Beyond Frontier Exploration

Arnoud Visser, Xingrui-Ji, Merlijn van Ittersum,
Luis A. González Jaime, and Laurenţiu A. Stancu

Intelligent Systems Laboratory Amsterdam,
Universiteit van Amsterdam, The Netherlands
http://www.science.uva.nl/research/isla

Abstract. This article investigates the prerequisites for a global exploration strategy in an unknown environment on a virtual disaster site. Assume that a robot equipped with a laser range scanner can build a detailed map of a previous unknown environment. The remaining question is how to use this information on this map for further exploration.

On a map several interesting locations can be present where the exploration can be continued, referred as exploration frontiers. Typically, a greedy algorithm is used for the decision which frontier to explore next. Such a greedy algorithm only considers interesting locations locally, focused to reduce the movement costs. More sophisticated algorithms also take into account the information that can be gained along each frontier. This shifts the problem to estimate the amount of unexplored area behind the frontiers on the global map. Our algorithm exploits the long range of current laser scanners. Typically, during the previous exploration a small number of laser rays already passed the frontier, but this number is too low to have major impact on the generated map. Yet, the few rays through a frontier can be used to estimate the potential information gain from unexplored area beyond the frontier.

1 Introduction

RoboCup Rescue is a competition in which (teams of) fully autonomous robots visit a hypothetical disaster site. This situation is either simulated in the real world [1] or a virtual world within the USARSim simulator [2]. The task for the robots in the competition is to explore the site and locate victims. There is a limited amount of time in which the robots can explore. Afterwards the competing teams will be scored on a various criteria, among them are the size of the explored area, the quality of the map and most importantly, the number of located victims (for a more detailed list and scoring see [3]).

An important problem in the competition is the *autonomous exploration* problem; to decide on the basis of the current map where to send the robot to improve the future map [4]. A correct choice would improve the competition score, which depends on the explored area and the quality of the map. Predictions about what would be visible on the edges of the current map could help to make better decisions for the robot.

In this work we build upon the contribution of the UvA Rescue team [5], which provided a fully autonomous agent system that controls up to eight virtual robots in the USARSim simulator [2]. The system already has a state of the art method [6] for simultaneously locating and mapping (SLAM) unknown environments, based on the Manifold approach [7] combined with the Weighted Scan Matching algorithm [8]. The exploration strategy of the robots is kept simple. The behavior is reactive and makes decision based on direct measurements, not on the current map. The goal of this work is to improve the exploration strategy by intelligently using global information that can be derived from the map. As demonstrated in [9], efficient allocation of the search effort can outperform simple exploration strategies. In the article the focus is on the exploration behavior of a single robot (or agent).

The outline of this paper is as follows. In Sect. 2 we will introduce some theoretical background behind this research. In Sect. 3 our algorithm will be worked out. The robustness of our method on maps from the RoboCup Rescue Competition will be demonstrated in Sect. 4. Finally, we draw our conclusions in Sect. 5.

2   Background

Exploration is the problem of directing a robot through the environment so that the knowledge about the external world is maximized [10]. Knowledge about the external world for a mobile robot is typically stored on a map m. Increasing the knowledge stored on a map can mean that the uncertainty about information on the map is reduced, or that new information is added to the map. The latter means that the map coverage is extended with parts of the external world that the robot has not seen before. Knowledge about the map m can be passively acquired, while the robot is wandering around busy with other tasks (for instance finding victims), as demonstrated by [11]. Here the focus is on autonomous exploration; the planning of the next exploration action a which will increase the knowledge about the world the most. Before this estimate is worked out in more detail, it should be noticed that such an exploration action can be quite complex from navigational point of view. Executing such an exploration action can mean that large parts of the current map are traversed, which can only be efficiently done with the availability of on-line path-planning functionality.

