Multi-robot Exploration and Mapping Using Self Biddings and Stop Signals

Chapter · October 2008

DOI: 10.1007/978-3-540-88636-5_59 · Source: DBLP

3 authors, including:

Some of the authors of this publication are also working on these related projects:

Project   Exact solution of the cyclic bandwidth problem View project

Project   Metaheuristic Algorithms for Solving Graph Labeling/Embedding Problems View project

Multi-robot Exploration and Mapping Using Self Biddings and Stop Signals⋆

Juan C. Elizondo-Leal, Gabriel Ramírez-Torres,
and Gregorio Toscano Pulido

Centro de Investigación y Estudios Avanzados del IPN
Laboratorio de Tecnologías de Información
Carretera a Monterrey KM 6, Ciudad Victoria, Tamaulipas México
{jelizondo, grtorres, gtoscano}@tamps.cinvestav.mx

Abstract. Exploration and mapping of the environment is a critical issue in mobile robotics. In this paper we present an approach to multi-robot exploration and mapping, which is based in a distributed bidding process, where the bids are calculated by each robot. When a robot reaches the target it makes a decision by itself, which involves every one of the team members without lost of time in a totally distributed behavior and without the necessity of a central module. Furthermore, if the selected goal is discovered for another robot, this robot send a stop signal and the process of bidding is restarted.

The bid is computed such each robot will be assigned to a near unexplored frontier, while keeping the robots far from each other. The bid includes the distance of robot towards one of the frontiers, and the distances to the others robots toward their objectives plus the distance of these objectives toward the frontier. The numerical results demonstrate that our approach is efficient and fault tolerant.

1 Introduction

Exploration and mapping of the environment is a critical issue in mobile robotics. Since autonomous robots must be able to navigate effectively in environments, must possess the ability to build representations of those environments. In general, this requires a good exploration strategy. There exist several applications in which the exploration task is an integral part of the robotic mission, such as search and rescue, hazardous material handling, planetary exploration, etc. There are many advantages in mapping and exploration with multiple robots compared to a single robot, like fault tolerant, concurrency, which can greatly reduce the time needed for the exploration and merging of overlapping maps so the error can be diminished.

The problem of multi-robot exploration can be stated as: *n robots set out in unknown area equipped with sensing, localization and communication capabilities; design an efficient coordinated exploration algorithm to make a map of the unknown area.*

Recently, bidding based approaches have arisen, which have demonstrated to reduce the time and traveling distance in previous works on multi-robot exploration [1][2][3][4]. These approaches treat all robots as economic agents, which use a bidding mechanism in order to locally minimize the time and traveling distance. However, most of the proposals that are based on biddings require a central module that sometimes do not include all the team members.

In our approach, the bidding mechanism involves every one of the team members without lost of time in a totally distributed behavior and without the necessity of a central module. When a robot reaches the target it makes a decision by itself. Furthermore, if the selected goal is discovered by another robot, this robot send a stop signal and the process of bidding is restarted. The bid is computed such each robot will be assigned to a near unexplored frontier, while keeping the robots far from each other. The bid includes the distance of robot towards one of the frontiers, and the distances to the others robots toward their objectives plus the distance of these objectives toward the frontier. We uses occupancy grid maps to represent the environment. Each cell of such an occupancy grid map has one of three status: occupied, free or unknown. The numerical results demonstrate that our approach is efficient and fault tolerant.

The remainder of the paper is arranged as follows: Section 2 discusses previous work in the area of multi-robot exploration. Section 3 describe our approach to multi-robot exploration and section 4 presents numerical results that show the performance of our method. In section 5, we present our conclusions and discuss future research.

2   Related Work

There has been a wide variety of approaches to robotic exploration. Many approaches have been proposed for exploring unknown environments with single robots [5][6][7], much work remains to be done in coordinated multi-robot mapping and exploration. Yamauchi [8] in 1997 developed an approach for exploration with a single robot based on the concept of frontiers, regions on the boundary between open space and unexplored space. The basic idea is to keep the robot moving to the closest frontier cell. In [9] Yamauchi applies the concept of frontiers to a distributed, asynchronous multirobot exploration algorithm, where robots exchange perceptual information continuously. This approach brings the fault tolerance capability, however, there is no explicit coordination so that multiple robots may end up covering the same area and may even physically interfere with each other. Simmons et al. [1] in 2000 developed a multi-robot approach which uses a concept of frontier and a simple bidding protocol. The team requires a central agent to evaluate bidding from all the other robots to obtain the most information gain while reducing the cost or the travelling distance. Zlot et al. [2] in 2002 presented a multi-robot exploration approach which exploits a market architecture. The team of robots continuously negotiate with each other, improving their current plans and sharing information about which regions have and have not already been covered. Their algorithm can improve the reliability, robustness and efficiency of the exploration. However, the goal generation

