Collaborative Multi-Robot Exploration

Wolfram Burgard[†] Dieter Fox[‡] Mark Moors[††] Reid Simmons[‡] Sebastian Thrun[‡]

[†]Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany
[‡]School of Computer Science, Carnegie Mellon University, Pittsburgh, PA 15213
[††]Department of Computer Science III, University of Bonn, 53117 Bonn, Germany

Abstract

In this paper we consider the problem of exploring an unknown environment by a team of robots. As in single-robot exploration the goal is to minimize the overall exploration time. The key problem to be solved therefore is to coordinate the robots so that they simultaneously explore different regions of their environment. We present a probabilistic approach for the coordination of multiple robots which, in contrast to previous approaches, simultaneously takes into account the costs of reaching a target point and the utility of target points. Since the environment is unknown, we provide a technique to estimate the area expected to be visible when a robot reaches the target position. The utility of unexplored areas which are close to a selected target point are reduced according to the expected visibility. This way, a team of multiple robots assigns different target points to the individual robots. The technique has been implemented and tested extensively in real-world experiments and simulation runs. The results given in this paper demonstrate that our coordination technique significantly reduces the exploration time compared to previous approaches.

1 Introduction

The problem of exploring an environment belongs to the fundamental problems in mobile robotics. In order to construct a model of their environment mobile robots need the ability to efficiently explore it. The key question during exploration is where to move the robot in order to minimize the time needed to completely explore an environment. This problem unfortunately is already NP-hard for known, graph-like environments. In this case it directly corresponds to the problem of finding the shortest roundtrip through all nodes of the graph, which is the well-known traveling salesman problem.

The use of multiple robots is often suggested to have several advantages over single robot systems [5, 4]. First, cooperating robots might be able to accomplish a single task faster than a single robot. For example, [8] built a system of collaborative robots that jointly schedule a meeting which outperformed several single robot systems designed to accomplish the same task. Furthermore, multiple robots can localize themselves more efficiently if the exchange information about their position whenever they sense each other [6]. Finally, using several cheap robots introduces redundancy and therefore can be expected to be more fault-tolerant than having only one powerful and expensive robot.

In this paper we consider the problem of collaborative exploration of an unknown environment by multiple robots. The problem to be solved when using multi-robot systems is to coordinate the actions of the robots. Without any coordination, all robots might follow the same exploration path so that the whole group of robots requires the same amount of time as a single robot would need. Therefore, the key problem in multi-robot exploration is to choose different actions for the individual robots so that they simultaneously explore different areas of their environment.

In this paper we present a technique for coordinating a group of robots while they are exploring their environment. This approach uses a map which is built based on the data sensed by the individual robots. Instead of just moving with all robots to the target points which have the minimum travel cost, as previous approaches do, our approach additionally considers the utility of unexplored positions. This utility is reduced as soon as one robot chooses a target position in the visibility range. By trading off the utility and costs of unexplored positions our approach achieves the coordination in an elegant way.

Whereas the exploration problem has been studied in detail for single robot systems [1, 7] there are only a few approaches for multi-robot systems. Concerning the collaborative exploration by multiple robots Rekleitis et al. [13, 14] focus on the problem of reducing the odometry error during exploration. They separate the environment into stripes which are explored successively by the robot team. Whenever one robot moves the other robots are kept stationary and observe the moving robot, a strategy similar to [10]. Whereas this approach can significantly reduce the odometry error during the exploration process, it is not designed to distribute the robots over the environment. Rather, the robots are forced to stay close to each other in order to remain in the visibility range. Thus, us-

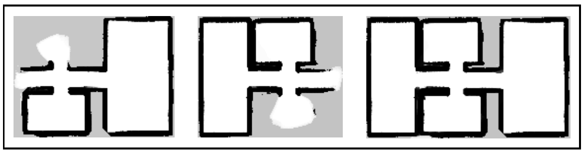

Figure 1: Integration of two individual maps into a global map

ing these strategies for multi-robot exploration one cannot expect that the exploration time is significantly reduced.

