

Coordinated Multi-Robot Exploration

Wolfram Burgard, *Member, IEEE*, Mark Moors, Cyrill Stachniss, *Student Member, IEEE*, and Frank E. Schneider

Abstract—In this paper, we consider the problem of exploring an unknown environment with 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 simultaneously takes into account the cost of reaching a target point and its utility. Whenever a target point is assigned to a specific robot, the utility of the unexplored area visible from this target position is reduced. In this way, different target locations are assigned to the individual robots. We furthermore describe how our algorithm can be extended to situations in which the communication range of the robots is limited. Our technique has been implemented and tested extensively in real-world experiments and simulation runs. The results demonstrate that our technique effectively distributes the robots over the environment and allows them to quickly accomplish their mission.

Index Terms—Coordinated behavior, limited communication, mobile robotics, multi-robot exploration.

I. INTRODUCTION

THE PROBLEM of exploring an environment belongs to the fundamental problems in mobile robotics. There are several applications like planetary exploration [3], reconnaissance [27], rescue [47], [62], mowing [30], or cleaning [19], [31], [55] in which the complete coverage of a terrain belongs to the integral parts 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 [26]. 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], [23]. 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. Also, 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 so as to efficiently explore their environment. Our method, which has originally been presented in [45] and has been integrated into two different systems [8], [54], follows a decision-theoretic approach to explicitly coordinate the robots. It does so by maximizing the overall utility and 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 toward this area, coordination is achieved in a very elegant way. In practice, one also has to deal with a limited communication range that restricts the communication abilities of the vehicles. Naturally, the task of exploring a terrain with limited communication range is harder than without this constraint. If the distance between the robots becomes too large to be bridged by the wireless network or if a temporal network error occurs, robots may explore an area another robot has already explored before, which can lead to a suboptimal behavior. In this paper, we also describe an extension of our algorithm to robot teams with a limited communication range.

Our technique has been implemented on teams of heterogeneous robots and has been proven effectively in 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 compared to an approach which lacks our centralized coordination. Furthermore, we describe experiments in which we analyze our algorithm in the context of teams of mobile robots with a limited communication range.

This paper is organized as follows. In Section II, we present our decision-theoretic approach to coordinated exploration with mobile robots. In Section III, we briefly describe the technique used by our system to acquire and communicate maps of the environment. Section IV presents series of experiments carried out with real robot systems and in simulation. Finally, we discuss related work in Section V.

II. 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 environ-

Manuscript received June 6, 2003; revised May 14, 2004. This paper was recommended for publication by Associate Editor L. Parker and Editor S. Hutchinson upon evaluation of the reviewers' comments. This work was supported in part by the German Science Foundation (DFG) under Contract SFB/TR8, project A3, and by the EC under Contract FP6-004250-CoSy.

W. Burgard and C. Stachniss are with the Department of Computer Science, University of Freiburg, 79110 Freiburg, Germany (e-mail: burgard@informatik.uni-freiburg.de).

M. Moors and F. E. Schneider are with the Department of Computer Science, University of Bonn, 53117 Bonn, Germany.

Digital Object Identifier 10.1109/TRO.2004.839232

ment 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 first 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 can be communicated between the robots. We will focus on the question of how to coordinate the robots in order to efficiently cover the environment. At the end of this section, we will consider the situation in which the robots have a limited communication range. The mapping system will briefly be described in Section III.

Our system uses occupancy grid maps [46], [61] to represent the environment. Each cell of such an occupancy grid map contains a numerical value representing the posterior 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 it has been intercepted by a sensor beam. At this point, we would like to mention that the approach presented here is not restricted to occupancy maps. The only requirement is that the underlying representation of the environment must allow the distinction between known and unknown areas and to compute travel costs for the individual robots. Therefore, our algorithm can also be applied to alternative representations like topological maps [11] or coverage maps [57].

When exploring an unknown environment we are especially interested in “frontier cells” [63]. 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 of how to assign exploration tasks represented by frontier cells to the individual robots. If multiple robots are involved, we want to avoid several of them moving to the same location. To deal with these problems and to determine appropriate target locations for the individual robots our system uses a decision-theoretic approach. We simultaneously consider the cost of reaching a frontier cell and the utility of that cell. For each robot, the cost of a cell is 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 subsections, 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 frontier cells to robots.