algorithm they used is not as efficient as the frontier based and the negotiation process is complex. Baglietto et al. [10] in 2002 apply the concepts of entropy and frontier, where entropy quantifies the information gain associated with the frontier cell with the greatest information gains. Burgar et al. [3] in 2005 developed a coordinated multi-robot exploration algorithm which simultaneously considers the utility of unexplored areas and the cost for reaching these areas, using evidence grids maps to represent the environment and the concept of frontiers. They investigate the impact of the limited-range communication. They use a central module dynamically selected which receives the local maps and combine them into a single global map which then is broadcasted to all robots. But using a central module suffers of a single point fault. Sheng et al. [4] in 2006 presented a coordinated multi-robot exploration algorithm based on a distributed bidding model. They address the problem caused by the limited-range communication. The authors introduce a nearness measure, and therefore the robots tend to stay close to each other. This approach does not guarantee that all the members of the team participate in the bidding. Stachniss et al. [11] in 2006 developed a multi-robot exploration strategy which considers the semantic place information, they apply a classifier learned with AdaBoost. This approach required learning and consider similar places. Poernomo and Shell Ying [12] in 2006 proposed the separation of robots in the frontier selection, which reduce the chances that the robots interfere with each other. However, they assumed that the size of robot is small enough, such that the collision is negligible.

In general, the proposals that are based on biddings require a central module that sometimes do not include all the team members. So we propose an algorithm where the team members are all included without the necessity of selecting a central module.

3 Approach

Our approach uses a bidding mechanism based on target assignment to coordinate the actions of the robots, which is totally distributed since each robot calculates the bid of the others and its own bid, and make its own decision, furthermore, if the selected goal is discovered by another robot, this robot send a stop signal and the process of bidding is restarted. Thus adding efficiency and robustness, since it eliminates times of delay, single points of fault and reduces unnecessary travels.

During the exploration the sensing and mapping is made continuously by each robot and the bidding is carry on when a frontier is reached. The frontier is considered reached when the robot is at a short distance and the frontier is detected explored.

3.1 Bid Compute

The bid is computed such each robot will be assigned to a near unexplored frontier, while keeping the robots far from each other. The bid of all robots

(bid_k) includes the distance of robot towards its goal ρ_k plus the distance of this goal toward the frontier α_k, and the distances to the others robots toward their objectives (ρ_i) plus the distance of these objectives toward the frontier (α_i).

$$bid_k = \rho_k + \alpha_k - \sum_{i=0, i\neq k}^{n} (\rho_i + \alpha_i); \text{ for } k = 0, \ldots, n-1 \qquad (1)$$

where i and k represent the index of each one of the robots.

With the inclusion of the team data in the bid we managed to separate them from each other.

The current robot computes its bid and all the team's bids to reach each one of the frontiers. The complexity of the calculation of all the bids is $O((n^2 - n)F)$, where n is the number of robots and F is the number of frontiers. Once all the bids have been computed the frontiers are sorted (according to its bid value associated to the present robot) for the process of bidding.

3.2   The Totally Distributed Bidding

This computation is used as follows: When a robot has reached its goal, it requests maps and targets of other robots and it waits their answers. Then the robot merges the maps to its local map and compute the bids to the nearest

---

Algorithm 1: Robot exploration

```
start_MappingOnline() My_Target = My_Pose Send_MyTarget while the
Map is not Complete
   broadcast(Request MapTarget)
   update_MyMap
   update_OtherRobotTarget
   detectFrontier()
   bid_compute()
   for i=0 to number of frontiers
      if I win this frontier
         My_Target=thisFrontier
         break
   if does not win any frontier
      wait other new pose
      continue
   send my target
   while not reach my target or not receive stop signal
      travel to My_Target
      if i view another target
         send stop signal
broadcast(MyMap)
```