The most advanced technique for multi-robot exploration has recently been presented by Yamauchi [16]. In this approach, the robots share a common map which is built during the exploration phase. Each robot moves to the closest frontier [17] which is the closest unknown area around the robot. There is no coordination component which chooses different frontiers for the individual robots. Our approach, in contrast, is especially designed to coordinate the robots so that they do not choose the same frontier. As a result, our approach needs significantly less time to accomplish the task.

2 Exploration of Unknown Environments

The goal of an exploration process is to cover the whole environment in a minimum amount of time. Therefore, it is essential that the robots keep track of which areas of the environment have already been explored. Furthermore, the robots have to construct a global map in order to plan their paths and to coordinate their actions. As in [16], our approach uses occupancy grid maps [11, 15] to represent the environment. We also keep track of the already explored area in order to identify possible target locations. Since we do not have any prior knowledge about the structure of the environment, we estimate the area which is expected to be covered by the robot's sensors when it reaches its target point. Based on this information we choose different target positions for the remaining robots. The only assumption we make is that the robots know their relative positions during the exploration process.

2.1 Integrating Occupancy Grid Maps

The idea of occupancy grid maps is to use a grid of equally spaced cells and to store in each cell the probability $P(occ_{x,y})$ that this cell is occupied by an obstacle. Due to this probabilistic nature, occupancy grid maps built by different robots can easily be integrated if their relative positions are known. Suppose there are N robots which all have an individual map m_i. Furthermore, let $P(occ^i_{x,y})$ denote the probability that the location $\langle x, y\rangle$ in the global coordinate frame is occupied in the map of robot i. Then we integrate the maps of the different robots according to the following formula [11, 15, 3]:

$$P(occ_{x,y}) = \frac{odds_{x,y}}{1 + odds_{x,y}} \tag{1}$$

where

$$odds_{x,y} = \prod_{i=1}^{n} odds^i_{x,y} \tag{2}$$

and

$$odds^i_{x,y} = \frac{P(occ^i_{x,y})}{1 - P(occ^i_{x,y})} \tag{3}$$

As an example consider the maps depicted in Figure 1. Here the two local maps shown on the left are integrated into one global map shown on the right side of the figure.

2.2 Target Point Selection

The key question during the exploration of unknown environments is to guide the robots to target points so that the overall time needed to explore the complete environment is minimized. Our approach uses the concept of frontier cells [17]. A frontier cell is a known, i.e. already explored cell which is an immediate neighbor of an unknown, i.e. unexplored cell.

Our technique constructs a map of the environment and iteratively chooses target points for the individual robots based on the trade-off between the costs of reaching the target point and its utility. Since the environment is not known, it estimates the expected area which will be explored when a robot reaches its target position. It then reduces the utility of unexplored points close to the chosen target position and uses the reduced utility to compute goal positions for the remaining robots.

2.2.1 Costs

To determine the cost of reaching the current frontier cells, we compute the optimal path from the current position of the robot to all frontier cells based on a deterministic variant of *value iteration*, a popular dynamic programming algorithm [2, 9]. In this approach, the cost for traversing a grid cell $\langle x, y\rangle$ is proportional to its occupancy value $P(occ_{xy})$. The minimum-cost path is computed using the following two steps.

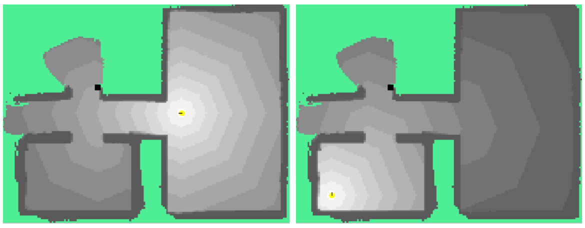

Figure 2: Typical value functions obtained for two different robot positions. The black rectangle indicates the target points in the unknown area with minimum cost

1. **Initialization.** The grid cell that contains the robot location is initialized with 0, all others with ∞:

$$V_{x,y} \quad \longleftarrow \quad \begin{cases} 0, & \text{if } \langle x, y \rangle \text{ is the robot position} \\ \infty, & \text{otherwise} \end{cases}$$

