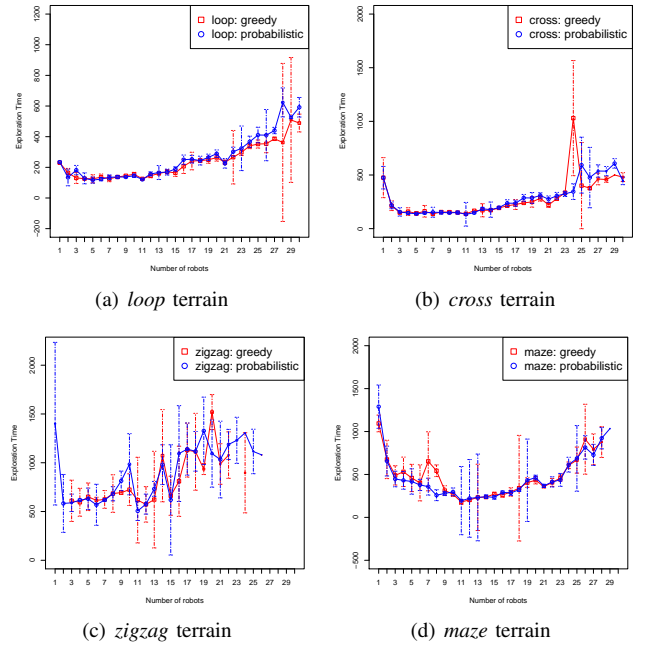
CPUs and 2GB RAM. Each robot is equipped with a SICK LMS500 laser scanner, which provides 180 sample points with 180 degrees field of view and a maximum range of 30 meters. Consistent with real one, the maximum speed of each simulated robot has been fixed to 1.2 meter per second for linear motion and 5.24 radians per second for rotational motion. A zero mean Gaussian white noise has been added to the odometry data. The standard deviation is 0.022 meters for position noise (x and y) and 0.02 radians for rotation noise. This noise is close to the actual one in Pioneer 3-DX robot, making our simulation more realistic.

The robots are initially placed along a vertical line starting from the top left corner of the terrain to the bottom left corner. The distance between robots' initial positions is set to 2 meters. The robots communicate with each other through a gigabit wired Ethernet network. The maximum range of communication between them is set to 200 meters based on their relative position in the simulated environment. The impact of obstacles on communication is currently ignored. Although this setting does not enable the comparison of the algorithms under realistic communication, we only focus on evaluating the impacts of the map merging algorithms. In fact, this setting is a direct consequence of the fact that the MORSE simulator we used does not support this feature. Nevertheless, we planned to tackle this important point in our future work and support different models of communication in our testbed.

All terrains are 80 meters long and 80 meters wide. The height of the obstacles is set to 2 meters and the width of corridors is fixed to 8 meters.

D. Results

The results can be seen in Figures 4, 5, 6 and 7. Four metrics are selected for benchmark testing, i.e. *exploration time*, *exploration cost*, *exploration efficiency*, and *map quality*. Since the completeness of the collaborative built map is a prerequisite to compare different map merging algorithms, we did not measure the *map completeness*. Each figure contains four plots, each corresponding to one terrain. In each plot, the abscissa denotes the fleet size of the mobile robots, the ordinate denotes the metric measurements, and the error

Fig. 4. Results with the *exploration time* metric.

bar indicates the confidence interval of each corresponding measurement of fleet size with the 0.95 confidence level. The red square represents the greedy algorithm and the blue circle represents the probabilistic algorithm.

Since, 1) there are several non-deterministic components in the simulated environment such as noises on laser scan and odometry; 2) the algorithm implemented for SLAM is a probabilistic algorithm; 3) a shared computer cluster with no-constant network bandwidth is used as it is often the case in many universities, we performed five runs for each fleet size, and display the median value of these runs. A monitor (a ROS package) is deployed on the workstation, which end each run when 99% terrain is discovered (successful run) or when exploration time exceeds 2000 seconds (failed run).

The size of symbol for each median value in the plot varies with the number of successful run (one to five runs). The median value will not be displayed if all five runs fail. Such no-displays occur for example when the fleet size is greater than 26 robots in the *zigzag* terrain. Causes of the failed run may come from two aspects, including the use of ROS-based robot controller and the uncertainty of network traffic:

- ROS is a system being forward along with its distributed architecture, variety packages and support for multi-robot systems. The loss of robot localization and the failure of long-distance path planning occasionally appeared in our experiments.
- As mentioned earlier, the computer cluster is shared use. When the experiment runs on network peak periods, we may get a failed run, or an outlier result (*cf.* 24 robots in the *cross* terrain by implementing the greedy algorithm in terms of *exploration time*, *exploration cost* and *exploration efficiency*).

