Team Size Optimization for Multi-robot Exploration

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

Ecole des Mines de Douai
59508 Douai, France
{firstname.lastname}@mines-douai.fr

Abstract. This paper analyzes and discusses the problem of optimizing the size of a team of robots for multi-robot exploration. We are concerned with the number of robots for a given exploration task that minimizes both exploration time and cost. Minimizing time means that the exploration should be done as fast as possible. Minimizing cost means that the number of robots and their energy consumption should be as low as possible. To solve this problem, we report in this paper, on a series of exploration simulations based on ROS and MORSE using a cluster of computers. The simulated code is exactly the same as that which would run on the actual robots. Such a simulation infrastructure is crucial to "quickly" execute experiments with different parameters such as the number of robots or their initial positions.

Keywords: Multi-robot systems, exploration, simulation, ROS, MORSE

1 Introduction

The problem of multi-robot exploration is a primary research topic within multi-robot systems. It requires a group of robots to explore an unknown environment in cooperation, and usually also needs the construction of a map of this environment.

In recent years, the manufacturing of the robot has been considerably developed. Therefore, finding a suitable robot team size for exploration missions becomes a meaningful question. For example, in case of an earthquake, robots can help rescuers to evaluate the damage to the interior of a building. In this case, it is important to do this evaluation as quickly as possible. Consequently, a multi-robot system is a solution. The question is how many robots do we need in such system. Having only a few robots will require a long exploration time and the risk of failure is important: if one robot stops its exploration, a large part of the system is impacted. If rescuers deploy many robots, the system's robustness is increased, but the robots may take too much time to explore because they have to avoid a lot of other teammate robots.

In this paper, we address the issue of team size optimization using realistic simulations based on the robotic middleware ROS (Robot Operating System) [8] and the 3D simulator MORSE [4]. We show how to determine the optimal size of a team of robots in order to complete an exploration mission in the shortest time possible and with the lowest cost. We consider two metrics to measure the optimal size of the robot team:

- The time metric. It is the total time required to complete an exploration mission.
- The cost metric. It is the sum of energy consumed by all robots in the team.

The remainder of the paper is organized as follows: Section 2 describes an overview of related work; Section 3 describes our multi-robot exploration system; Section 4 describes our evaluation metrics to the team size optimization problem. Section 5 describes the experimental results obtained with our system. We discuss this work in Section 6, and conclude the paper with Section 7.

2 Related Work

Yamauchi [10] introduced an approach for robotic exploration based on the concept of frontiers. In this approach, a robot can build a grid map with information obtained from laser and sonar sensors, detect the frontier which is the region on the boundary between open space and unexplored space in the map, then navigate to the nearest accessible frontier. By using the proposed approach, a Nomad 200 mobile robot was able to map the open spaces quickly, mapping an environment with 45 feet long and 25 feet wide in about half an hour.

Yamauchi [11] then extended this frontier-based approach to multi-robot systems. He constructed a decentralized system in which each robot has its own global grid map representing its knowledge about the environment. Whenever a robot arrives at a new frontier, it constructs a local grid map representing its current surroundings. This local map is integrated with the robot's global map, and also broadcasted to all of the other robots. Then each robot integrates the local map received from its teammates with its global map. This strategy requires robots to know their relative positions at the beginning of exploration, and use dead reckoning alone for position estimation so as to properly blend the local map and the global map. A limitation of the proposed approach is that robots may waste time by navigating to the same frontier since there is no coordination.

Burgard *et al.* [2] designed a coordination component based on the approach of Yamauchi. This component applies a probabilistic method which takes the cost of reaching a frontier and its utility into account simultaneously. The cost is given by the distance of traveling to a frontier (by using a value iteration algorithm) and the utility is given by the size of the unexplored area that a robot can cover from this frontier using its sensors. Whenever a frontier is assigned to a robot, the utility of the visible unexplored area of this frontier is reduced to all its teammates, making all other robots explore different areas. Their experimental results show that the coordinated robots can accomplish an exploration task significantly faster than uncoordinated robots.

Howard [5] described a multi-robot simultaneous localization and mapping (SLAM) approach by using a particle filter. The proposed approach is able to handle the case in which the initial position of robots are unknown. They start mapping with only one robot (whose initial pose is arbitrary) and wait until this first robot encounters other robots before incorporating their data into the global map.