2. **Update loop.** For grid cells $\langle x, y \rangle$ do:

$$V_{x,y} \quad \longleftarrow \quad \min_{\substack{\Delta x = -1,0,1 \\ \Delta y = -1,0,1}} \Big\{ V_{x+\Delta x, y+\Delta y} + \sqrt{\Delta x^2 + \Delta y^2} \cdot P(occc_{x+\Delta x, y+\Delta y}) \Big\}$$

Value iteration updates the value of all grid cells by the value of their best neighbors, plus the cost of moving to this neighbor (just like A* [12]). Cost is here equivalent to the probability $P(occc_{x,y})$ that a grid cell $\langle x, y \rangle$ is occupied times the distance of the cell. The update rule is iterated. When the update converges, each value $V_{x,y}$ measures the *cumulative cost* for moving to the corresponding cell. The resulting value function V can also be used to efficiently derive the the minimum-cost path from the current location of the robot to arbitrary goal positions. This is done by steepest descent in V, starting at the desired goal position.

Figure 2 shows the resulting value functions for two different robot positions in the rightmost map of Figure 1. The black rectangle indicates the target point in the unknown area with minimum travel cost. Please note that the same target point is chosen in both situations.

2.2.2 Expected Visibility Range

As already mentioned above, a naive approach to multi-robot exploration would be to move every robot to the closest frontier cell. This however would not prevent two different robots to approach the same target position (see Figure 2). To achieve a coordinated exploration of the environment it is highly important to avoid that two robots choose the same target position (or one which is in the visibility range of another robot's target point). Thus, we need to know which part of the environment will be covered by the robot's sensors when it reaches its designated target position. Unfortunately, the exact area that a robot's sensors will cover is unpredictable — otherwise there would be no exploration problem. In this section we will devise a heuristic to estimate the covered area. It is based on probabilistic considerations and has been found to work well in practice. The key idea of this heuristic is based on the observation that a robot exploring a big open terrain can cover much larger areas than a robot exploring a narrow part of the environment.

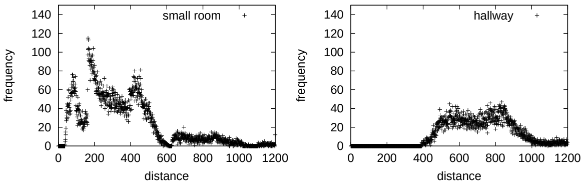

Figure 3: Distance histograms $h(d \mid s)$ obtained in a small room (left) and in a hallway (right)

During exploration we count for a discrete set of distances $d_1, \ldots, d_n$, the number of times $h(d_i)$ the distance d_i was measured by any of the robots. Based on this histogram we can compute the probability that a cell in a certain distance d will be covered by a sensor beam and thus will be explored after the robot reached its target. In essence, we are interested in the quantity $P(d)$ which is the probability that the robot's sensors cover objects at distance d:

$$P(d) \quad = \quad \frac{\sum_{d_i \geq d} h(d_i)}{\sum_{d_i} h(d_i)} \tag{4}$$

The advantage of this approach is that it automatically adapts itself according to the free space around the robot. For example, in an area with wide open spaces such as a hallway, the robots will sense more long readings than in narrow areas or small rooms. As an example consider the different histograms depicted in Figure 3. Here the robot started in a large open hallway and in a typical office room. Obviously the robot measures shorter readings in rooms than in a hallway. Correspondingly, the probability of measuring at least $4m$ is almost one in the hallway whereas it is comparably small in a room (see Figure 4).

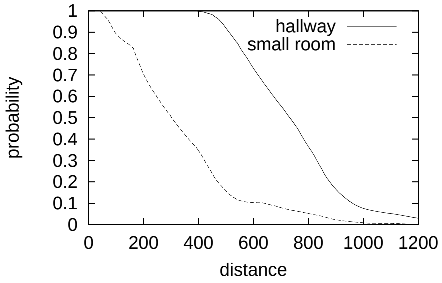

Figure 4: Probability $P(d)$ of measuring at least d given the histograms in Figure 3.

