Balancing the Information Gain Against the Movement Cost for Multi-robot Frontier Exploration

Arnoud Visser and Bayu A. Slamet

Intelligent Systems Laboratorium Amsterdam, Universiteit van Amsterdam (UvA) Kruislaan 403, 1098 SJ, Amsterdam, The Netherlands
`arnoud@science.uva.nl`

Summary. This article investigates the scenario where a small team of robots needs to explore a hypothetical disaster site. The challenge faced by the robot-team is to coordinate their actions such that they efficiently explore the environment in their search for victims.

A popular paradigm for the exploration problem is based on the notion of frontiers: the boundaries of the current map from where robots can enter yet unexplored area. Coordinating multiple robots is then about intelligently assigning frontiers to robots. Typically, the assignment of a particular frontier to a particular robot is governed by a cost measure, e.g. the movement costs for the robot to reach the frontier. In more recent approaches these costs are traded off with the potential gain in information if the frontier would be explored by the robot.

In this paper we will further investigate the effect of balancing movement costs with information gains while assigning frontiers to robots. In our experiments we will illustrate how various choices for this balance can have a significant impact on the exploratory behavior exposed by the robot team.

1 Introduction

This paper will investigate a multi-robot exploration approach that was designed with the disaster sites of the RoboCup Rescue Competitions in mind. These scenarios can be simulated in the real world [7] or virtually within the USARSim simulator [1]. In either case, the team of robots is challenged to explore the site and locate victims in a constrained amount of time. Afterwards, the efforts of the robot team are evaluated on the amount of covered area, the quality of the produced map and most importantly the number of located victims. See [2] for a more detailed discussion on this scoring process.

The exploratory efforts exposed by our robots have so far been governed by strictly reactive behavior (2006) and tele-operation (2007). Although the autonomous behavior has demonstrated good robustness and obstacle avoidance, any seemingly 'intelligent' autonomous exploration effort was due to randomizations that were inherent to the behavior control design [11].

A well-known paradigm to address the multi-robot exploration challenge in a more intelligent fashion is *frontier*-based exploration. The frontiers are typically defined as the boundaries of the currently mapped free area where the robot can enter yet unexplored area [14]. Collaborating robots can use these frontiers to coordinate their actions

[3, 14, 15], i.e.: assign robots to frontiers such that the robots simultaneously explore multiple yet unexplored parts of the environment. This shifts the exploration problem to a frontier assignment problem.

Most approaches use a *cost* measure to evaluate the utility of a frontier. The anticipated traveling distance to reach a frontier or the associated motion costs are examples of such cost measures. In [6] and [13] however, frontier evaluation approaches were presented that focus on the opposite measure: the *information gain* that can be expected if the frontier would be explored. This gain is expressed as an estimate for the amount of area that lies beyond the frontier. While [6] uses a sampling method that extrapolates the current map to estimate the information gain, the approach of [13] directly measures the expected gain from the *current* map.

This paper will look more carefully at the balance between the cost and the gain. In the following Section we will present our multi-robot exploration strategy. Subsequently, the balance between information gain and movement costs is varied in a number of the experiments in Section 3. This will lead to the conclusions in Section 4.

2   Multi-robot Rescue Site Exploration

Exploration addresses the challenge of directing a robot through an environment such that its knowledge about the environment is maximized [12]. A mobile robot typically maintains its knowledge about the external world in a map m. Increasing the knowledge represented by m is achieved by either reducing the uncertainty about current information, or by inserting new information. The latter occurs when the map coverage is extended as the robot visits areas in the external world not yet covered by m before.

The approach in [11] was to passively acquire the information to store in the map, i.e. while the robot was wandering around pursuing other objectives like finding victims. In this work however, the focus is on *active* exploration: to explicitly plan the next exploration action a which will increase the knowledge about the world the most. In this paradigm victim finding becomes the side-effect of efficient exploration.