Fig. 1. Algorithm for robot exploration

Algorithm 2: Robot communication

```
while the Map is not Complete
   idRobot=receiveData()
   if request map and target
      sendTarget(idRobot, My_Target)
      sendMap(idRobot, My_map)
   else if Receive target
      saveTarget(idRobot, Target)
      if(Target is equal My_Target)
         stopTravel
   else if Receive stop signal
      stopTravel
```

Fig. 2. Algorithm for robot communication

frontier. If the robot provide the best bid, then this robot wins the bidding and broadcast this frontier as its new goal. If this robot does not win, the process is repeated with the next nearness frontier until it wins, or no frontier is left. If the robot does not win any case then the robot stops until new data is received. Furthermore if the selected frontier is discovered by any robot, this robot send a stop signal and the process of bidding is restarted.

The pseudocode of our approach is shown in Figure 1 and run in parallel with the communication protocol in Figure 2. The complexity of this algorithm including calculation of bids is $O((n^2 - n)F^2)$, in the worst case where n is the number of robots and F is the number of frontiers.

Algorithm 3: Clustering frontier

```
PROCEDURE RecursiveDetection(x, y)
   for i=x-1 to x+1
      for j=y-1 to y+1
         if cell[i][j] is frontier and not considered before
            RecursiveDetection(i, j)
MAIN
   for i=0 to width of map
      for j=0 to height of map
         if cell[i][j] is frontier and not considered before
            RecursiveDetection(i, j)
            if size of cluster > threshold
               search geometrical center
               save frontier
```

Fig. 3. Algorithm for clustering frontier

3.3   Maps Updating

When a robot receives a map, it makes an average with its own map. The map computed by a robot can contain incorrect data, because of the own errors of the sensors. We exploit the advantage that multiple robots may produce more accurate maps by merging the overlapping information. When a robot receives a map, each point is overlapped with the local map, but if the local point is marked as explored, the point is averaged with the new point, so the errors can be diminished.

Once the map was updated, the detection of frontiers is carried out, which closer frontiers are considered as a single one located in the geometrical center of that cluster. The algorithm of clustering is showing in Figure 3.

4   Numerical Results

In this section we evaluate the performance of our approach. The simulator used in our approach is the player/stage simulator[1] [13] which models the interferences betwen robots. The environment is represented with a grid-based map. To carry out our experiments we simulated laser sensors with 360 degree of view in order to determine the occupied area and a ring of ultrasound sensors which determines the free area, the sensors radius range is 4 meters, mounted on the robot Pioneer 2-DX.

An example environment is shown in the figure 9. A number of robots are simulated by identical programs with each representing one robot. We performed four experiments.

First, we show the fault tolerance of this approach, using the map shown in figure 4. We identify two cases: the robot fails while traveling to its goal, and the robot fails during its bidding process. In the figure 5 the robot 2 has a failure during its travel, when the others robots begin the bid process they can not receive any new information from the robot 2 and they do not include it in its bidding process and can complete the task. In the figure 6 the robot 4 has some failure during its bid process, but this do not affect the other robots since each robot makes a decision by itself.

The results of the previous cases prove that the multi-robot exploration using self biddings and stop signals is tolerant to faults. When some robot has a failure, the other robots can continue to explore the unknown environment.

Second, we show the behaviour when a robot discovers another target. In the figure 7 the robot 4 discovers a goal assigned to robot 1, this last one is informed so it stops and bids for a new goal. The result of the second experiment prove that when stop signals are added the redundant work is reduced.

Third, we show the behaviour when a robot is added a long time after the beginning of exploration. In the figure 8 the robot 3 is added after 3.5 minutes, so when the other robots begin the bid process receives new information from