2.2.3 The Target Point Selection Algorithm

Given the expected visible area we now can estimate the utility $U_{x,y}$ of frontier cells $\langle x,y\rangle$. Initially, the utility is set to 1. Whenever a target point is selected for a robot, we reduce the Utility of the adjacent points in distance d according to their visibility probability $P(d)$. The target point is selected by trading off the utility $U_{x,y}$ and the cost $V^i_{x,y}$ of moving there. This results in the following algorithm shown in Table 1.

```
1. Determine the set of frontier cells
2. Compute for each robot $i$ the cost $V^i_{x,y}$ for reaching each frontier cell
3. Set the utility $U_{x,y}$ of all frontier cells to 1
4. While there is one robot left without a target point
    (a) Determine a robot $i$ and a frontier cell $\langle x,y\rangle$ which satisfy
        $(i,\langle x,y\rangle) = \underset{(i',\langle x',y'\rangle)}{\operatorname{argmax}} U_{x',y'} - V^{i'}_{x',y'}$   (5)
    (b) Reduce the utility of each target point $\langle x',y'\rangle$ in the visibility area according to
        $U_{x',y'} \leftarrow U_{x',y'} \cdot (1 - P(||\,\langle x,y\rangle - \langle x',y'\rangle\,||))$   (6)
```

Table 1: The Target Point Selection Algorithm

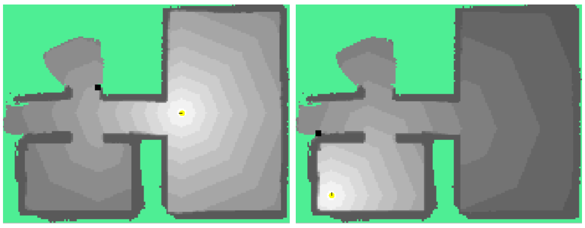

Figure 5: Target positions obtained using the coordination approach. In this case the target point for the second robot is to the left in the corridor.

Please note that in step 3.a this approach chooses a robot and target point pair $(i,\langle x,y\rangle)$ which has the best overall evaluation. Figure 5 illustrates the effect of our coordination technique. Whereas uncoordinated robots would choose the same target position (see Figure 2), the coordinated robots select different frontier cells as the next exploration targets.

3 Experimental Results

The approach described has been implemented and extensively tested on real robots and in real environments. Additionally to the experiments with real robots we performed a series of simulation experiments to get a quantitative assessment of the improvements of our approach over previous techniques.

3.1 Implementation Details

Our current system uses an efficient implementation of value iteration. It requires less than .2 seconds until convergence in environments with a size of 27 × $20m^2$ as it is used in the simulation experiments. However, the value iteration technique described in Section 2.2.1 is a deterministic variant of the original value iteration approach [2, 9]. It assumes that the actions of the robot are always executed with absolute certainty. The advantage of this approach is that it can be implemented much more efficient than the original value iteration. To deal with the uncertainty of the robots motions and benefit from the efficiency of the deterministic variant, we smooth the input maps by a convolution with a Gaussian kernel. This has a similar effect as generally observed when using the non-deterministic approach: It introduces a penalty for traversing narrow passages or staying close to obstacles. Therefore, the robots generally prefer target points in open spaces rather than behind narrow doorways.