Occupancy grids [8] are a convenient representation for m in order to address the exploration challenge as they lend themselves excellently for storing probabilistic information about past observations. Each cell x corresponds to a region in the external world and holds the value $p(x)$ that denotes the aggregated probability that this region in the real world is 'occupied', i.e. is (part of) an obstacle. The objective of active exploration can then be seen as to minimize the *information entropy* $H(m)$ [5] of the probability distribution defined over all $x \in m$:

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

Initially each grid cell has unknown occupancy, so $p(x) = 0.5$ for all $x \in m$ and the entropy of the map $H(m)$ is maximum. For exploration purposes the absolute value of $H(m)$ is not of interest, what is relevant is the difference in entropy before $H(m)$ and after $H(m|a)$ a particular exploration action a: the *information gain* $\Delta I(a)$ [9, 10].

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

Note that the exploration action a could be a complex maneuver, consisting of a number of controls u_i and observations z_i that spans multiple time steps i. Hence, for predictions about $\Delta I(a)$ that lie multiple timestamps in the future, the set of possible exploration actions can grow rather fast. In many current exploration strategies this issue is addressed by evaluating only a limited set of future states. These approaches consider only the situations where a robot actually enters yet unexplored area, which are by definition the locations where open area borders on unknown area: the *frontiers* [14]. The borders can be easily derived from the occupancy grid map m; the unknown area involves all the cells x for which the occupancy $p(x)$ is still at its initial value $p(x) = 0.5$ and the open area involves all the cells for which $p(x)$ is sufficiently close to zero.

2.1   Estimating Beyond Frontiers on Occupancy Grids

A good autonomous exploration algorithm should navigate the robot to optimal target observation points. The approach presented in [13] enables a robot to distinguish these locations using a method that is based on 'safe regions'. The idea is that the robot simultaneously maintains two occupancy grids: one based on the maximum sensing range r_{max} of the range sensing device and another one based on a more conservative safety distance r_{safe}. Typical values for r_{max} and r_{safe} are 20 meters and 3 meters respectively. The result is that the safe region is a subset of the open area. Frontiers can then be extracted on the boundaries of the safe region where the robot can enter the free space. Subsequently, the area beyond the frontier can be estimated directly from the current map by measuring the amount of free space beyond the safe region.

Greedy exploration could continuously focus the robot to the frontier f with largest area $A(f)$ and which will ultimately lead to a complete coverage of the environment. More efficient exploration can be expected when also the distance $d(f)$ is considered in a utility function $U(f)$ that trades off the costs of moving to the frontier with the expected information gain. In our experiments we used the equation:

$$U(f) = A(f)/d^n(f) \tag{3}$$

The parameter n can be tuned to balance the costs against the gains, simular to the constant λ used in the utility function of [6]:

$$U(f) = A(f)e^{-\lambda d(f)} \tag{4}$$

2.2   Multi-robot Coordination

After the frontier extraction method illustrated in Section 2.1 and the utility function from Equation (3) we are left with the challenge to intelligently assign frontiers to the members of a multi-robot team.

Given the set of frontiers $F = \{f\}$ and team of robots $R = \{r\}$ the full utility matrix $U = [u_{ij}]$ can be computed that stores the utility u_{ij} for all possible assignment of robots $r_i \in R$ to frontiers $f_j \in F$. This matrix U is calculated with an Euclidian distance measure $d_{eu}(f)$. The Euclidean distance d_{eu} gives a lower bound of the actual distance to be

---

Algorithm 1. The algorithm for the assignment of frontiers to robots

Data: the identity of the current robot $r_c \in R$ and the map m as known by r_c.

Data: the set of robots r_i in R. Each r_i consist of the tuple $(x_{r_i}, y_{r_i}, \theta_{r_i})$.

Data: the set of frontiers f_j in F. Each f_j consist of the tuple $(x_{f_j}, y_{f_j}, A_{f_j})$.