Stachniss [9] presented their work on collaborative mapping with teams of mobile robots. Their multi-robot mapping system needs to place the robots in nearby locations. Robots also need to know the relative initial poses of their team members. During exploration, robots within communication range can exchange maps. We have implemented this solution in our simulated multi-robot mapping system.

Lass *et al.* [6] surveyed several evaluation metrics for multi-agent systems. They classified the metrics along two axes: the effectiveness or performance of metrics and the types of data they represent. Measures of effectiveness quantify the system's ability to complete its task in a given environment, while measures of performance are quantitative mesures of some secondary performance characteristics, usually being resource consumption of the system, such as bandwidth usage, energy consumption, communication range or task runtime.

3 Multi-robot Exploration System

Our robots rely on laser sensing for both localization and mapping. We use the ROS (Robot Operating System) middleware for communication both between control software and simulated robots. We also use ROS for inter-robot communication and more specifically for map sharing.

3.1 Single-robot Setup

The main functions are achieved by the following packages:

- *gmapping*: This package is provided by ROS, which realizes the function of laser-based SLAM. It is used for mobile robot localization. Specifically, it sends pose data to the *explore* package.
- *explore*: The original package is provided by ROS which realizes the frontier-based exploration approach. It has been modified by our research team to be compatible with multi-robot systems. Specifically, a subscriber has been added to receive the map generated from the *map_fusion* package, so as to update the robot's current exploration map.
- *map_fusion*: This package is realized by our research team, which merges multiple exploration maps by considering the relative initial position of the robots, then transfers the fused map to the *explore* package.
- *move_base*: This package is provided by ROS, which navigates the robot to a goal location.

The relationships between the packages are illustrated in Figure 1.

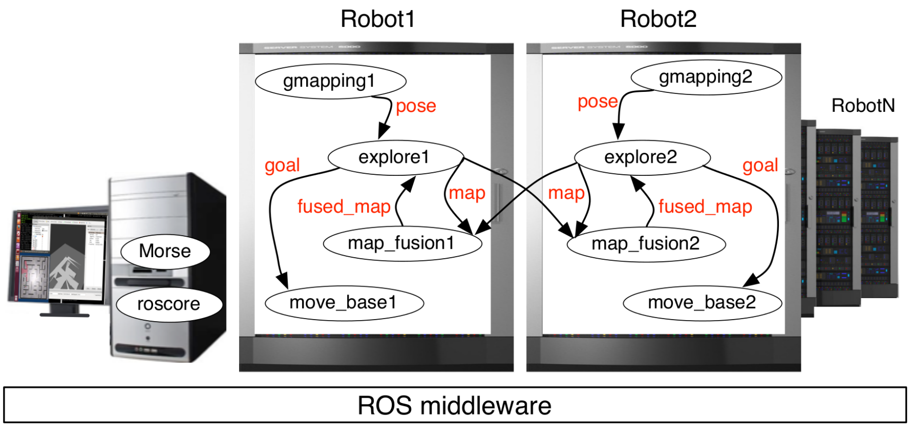


Fig. 1. Our distributed multi-robot exploration system relies on the ROS middleware and the MORSE 3D simulator. Each robot is simulated by a computer that runs 4 ROS nodes.

3.2 Multi-robot Communication

After preparing a single robot with exploration capabilities, the next problem we had to solve is the communication in our multi-robot system. Our current exploration system is a decentralized one, in which each robot can make its own decisions according to the local information with limited communication. In order to cooperate, we introduce some level of communication between neighboring robots [12]. To simulate network range, we introduce a discovery algorithm based on distance between simulated robots. Algorithm 1 illustrates our connection establishment process for $robot_i$.

Algorithm 1 Communication Connection for $robot_i$

1: Querying all published ROS topics
2: Subscribing to robot pose topics
3: **if** $\exists robot_j \in$ exploration team : $distBetween(robot_j - robot_i) < max\_comm\_distance$ **then**
4: Establishing connection with $robot_j$
5: **end if**

In order to have a quite realistic simulation at least regarding the scale factors, we set a value that indicates the maximum communication distance (i.e. $max\_comm\_distance$) for our multi-robot system, but the impact of obstacles on communication is currently ignored. Moreover, to calculate the distance between two robots, we supposed that the relative initial positions of robots are known.