A. 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 the *value iteration*, a popular dynamic programming algorithm [5], [29]. In the following, a tuple (x, y) corresponds to the x th cell in the direction of the x -axis and the y th cell in direction of the y -axis of the two-dimensional occupancy grid map. In our approach,

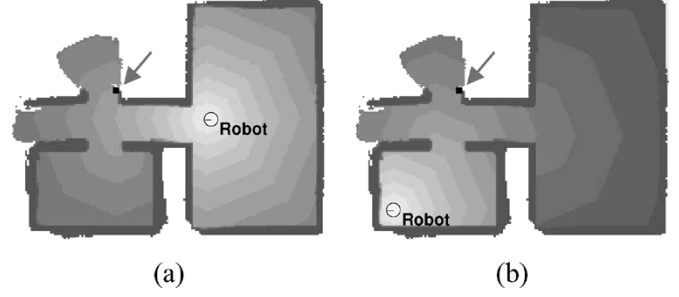


Fig. 1. Typical value functions obtained for two different robot positions. The black rectangle indicates the target points in the unknown area with minimum cost.

the cost for traversing a grid cell (x, y) 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 ∞ :

$$V_{x,y} \leftarrow \begin{cases} 0, & \text{if } (x, y) \text{ is the robot position} \\ \infty, & \text{otherwise.} \end{cases}$$

- 2) **Update loop.** For all grid cells (x, y) do:

$$V_{x,y} \leftarrow \min \left\{ V_{x+\Delta x, y+\Delta y} + \sqrt{\Delta x^2 + \Delta y^2} \cdot P(occ_{x+\Delta x, y+\Delta y}) \mid \Delta x, \Delta y \in \{-1, 0, 1\} \wedge P(occ_{x+\Delta x, y+\Delta y}) \in [0, occ_{\max}] \right\}$$

where occ_{\max} is the maximum occupancy probability value of a grid cell the robot is allowed to traverse. 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 $P(occ_{x,y})$ that a grid cell (x, y) is occupied times the distance to the cell. The update rule is repeated until convergence. Then each value $V_{x,y}$ corresponds to the *cumulative cost* of moving from the current position of the robot to (x, y) . The convergence of the algorithm is guaranteed as long as the cost for traversing a cell is not negative and the environment is bounded. Both criteria are fulfilled in our approach. 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 (x, y) . This is done by steepest descent in V , starting at (x, y) .

Fig. 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. 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, which seriously speeds up the computation. To incorporate the uncertainty of the robots' motions into the process 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 nondeterministic 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. Note that the maps depicted in Fig. 1 have not been smoothed to allow the reader to distinguish between walls (dark grey) and the values of the final value function (light grey).

B. 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 moves 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 the sensors of a robot typically cover a certain region around a particular 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 and visibility to cells that are assigned to other 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 . One can estimate $P(d)$ by maintaining a posterior about the estimated distances to be measured. While the robot moves through the environment, this posterior is updated.

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 $U(t_n|t_1, \dots, t_{n-1})$ of a frontier cell t_n given that the cells t_1, \dots, t_{n-1} have already been assigned to the robots $1, \dots, n-1$ as

$$U(t_n|t_1, \dots, t_{n-1}) = U_{t_n} - \sum_{i=1}^{n-1} P(\|t_n - t_i\|). \quad (1)$$

According to (1), the more robots move to a location from where t_n is likely to be visible, the lower is the utility of t_n . Note that we also take into account whether there is an obstacle between two frontier cells t and t' . This is achieved by a ray-casting operation on the grid map. If there is an obstacle between two frontier cells t and t' , we set $P(\|t - t'\|)$ to zero.

In extensive experiments we could not find a significant difference in the resulting exploration time depending on in which environment the posterior $P(d)$ has been learned. We therefore use the following approximation:

$$P(d) = \begin{cases} 1.0 - \frac{d}{\text{max_range}}, & \text{if } d < \text{max_range} \\ 0, & \text{otherwise} \end{cases} \quad (2)$$

where max_range is the maximum range reading provided by the range sensor.

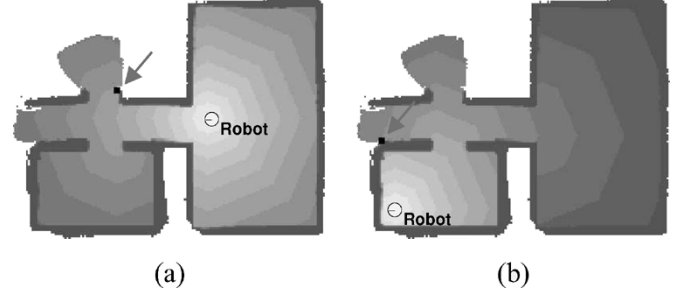


Fig. 2. Target positions obtained using the coordination approach. In this case the target point for the second robot is to the left in the corridor.

C. Target Point Selection

To compute appropriate target points for the individual robots we need to consider for each robot the cost of moving to a location and 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 .

Algorithm 1: Goal Assignment for Coordinated Multi-Robot Exploration.