Result: the pair r_c, f_c and the path p_c to the location (x_{f_c}, y_{f_c})

for *each robot r_i in R* **do**

    for *each frontier f_j in F* **do**

        $d_{eu} = \sqrt{(x_{f_j} - x_{r_i})^2 + (y_{f_j} - y_{r_i})^2}$;

        $u_{ij} = A(f_j)/d_{eu}^n$;

    end

end

$u_{max} = \max u_{ij}$;

repeat

    for *robot r_i and frontier f_j of u_{max}* **do**

        p=PathPlanning from (x_{r_i}, y_{r_i}) to (x_{f_j}, y_{f_j}) on map m;

        d_{pp}=length of path p;

        $u_{ij} = A(f_j)/d_{pp}^n$;

    end

    if $\max u_{ij} = u_{max}$ **then**

        Assign f_j to r_i;

        Prune U from i and j;

    end

    $u_{max} = \max u_{ij}$;

until *robot r_c is Assigned*;

p_c=last path p

---

traveled. This means that the utility values $[u_{ij}]$ are optimistic when $n > 1$. The actual distance to be traveled can be calculated by performing a path-planning operation, but this is typical a computation intensive operation. The efficiency of our algorithm is optimized by recalculating only a few elements of the matrix; only the highest utility $u_{ij} \in U$ is recalculated. Thereafter it is checked if this value is still the maximum value. Otherwise the new maximum value is recalculated with a distance measure based on path-planning. When the path-planned value is the maximum, the frontier f_j is assigned to robot r_i and the rows and columns of the utility matrix U are pruned. This process continues until a frontier is assigned to the current robot r_c. The pseudo code of this algorithm is given in Algorithm 1.

Note that the algorithm makes no assumption about the numbers of frontiers and robots. When this algorithm is applied in real time the computational consequences of a large number of frontiers and robots have to be studied, but typically the two step approach described above limits the path-planning operation to one or two frontiers per robot. So, this algorithm scales nearly linearly with the number of robots. Also, when there are less frontiers than robots some robots will not be assigned a frontier. However,

in our experience these occasions are rather rare as robots usually find more frontiers than they can close.

2.3   Planning Safe Paths

In the second part of the algorithm a check is made if an obstacle free path exists to a frontier. The same occupancy grid that was used to extract the frontiers can also be used to plan safe paths that avoid obstacles. If paths would planned on the free-space robots may be guided to locations that are dangerously close to obstacles. This is a well-studied problem in robotics [4] and several solutions exist.

Because path-planning has to be performed for several robot-frontier combinations, a simple method has been applied that gives fast reasonable path. The occupancy map is converted into a safety map by convoluting the obstacles with the shape of the robot. When the shape of the robot is non-holonomic, the convolution can be performed by taking the Minkowski sum. For holonomic robots, as used in this study, the convolution can be approximated by employing a Gaussian convolution kernel. On this approximated safety map path-planning is performed with a breath-first algorithm.

This completes the set of tools necessary to enable coordinated frontier-based exploration by a team of robots. In the subsequent section the method will be applied to guide robots through to a virtual disaster site from two start positions.

3   Experiments and Results

Our experiments will be based on the 'Hotel Arena' that was used extensively during the RoboCup Rescue competitions of 2006[1]. Figure 1 shows a blue-print of this office-like environment. The wide vertical corridor in the center connects the lobby in the south with several horizontal corridors that go east and west. Numerous rooms border on all

Fig. 1. A blue-print of the Hotel Arena, the virtual environment used for our experiments

---

the corridors. For the competition runs, robots would typically be spawned in the lobby or at the far ends of corridors, e.g. in the north-east, north-west or south-east corners.

Following the same setup as in the RoboCup competition each experiment will involve a run of 20 minutes. So the comparisons will focus on the amount of area that the robots were able to explore in this fixed time-window.