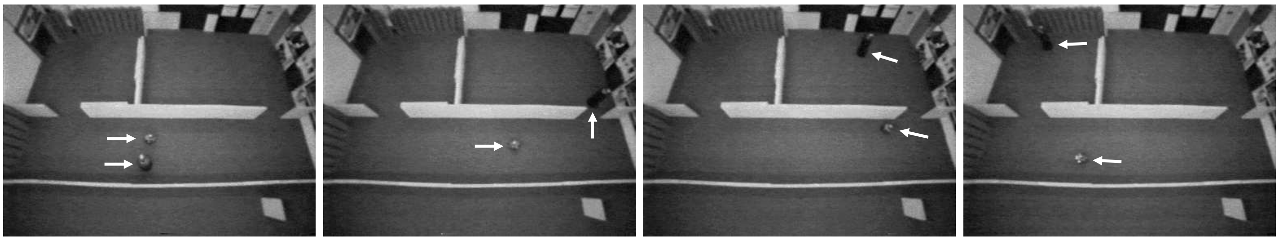

Figure 6: Uncoordinated exploration by the robots two robots Defiant and Yang

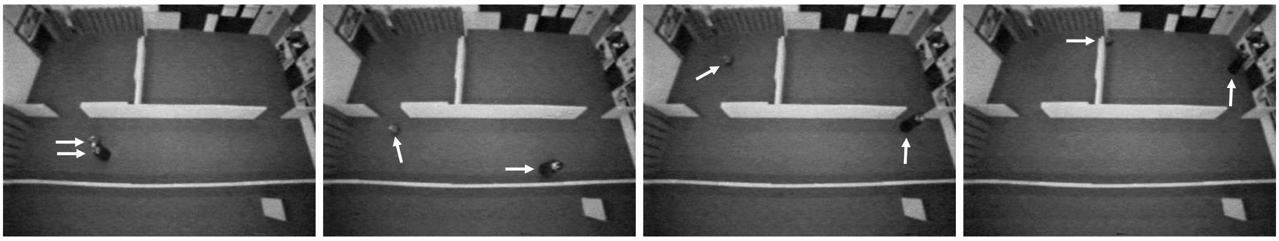

Figure 7: Coordinated exploration by the robots two robots Defiant and Yang

3.2 Exploration with Two Robots

The first experiment described in this section is designed to illustrate the advantage of our coordination technique over the uncoordinated approach in which the robots share a map and each robot approaches the frontier position with minimum cost. For this experiment we used the robots Defiant and Yang. Defiant is an RWI B21 robot equipped with two laser range-finders. Yang is a Pioneer I robot equipped with a single laser range-finder. The range of laser range-finders was limited to 5m in this experiment. Figure 6 shows the typical behaviour of the two robots when they explore their environment without coordination, i.e. when they always move to the unexplored location with minimum cost. The white arrows indicate the positions and directions of the two robots. Since the cost for moving through the narrow doorway in the upper left room are higher than the cost for reaching a target point in the corridor, both robots decide first to explore the corridor. After reaching the end of the corridor Defiant enters the upper right room. At that point Yang assigns the highest utility to the upper left room and therefore turns back. Before Yang reaches the upper left room Defiant already entered it and completed the exploration mission. In this example Defiant explored the whole environment on its own and Yang did not contribute anything. Accordingly, the exploration time is worst in this case.

If, however, both robots are coordinated, then they perform much better (see Figure 7). As in the previous example, Defiant moved to the end of the corridor. Since the utilities of the frontier cells in the corridor are reduced, Yang decides to enter the upper left room. As soon as Defiant entered the upper right room, the exploration mission is finished. As a result, the exploration is finished much earlier than in the uncoordinated case.

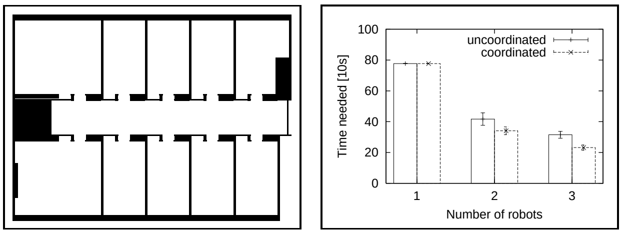

Figure 8: Environment used for the simulation experiments and average time needed by robot teams to explore this environment.

3.3 Simulation Experiments

The previous experiment gives only a qualitative illustration of the different behaviours of coordinated and uncoordinated robot teams. To get a more quantitative assessment we performed several simulation experiments. We used the $27 \times 20m^2$ large environment depicted on the left side of Figure 8 which is an outline of our office environment. We performed ten different experiments using two and three robots. In each experiment we randomly chose the initial positions of the robots in the map. Each experiment was carried out with and without coordination. The right part of Figure 8 shows the average time needed to explore the environment by several robots. The error bars indicate 95% confidence intervals. As expected, even an uncoordinated team of robots is faster than a single robot. However, the coordinated robots require significantly less time than the uncoordinated robots. Please note that in this experiment, two coordinated robots take about the same time to explore the area as three uncoordinated robots.