For a mobile robot it is important to remember were obstacles are located. This information can be represented with an occupancy grid map [12], where each grid cell indicates the probability $p(x)$ if that location x is occupied or free. Active exploration can been seen as minimizing the *information entropy $H(m)$* [13] of the probability distribution $p(x)$ for all x on the map m, which requires an integration over the complete occupancy grid map:

$$H(m) = -\int_{x \in m} p(x)log(p(x)) \tag{1}$$

When all grid cells are initialized as unknown by giving them a uniform value of $p(x) = 0.5$, the entropy of the map $H(m)$ is maximal. When the boundaries of

the map m are not known, the limits of the integral are slowly extended when new areas are discovered. The interest is not in the absolute value of the entropy, but in the difference in entropy before $H(m)$ and after $H(m|a)$ an exploration action a; the *information gain* $\Delta I(a)$ [14,15,16]. Remember that the exploration action a could be a complex maneuver, consisting of a number of controls u_i and observations z_i for multiple timesteps i.

$$\Delta I(a) = H(m|a) - H(m) \tag{2}$$

Because the set of possible exploration actions can grow very fast when predictions are needed multiple timesteps in the future, this set is approximated. In existing exploration methods the number of exploration actions is reduced by considering only the path to a finite number of candidate observation points. Typically, those candidate observation points are chosen on the boundary of explored and unexplored areas; frontier-based exploration [17].

One of the most interesting approaches to generate and select those candidate observation points is the presented by González-Baños [18]. They model exploration frontier with free curves; polylines which indicate where the laser range scanner reported values larger than a threshold $r_{\max}$. Near those free curves a number of candidate observation points are considered . This number of candidate points is generated randomly with a Monte-Carlo method, and for each point q they simulate a number of laser scans through the free curve. The amount of area $A(q)$ covered by those rays (with a maximum length $r_{\max}$) is taken in account as a measure of the potential information gain $\Delta I(a_q)$ for the observation z_q at the observation point q.
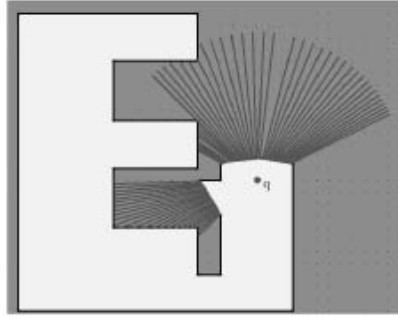

Fig. 1. The potential information gain of a candidate observation point q is the area $A(q)$ that may be visible through the two free edges; this area is estimated by casting rays from q. Courtesy from [18].

The area $A(q)$ is an estimation for the information gained from the observation z_q. This implicitly ignores the information that could be gained by the observations $z_1, \ldots, z_{q-1}$ along the path to the observation point q. When the robot traverses mainly well known regions on its path to point q this is a reasonable assumption. Yet, the exploration action a_q consists not only of a number

of observations $z_1, \ldots, z_q$, but also of a number of controls $u_1, \ldots, u_q$ to drive the path. Because control is never perfect, confidence about the location of the robot is lost for every control step u_i. Probabilities spread out over the map, resulting in a loss of information. An optimal exploration action a^* can only be chosen when the cost of traveling along a path $u_1, \ldots, u_q$ is taken into account. Typical traveling cost functions are the distance traveled, the time taken or the energy expended. González-Baños has chosen to use as cost-function the length $L(q)$ of the path $u_1, \ldots, u_q$ to a candidate observation point q. They combine the cost of the path $u_1, \ldots, u_q$ and the estimated gain of the observation z_q for the evaluation action a_q into the following value function $V(q)$:

$$V(q) = A(q)e^{-\lambda L(q)} \tag{3}$$

The constant λ can be used balance the cost of motion $L(q)$ against the expected gain of information $A(q)$.

For the observation points found in this article an equivalent value function could be calculated. Note, however, that the area $A(q)$ of González-Baños is extrapolated from the current map by simulating a number of laser-rays through the frontier, while in our case the area $A(q)$ is directly estimated from the laser range measurements. Another difference is the generation of observation points. González-Baños generates multiple candidate observation points on a short random distance from the exploration frontiers. In our approach, per frontier a single candidate observation point is generated, in the center of the exploration frontier.

3   Estimation of Exploration Frontiers and Observation Points

A good autonomous exploration algorithm should navigate the robot to an optimal observation point. This point will be close to an exploration frontier. To find such an exploration frontier is not trivial. Exploration frontiers can be found based on an occupancy grid map. The probability that a point is an obstacle or not on a certain location can be stored in occupancy grid with arbitrary resolution. An example of such occupancy grid map is given in Fig. 2, an actual map produced during the Virtual RoboCup Rescue competition by the UvA Rescue team with a resolution of 1 centimeter. The map gives a top view of an office environment, where clearly three corridors are visible that are well explored, and a number of adjacent rooms that are not entered yet.

