Collaborative Exploration of Unknown Environments with Teams of Mobile Robots

Wolfram Burgard[1], Mark Moors[2], and Frank Schneider[3]

[1] Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany
[2] Department of Computer Science, University of Bonn, 53117 Bonn, Germany
[3] Research Establishment for Applied Science, 53343 Wachtberg, 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 in the context of multiple robots is to choose appropriate target points for the individual robots so that they simultaneously explore different regions of the environment. We present an approach for the coordination of multiple robots which, in contrast to previous approaches, simultaneously takes into account the cost of reaching a target point and its utility. The utility of a target point is given by the size of the unexplored area that a robot can cover with its sensors upon reaching that location. Whenever a target point is assigned to a specific robot, the utility of the unexplored area visible from this target position is reduced for the other robots. 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. There are several applications like planetary exploration [3], reconnaissance [26], rescue, mowing [28], or cleaning [19, 48] in which the complete coverage of a terrain belongs to the inherent goals of a robotic mission.

In this paper, we consider the problem of exploring unknown environments with teams of mobile robots. The use of multiple robots is often suggested to have several advantages over single robot systems [9, 17]. First, cooperating robots have the potential to accomplish a single task faster than a single robot [25]. Furthermore, using several robots introduces redundancy. Teams of robots therefore can be expected to be more fault-tolerant than only one robot. Another advantage of robot teams is due to merging of overlapping information, which can help compensate for sensor uncertainty. For example, multiple robots have been shown to localize themselves more efficiently, especially when they have different sensor capabilities [21]. However, when robots operate in teams there is the risk of possible interferences between them [20, 22]. For example, if the robots have the same type of active sensors such as ultrasound sensors, the overall performance can be reduced due to cross-talk between the sensors. Furthermore, the more robots are used the longer detours may be necessary in order to avoid collisions with other members of the team.

In this paper we present an algorithm for coordinating a group of robots while they are exploring their environment. Our method, which has originally been presented in [40] and has been integrated into two different systems [8, 47], follows a decision-theoretic approach. Instead of greedily guiding every robot to the closest unexplored area, our algorithm explicitly coordinates the robots. It tries to maximize overall utility by minimizing the potential for overlap in information gain amongst the various robots. Our algorithm simultaneously considers the utility of unexplored areas and the cost for reaching these areas. By trading off the utilities and the cost and by reducing the utilities according to the number of robots that already are heading towards this area, coordination is achieved in a very elegant way. The underlying mapping algorithm, which is described in detail in [52], is an on-line solution to the *simultaneous localization and mapping problem (SLAM)* [10, 15]. In a distributed fashion it computes a consistent representation of the environment explored so far and also determines the positions of the robots given this map.

Our technique has been implemented on teams of heterogeneous robots and has been proven effectively in realistic real-world scenarios. Additionally we have carried out a variety of simulation experiments to explore the properties of our approach and to compare the coordination mechanism to other approaches developed so far. As the experiments demonstrate, our technique significantly reduces the time required to completely cover an unknown environment with a team of robots.

2 Coordinating a Team of Robots during Exploration

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. Throughout this section we assume that at every point in time both, the map of the area explored so far and the positions of the robots in this map are known. The focus of this section lies in the question of how to coordinate the robots in order to efficiently cover the environment. The mapping system will briefly be described Section 3.

Our system uses occupancy grid maps [41, 52] to represent the environment. Each cell of such an occupancy grid map contains a numerical value representing the probability that the corresponding area in the environment is covered by an obstacle. Since the sensors of real robots generally have a limited range and since often parts of the environment are occluded by objects, a map generally contains certain cells whose value is "unknown" since they have never been updated so far. Throughout this paper, we assume that exploredness is a binary concept and we regard a cell as explored as soon as they have been covered by a sensor beam.

When exploring an unknown environment we are especially interested in "frontier cells" [53]. As a frontier cell we denote each already explored cell that is an immediate neighbor of an unknown, unexplored cell. If we direct a robot to such a cell, we can expect that it gains information about the unexplored area when it arrives at its target location. The fact that a map generally contains several unexplored areas raises the problem that there often are multiple possibilities of directing robots to frontier cells. On the other hand, if multiple robots are involved, we want to avoid that several of them move to the same location. Our system uses a decision-theoretic framework approach to determine appropriate target locations for the individual robots. We simultaneously consider the cost of reaching a frontier cell and the utility of that cell. For each robot, the cost of a cell are proportional to the distance between the robot and that cell. The utility of a frontier cell instead depends on the number of robots that are moving to that cell or to a place close to that cell.

In the following sections we will describe how we compute the cost of reaching a frontier cell for the individual robots, how we determine the utility of a frontier cell and how we choose appropriate assignments of robots to frontier cells.

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 [5, 27]. In our 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.

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

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

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

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

This technique updates the value of all grid cells by the value of their best neighbors, plus the cost of moving to this neighbor. Here, cost is equivalent to the probability $P(occc_{x,y})$ that a grid cell $\langle x, y\rangle$ is occupied times the distance to 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 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 1 shows the resulting value functions for two different robot positions. 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. Accordingly, if the robots are not coordinated during exploration, they would move to the same position which obviously is not optimal.