3.1   Results Based on Coordinated Exploration

These experiments involve multi-robot exploration using the presented approach. We used two spawn positions which were frequently used in the competitions of 2006. For each set of spawn positions a number of runs with a team of two collaborating robots were performed, each time with another formula for the utility function $U(f) = A(f)/d^n(f)$. In Figure 2 and 3 the resulting maps are given, for an inverse linear ($n = 1$), quadratic ($n = 2$) and cubic ($n = 3$) dependence on the distance $d(f)$.

On the maps the following color-coding is used:

- blue (dark grey) indicates unknown terrain,
- shades between light-blue (light grey) and white indicate the probability that the area is free from obstacles
- black indicates obstacles
- solid grey indicates 'safe region', as introduced in Section 2.1
- red indicates a victim
- light-green (light grey) line inside safe region indicates the path of the robots

In Figure 2 the robots started in the north-east corner. In Figure 2(a) the first robot (called Hercules) starts exploring the north-east corridor, while the second robot (called Achilles) explores the first two rooms. Hercules arrives at the T-junction, and decides to go south in the direction of the lobby. In the mean time Achilles is ready with the second room, and favors the unexplored corridors above exploring the nearby third room. The east corridor is chosen, which is explored till the end. Hercules explores firstly the west side and secondly the east side of the lobby. When the lobby is well covered, Hercules continues the exploration in the south-east corridor. This corridor is explored till the end, and the robot has time enough to enter a last chamber (a maze called the Yellow arena). This was a very successful run. More area can only be covered when Achilles would not have explored the first two rooms, but would instead explore the north-west corridor. Such a decision should be based on heuristics, because the existence of this corridor can only be known when a robot arrives at the T-junction. Until that time, the exploration of the nearby rooms is a good choice.

With an inverse quadratic dependence, the robot behavior is more tuned to nearby frontiers. As can be seen in Figure 2(b), with $n = 2$ the first three rooms are explored (one by Hercules, two by Achilles). Hercules goes again towards the lobby, while Achilles explores the north-west corridor. Hercules enters first the Yellow arena, before he continues with the south-east corridor. Notice in the north-west corridor that the localization is a few degrees off, because the robot Achilles had a lot of trouble to navigate in a confined space near the victim in the third room.

With an inverse cubic dependence, even the fifth room along the north-east corridor is entered (the entrance to the fourth room is blocked by a victim). Unfortunately, Achilles

got stuck at that location, and the exploration has to be continued by the other robot. Hercules enters the large room west of the north-south corridor, and finds two victims. Afterwards, the north-south corridor and even the lobby is explored.

It should be clear from these examples that for a smaller n the robots have a tendency to stay in the corridors. This is a good strategy to cover a large area. On the other hand, a larger n can be used to direct the robots into the rooms. In those rooms victims can be found, but the confined space is more difficult to navigate, and the chance exists that the robot will not be able to leave the room anymore.

(a) $A(f)/d(f)$ (629 m^2, 6 victims)   (b) $A(f)/d^2(f)$ (619 m^2, 8 victims)


(c) $A(f)/d^3(f)$ (486 m^2, 8 victims)
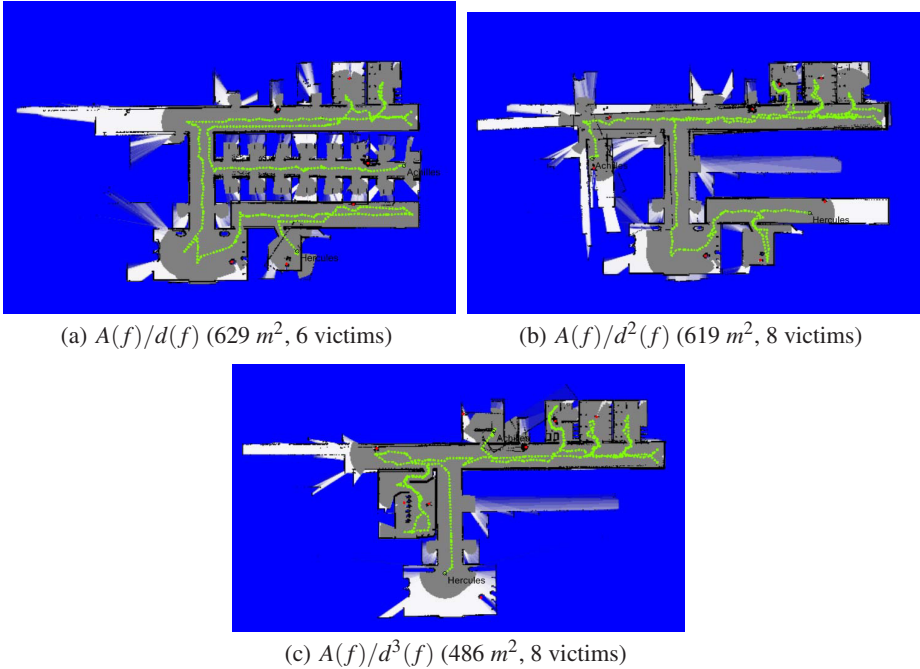
Fig. 2. Exploration from the north-east corner with two robots