3.3 Multi-robot Mapping

Each robot in our simulated multi-robot system needs to exchange the grid map with its teammates in order to perform exploration mission cooperatively. Our current map fusion algorithm is lightweight and straightforward, by still supposing that the relative initial positions of robots are known (see Algorithm 2).

Algorithm 2 Map Fusion for $robot_i$

$\delta \leftarrow (robot_i.init\_pose - robot_j.init\_pose) \times map\_scale$
$robot_i.fused\_map \leftarrow robot_i.map$
for all $grid$ in $robot_i.fused\_map$ **do**
  if $grid = NO\_INFORMATION$ **then**
    $grid \leftarrow robot_j.map_{grid.pose+\delta}$
  end if
end for

3.4 Multi-robot Motion Planning

In our current implementation, map exchange is the only cooperative task done by the robot team. Each robot decides autonomously where to go based on its own grid map. Once the robot has updated its current map, it will select the nearest frontier and move towards it. This solution is not optimal, because different robots may go to explore the same frontier, resulting into redundant and useless exploration and possibly obstructing pathways.

Figure 2 shows three Pioneer 3-DX robots equipped with a SICK LMS500 laser scanner during an exploration mission. The left half of the figure is derived from the 3D simulator MORSE, and the right half is derived from the 3D visualization tool RVIZ [1].

4 Evaluation Metrics

Our goal is to find the optimum size of a robot team (denoted by n) for the purpose of exploring a given terrain. Optimization targets identify the shortest exploration time (denoted by $time$) and the lowest energy cost (denoted by $cost$). The cost refers to the total of energy consumed by a robotic team to perform an exploration mission. We supposed that the energy consumption is proportional to the distance traveled of all the robots in the team. For example, a team of two robots that move forward 10 meters each, consumes 20 units of energy.

$$cost(n) = \sum_{i=1}^{n} (distanceTraveled(robot_i)) \tag{1}$$

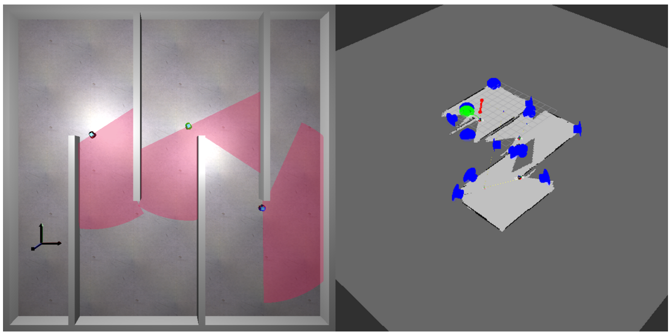


Fig. 2. Three Pioneer 3-DX robots explore an unknown environment cooperatively. In the right part of the figure: the map shown results from fusing local maps provided by three robots; the green arrow indicates the exploration goal (a frontier); the blue arrows indicates the potential exploration targets (frontiers); and the red sphere indicates the loop closure.

Due to the complexity of multi-robot exploration problem, the time and energy cost of a fleet of robots depend not only on the number of robots n, but it is also influenced by several other parameters:

- Robot characteristics. Absolute performances (e.g. exploration time) vary depending on these characteristics. More importantly, repeatability of experiments depends on the homogeneity of used robots. A fleet built out of heterogeneous robots with different capabilities, may lead to very different results from a test run to the others for various reasons such as simply the relative position of robots. This is why we prefer using a homogeneous robot team.
- Terrain properties. These include:
  - Terrain size. More robots are required to quickly explore a large area than a smaller one.
  - 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.
  - 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.
- Robot initial positions. Depending on the environment and obstacles, location of robots at start up, the exploration runtime and/or the energy consumption may be significantly impacted.
- Coordination strategies. For a given set up (terrain, robotic fleet, and initial conditions), results may significantly vary depending on the implemented coordination strategies. As a result, the optimal size of the fleet can be used as an objective value to compare different coordination solutions.

- Wireless range. Cooperation often requires communication which in turn depends on the wireless range. While the wireless range can impact a team's performance, this can be mitigated by path planning strategies that take into account robotic network connectivity [3, 7].
- Dynamicity of the environment. 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. Path planning and obstacle avoidance strategies interfere with coordination resulting in an NP-hard optimization problem.