Our algorithm differs from standard value iteration in that it regards all actions of the robots as deterministic. This way, the value function can be computed faster than

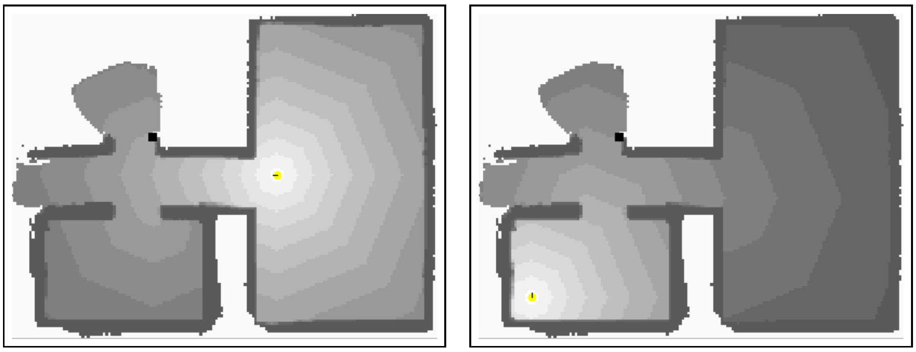

Fig. 1. Typical value functions obtained for two different robot positions. The black rectangle indicates the target points in the unknown area with minimum cost.

with value iteration. To incorporated the uncertainty of the robots motions into our approach and to 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. Please note that the maps depicted in Figure 1 are not smoothed.

2.2 Computing Utilities of Frontier Cells

Estimating the utility of frontier cells is more difficult. In fact, the actual information that can be gathered by moving to a particular location is impossible to predict, since it very much depends on the structure of the corresponding area. However, if there already is a robot that is moving to a particular frontier cell, the utility of that cell can be expected to be lower for other robots. But not only the designated target location has a reduced utility. Since sensors of a robot also cover the terrain around a frontier cell as soon as the robot arrives there, even the expected utility of frontier cells in the vicinity of the robot's target point is reduced.

In this section we will present a technique that estimates the expected utility of a frontier cell based on the distance to cells that are assigned to other robots. To adapt the system to the structure of the environment, we permanently and on-line estimate the visibility range of the sensors of all robots. Suppose in the beginning each frontier cell t has the utility U_t which is equal for all frontier cells if no additional information about the usefulness of certain positions in the environment is available. Whenever a target point t' is selected for a robot, we reduce the utility of the adjacent frontier cells in distance d from t' according to the probability $P(d)$ that the robot's sensors will cover cells in distance d.

To compute the quantity $P(d)$ while the robots are exploring the environment we count for a discrete set of distances $d_1, \ldots, d_n$ the number of times $h(d_i)$ the distance

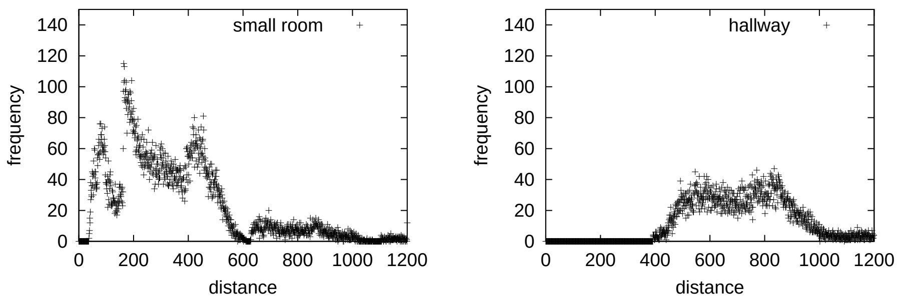

Fig. 2. Distance histograms $h(d \mid s)$ obtained in a small room (left) and in a hallway (right).

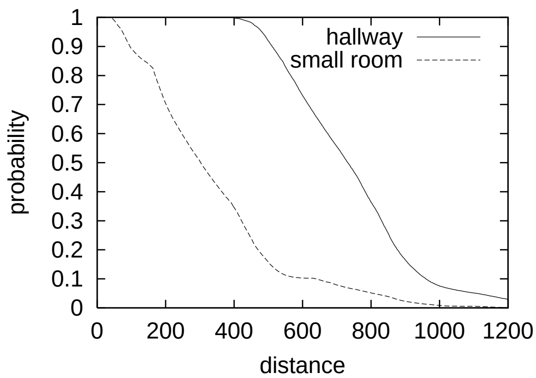

Fig. 3. Resulting likelihood $P(d)$ of measuring at least distance d for the histograms depicted in Figure 2.

d_i was measured by any of the robots. Based on this histogram we can compute the probability $P(d)$ that a cell in distance d will be covered by a sensor beam:

$$P(d) = \frac{\sum_{d_i \geq d} h(d_i)}{\sum_{d_i} h(d_i)} \tag{1}$$

Thus, any cell t in distance d from the designated target location t' will be covered with probability $P(d)$ when the robot reaches t'. Accordingly, we compute the utility $U(t_n \mid t_1, \ldots, t_{n-1})$ of a frontier cell t_n given that the cells $t_1, \ldots, t_{n-1}$ have already been assigned to the robots $1, \ldots, n-1$ as

$$U(t_n \mid t_1, \ldots, t_{n-1}) = U_{t_n} - \sum_{i=1}^{n-1} P(||t_n - t_i||) \tag{2}$$

According to Equation 2, the more robots move to a location from where t_n is likely to be visible, the lower is the utility of t_n.