The experiments from the other start location show that tuning the parameter n not always leads to different behavior. For instance, in Figure 3(a-c) it can be seen that Hercules always explores the lobby, enters the south-east corridor and part of the Yellow arena. The south-east corridor is a dead-end, with enough free space to explore. Against the time that one is finished with this corridor, no time is left for any other choices.

The difference between the maps is mainly due to Achilles. In Figure 3(a) Achilles enters the north-south corridor and explores the east corridor. At the end of the corridor (again an dead-end), Achilles turns and goes back to the north-south corridor. In Figure 3(b) Achilles explores the north-east corridor and three rooms at the last corridor. In Figure 3(c) Achilles first checks the south-west corridor, before it heads towards the north-east corridor. Due to the detour there is only time left to explore a single room at the north-east corridor, but the area seen in the south-west corridor compensates more

than enough. During its exploration, Achilles mainly encounters large corridors. Increasing the parameter *n* resulted in different behavior, because different corridors are chosen. This illustrates that difference in behavior doesn't always have the same effect. In Figure 3 an increase of *n* has as result more area and fewer victims, but this was mainly due to the particular layout of the corridors.

(a) $A(f)/d(f)$ (454 m^2, 6 victims)

(b) $A(f)/d^2(f)$ (512 m^2, 6 victims)

(c) $A(f)/d^3(f)$ (570 m^2, 4 victims)
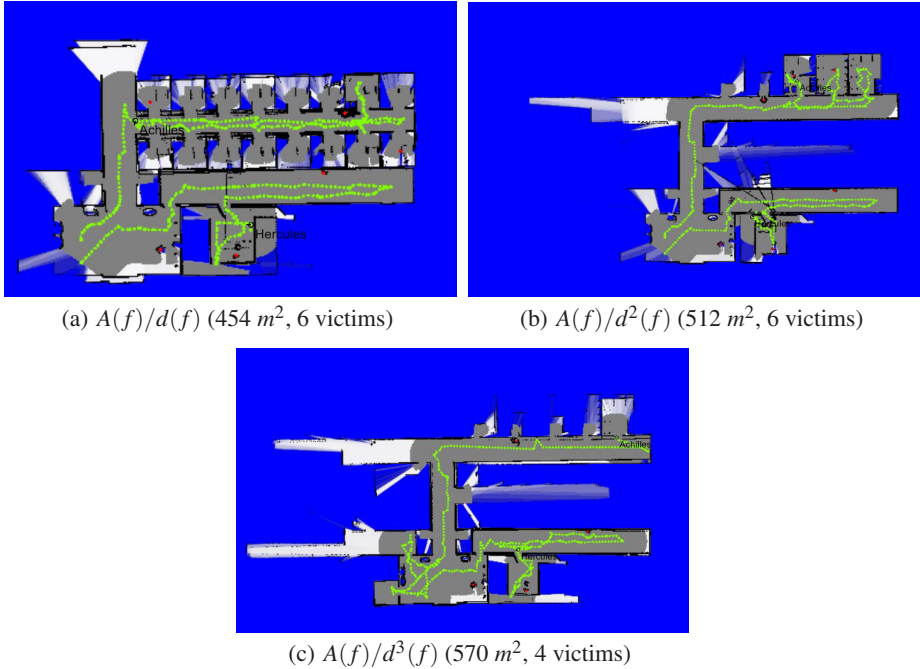
Fig. 3. Exploration from the lobby with two robots