5 Experiments

5.1 Simulation Infrastructure

For our experiments we used the 3D robotics simulator MORSE. Our simulations are run on a cluster computer that copes with the important amount of computations required for the multi-robot 3D simulation. The cluster consists of 70 computing nodes and a master node (entry point). Each computing node contains multiple processors varying from 8 to 12, and RAM varying from 16 Go to 48 Go.

This configuration gives us the possibility to launch the robots simultaneously, but each robot has an initialization phase that takes a different amount of time. It means that the robots start the exploration at different moments, as in actual multi-robot systems where robots are turned on by human operators.

5.2 Setup

As explained in the previous section, the multi-robot exploration is complex due to the number of parameters to be considered. In the following experiment, we decided to fix several parameters and focus our question on the optimal number of robots needed to explore an environment.

Regarding robots characteristics, we work with a homogeneous fleet of robots. We used simulated Pioneer 3-DX robots equipped with a SICK LMS500 laser scanner providing 180 sample points with 180 degrees field of view and a maximum range of 30 meters. The maximal speed of the robot is fixed to 1.2 meter per second and 5.24 radians per second. The odometry is considered as perfect. The robots exchange the exploration map once every 5 seconds and the maximal distance for communication is fixed to 200 meters. This distance value is to avoid the problem of communication between the robots which is not the topic of this paper.

The simulation terrain is an enclosed space, manually generated in Blender (the 3D engine for MORSE). It is 80 meters long and 80 meters wide, and contains several fixed obstacles (a maze-like space, see Figure 3). The distance between walls (or width of corridors) is fixed to 8 meters. Besides, the environment is static, meaning that the exploration robots are the only mobile entities.

For measuring the duration and the energy consumption, we run simulations until the full terrain is covered. Actually, we have considered that the exploration is finished when 99% of the map is discovered.

5.3 Robot Initial Positions

We run three series of simulation each corresponding to an experiment with specific initial positions for the robots. Figure 3 shows the initial position for the three experiments.

Experiment A: Blind exploration without any prior knowledge on the terrain. The robots are placed along a vertical line starting from the top left corner of the terrain to the bottom left corner. The first robot is placed on the top left corner, then the other robots are placed every 4 meters from the previous one. We run simulations in this experiment with fleet sizes ranging from 1 robot, and up to 14 robots.

Experiment B: Exploration with knowledge of maze entry points (1 robot/entry point). The robots are placed at the entry points of the maze terrain. One simulation is run with 2 robots, one robot on top left corner and one on bottom left corner. The second simulation is run with 3 robots, the third robot is placed on the middle left border, at a maze entry point.

Experiment C: Exploration with knowledge of maze entry points (2 robots/entry point). The robots are placed at the same entry points like in the second experiment, but we placed 2 robots at each position. It means that we run simulations with 4 and 6 robots.

5.4 Results and Interpretation

Figure 4 shows the results of our simulation experiments with different sizes of robot teams. We performed 5 runs for each team size, and display the median value of these 5 runs. The figure contains two sets of experimental data corresponding to the exploration time and the exploration cost. The abscissa in the plot denotes the team size, and the ordinate denotes the time (exploration duration) or the cost (total energy consumption).

From the figure we can see that, in general, the more robots in a team the less exploration time is needed, while the changes in the exploration cost is slightly more complex. But it does not mean the more the number of robots, the better. The best results occur here with 12 robots. The exploration time and cost are both minimized with a fleet size of 12 robots.

With the simple share of maps, exploration time and cost are highly dependent on the initial positions of the robots. To verify this hypothesis, we conducted 2 additional experiments (i.e. experiments B and C). In Figure 4, the results of experiment B are shown by red diamonds and the results of experiment C are indicated by blue triangles. Obviously, with some knowledge of the terrain, one