The advantage of this approach is that it automatically adapts itself according to the free space in the environment. For example, in an area with wide open spaces, such as a hallway, the robots are expected to sense a higher number of long readings than in narrow areas or small rooms. As an example consider the two different histograms depicted in Figure 2. Here a team of robots started in a large open hallway (left image) and in a typical office room (right image). Obviously the robots measure 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 3). Please note that we also take into account whether there is an obstacle between two frontier cells t and t'. This is achieved using a ray-tracing on the grid map. If there is an obstacle in between, we set $P(||t - t'||)$ to zero.

2.3 Target Point Selection

To compute appropriate target points for the individual robots we need to consider for each robot (1) the cost of moving to a location and (2) the utility of that location. In particular, for each robot i we trade-off the cost V_t^i to move to the location t and the utility U_t of t.

Table 1. The Target Point Selection Algorithm with Greedy Assignment.

1. Determine the set of frontier cells
2. Compute for each robot i the cost V_t^i for reaching each frontier cell t
3. Set the utility U_t 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 t which satisfy
$$(i, t) = \underset{(i', t')}{\operatorname{argmax}} \left(U_{t'} - \beta \cdot V_{t'}^{i'} \right) \tag{3}$$
   (b) Reduce the utility of each target point t' in the visibility area according to
$$U_{t'} \leftarrow U_{t'} - P(||t - t'||) \tag{4}$$

To determine appropriate target points for all robots, we use an iterative approach together with a greedy strategy. In each round we compute that tuple of a robot i and a target point t, which has the best overall evaluation $U_t - \beta \cdot V_t^i$. Here $\beta \geq 0$ determines the relative importance of utility versus cost. In the decision-theoretic context the choice of β usually depends on the application. In our system, β generally was set to 1. We then recompute the utilities of all frontier cells given the new and all previous assignments according to Equation 2. This results in the algorithm shown in Table 1.

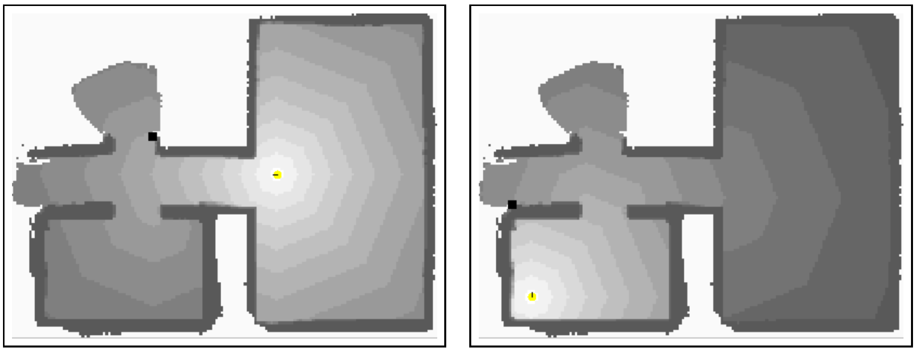

Fig. 4. Target positions obtained using the coordination approach. In this case the target point for the second robot is to the left in the corridor.

Figure 4 illustrates the effect of our coordination technique. Whereas uncoordinated robots would choose the same target position (see Figure 1), the coordinated robots select different frontier cells as the next exploration targets.

Please note that in Step 4.a the assignment is computed quite efficiently. The complexity is $O(n^2m)$ where n is the number of robots and m is the number of frontier cells. In principle, one could also try to find the optimal assignment instead. This however, introduces a problem that one has to iterate over all possible assignments of n robots to m frontier cells. In the experimental section we will also consider an approach that optimizes the assignment in such a way. Whereas this method has been found to yield slightly better results in certain environments it is much more time-consuming.

3 Collaborative Mapping with Teams of Mobile Robots

To explore their environment and to coordinated their actions, the robots need a detailed map of the environment. Furthermore, the robots must be able to build maps online, while they are in motion. The online characteristic is especially important in the context of the exploration task, since mapping is constantly interleaved with decision making as to where to move next.

To map an environment, a robot has to cope with two types of sensor noise: Noise in perception (e.g., range measurements), and noise in odometry (e.g., wheel encoders). Because of the latter, the problem of mapping creates an inherent localization problem, which is the problem of determining the location of a robot relative to its own map. The mobile robot mapping problem is therefore often referred to as the *concurrent mapping and localization problem (CML)* [36], or as the *simultaneous localization and mapping problem (SLAM)* [10, 15].

Our system applies the statistical framework presented in detail in [52] to compute consistent maps while the robots are exploring the environment. Each robot simultaneously performs two tasks: It determines a maximum likelihood estimate for its own position and a maximum likelihood estimate for the map (location of surrounding objects). To recover from possible localization errors, each robot maintains a posterior density characterizing its “true” location ([52]). The whole process is carried out in a distributed fashion. A central module receives the local maps and combines them into a single, global map which then is broadcasted to all robots. The current version of the system relies on the following two assumptions:

1. The robots must begin their operation in nearby locations, so that their range scans show substantial overlap.
2. The software must be told the approximate relative initial pose of the robots. Thereby errors up to 50 cm and 20 degrees in orientation are admissible.