Notice that the number of victims found by the two robots in the previous experiments is comparable to the number of victims found in the semi-final by teams with four to eight robots [2]. Experiments with only two robots are not enough to make strong claims about the scalability of the algorithm to larger teams, but detailed analysis of the experiments indicated that in our current implementation the performance bottleneck is in merging the observations of multiple robots into a shared map. Planning the exploration on this shared map goes rather efficiently and is more a function of the length of the resulting path (e.g. when backtracking from a long dead-end corridor) than the number of robots or frontiers. Future experiments have back up this claim.

4   Conclusions

This paper investigated a frontier-based exploration approach that can be used to coordinate a team of robots. The approach assigns utilities to frontiers using a measure of the

information gain that can be estimated directly from the *current* map. This information gain is balanced by the movement costs. In a first approximation these movement costs are estimated by the Euclidian distance. The actual movement costs are checked by performing path-planning on the map. Subsequently, a frontier with the highest utility is assigned to the members of a robot team.

The information gain and the movement costs can be balanced by a parameter. In the presented experiments is shown that tuning this parameter can change the overall behavior from exploring mainly corridors towards exploring nearby rooms. This parameter could be further tuned towards the scoring-function as applied in the RoboCup [2], but before this tuning is applied, the underlying navigation should be optimized. Currently, corridors and rooms are explored with the same care and speed. Inside the corridors the speed can be increased, to allow fast coverage of large areas. Inside the rooms the care could be increased, to guarantee that a room once entered could also be leaved.

In our experiments we have shown that this approach leads to efficient rescue site coverage. In future work we would like to investigate the possibilities for multi-robot coordinated exploration with more than two robots, study the influence of a-priori data, the effect of distributed decision making and the conditions where only limited communication is possible.

Acknowledgments

References

1. Balakirsky, S., Scrapper, C., Carpin, S., Lewis, M.: Usarsim: providing a framework for multi-robot performance evaluation. In: Proceedings of PerMIS 2006, August 2006, pp. 98–103 (2006)
2. 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 24(11-12), 943–967 (2007)
3. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. IEEE Transactions on Robotics 21(3), 376–378 (2005)
4. Choset, H., Lynch, K.M., Hutchinson, S., Kantor, G.A., Burgard, W., Kavraki, L.E., Thrun, S.: Principles of Robot Motion: Theory, Algorithms, and Implementations, June 2005. MIT Press, Cambridge (2005)
5. Fox, D., Burgard, W., Thrun, S.: Active markov localization for mobile robots. Robotics and Autonomous Systems 25, 195–207 (1998)
6. 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)
7. Jacoff, A., Messina, E., Weiss, B.A., 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 (October 2003)

8. Moravec, H.: Sensor fusion in certainty grids for mobile robots. AI Magazine 9, 61–74 (1988)
9. 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)
10. Simmons, R.G., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H.: Coordination for multi-robot exploration and mapping. AAAI/IAAI, 852–858 (2000)
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 (December 2006)
12. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents), September 2005. MIT Press, Cambridge (2005)
13. Visser, A., Xingrui-Ji, v.I.M., Jaime, L.A.G.: Beyond frontier exploration. In: Proceedings of the 11th RoboCup International Symposium (July 2007)
14. Yamauchi, B.: Frontier-based exploration using multiple robots. In: AGENTS 1998: Proceedings of the second international conference on Autonomous agents, pp. 47–53. ACM Press, New York (1998)
15. 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 (2002)