---

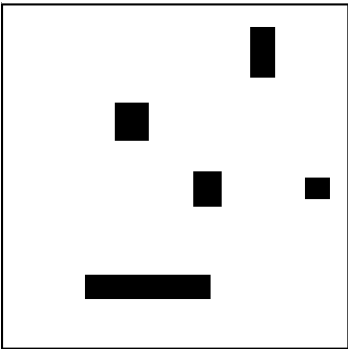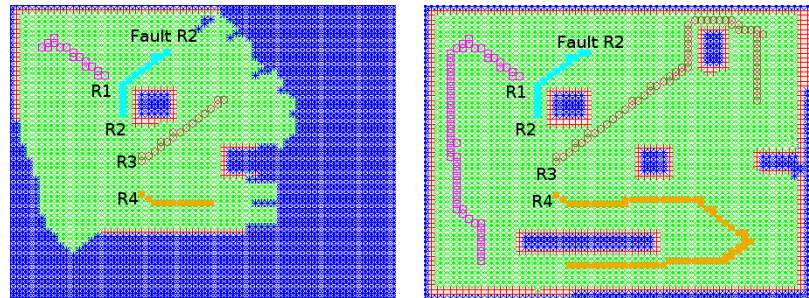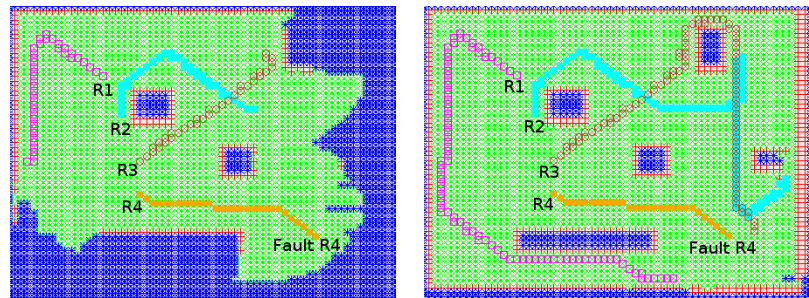[1] Player is freely available under the GNU General Public License from
http://playerstage.sourceforge.net.

Fig. 4. Map used for the demonstration of fault tolerant (15×15 meters)

(a) The robot 2 has some failure during
its travel

(b) Complete exploration

Fig. 5. Fault during travel

(a) The robot 4 has some failure during
its bid process

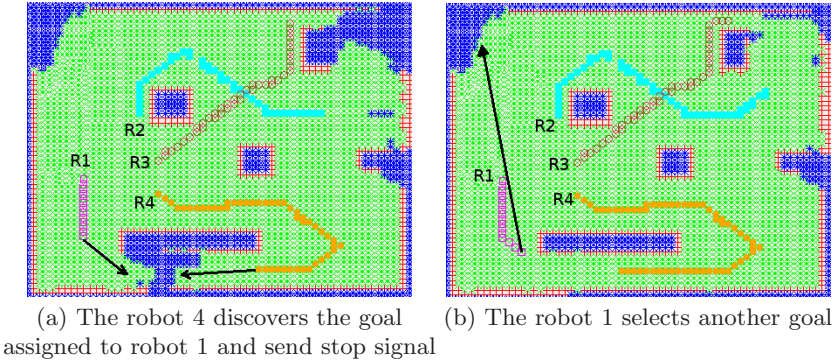(b) Complete exploration

Fig. 6. Fault during bid process

(a) The robot 4 discovers the goal assigned to robot 1 and send stop signal

(b) The robot 1 selects another goal

Fig. 7. Discovery of the goal of another robot

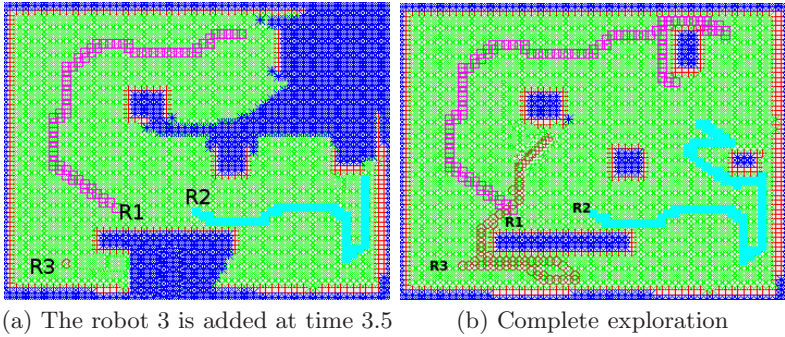(a) The robot 3 is added at time 3.5

(b) Complete exploration

Fig. 8. The robot 3 is added a time after the beginning of exploration
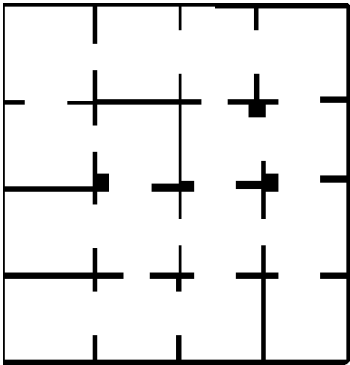
Fig. 9. Map used for the simulation (25×25 meters)