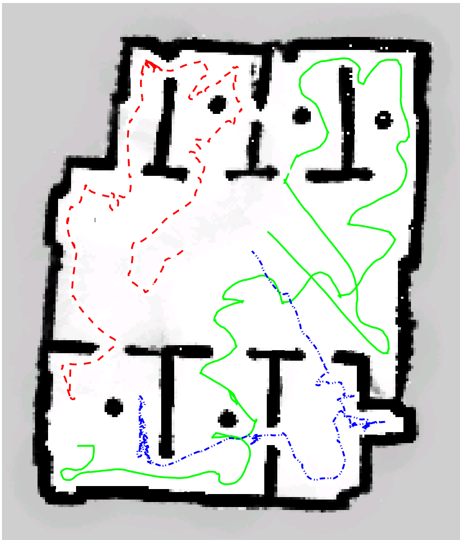

Fig. 5. Coordinated exploration by a team of three robots.

4 Experimental Results

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

4.1 Exploration with a Team of Mobile Robots

The first experiment is designed to demonstrate the capability of our approach to efficiently cover an unknown environment with a team of mobile robots. To evaluate our approach we installed three robots (two Pioneer I and one RWI B21) in an empty office environment. Figure 5 shows the map of the environment. The size of this environment is $18 \times 14m$. Also shown are the paths of the robots which started in the upper left office. As can be seen from the figure, the robots were effectively distributed over the environment. This demonstrates that our approach can effectively guide a team of mobile robots to collaboratively explore an unknown environment.

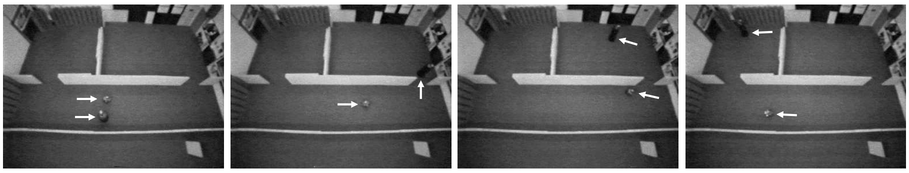

Fig. 6. Uncoordinated exploration with two robots.

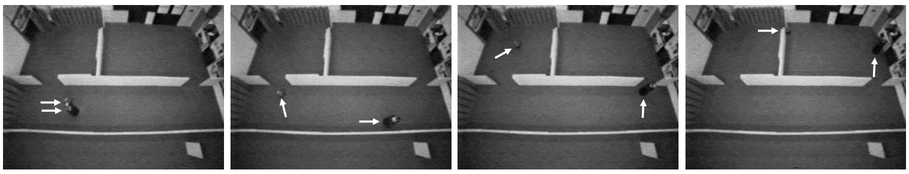

Fig. 7. Coordinated exploration by two robots.

4.2 Comparison between Greedy and Coordinated Exploration

The experiment described in this section is designed to illustrate the advantage of our coordination technique over an approach in which the robots share a map but in which there is no arbitration about target locations so that each robot approaches the closest frontier cell. Typical techniques belonging to this class are described in [53, 49]. For this experiment we used two different robots: An RWI B21 robot equipped with two laser-range scanners and a Pioneer I robot equipped with a single laser scanner. The size of the environment to be explored in this experiment was $14 \times 8m$ and the range of the laser sensors was limited to $5m$.

Figure 6 shows the typical behavior of the two robots when they explore their environment without coordination, i.e. when each robot moves to the closest unexplored location. 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 one robot enters the upper right room. At that point the other robot assigns the highest utility to the upper left room and therefore turns back. Before this robot reaches the upper left room, the B21 platform has already entered it and has completed the exploration mission. As a result, the B21 system explores the whole environment on its own and the Pioneer I robot does not contribute anything. The overall time needed to complete the exploration was 49 seconds in this case.

If, however, both robots are coordinated they perform much better (see Figure 7). As in the previous example, the B21 system moves to the end of the corridor. Since the utilities of the frontier cells in the corridor are reduced, the Pioneer I platform directly enters the upper left room. As soon as both robots have entered the rooms, the exploration mission is completed. This run lasted 35 seconds.

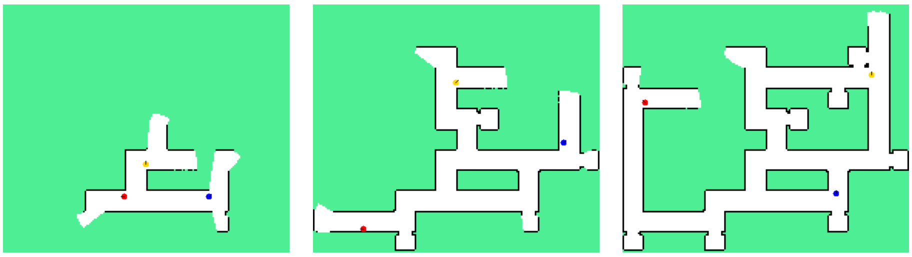

Fig. 8. Simulated exploration with three robots.

4.3 Simulation Experiments

The previous experiments demonstrate that our approach can effectively guide robots to collaboratively explore an unknown environment. To get a more quantitative assessment we performed a series of simulation experiments in using different environments. For this purpose, we developed a simulation system, that allows us to consider the effects of various parameters on the exploration performance. The simulator can handle an arbitrary number of robots. It uses a discretized representation of the state space into equally sized cells of $15 \cdot 15cm$ and 8 orientations. Additionally, it models interferences between the robots using a randomized strategy. Whenever the robots are close to each other, the system performs the planned movement with a probability of 0.7. Thus, robots that stay close to each other move slower than robots that are isolated.