Grid points on a map can be combined to regions, when the edges of the regions can be found. As can be seen from the example (Fig. 2), the boundaries of the safe region are only sharp along walls. Inside the rooms and at the end of the corridors the boundaries of the safe region are fuzzy. Selecting an absolute threshold for this boundary is difficult [19]. Still, a human can clearly distinguished safe regions and indicate the regions that should be further explored (observation regions). What is difficult, also for a human, is the precise location of the boundary between those regions; the exploration frontier.
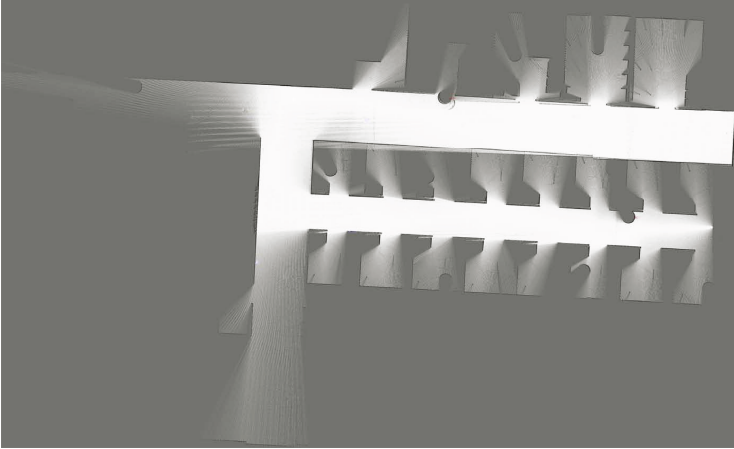
Fig. 2. The occupancy grid map produced during the semi-final by one the robots

The method used in this paper to distinguish safe regions from observation regions is based on a simple ray-casting technique. Ray-casting is used to generate an occupancy grid from the scan-data stored in the manifold [11]. The trick is to generate two occupancy grids at the same time; one with a short range constraint r_{safe} and one with a long range constraint r_{max} equal to maximum range of the laser scanner. A typical value for r_{safe} is 2 meters and for r_{max} 20 meters. The occupancy grid with a short range constraint r_{safe} generates a conservative estimate of the obstacle free space; the safe region.

An example of such safe region is given in Fig. 3.a. The three corridors of the office environment can be recognized in this picture. From the contour of the safe region the exploration frontier can be derived. The contour of the safe region is indicated in Fig. 3.b. The exploration frontier is only a part of the contour, the other part of the contour are walls (Fig. 3.c). The part of the safe region contour that is no wall can be identified as the exploration frontier (Fig. 3.d).