4 Summary and Conclusions

In this paper we presented a technique for coordinating a team of robots while they are exploring their environment. The key idea of this technique is that it simultaneously takes into account the cost of reaching a so far unexplored location and its utility. The utility of a target location depends on the probability that this location is visible from a target location assigned to another robot. It always assigns that target location to a robot which has the best trade-off between the utility of the location and the cost for the robot to reach this location. Our technique has been implemented and tested on real robots. The experiments presented in this paper demonstrate that our approach is able to coordinate a team of robots so that they choose different target points during exploration.

Our approach differs from previous techniques in that it has an explicit coordination mechanism which is designed to assign different target locations to the robots. Previous approaches to multi-robot exploration either forced the robots to stay close to each other, or used a greedy strategy which assigns to each robot the target point with minimum cost. This, however, does not prevent different robots to select the same target location. According to that, our approach is significantly faster than previous approaches.

Despite these encouraging results, there are several aspects which could be improved. In this paper we proposed a greedy strategy to the NP-hard exploration problem. It is likely that more sophisticated strategies perform better. Additionally, one could use improved techniques for estimating the area that can be expected to be visible when a robot reaches its target location. Another interesting research direction is to consider situations in which the robots do not know their relative positions. In this case the exploration problem becomes even harder, since the robots now have to solve two problems. On one hand they have to extend the map and on the other hand they need to find out where they are relative to each other.

Acknowledgments

We would like to thank Frank Schneider and Dennis Wildermuth from the Research Institute for Applied Science (FGAN) for their help and support while we carried out the experiments.

References

[1] S. Albers, K. Kursawe, and S. Schuierer. Exloring unknown environments with obstacles. In *Proc. of the 10th Symposium on Discrete Algorithms*, 1999.

[2] R. E. Bellman. *Dynamic Programming*. Princeton University Press, Princeton, NJ, 1957.

[3] W. Burgard, D. Fox, H. Jans, C. Matenar, and S. Thrun. Sonar-based mapping with mobile robots using EM. In *Proc. of the International Conference on Machine Learning*, 1999.

[4] Y.U. Cao, A.S. Fukunaga, and A.B. Khang. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4, 1997.

[5] G. Dudek, M.. Jenkin, E.E. Milios, and D. Wilkes. A taxonomy for multi-agent robotics. *Autonomous Robots*, 3(4), 1996.

[6] D. Fox, W. Burgard, H. Kruppa, and S. Thrun. Collaborative multi-robot localization. In *Proceedings of the 23rd German Conference on Artificial Intellige nce*. Springer Verlag, 1999.

[7] H.H. Gonzalez-Banos and J.C. Latombe. Planning robot motions for range-image acquisition and automatic 3d model construction. In *AAAI Fall Symposium*, 1998.

[8] D. Guzzoni, A. Cheyer, L. Julia, and K. Konolige. Many robots make short work. *AI Magazine*, 18(1):55–64, 1997.

[9] R. A. Howard. *Dynamic Programming and Markov Processes*. MIT Press and Wiley, 1960.

[10] R. Kurazume and N. Shigemi. Cooperative positioning with multiple robots. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1994.

[11] H.P. Moravec. Sensor fusion in certainty grids for mobile robots. *AI Magazine*, pages 61–74, Summer 1988.

[12] N. J. Nilsson. *Principles of Artificial Intelligence*. Springer Publisher, Berlin, New York, 1982.

[13] I.M. Rekleitis, G. Dudek, and E.E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *International Joint Conference in Artificial Intelligence*, pages 1340–1345, Nagoya, Japan, August 1997. Morgan Kaufmann Publishers, Inc.

[14] I.M. Rekleitis, G. Dudek, and E.E. Milios. Accurate mapping of an unknown world and online landmark positioning. In *Proceedings of Vision Interface 1998*, pages 455–461, Vancouver, Canada, June 1998.

[15] S. Thrun. Learning maps for indoor mobile robot navigation. *Artificial Intelligence*, 1998.

[16] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 47–53, Navy Research Laboratory, Washington, DC 20375-5337, May 1998. Navy Center for Applied Research in Artificial Intelligence.

[17] B. Yamauchi, P. Langley, A. C. Schultz, J. Grefenstette, and W. Adams. Magellan: An integrated adaptive architecture for mobile robots. Technical Report 98-2, Institute for the Study of Learning and Expertise (ISLE), Palo Alto, CA, May 1998.