Throughout these experiments we compared three different strategies. The first approach is the technique used by Yamauchi et al. [53] as well as [49], in which all robots share a joint map and greedily approach the closest unexplored part of the map. The second approach is our coordination algorithm shown in Table 1.

Additionally, we evaluated an alternative approach that seeks to optimize the assignments computed in Step 4 of our algorithm. For example, consider the situation depicted in Figure 9. Here two robots are exploring a corridor with two offices. The already explored area is depicted in grey/yellow. The assignment resulting from an application of our algorithm is depicted in the left image of this figure. Suppose both target points a and b have the same utility. Then in the first round our algorithm assigns robot 2 to a since this assignment has the least cost of all other possible assignments. Accordingly, in the second round, 1 is assigned to b.

If we assume that both robots require the same amount of time to explore a room, this assignment is clearly sub-optimal. A better assignment is shown in the right image

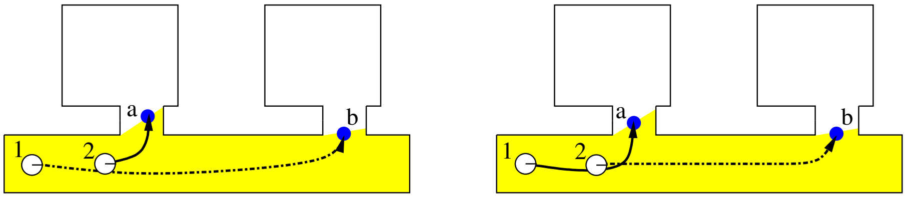

Fig. 9. The trajectories depicted in the left image that result from algorithm 1 are sub-optimal. If robot 1 moves to point a and robot 2 moves to the location b as illustrated in the right figure, the time needed to finish the exploration task is reduced, since the maximum time needed to reach the rooms is lower.

of Figure 9. By directing robot 1 to the left room and robot 2 to the right room, the whole team can finish the job earlier, because the time required to reach the rooms is reduced.

As already mentioned above, one approach to overcome this problem is to consider all possible combinations of target points and robots. Again we want to minimize the trade-off between the utility of frontier cells and the distance to be traveled. However, just adding the distances to be traveled by the two robots does not make a difference in situations like that depicted in a situation like that depicted in Figure 9. To minimize the completion time we therefore modify the evaluation function so that it considers squared distances to choose target locations $t_1, \ldots, t_n$:

$$\underset{(t_1,\ldots,t_n)}{\operatorname{argmax}} \sum_{i=1}^{n} \left[U(t_i \mid t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n) - \beta \cdot (V_{t_i}^i)^2 \right]. \tag{5}$$

The resulting algorithm that determines in every round the optimal assignment of robots to target locations according to this evaluation function is given in Table 2. Compared to our greedy selection scheme, the major problem of this approach lies in the fact that in the worst case one has to figure out $\frac{m!}{(m-n)!}$ possible assignments where m is the number of possible target locations, n is the number of robots, and $m \leq n$. Whereas this number can be handled for small numbers of robots, it becomes intractable for larger numbers, because the number of possible assignments grows exponentially in the number of robots. In practice one therefore needs appropriate search techniques to find good assignments in a reasonable amount of time. In the experiments described here, we applied a randomized search technique with hill-climbing to search for optimal assignments.

To compare these three strategies we chose a set of different environments depicted in Figure 10. For each environment and each number of robots we performed 8 different experiments. Thereby we varied over the points where the team was deployed at the beginning of each run. We then evaluated the average number of time steps the system needed to complete the job. The resulting plots are shown in Figure 11. The error bars indicate the 5% confidence level. As can be seen from the figure, the team using our algorithm significantly outperforms the uncoordinated system. It is worth noting that

Table 2. Target point selection determining th optimal assignment.

1. Determine the set of frontier cells
2. Compute for each robot i the cost V_t^i for reaching each frontier cell
3. Determine target locations $t_1, \ldots, t_n$ for the robots $i = 1, \ldots, n$ that maximizes the following evaluation function

$$\sum_{i=1}^{n} U(t_i \mid t_1, \ldots, t_{i-1}, t_{i+1}, \ldots, t_n) - \beta \cdot (V_{t_i}^i)^2 \tag{6}$$

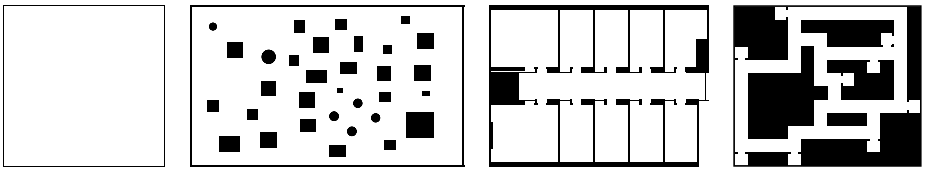

Fig. 10. Maps used for the simulation experiments. From left to right: Empty environment, unstructured environment, office environment, and corridor environment.

the optimization strategy on average is slightly better in the office environment and in the corridor environment, although the results are not significant.

These plots illustrate two advantages of our coordination approach. First, the robots are distributed over the environment so that they explore it much faster than uncoordinated robots. On the other hand, the robots are kept away from each other so that the number of interferences between them is minimized. For example, consider the results for 20 robots. In principle, coordination is less important the more robots are involved, since the probability that a certain area is explored quickly raises with the number of robots involved. However, if the robots are not distributed, the interferences between them result in longer execution times.