The figures show that differences on the results between

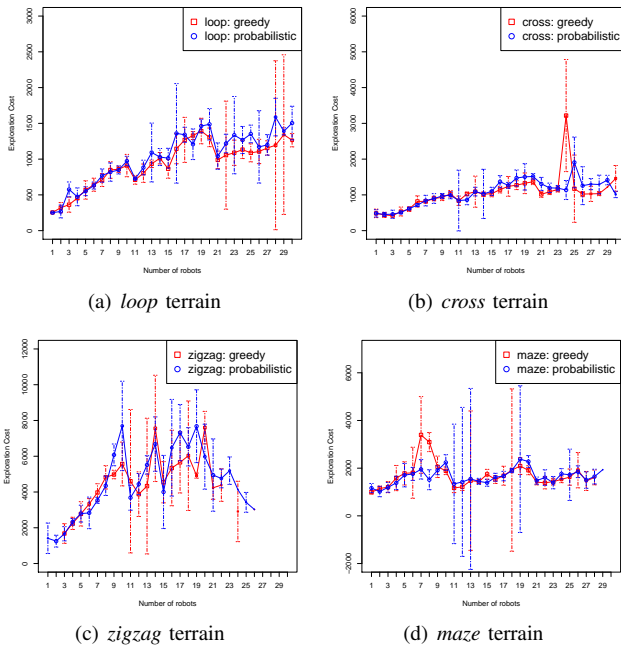


Fig. 5. Results with the *exploration cost* metric.

the greedy algorithm and the probabilistic one in terms of the *exploration time*, the *exploration cost* and the *exploration efficiency* are not significant. In general, the tendency of the *exploration time* and the *exploration cost* is to increase and the *exploration efficiency* is to decrease when the number of robots is increased, except for the *maze* terrain. The main reason is that more robots in the fleet leads them spend more *time* to avoid collision with others. To this end, robots usually need to replan their trajectories, thus the *exploration cost* is increased. Since the simulation stops when a 99% of the required area has been explored, then the term M in the expression for the *exploration efficiency* is a constant value for each terrain, so the *exploration efficiency* is inversely proportional to the *exploration cost*. An interesting aspect based on these results is that the optimal size of the robot fleet can be assessed for a given terrain. For example, when exploring the *maze* terrain, the ideal fleet should have 11 robots. Indeed, this fleet size minimizes the *exploration time* and the *exploration cost* while ensuring a high *exploration efficiency*.

Figure 7 shows that the influence of the greedy algorithm and the probabilistic one is mainly on the quality of the map. This result is expected and demonstrates the sensitivity of our defined metrics.

Furthermore, results show that, with the *zigzag* terrain, performance metrics are the worse, which are hugely influenced by the parameter of robots initial positions. Simulations showed that exploration is mainly performed by a single robot. Indeed, there is only a single frontier and it is always close to the same robot which is closer to the top left corner of the terrain.

Based on these experimental results, it is clear that the robot fleet size and the terrain layouts make a strong impact

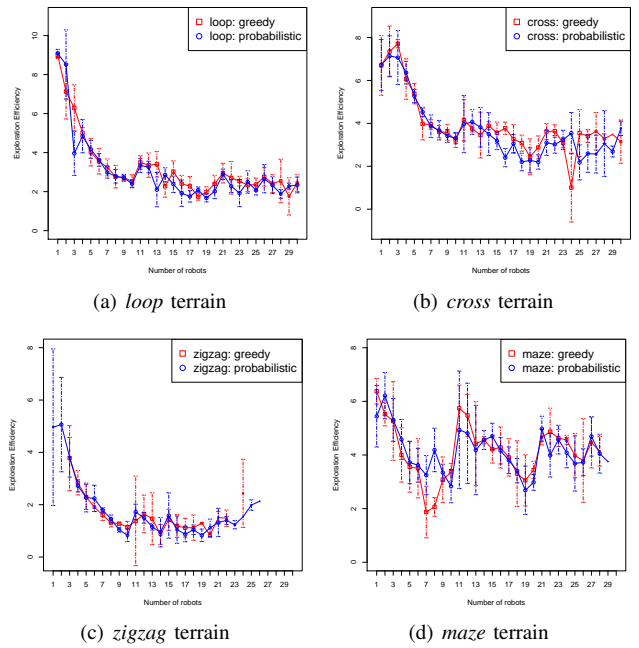


Fig. 6. Results with the *exploration efficiency* metric.

on the *exploration time*, the *exploration cost*, and the *exploration efficiency* of the multi-robot system, while the major effect of the map merging algorithms is found to be the *map quality*.