(a) Exploration time   (b) Total distances
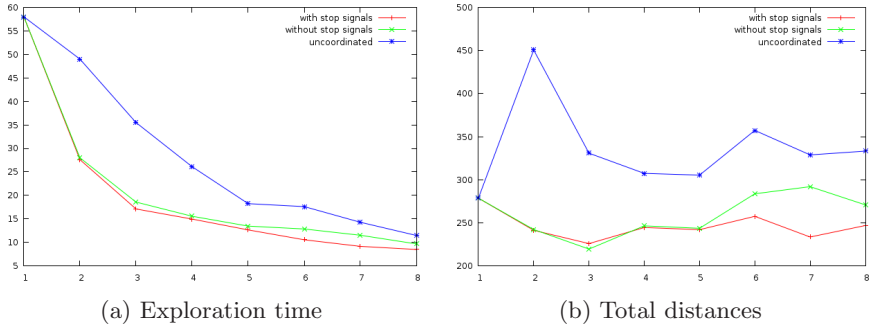
Fig. 10. Performances of the different coordination strategies for the environments shown in Figure 9

(a)1.6 minutes   (b) 7.2 minutes

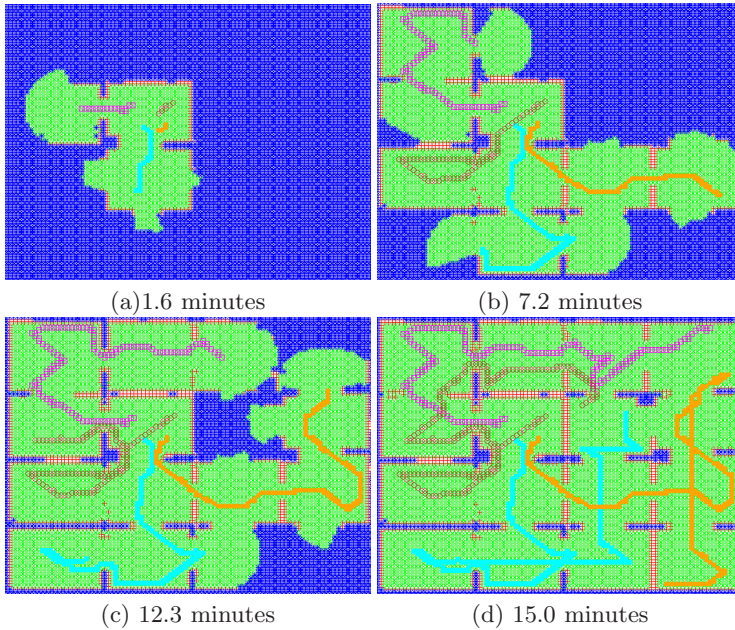(c) 12.3 minutes   (d) 15.0 minutes

Fig. 11. Simulated exploration with four robots

the robot 3 and they include it in its bidding process and may complete the task in a smaller time.

Fourth, we compared our approach with the technique where the process of bidding is eliminated and each robots goes to the nearest frontier, like used by Yamauchi. From now on, this approach will be denoted as uncoordinated exploration since it lacks a component that arbitrates between the robots whenever they choose the same frontier cells. Moreover we also compare it with our own

approach without stop signals. For each case and a fixed number of robots, ten test runs are conducted and the total exploration times are recorded. Figure 10 compares the total exploration time needed versus the number of robots and the total distances needed versus the number of robots. For each comparison of the strategies, the robot team was started at the same randomly chosen location. The map used in our experiments are shown in the figure 9. Screen shots of a simulation with four robots exploring the environment are shown in Figure 11 the robots started in the center of map.

The results of the fourth experiment demonstrate a great efficiency in the selection of the frontier to explore by every robot.