- 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 **do**
- 5) Determine a robot i and a frontier cell t which satisfy: $(i, t) = \arg \max_{(i', t')} (U_{t'} - \beta \cdot V_{t'}^{i'})$.
- 6) Reduce the utility of each target point t' in the visibility area according to $U_{t'} \leftarrow U_{t'} - P(\|t - t'\|)$.
- 7) **end while**

To determine appropriate target points for all robots, we use an iterative approach. In each round, we compute that tuple (i, t) where i is the number of a robot and t is a frontier cell, which has the best overall evaluation $U_t - \beta \cdot V_t^i$. We then recompute the utilities of all frontier cells given the new and all previous assignments according to (1). Finally, we repeat this process for the remaining robots. This results in Algorithm 1. The complexity of this algorithm is $O(n^2T)$ where n is the number of robots and T is the number of frontier cells.

The quantity $\beta \geq 0$ determines the relative importance of utility versus cost. Experiments showed that the exploration time stays nearly constant if $\beta \in [0.01, 50]$. For bigger values of β the exploration time increases because the impact of the coordination is decreased. If β is close to 0 the robots ignore the distance to be traveled which also leads to an increased exploration time. Therefore, β generally is set to 1 in our current implementation.

Fig. 2 illustrates the effect of our coordination technique. Whereas uncoordinated robots would choose the same target position (see Fig. 1), the coordinated robots select different frontier cells as the next exploration targets. When coordinating a team of robots during exploration one question is when to re-compute the target locations. In the case of unlimited communication, we compute new assignments whenever one robot has reached its designated target location or whenever the distance traveled by

the robots or the time elapsed after computing the latest assignment exceeds a given threshold.

D. Coordination With Limited Communication Range

In practice, one cannot assume that the robots can exchange information at any point in time. For example, the limited range of nowadays wireless networks can prevent robots from being able to communicate with other robots at a certain point in time. If the distance between the robots becomes too large so that not all robots can communicate with each other, a centralized approach as described above can no longer be applied. However, our algorithm can easily be adapted to cope with a limited communication range. In our system, we apply our approach to each sub-team of robots which are able to communicate with each other. Obviously, this can, at least in the worst case, lead to a situation in which all robots individually explore the whole environment. In practical experiments, however, we found that this approach still results in a quite efficient exploration process, since the robots can quickly exchange necessary information and coordinate with each other again as soon a connection between them has been reestablished.

In the case of limited communication, we apply a slightly different strategy to determine when to compute new assignments. In our experiments, we found that the risk of redundant work is increased if the robots forget about the assignments of other robots as soon as the communication breaks down. Instead, if each robot stores the latest target locations assigned to other robots the overall performance is increased especially in situations in which the communication range has been exceeded, since the robots avoid going to places already explored by other robots. This approach turned out to be useful especially in the context of small robot teams.

III. COLLABORATIVE MAPPING WITH TEAMS OF MOBILE ROBOTS

To explore their environment and to coordinate 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)* [41] or as the *simultaneous localization and mapping problem (SLAM)* [10], [15].

Our system applies the statistical framework presented in detail in [61] to compute consistent maps while the robots are exploring the environment. Each robot starts with a blank grid map. During exploration each robot simultaneously performs two tasks: It determines a maximum likelihood estimate for its own position and a maximum likelihood estimate for the map

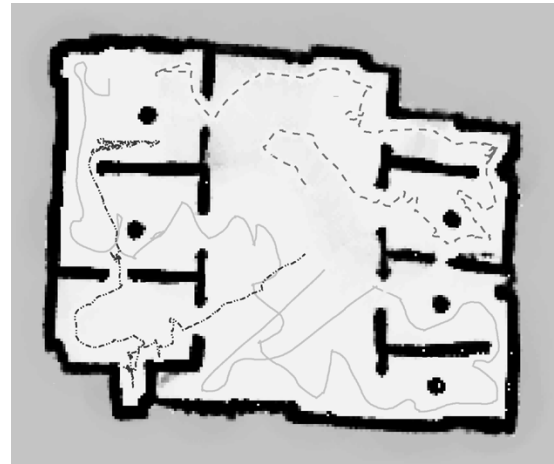


Fig. 3. Coordinated exploration by a team of three robots with unlimited communication abilities in a real-world experiment.

(location of surrounding objects). To recover from possible localization errors, each robot maintains a posterior density characterizing its “true” location (see [61]). 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.

To achieve the coordination, the team must be able to communicate the maps of the individual robots during exploration. In our current system, we assume that the robots set up an ad hoc network which forms clusters. The messages sent by a robot are forwarded to all team-mates within the corresponding cluster. Whenever two clusters are merged, care has to be taken to avoid that robots become overly confident in the state of the environment. Suppose that each cluster maintains an occupancy grid map built from all observations made by the robots of that team. As an example, let us assume that two robots that share a map m leave their communication range. As long as they explore the environment individually they update their maps and obtain two different maps m_1 and m_2 . Now suppose the robots can communicate again and exchange their maps. If they use the recursive update rule for occupancy grids to combine m_1 and m_2 the information originally contained in m is integrated twice in the resulting map, which is not admissible.