5 Related Work

The various aspects of the problem of exploring unknown environments with mobile robots have been studied intensively in the past. Different techniques for single robots have been presented in [32, 39, 51, 18, 23, 11, 16, 54, 50]. Whereas most of these approaches follow a greed strategy to acquire unknown terrain, they mainly differ in the way the environment is represented. Furthermore, there is a serious amount of theoretical work providing a mathematical analysis of the complexity of exploration strategies including comparisons for single robots [37, 31, 13, 14, 1, 2, 42]. Additionally [35] provides an experimental analysis of the performance of different exploration strategies for one mobile robot.

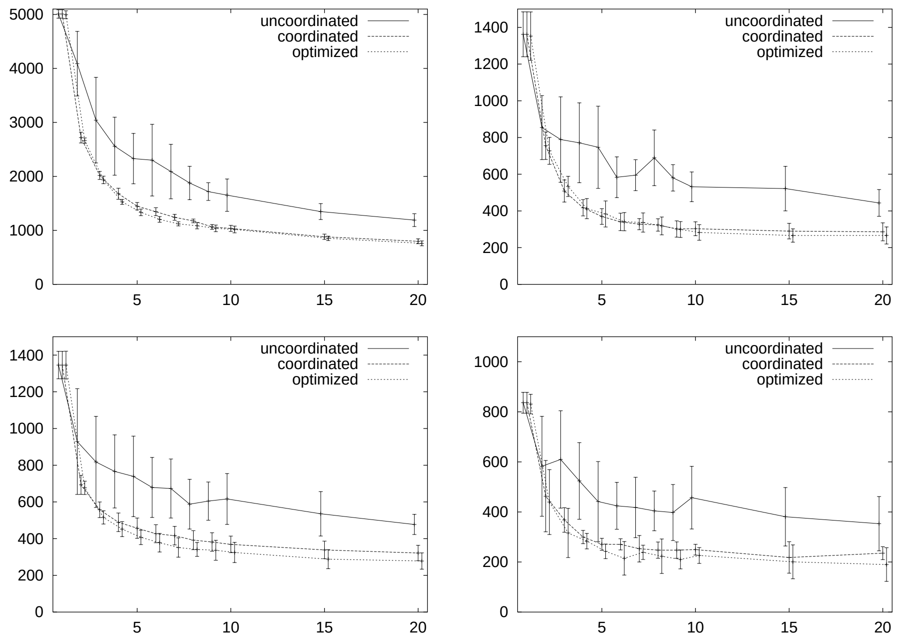

Fig. 11. Performances of the different coordination strategies for the environments shown in Figure 10: Empty environment (top left), unstructured environment (top right), office environment (lower left), and corridor environment (lower right).

Also the problem of exploring terrains with teams of mobile robots has received considerable attention in the past. For example, Rekleitis et al. [43, 44, 45] focus on the problem of reducing the odometry error during exploration. They separate the environment into stripes that 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 [34]. 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, using these strategies for multi-robot exploration one cannot expect that the exploration time is significantly reduced.

Cohen [12] considers the problem of collaborative mapping and navigation of teams of mobile robots. The team consists of a *navigator* that has to reach an initially unknown target location and a set of *cartographers* that randomly move through the environment to find the target location. When a robot discovers the goal point, the location is communicated among the cartographers to the navigation robot which then starts to move to the target location. In extensive experiments, the author analyzes the performance of this approach and compares it to the optimal solution for in different environments and different sizes of robot teams.

Koenig et al. [30] analyze different terrain coverage methods for ants which are simple robots with limited sensing and computational capabilities. They consider environments that are discretized into equally spaced cells. Instead of storing a map of the environment in their memory, the ants maintain markings in the cells they visit. The authors consider two different strategies for updating the markings. The first strategy is Learning Real-Time A* (LRTA*), which greedily and independently guides the robots to the closest unexplored areas and thus results in a similar behavior of the robots as in [53]. The second approach is Node Counting in which the ants simply count the number of times a cell was visited. The paper shows that Learning Real-Time A* (LRTA*) is guaranteed to be polynomial in the number of cells, whereas Node counting can be exponential.

Billard et al. [7] introduce a probabilistic model to simulate a team of mobile robots that explores and maps locations of objects in a circular environment. In several experiments they demonstrate the correspondence of their model with the behavior of a team of real robots.

In [4] Balch and Arkin analyze the effects of different kinds of communication on the performance of teams of mobile robots that perform tasks like searching for objects or covering a terrain. The "graze task" carried out by the team of robots corresponds to an exploration behavior. One of the results is that the communication of goal locations does not help if the robots can detect the "graze swathes" of other robots.

The technique presented in [33] is an off-line approach, which, given a map of the environment, computes a cooperative terrain sweeping technique for a team of mobile robots. In contrast to most other approaches this method is not designed to acquire a map. Rather the goal is to minimize the time required to cover a known environment which can lead to a more efficient behavior in the context of cleaning or mowing tasks.

Yamauchi et al. [53] present a technique to learn maps with a team of mobile robots. In this approach the robots exchange information about the map that is continuously updated whenever new sensor input arrives. They also use map-matching techniques [54] to improve the consistency of the resulting map. To acquire knowledge about the environment all robots follow a greedy strategy and move to the closest frontier cell. They are not applying any strategies to distribute the robots over the environment or to avoid that two or more robots explore the same areas.