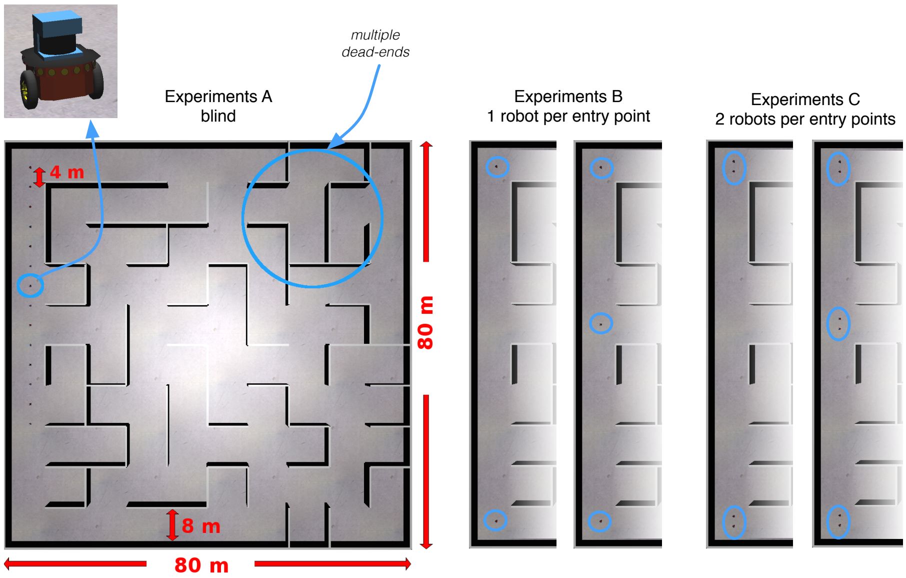


Fig. 3. Simulated environment (a maze-like space) and simulated Pioneer 3-DX robot equipped with a SICK LMS500 laser scanner in the simulator MORSE. The figures show the initial robot pose in the 3 different experiments A, B and C.

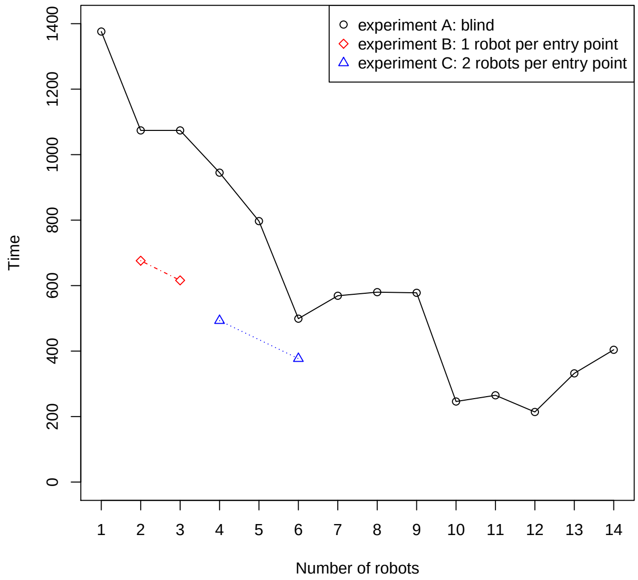
can choose better initial positions for the robots. As a result, exploration time and cost are significantly decreased with respect to the experiment A.

In our experiments there was no cooperation between robots, except exchanging maps. Thus, robots might block the path of each other during exploration, and waste time by replanning their own local paths. This results in a longer exploration time and increased exploration cost when robots are too close to each other as in experiment A.

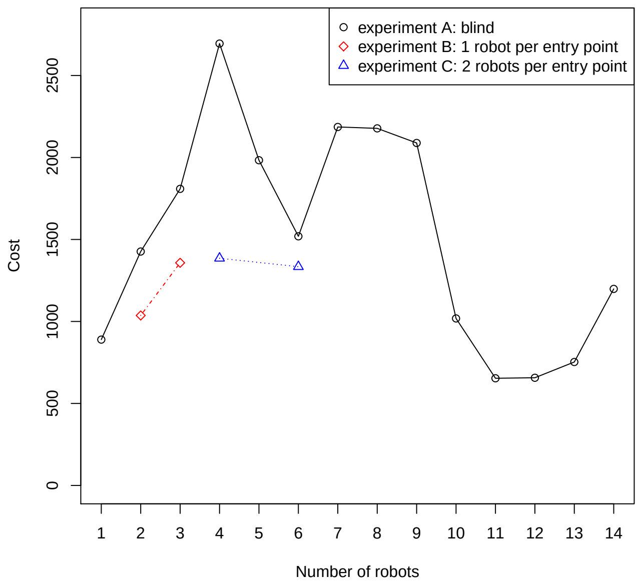
Moreover, the terrain properties are also an important factors affecting the experimental results. Our simulated environment is quite large and complex: it contains a significant number of dead ends. Typically in the top right corner of the maze (Figure 3), the robots need more time to plan their trajectories than in other areas of the terrain. We can suspect that the total time and cost are also bound to the terrain properties.