There are several ways to avoid the multiple use of sensor information. One solution is to prevent the robots from exchanging information more than once [22], which reduces the benefit of a multi-robot system. An alternative solution is that each robot maintains an individual map for each other robot. These maps, which can be combined to a joint map, can be updated separately. In our current system, we apply a different approach that we found to be less memory intensive. Furthermore, it reduces the communication overhead. In this approach, each robot stores for each other robot a log of sensor measurements perceived by this robot. A robot only transfers those measurements that have not been transmitted to the corresponding robot so far. Additionally, the robots maintain a small data structure containing the time stamp of the latest

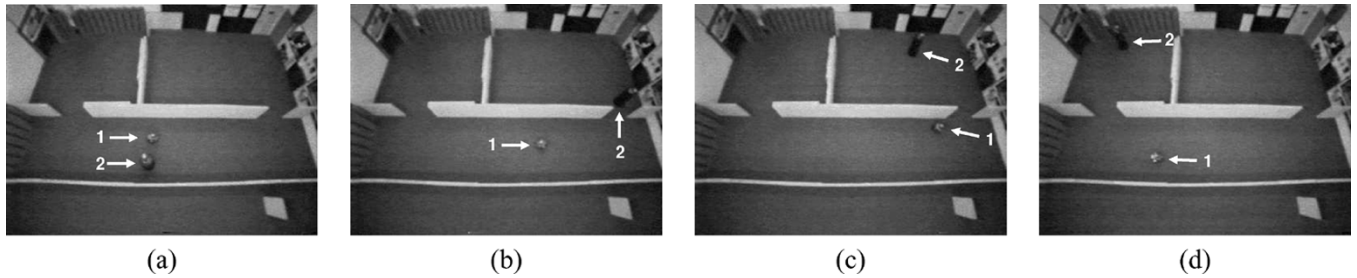


Fig. 4. Uncoordinated exploration with two robots. In (a) and (b), both robots drive along the corridor, but robot 1 is slower than robot 2. In (c), robot 1 reached the end of the corridor, but robot 2 already has explored the right room. Due to the convolution of the maps, the path from robot 1 to the left room through the corridor has lower cost (one doorway) compared to the path through the right room (two doorways). Therefore, robot 1 turns around and follows the corridor. In (d), robot 2 has entered the left room from the right-hand side and explored it.

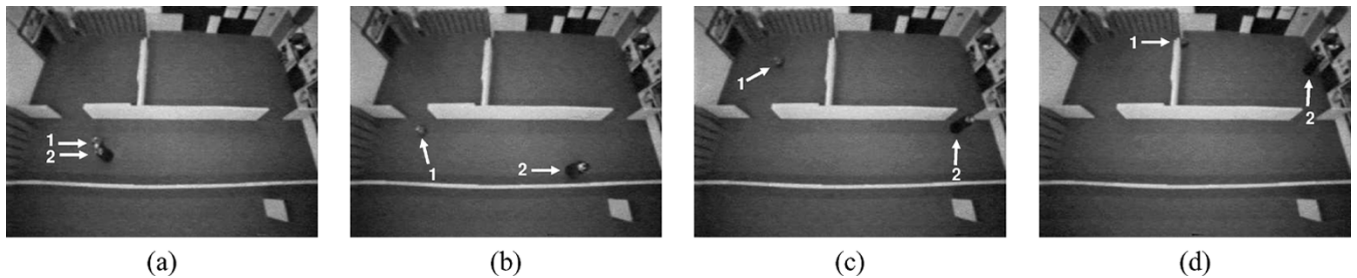


Fig. 5. Coordinated exploration by two robots. In (b), both robots focus on different frontiers due to the coordination strategy. Therefore, robot 1 explores the left room and robot 2 the right one. This leads to a better performance compared to the uncoordinated behavior.

sensor measurement of a robot that was transmitted to all other robots. This allows the robots to discard those measurements which have been received by all other robots already.

IV. EXPERIMENTAL RESULTS

The approach described has been implemented and extensively tested on real robots and in different environments. Additionally, we performed a series of simulation experiments.

A. 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 iRobot B21) in an empty laboratory environment. Fig. 3 shows the map of this environment. The size of the environment is $18 \text{ m} \times 14 \text{ m}$. Also shown are the paths of the robots which started in the upper left room. 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.

B. Comparison Between Uncoordinated and Coordinated Exploration

The goal of the second experiment described here is 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. For this experiment we used two different robots: An iRobot B21 robot equipped with two laser-range scanners covering a 360° field of view (robot 1) and

a Pioneer I robot equipped with a single laser scanner covering a 180° field of view (robot 2). The size of the environment to be explored in this experiment was $14 \text{ m} \times 8 \text{ m}$ and the range of the laser sensors was limited to 5 m.