The same ray-tracing can be repeated with the long range constraint r_{max}. With the long range constraint a less conservative estimate of free space is generated. The areas are probably free of obstacles, but not guaranteed to be safe. The result is visualized in Fig. 4. Outside the corridors new contours are visible: the rooms along the corridor. These contours are the areas which are probably free, but not guaranteed to be safe: the observation regions (indicated in yellow). These observation regions are not equivalent with unknown areas; there also exist large parts of the map where the probability $p(x)$ is still on its initial value. The frontiers between the observation regions and the safe regions, as shown in Fig. 3.d, are also given (indicated in grey). The convex frontiers contours are extended with a small white point which is the center of the contour. These centers can be associated with potential observation points. From each point in the safe region the path to such a potential observation point q can be estimated with a breath-first algorithm. An example of such path is indicated in green, from the start position indicated with a large white point in the upper-right corner.
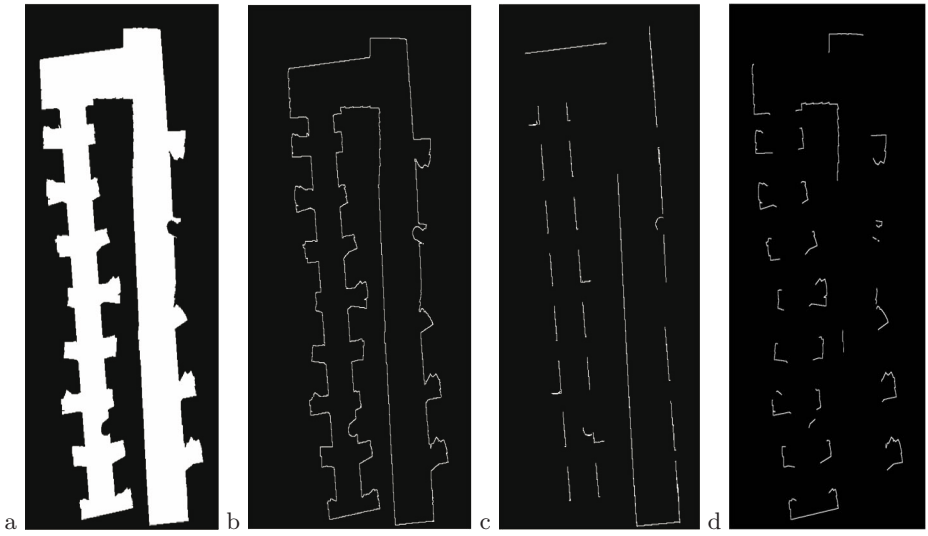
Fig. 3. The safe region of Fig. 2 (90° degrees rotated). From left to right respectively a) The surface. b) The contour. c) The part of the contour which is an obstacle (wall). d) The part of the contour which is free (frontier).
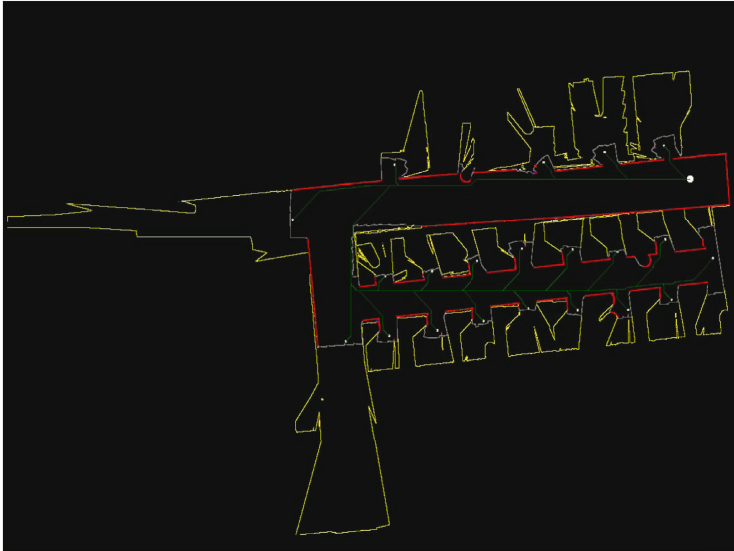

Fig. 4. The interpreted map of Fig. 2. The yellow contours indicate the observation regions. The grey contours indicate the exploration frontiers. The large white circle indicates the current position of the robot. The small white circles indicates potential observation points. The green line indicates a possible path to those observation points. Red lines indicate the walls.

In this example one can also see that the process of frontier estimation is not completely failsafe. Going from one corridor to another, the robot makes a sharp turn (top of Fig. 3). During such turn the confidence in the location estimate can drop. At that moment multiple laserscans of the same wall do not completely overlap, and the wall as edge is not sharp. In that case a part of the wall is seen as unknown, and such a contour could be incorrectly identified as frontier. The effect is visible near the upper right corner in Fig. 3.d. To prevent false positives like this, the following consistence check is designed. All frontiers are tested if they are concave or convex. Only convex frontiers generate candidate observation points q in Fig. 4. Many of the false frontiers can be removed because they are concave. The result is a slight increase of the number of false negatives, as indicated in the Sect. 4.

4   Results

In the previous sections we have illustrated our methods on the map given in Fig. 2. In this map several potential observation points were identified, as shown in Fig. 4. To test the reliability of our method the number of observation points is compared against the number of points that should have been found. After the 2006 competition the environment used during the RoboCup was made available for inspection[1]. A top view of this environment is visible in Fig. 5.a. The three corridors explored on the map can be found in the upper-right corner of Fig. 5.a.

With the provided environment as reference, the number of doorways and corridors that the robot has passed during its exploration can be counted. For the map given in Fig. 2 in total 2 corridors and 20 doorways to 20 rooms are passed. There is another doorway to a 21st room, but this doorway is blocked by a victim. The algorithm skipped this doorway correctly. The results are summarized in Table 1. Next to the expected and found number of doorways, the number of false positives and false negatives are given. This is done for both the exploration frontiers (both convex and concave) and the observation points (center of convex exploration frontier). One can see that the number of false positives is reduced by only selecting convex exploration frontiers.