6 Discussion and Future Work

This work was started by considering the parameter of the number of robots, and subsequently the parameter of initial positions of robots was also considered. The other variables are fixed to ensure that they do not interfere in the experiment. We consider five main variables that should be discussed and integrated for future work.



(a)



(b)

Fig. 4. Exploration time and cost.

- Robot characteristics: we consider in this paper only one kind of robot and that all robots have the same characteristics. In realistic situations, the robots may be different in terms of configuration (different version of the same robot), or in terms of the kind (an exploration could be executed well with robots coupled with drones, or biped robots).
- Environment: in this experiment, we consider only one map, that is a kind of maze. We would like to run experiments with other kinds of terrains to compare results.
- Communication issues: for the experiment, we explicitly defined the communication distance to a large number (higher than the size of the map), which avoid the problem of communication between two robots. We need in future work to vary this parameter.
- The odometry precision: we do not consider the odometry noise in this paper. We will integrate it in future work. This usually needs more complex and efficient map fusion algorithms.
- Multi-robot cooperative map building algorithms: cooperation is highly desirable for multi-robot systems. The infrastructure that we have built for this study will be useful to try and compare other algorithms.

7 Conclusions

In this paper, we considered the optimization problem of the fleet size for multi-robot exploration. Our concern is, how many robots should be used for an exploration mission, so as to minimize both the exploration duration time and its cost. It is not easy to address this question due to the complexity of multi-robot systems. To provide a first answer, we conducted three series of simulations in a maze-like terrain. While they confirmed that adding more robots is usually better, they also show that the performance of the system can be significantly improved by selecting better initial positions.

To perform our experiments, we had set up an ROS-based infrastructure that runs on a cluster computer. It includes several essential nodes such as SLAM, map fusion, frontier-based exploration and motion planning. We plan to extend this infrastructure by introducing support for coordinated motion planning. Our goal is to build a test bed for evaluating different coordination algorithms in different conditions.

ACKNOWLEDGMENT

This work is part of the SUCRé project that is supported by Région Nord Pas-de-Calais.

References

1. rviz, `http://wiki.ros.org/rviz/`

2. Burgard, W., Moors, M., Fox, D., Simmons, R.G., Thrun, S.: Collaborative multi-robot exploration. In: Proc. IEEE ICRA'00. pp. 476–481. San Francisco, CA, USA (April 2000)
3. Doniec, A., Bouraqadi, N., Defoort, M., Le, V.T., Stinckwich, S.: Distributed constraint reasoning applied to multi-robot exploration. In: Proceedings of ICTAI 2009, 21st IEEE International Conference on Tools with Artificial Intelligence. pp. 159–166 (2009)
4. Echeverria, G., Lemaignan, S., Degroote, A., Lacroix, S., Karg, M., Koch, P., Lesire, C., Stinckwich, S.: Simulating complex robotic scenarios with morse. In: SIMPAR. pp. 197–208 (2012), `http://morse.openrobots.org`
5. Howard, A.: Multi-robot simultaneous localization and mapping using particle filters. The International Journal of Robotics Research 25, 1243–1256 (December 2006)
6. Lass, R.N., Sultanik, E.A., Regli, W.C.: Metrics for multiagent systems. In: Performance Evaluation and Benchmarking of Intelligent Systems, pp. 1–19. Springer (2009)
7. Le, V.T., Bouraqadi, N., Stinckwich, S., Moraru, V., Doniec, A.: Making networked robot connectivity-aware. In: Proceedings of ICRA (International Conference on Robotics and Automation). Kobe, Japan (May 2009)
8. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source robot operating system. In: Proc. IEEE ICRA'09 Workshop on Open Source Software. Kobe, Japan (May 2009)
9. Stachniss, C.: Robotic Mapping and Exploration. Springer (2009)
10. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: Proc. IEEE CIRA'97. pp. 146–151. Monterey, CA, USA (July 1997)
11. Yamauchi, B.: Frontier-based exploration using multiple robots. In: Proc. ACM Agents'98. pp. 47–53. St. Paul, MN, USA (May 1998)
12. Yan, Z., Jouandeau, N., Cherif, A.A.: A survey and analysis of multi-robot coordination. International Journal of Advanced Robotic Systems 10 (December 2013)