Fig. 4 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, robot 2 enters the upper right room. At that point, robot 1 assigns the highest utility to the upper left room and therefore turns back. Before robot 1 reaches the upper left room, robot 2 has already entered it and has completed the exploration mission. As a result, robot 2 explores the whole environment on its own while robot 1 does not contribute anything. The overall time needed to complete the exploration was 49 s in this case.

However, if both robots are coordinated, they perform much better (see Fig. 5). As in the previous example, robot 2 moves to the end of the corridor. Since the utilities of the frontier cells in the corridor are reduced, robot 1 directly enters the upper left room. As soon as both robots have entered the rooms, the exploration mission is completed. This run lasted 35 s.

C. 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 different environments.

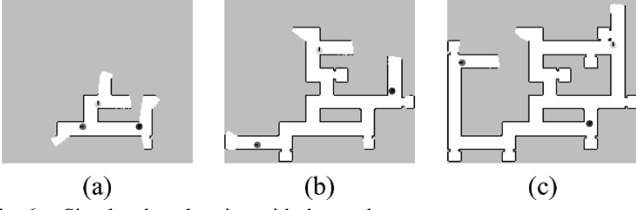
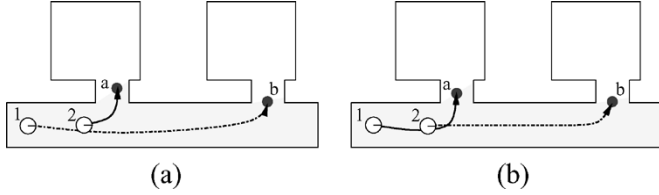


Fig. 6. Simulated exploration with three robots.

Fig. 7. Situation in which the assignments resulting from the algorithm given in Algorithm 1 are sub-optimal (a). If robot 1 moves to point *a* and robot 2 moves to the location *b* as illustrated in (b), the time needed to finish the exploration task is reduced, since the maximum time needed to reach the rooms is lower.

To carry out these experiments, 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 cm × 15 cm and 8 orientations. Additionally, it models interferences between the robots. Whenever 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. This approach is designed to model cross-talk between active sensors such as ultrasound devices as well as time delays introduced by necessary collision avoidance maneuvers.

Screen shots of this simulation system during a run in which three robots explore the environment are shown in Fig. 6. The simulator also allows the specification of different properties of the robot systems and sensors. To carry out these experiments, we used sensors with a 360° field of view as is the case, for example, for robots equipped with two laser range sensors or with a ring of ultrasound sensors. Note that our approach does not require a 360° field of view. In the past, we successfully applied our approach even to robots with a limited field of view, equipped only with a single laser scanner [8], [54].

Throughout these experiments, we compared three different strategies. The first approach is the technique used by Yamauchi *et al.* [63] as well as Singh and Fujimura [56], in which each robot always approaches the closest unexplored area of a joint map. In the sequel, this approach will be denoted as uncoordinated exploration since it lacks a component that arbitrates between the robots whenever they choose the same frontier cells. The second approach is our coordination approach specified by Algorithm 1. Additionally, we evaluated an alternative approach that seeks to optimize the assignments computed in lines 4–7 of our algorithm. For example, consider the situation depicted in Fig. 7. Here two robots are exploring a corridor with two rooms. The already explored area is depicted in grey/yellow. Suppose both target points *a* and *b* have the same utility. In the first round of the iteration (see while loop in Algorithm 1), our algorithm assigns robot 2 to *a* since this assignment has the least cost of all

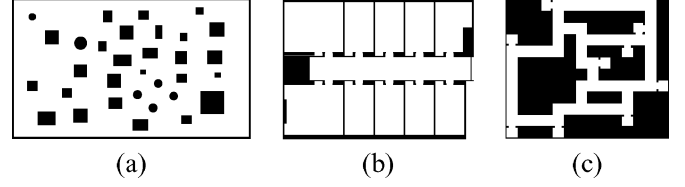


Fig. 8. Maps used for the simulation experiments: (a) unstructured, (b) office, and (c) corridor environment.

other possible assignments. Accordingly, in the second round, robot 1 is assigned to *b*. The resulting assignments are depicted in Fig. 7(a). 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 Fig. 7(b). 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.

One approach to overcome this problem is to consider all possible combinations of target points and robots. Again, we want to minimize the tradeoff 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 Fig. 7. Since the robots execute their actions in parallel the time to complete the whole task depends on the longest trajectory. To minimize the completion time (by choosing more balanced trajectories for the individual robots), we therefore modify the evaluation function so that it considers squared distances to choose target locations t_1, \dots, t_n :

$$\arg \max_{(t_1, \dots, t_n)} \sum_{i=1}^n \left[U(t_i | t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n) - \beta \cdot (V_{t_i}^i)^2 \right].$$

Algorithm 2: Goal selection determining the best assignment over all permutations.