One approach towards cooperation between robots has been presented by Singh and Fujimura [49]. This approach especially addresses the problem of heterogenous robot systems. During exploration each robots identifies "tunnels" to the so far unexplored area. If a robot is too big to pass through a tunnel it informs other robots about this tunnel. Whenever a robot receives such a message it either accepts this new task or further delegates it to smaller robots. In the case of homogeneous robots, the robots perform a greedy strategy similar to the system of Yamauchi et al. [53].

Furthermore, there has been several work focusing on the coordination of two robots. The work presented by Bender and Slonim [6] theoretically analyzes the complexity of exploring strongly-connected directed graphs with two robots. Roy and Dudek [46] focus on the problem of exploring unknown environments with two robots. Specifically, this paper presents an approach allowing the robots with a limited communication range to schedule rendezvous. The algorithms are analyzed analytically as well as empirically using real robots.

Finally, several researchers have focused on architectures for multi-robot cooperation. For example, Grabowski et al. [24] consider teams of miniature robots that overcome the limitations imposed by their small scale by exchanging mapping and sensor information. In this architecture, a team leader integrates the information gathered by the other robots. Furthermore, it directs the other robots to move around obstacles or to direct them to unknown areas. Jung and Zelinsky [29] present a distributed action selection scheme for behavior-based agents which has successfully been applied to a cleaning task. Matarić and Sukhatme [38] consider different strategies for task allocation in robot teams and analyze the performance of the team in extensive experiments.

In contrast to all approaches discussed above, the technique presented in this paper explicitely coordinates the actions of the robots so that they are distributed over the environment while they are exploring the environment. Accordingly the time needed to complete the exploration task is significantly reduced.

6 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 to simultaneously take into account the cost of reaching a so far unexplored location and its utility. Thereby, the utility of a target location depends on the probability that this location is visible from target locations assigned to other robots. Our algorithm always assigns that target location to a robot which has the best trade-off between utility and costs. Our method differs from previous techniques in an explicit coordination mechanism that assigns different target locations to the robots. Some of the 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 from selecting the same target location.

Our technique has been implemented and tested on real robots and in extensive simulation runs. Experiments presented in this paper demonstrate that our algorithm is able to effectively coordinate a team of robots during exploration. They further demonstrate that our coordination technique outperforms other methods developed so far.

Despite these encouraging results, there are several aspects which could be improved. One 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. A further possible research direction is towards systems with a limited communication range. In this case the systems also have to plan rendezvous to exchange information and to avoid redundant operations.

Acknowledgments

This work is sponsored in part by the EC (contract number IST-2000-29456). The authors would furthermore like to thank Dieter Fox, Reid Simmons, and Sebastian Thrun for fruitful discussions.

References

[1] S. Albers and M.R. Henzinger. Exploring unknown environments. *SIAM Journal on Computing*, 29:1164–1188, 2000.

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

[3] D. Apostolopoulos, L. Pedersen, B. Shamah, K. Shillcutt, M.D. Wagner, and W.R.L. Whittaker. Robotic antarctic meteorite search: Outcomes. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 4174–4179, 2001.

[4] T. Balch and R.C. Arkin. Communication in reactive multiagent robotic systems. *Journal of Autonomous Robots*, 1(1):27–52, 1994.

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

[6] M. Bender and D. Slonim. The power of team exploration: two robots can learn unlabeled directed graphs. In *Proc. of the 35rd Annual Symposium on Foundations of Computer Science*, pages 75–85, 1994.

[7] A. Billard, A.J. Ijspeert, and A. Martinoli. A multi-robot system for adaptive exploration of a fast changing environment: probabilistic modelling and experimental study. *Connection Science*, 11(3/4):357–377, 2000.

[8] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000.

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

[10] J.A. Castellanos, J.M.M. Montiel, J. Neira, and J.D. Tardós. The SPmap: A probabilistic framework for simultaneous localization and map building. *IEEE Transactions on Robotics and Automation*, 15(5):948–953, 1999.

[11] H. Choset. Topological simultaneous localization and mapping (slam): Toward exact localization without explicit localization. *IEEE Transactions on Robotics and Automation*, 17(2), April 2001.

[12] W. Cohen. Adaptive mapping and navigation by teams of simple robots. *Journal of Robotics & Autonomous Systems*, 18:411–434, 1996.

[13] X. Deng, T. Kameda, and C. Papadimitriou. How to learn in an unknown environment. In *Proc. of the 32nd Symposium on the Foundations of Comp. Sci.*, pages 298–303. IEEE Computer Society Press, Los Alamitos, CA, 1991.

[14] X. Deng and C. Papadimitriou. How to learn in an unknown environment: The rectilinear case. *Journal of the ACM*, 45(2):215–245, 1998.

[15] G. Dissanayake, H. Durrant-Whyte, and T. Bailey. A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem. Working notes of ICRA'2000 Workshop W4: Mobile Robot Navigation and Mapping, April 2000.

[16] G. Dudek, M. Jenkin, E. Milios, and D. Wilkes. Robotic exploration as graph construction. *IEEE Transactions on Robotics and Automation*, 7(6):859–865, 1991.

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

[18] T. Edlinger and E. von Puttkamer. Exploration of an indoor-environment by an autonomous mobile robot. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1994.