To demonstrate the robustness of the algorithm, the procedure is repeated for two other maps that could have been encountered during the competition. The first map is tour that begins and ends in the lobby, the light-grey area at the bottom of Fig. 5.a. During this tour 6 corridors and 7 doorways to rooms should have been found. For the majority of the corridors and rooms an observation point is found, as can be seen from Table 1 and Fig. 5. The missed corridor and room are located in the left lower corner. The robot came through the narrow passage at the left and turned back towards the lobby. Due to this turn the robot did not get a clear view into the corner. The doorway to the room is visible; the $6^{t}h$ corridor stays mainly hidden behind the robot. The combined frontier of the corner, doorway and corridor was irregular of shape and not convex, which resulted in a false negative.

---

[1] http://sourceforge.net/projects/usarsim

Table 1. Experimental results

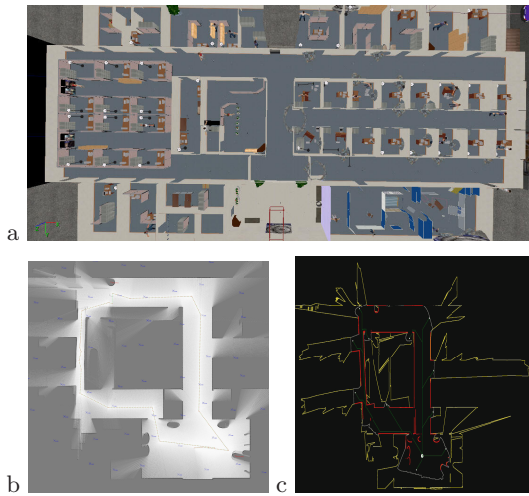|	*expected frontiers & observation points*	**exploration frontiers**			**observation points**		
		found	*false positives*	*false negatives*	**found**	*false positives*	*false negatives*
Three corridors (Fig. 2)	22	27	6	1	21	0	1
Lobby loop (Fig. 5)	13	17	5	1	11	0	2
Yellow arena (Fig. 6)	9	17	8	0	9	1	1

Fig. 5. Overview of the indoor area used for the 2006 RoboCup Virtual League Competition. Fig. b and c show the results of exploring a loop starting and ending in the large lobby at the bottom of Fig. a. On this map 11 of the 13 observation points are found.

Last, but not least, the algorithm was tested on the Yellow arena. This is classical benchmark in the RoboCup Rescue competition, where an irregular office-maze is build with flexible walls. The Yellow arena is also visible in Fig. 5.a, the large room to the right of the lobby. Fig. 6.a gives a closer look at this environment. In this office-maze it is less obvious to indicate what the frontiers are that should have been explored. For instance, central in the Yellow arena is a bed. The algorithm indicated with the three green paths that an observation should be made at the left, at the right and under the bed. This was classified as a correct decision. Another aspect is the open space in the rooms. The rooms were sometimes so large that frontiers appeared in the corners. These frontiers in the corner have a convex shape, and could be selected if the observation space behind the frontier was large enough. Checking the corners of a room is probably quite
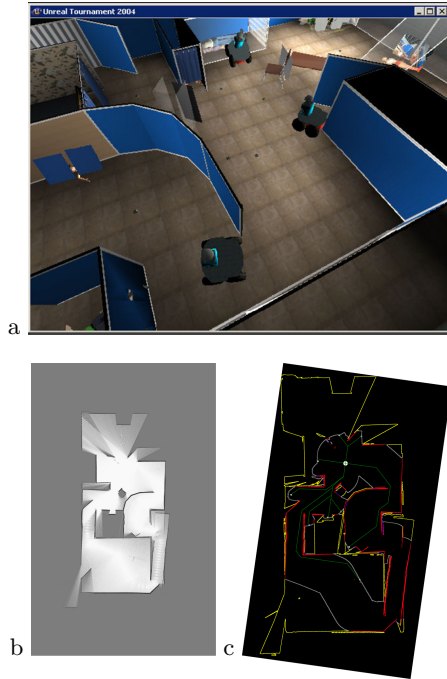
Fig. 6. Results of the Yellow-Arena map. Fig. b and c show the results of exploring a loop starting and ending in the curved wall in the center of the figures. On this map 9 observation points are found (one false positive).

robust, but probably not highly efficient. Both the false positive and negative were related with a corner. At the bottom left of Fig. 6.c a doorway is missed, because that corner was not explored from close enough distance. On the other hand an observation point is generated to check the tiny space behind the 'W'-shaped obstacle. Fortunately many other observation points are generated which much more area behind the frontier, which make them far more attractive for exploration. This observation point was classified as a false positive.