- 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, \dots, t_n for the robots $i = 1, \dots, n$ that maximizes the following evaluation function: $\sum_{i=1}^n U(t_i | t_1, \dots, t_{i-1}, t_{i+1}, \dots, t_n) - \beta \cdot (V_{t_i}^i)^2$.

The resulting algorithm that determines in every round the optimal assignment of robots to target locations according to this evaluation function is given in Algorithm 2. Compared to the selection scheme of our algorithm, the major problem of this approach lies in the fact that one has to figure out $(T!/(T-n)!)$ possible assignments in the worst case where *T* is the number of possible target locations, *n* is the number of robots, and $n \leq T$. Whereas this number can be handled for small teams of robots, it becomes intractable for larger teams, 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 combined with hill-climbing to search for optimal assignments of frontiers to robots.

To compare these three strategies, we chose a set of different environments depicted in Fig. 8. For each environment and each

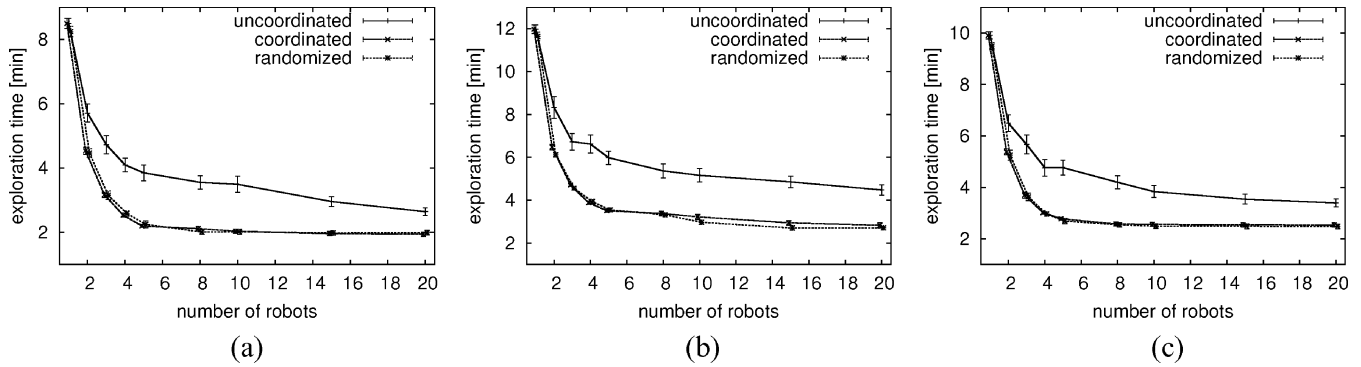


Fig. 9. Performances of the different coordination strategies for the environments shown in Fig. 8: (a) unstructured environment, (b) office environment, and (c) corridor environment.

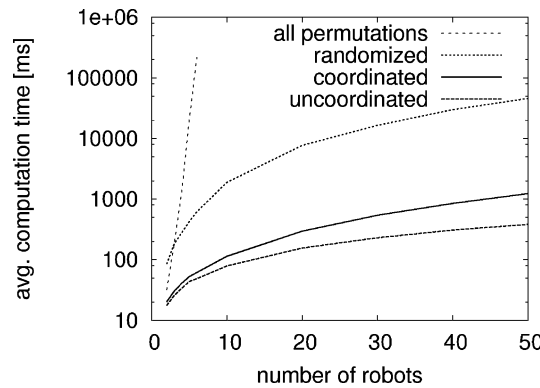


Fig. 10. Time required on a Pentium 4 2.8-GHz machine to compute the assignment of target locations to robots for three different strategies.

number of robots we performed 45 different simulation experiments for each strategy. In each comparison of the three strategies, the robot team was started at the same randomly chosen location. We then evaluated the average number of time steps the system needed to complete the job. The resulting plots are shown in Fig. 9. 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 with respect to the exploration time. This is mainly due to the fact that Algorithm 1 provides a better distribution of the robots over the environment. We repeated the experiments without modeling the interference between the robots. The results of these experiments were quite similar and revealed a similar relative improvement of our algorithm compared to the uncoordinated approach.

It is worth noting that the randomized optimization strategy usually yields slightly better results than our coordination technique although the improvement is not significant. Thus, the transition from our algorithm, which has complexity $O(n^2T)$, to a complex search that seeks to determine the optimal assignment from all $(T!/(T-n)!)$ permutations appears to yield only slight improvements. Given the computational overhead introduced by the randomized search in the space of all permutations (see Fig. 10), especially for large teams of robots Algorithm 1 appears to be preferable over Algorithm 2.