[19] H. Endres, W. Feiten, and G. Lawitzky. Field test of a navigation system: Autonomous cleaning in supermarkets. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 1998.

[20] M. Fontan and M. Mataric. Territorial multi-robot task division. *IEEE Transactions on Robotics and Automation*, 14(5), 1998.

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

[22] D. Goldberg and M. Mataric. Interference as a tool for designing and evaluating multi-robot controllers. *Journal of Robotics & Autonomous Systems*, 8:637–642, 1997.

[23] H.H. González-Baños, E. Mao, J.C. Latombe, T.M. Murali, and A. Efrat. Planning robot motion strategies for efficient model construction. In *Proc. Intl. Symp. on Robotics Research (ISRR)*, 1999.

[24] R. Grabowski, L.E. Navarro-Serment, C.J.J. Paredis, and P.K. Khosla. Heterogeneous teams of modular robots for mapping and exploration. *Journal of Autonomous Robots*, 8(3):293–308, 2000.

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

[26] D.F. Hougen, S. Benjaafar, J.C. Bonney, J.R. Budenske, M. Dvorak, M. Gini, H. French, D.G. Krantz, P.Y. Li, F. Malver, B. Nelson, N. Papanikolopoulos, P.E. Rybski, S.A. Stoeter, R. Voyles, and K.B. Yesin. A miniature robotic system for reconnaissance and surveillance. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000.

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

[28] Y. Huang, Z.Cao, S. Oh, E. Kattan, and E. Hall. Automatic operation for a robot lawn mower. In *SPIE Conference on Mobile Robots*, volume 727, pages 344–354, 1986.

[29] D. Jung and A. Zelinksy. An architecture for distributed cooperative planning in a behaviour-based multi-robot system. *Journal of Robotics & Autonomous Systems*, 26(2-3):149–174, 1999.

[30] S. Koenig, B. Szymanski, and Y. Liu. Efficient and inefficient ant coverage methods. *Annals of Mathematics and Artificial Intelligence*, In Print.

[31] S. Koenig, C. Tovey, and W. Halliburton. Greedy mapping of terrain. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001.

[32] B. Kuipers and Y.-T. Byun. A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations. *Journal of Robotics & Autonomous Systems*, 8:47–63, 1991.

[33] D. Kurabayashi, J. Ota, T. Arai, and E. Yoshida. Cooperative sweeping by multiple mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 1744–1749, 1996.

[34] 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.

[35] D. Lee and M. Recce. Quantitative evaluation of the exploration strategies of a mobile robot. *International Journal of Robotics Research*, 16(4):413–447, 1997.

[36] J.J. Leonard and H.J.S. Feder. A computationally efficient method for large-scale concurrent mapping and localization. In J. Hollerbach and D. Koditschek, editors, *Proceedings of the Ninth International Symposium on Robotics Research*, 1999.

[37] S. Lumelsky, S. Mukhopadhyay, and K. Sun. Dynamic path planning in sensor-based terrain acquisition. *IEEE Transactions on Robotics and Automation*, 6(4):462–472, 1990.

[38] M.J. Matarić and G. Sukhatme. Task-allocation and coordination of multiple robots for planetary exploration. In *Proc. of the International Conference on Advanced Robotics*, 2001.

[39] S.J. Moorehead, R. Simmons, and W.L. Whittaker. Autonomous exploration using multiple sources of information. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2001.

[40] M. Moors. Koordinierte Multi-Robot Exploration. Master's thesis, Department of Computer Science, University of Bonn, 2000. In German.

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

[42] N. Rao, S. Hareti, W. Shi, and S. Iyengar. Robot navigation in unknown terrains: Introductory survey of non-heuristic algorithms. Technical Report ORNL/TM-12410, Oak Ridge National Laboratory, 1993.

[43] I. Rekleitis, G. Dudek, and E. Milios. Multi-robot exploration of an unknown environment, efficiently reducing the odometry error. In *Proc. of International Joint Conference in Artificial Intelligence (IJCAI)*, 1997.

[44] I. Rekleitis, G. Dudek, and E. Milios. Accurate mapping of an unknown world and online landmark positioning. In *Proc. of Vision Interface (VI)*, 1998.

[45] I. Rekleitis, R. Sim, G. Dudek, and E. Milios. Collaborative exploration for the construction of visual maps. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2001.

[46] N. Roy and G. Dudek. Collaborative robot exploration and rendezvous: Algorithms, performance bounds and observations. *Journal of Autonomous Robots*, 11(2):117–136, 2001.

[47] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proc. of the National Conference on Artificial Intelligence (AAAI)*, 2000.

[48] M. Simoncelli, G. Zunino, H.I. Christensen, and K. Lange. Autonomous pool cleaning: Self localization and autonomous navigation for cleaning. *Journal of Autonomous Robots*, 9(3):261–270, 2000.

[49] K. Singh and K. Fujimura. Map making by cooperating mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 254–259, 1993.

[50] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 3310–3317, 1994.

[51] C.J. Taylor and D.J. Kriegman. Exloration strategies for mobile robots. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, pages 248–253, 1993.

[52] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. *International Journal of Robotics Research*, 20(5):335–363, 2001.

[53] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the Second International Conference on Autonomous Agents*, pages 47–53, 1998.

[54] B. Yamauchi, A. Schultz, and W. Adams. Integrating exploration and localization for mobile robots. *Adaptive Systems*, 7(2), 1999.