VI. CONCLUSIONS

In this paper, we considered the performance benchmarking for multi-robot exploration. Our concern is, how to quantitatively compare different algorithms and perform an objective evaluation on a common predefined settings. It is not easy to address this question due to the complexity of multi-robot systems and environments they explore.

To address this issue, we have introduced five metrics to quantify the exploration performance, namely: *exploration time*, *exploration cost*, *exploration efficiency*, *map completeness*, and *map quality*. These metrics, can be used in both simulated systems and real ones.

We also have identified several parameters that impact the exploration performance. Clearly stating these parameters allow to increase reproducibility and repeatability of experiments. We view these parameters as a first step towards the definition of standard environments and reference experiment setups that can be shared by the research community.

To illustrate our contribution, we benchmarked two map merging algorithms used for Yamauchi's frontier-based multi-robot exploration strategy. We relied on simulations conducted using the MORSE 3D simulator, with ROS-based robot controllers. While most parameters had fixed values, we varied a few of them: the number of robots and terrain layouts. By using some of the defined metrics, we thus showed the impact of different algorithms on the exploration performance.

As future work, we would like to define a collection of

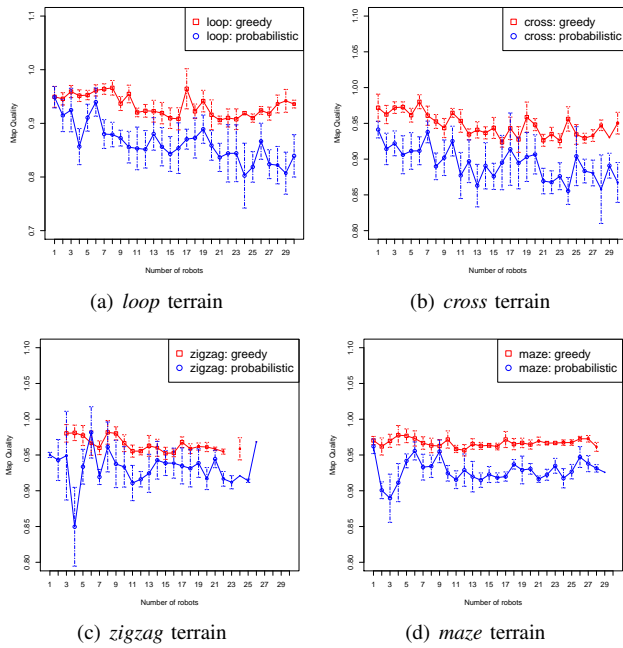


Fig. 7. Results with the *map quality* metric.

reference values for the parameters we have identified. The collection would consist in vectors, where each vector correspond to particular values of characteristics of the robots, the fleet, and the environment used for benchmarks. Our goal is to provide the community with the seed for a database that will be used for comparing algorithms. The next logical step would be to use this database to compare existing algorithms. This should ultimately provide us with insights on how existing solutions compare, and which exploration strategy to pick given a particular problem.

Another direction for future work could be to go a step further in making even more realistic simulations by introducing an engine for radio-wave propagation and absorption into obstacles such as walls.

ACKNOWLEDGMENT

This work is part of the SUCR  project that is supported by R gion Nord Pas-de-Calais.

REFERENCES

- [1] L. E. Parker, "Multiple mobile robot systems," *Springer Handbook of Robotics*, pp. 921–941, 2008.
- [2] Z. Yan, N. Jouandeau, and A. A. Cherif, "A survey and analysis of multi-robot coordination," *International Journal of Advanced Robotic Systems*, vol. 10:399, December 2013.
- [3] C. Stachniss, *Robotic Mapping and Exploration*. Springer, 2009.
- [4] Z. Yan, L. Fabresse, J. Laval, and N. Bouraqadi, "Team size optimization for multi-robot exploration," in *Proceedings of the 4th International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR 2014)*, Bergamo, Italy, October 2014, pp. 438–449.
- [5] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*, Minneapolis, MN, USA, May 1998, pp. 47–53.
- [6] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun, "Collaborative multi-robot exploration," in *Proceedings of the 2000 IEEE International Conference on Robotics and Automation (ICRA'00)*, San Francisco, CA, USA, April 2000, pp. 476–481.