D. Exploration Under Limited Communication

The final experiments are designed to analyze the performance of our coordination strategy if the robots only have a limited communication range. As explained above, if the commu-

nication range is limited the robots cannot globally coordinate their actions anymore. As a result, different robots may explore the same regions which reduces the overall efficiency.

The first experiment described in this section was carried out with three robots in our laboratory environment. Throughout this experiment, we limited the communication range to 5 m. Fig. 11 depicts the exploration process. Each row shows the maps of the individual robots at different points in time. The initial situation is depicted in the first row. The communication ranges of the robots are highlighted by colored/grey disks around each robot. As can be seen from the second row, the robots were quickly split up in this experiment and had to plan their trajectories individually. In row three, the robots R_1 and R_3 are able to communicate again and therefore can exchange their maps and coordinate their behavior again. Robot R_2 , however, still acts independently of the other two robots. In row five, R_1 and R_3 again leave their communication range, whereas R_2 and R_3 can merge their maps and approach the last unexplored area in the top left corner. In the last row, the robots R_2 and R_3 complete the exploration task.

To analyze the influence of the communication range, we performed a large series of simulation experiments. For different numbers of robots (1–5) and seven different communication ranges, we carried out 45 simulation runs. In each run, we chose a random starting point for the robot team. We regard the exploration task as completed as soon as the known area in the map of one robot covers the whole environment. The results are depicted in Fig. 12. The x -axis shows the communication range of the robots in relation to the maximum distance in the map, whereas the y -axis depicts the average exploration time. If the communication range is close to zero the coordinated and uncoordinated strategies behave similar, because all robots act independently most of the time. As the communication range increases, the benefit of the coordinated approach improves. An interesting result of this experiment is that a communication range of 30% of the diameter of the environment appears to be sufficient to yield the same performance as with unlimited communication.

V. RELATED WORK

The various aspects of the problem of exploring unknown environments with mobile robots have been studied intensively in the past. Many approaches have been proposed for exploring

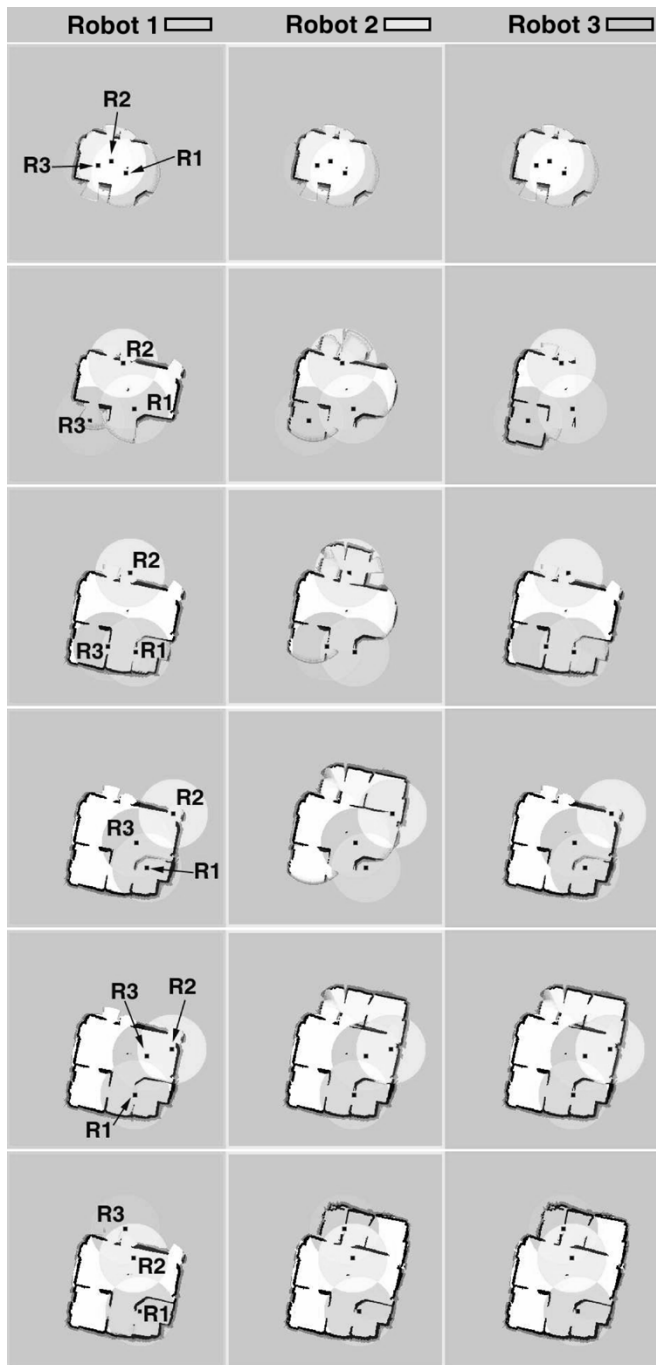


Fig. 11. Coordinated exploration by a team of three robots with limited communication abilities. Each column shows the evolution of the map of one robot over time.