5 Future Work and Conclusions

This work presents a novel completely distributed multi-robot coordination algorithm to explore and map terrain, based on a market bidding process. The key idea of this approach is take into account the cost of reaching a unexplored frontier and the distances to the others robots and theirs objectives. The result is an efficient and fault tolerant approach. From the theoretical analysis and simulation experiments, it is shown that the method is feasible and can effectively distribute multiple frontiers to multiple robots in a totally uncentralized form.
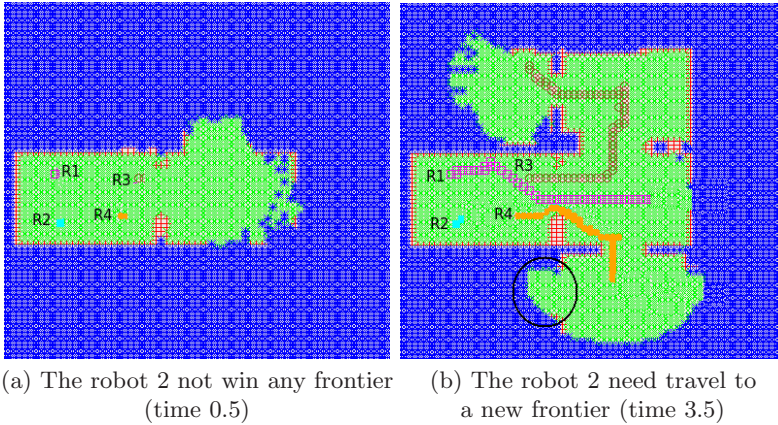

(a) The robot 2 not win any frontier (time 0.5)   (b) The robot 2 need travel to a new frontier (time 3.5)

Fig. 12. The robot 2 stops some time while new frontier has discovered

Since sometimes the number of frontiers is less than the number of robots, some robots can remain stopped as shown in figure 12. This may cause over time in some robots. We believe that the process of exploration can be improved by adding an opportunistic distribution, so these robots follow at a safe distance other robots, such if the leader robot discover a new frontier, the follower will be ready to take it. At present, we are working in its development of such strategy.

References

1. Simmons, R.G., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H.: Coordination for multi-robot exploration and mapping. In: AAAI/IAAI, pp. 852–858 (2000)
2. Zlot, R., Stentz, A., Dias, M., Thayer, S.: Multi-robot exploration controlled by a market economy. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 3016–3023 (2002)
3. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. IEEE Transactions on Robotics and Automation 21, 376–386 (2005)
4. Sheng, W., Yang, Q., Tan, J., Xi, N.: Distributed multi-robot coordination in area exploration. Robot. Auton. Syst. 54, 945–955 (2006)
5. Dudek, G., Jenkin, M., Milios, E., Wilkes, D.: Robotic exploration as graph construction. IEEE Transactions on Robotics and Automation, 859–865 (1991)
6. Thrun, S.: Learning metric-topological maps for indoor mobile robot navigation. Artificial Intelligence 99, 21–71 (1998)
7. Albers, S., Kursawe, K., Schuierer, S.: Exploring unknown environments with obstacles. In: SODA 1999: Proceedings of the tenth annual ACM-SIAM symposium on Discrete algorithms, Philadelphia, PA, USA, pp. 842–843. Society for Industrial and Applied Mathematics (1999)
8. Yamauchi, B.: A frontier-based approach for autonomous exploration. In: CIRA 1997: Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation, Washington, DC, USA, p. 146. IEEE Computer Society, Los Alamitos (1997)
9. Yamauchi, B.: Decentralized coordination for multirobot exploration. Robotics and Autonomous Systems 29, 111–118 (1999)
10. Baglietto, M., Paolucci, M.S., Zoppoli, R.L.: Information-based multi-agent exploration. In: Proceedings of the Third International Workshop on Robot Motion and Control, RoMoCo 2002, November 9-11, 2002, pp. 173–179 (2002)
11. Stachniss, C., Martínez-Mozos, O., Burgard, W.: Speeding-up multi-robot exploration by considering semantic place information. In: Proceedings 2006 IEEE International Conference on Robotics and Automation, ICRA 2006, pp. 1692–1697 (2006)
12. Poernomo, A., Ying, H.S.: New cost function for multi-robot exploration. In: 9th International Conference on Control, Automation, Robotics and Vision, ICARCV 2006, December 5-8, 2006, pp. 1–6 (2006)
13. Gerkey, B.P., Vaughan, R.T., Howard, A.: The player/stage project: Tools for multi-robot and distributed sensor systems. In: Proc. of the Intl. Conf. on Advanced Robotics (ICAR), July 2003, pp. 317–323 (2003)