- [7] R. Zlot, A. T. Stentz, M. B. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," in *Proceedings of the 2002 IEEE International Conference on Robotics and Automation (ICRA'02)*, Washington, DC, USA, May 2002, pp. 3016–2023.
- [8] A. Howard, "Multi-robot simultaneous localization and mapping using particle filters," *International Journal of Robotic Research*, vol. 25, no. 1, pp. 1243–1256, January 2006.
- [9] W. Sheng, Q. Yang, J. Tan, and N. Xi, "Distributed multi-robot coordination in area exploration," *Robotics and Autonomous Systems*, vol. 54, no. 12, pp. 945–955, January 2006.
- [10] A. Doniec, N. Bouraqadi, M. Defoort, V. T. Le, and S. Stinckwich, "Distributed constraint reasoning applied to multi-robot exploration," in *Proceedings of 21st IEEE International Conference on Tools with Artificial Intelligence (ICTAI'09)*, November 2009, pp. 159–166.
- [11] F. Bonsignorio, A. P. D. Pobil, and E. Messina, "Fostering progress in performance evaluation and benchmarking of robotic and automation systems," *IEEE Robotics & Automation Magazine*, vol. 21, no. 1, pp. 22–25, March 2014.
- [12] A. P. D. Pobil, "Why do we need benchmarks in robotics research," in *Proceedings of the IROS'06 Workshop on Benchmarks in Robotics Research*, Beijing, China, October 2006.
- [13] M. Quigley, K. Conley, B. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "ROS: an open-source robot operating system," in *Proceedings of the ICRA'09 workshop on open source software*, Kobe, Japan, May 2009.
- [14] M. S. Couceiro, P. A. Vargas, R. P. Rocha, and N. M. F. Ferreira, "Benchmark of swarm robotics distributed techniques in a search task," *Robotics and Autonomous Systems*, vol. 62, no. 2, pp. 200–213, January 2014.
- [15] S. Frank, K. D. Listmann, A. D. Haumann, and V. Willert, "Performance analysis for multi-robot exploration strategies," in *Proceedings of the 2nd International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR 2010)*, Darmstadt, Germany, November 2010, pp. 399–410.
- [16] F. Amigoni, "Experimental evaluation of some exploration strategies for mobile robots," in *Proceedings of the 2008 IEEE International Conference on Robotics and Automation (ICRA'08)*, Pasadena, CA, USA, May 2008, pp. 2818–2823.
- [17] B. Balaguer, S. Balakirsky, S. Carpin, and A. Visser, "Evaluating maps produced by urban search and rescue robots: lessons learned from RoboCup," *Autonomous Robots*, vol. 27, no. 4, pp. 449–464, November 2009.
- [18] C. Scrapper, R. Madhavan, R. Lakaemper, A. Censi, A. Godil, A. Wagan, and A. Jacoff, "Quantitative assessment of robot-generated maps," in *Performance Evaluation and Benchmarking of Intelligent Systems*. Springer, 2009, pp. 221–248.
- [19] R. N. Lass, E. A. Sultanik, and W. C. Regli, "Metrics for multiagent systems," in *Performance Evaluation and Benchmarking of Intelligent Systems*. Springer, 2009, pp. 1–19.
- [20] V. T. Le, N. Bouraqadi, S. Stinckwich, V. Moraru, and A. Doniec, "Making networked robots connectivity-aware," in *Proceedings of the 2009 IEEE International Conference on Robotics and Automation (ICRA'09)*, Kobe, Japan, May 2009, pp. 3502–3507.
- [21] A. Lampe and R. Chatila, "Performance measure for the evaluation of mobile robot autonomy," in *Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA'06)*, Orlando, FL, USA, May 2006, pp. 4057–4062.
- [22] G. Echeverria, S. Lemaignan, A. Degroote, S. Lacroix, M. Karg, P. Koch, C. Lesire, and S. Stinckwich, "Simulating complex robotic scenarios with MORSE," in *Proceedings of the 3rd International Conference on Simulation, Modeling, and Programming for Autonomous Robots (SIMPAR 2012)*, Tsukuba, Japan, November 2012, pp. 197–208.
- [23] G. Grisetti, C. Stachniss, and W. Burgard, "Improved techniques for grid mapping with rao-blackwellized particle filters," *IEEE Transactions on Robotics*, vol. 23, no. 1, pp. 34–46, February 2007.