Overall, these experiments, summarized in Table 1, demonstrate the robustness of the algorithm. The algorithm generated a limited number of potential observation points. The impact of the false negatives (4 of the 44 potential observation points were missed) on the exploration behavior will be minor. As long as there are enough candidate observation points, the robots can coordinate their actions and distribute the points over the team. They can optimize their effort by optimizing a joint value function equivalent with equation (3).

5   Conclusion

In this report an algorithm is proposed to generate a limited number of observation points, and to estimate the information that could be gained at each

location. The method generates exploration frontiers on the contours of safe regions. One observation point is generated per convex exploration frontier. The potential information gain for each observation point is estimated based on the area of obstacle free space beyond the exploration frontier. This estimate of the area is based on measurements, and not on extrapolations from the current map. The algorithm shows good results in office-environments.

In our future research it will be demonstrated how much the exploration efficiency will increase by selecting the observation point with highest potential information gain and the lowest travel costs.

Acknowledgments

References

1. Jacoff, A., Messina, E., Weiss, B., Tadokoro, S., Nakagawa, Y.: Test arenas and performance metrics for urban search and rescue robots. In: Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (2003)
2. Balakirsky, S., Scrapper, C., Carpin, S., Lewis, M.: Usarsim: providing a framework for multi-robot performance evaluation. In: Proceedings of PerMIS 2006 (2006)
3. Balakirsky, S., Carpin, S., Kleiner, A., Lewis, M., Visser, A., Wang, J., Ziparo, V.A.: Towards heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue. Journal of Field Robotics (to appear, 2007)
4. Hasegawa, B.R.: Continues observation planning for autonomous exploration. Master's thesis, Massachusetts Institute of Technology (2004)
5. Pfingsthorn, M., Slamet, B., Visser, A., Vlassis, N.: Uva rescue team 2006 robocup rescue - simulation league. In: Lakemeyer, G., Sklar, E., Sorrenti, D.G., Takahashi, T. (eds.) RoboCup 2006: Robot Soccer World Cup X. LNCS (LNAI), vol. 4434, Springer, Heidelberg (2007)
6. Pfingsthorn, M., Slamet, B., Visser, A.: A scalable hybrid multi-robot slam method for highly detailed maps. In: Robocup 2007: Robot Soccer World Cup XI. Lecture Notes on Artificial Intelligence (to be published, 2007)
7. Howard, A., Sukhatme, G.S., Matarić, M.J.: Multi-robot mapping using manifold representations. In: Proceedings of the IEEE - Special Issue on Multi-robot Systems (2006)
8. Pfister, S.T., Kriechbaum, K.L., Roumeliotis, S.I., Burdick, J.W.: A weighted range sensor matching algorithm for mobile robot displacement estimation. IEEE Transactions on Robotics and Automation (to appear, 2007)
9. Bourgault, F., Göktoğan, A., Furukawa, T., Durrant-Whyte, H.F.: Coordinated search for a lost target in a bayesian world. Advanced Robotics 18(10), 979–1000 (2004)

10. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press (2005)
11. Slamet, B., Pfingsthorn, M.: Manifoldslam: a multi-agent simultaneous localization and mapping system for the robocup rescue virtual robots competition. Master's thesis, Universiteit van Amsterdam (2006)
12. Moravec, H.: Sensor fusion in certainty grids for mobile robots. AI Magazine 9, 61–74 (1988)
13. Fox, D., Burgard, W., Thrun, S.: Active markov localization for mobile robots. Robotics and Autonomous Systems 25, 195–207 (1998)
14. Roy, N., Burgard, W., Fox, D., Thrun, S.: Coastal navigation - mobile robot navigation with uncertainty in dynamic environments. In: Proceedings of the IEEE International Conference on Robotics and Automation, pp. 34–40 (1999)
15. 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)
16. Sim, R., Roy, N.: Global a-optimal robot exploration in slam. In: Proceedings of the IEEE International Conference on Robotics and Automation (ICRA), Barcelona, Spain (2005)
17. Yamauchi, B.: A frontier based approach for autonomous exploration. In: Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation, Monterey, July 10-11, 1997 (1997)
18. González-Baños, H.H., Latombe, J.C.: Navigation Strategies for Exploring Indoor Environments. The International Journal of Robotics Research 21(10-11), 829–848 (2002)
19. van Ittersum, M., Xingrui-Ji, Gonzalez, L., Stancu, L.: Natural boundaries. Report, Universiteit van Amsterdam (2007)