unknown environments with single robots [11], [16], [18], [24], [37], [44], [58], [60], [64], [65]. Most of these approaches guide the robot to the closest unexplored area, just as our approach does when applied to a single robot system. These techniques mainly differ in the way the environment is represented. Popular representations are topological [11], [37], metric [18], or grid-based [64], [65]. Furthermore, there is a serious amount of theoretical work providing a mathematical analysis of the complexity of exploration strategies including comparisons for single robots [1], [2], [13], [14], [35], [42], [49]. Additionally, Lee and Recce [40] provide an experimental analysis of the

performance of different exploration strategies for one mobile robot.

Also the problem of exploring terrains with teams of mobile robots has received considerable attention in the past. For example, Rekleitis *et al.* [50]–[52] 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 the presented by Kurazume and Shigemi [39]. 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 that location. In extensive experiments, the author analyzes the performance of this approach and compares it to the optimal solution for different environments and different sizes of robot teams.

Koenig *et al.* [34] 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 leave markers in the cells they visit. The authors consider two different strategies for updating the markers. 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 the approach of Yamauchi *et al.* [63]. The second approach is Node Counting, in which the ants simply count the number of times a cell has been visited. The authors show that 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 by Kurabayashi *et al.* [38] 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

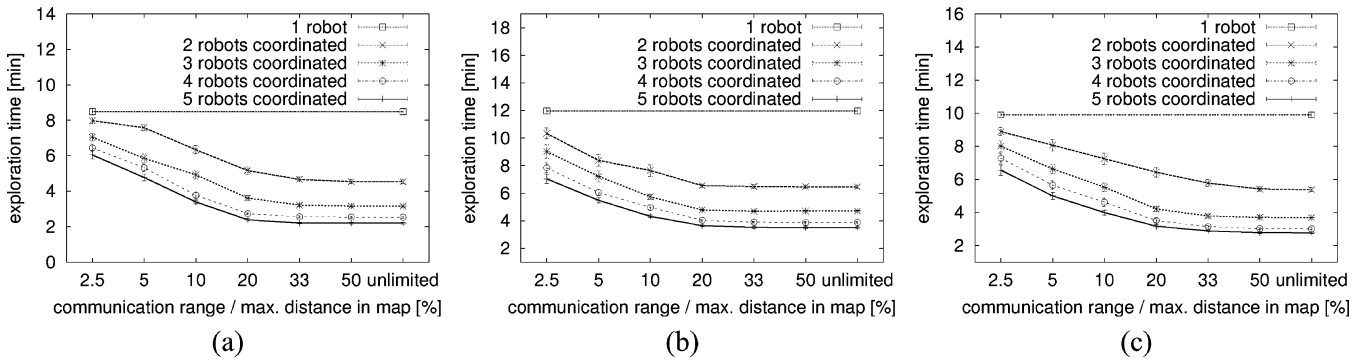


Fig. 12. Performances of the coordinated strategy with limited communication range for the different environments (a) (unstructured), (b) office, (c) and corridor environment). The x -axis shows the communication range in relation to the size of the environment, the y -axis the average exploration time. As can be seen, the results of these experiments look very similar in all tested environments.

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.* [63] 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 [64] to improve the consistency of the resulting map. To acquire knowledge about the environment all robots move to the closest frontier cell. The authors do not apply any strategies to distribute the robots over the environment or to avoid that two or more robots exploring the same areas.

One approach toward cooperation between robots has been presented by Singh and Fujimura [56]. This approach especially addresses the problem of heterogeneous robot systems. During exploration, each robot 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 task. Whenever a robot receives a message about a new task, it either accepts it or delegates it to smaller robots. In the case of homogeneous robots, the robots follow a strategy similar to the system of Yamauchi *et al.* [63]. Recently, Howard *et al.* [28] presented an incremental deployment approach that is similar to the technique described here. Whereas their approach explicitly deals with obstructions, i.e., situations in which the path of one robot is blocked by another, they do not consider the problem of limited communication. Zlot and colleagues [66] have recently proposed an architecture for mobile robot teams in which the exploration is guided by a market economy. In contrast to our algorithm, they consider sequences of potential target locations for each robot and trade tasks using single-item first-price sealed-bid auctions. In several experiments, we figured out that the treatment of the assignment problem as a multi-agent traveling sales-man problem yields advantages if the number of robots is small compared to the number of frontier cells. However, in the case of multiple robots this TSP-approach can be disadvantageous. Whenever a robot discovers a new frontier during exploration, this robot will often be the best suited to go on it [66]. We found that this can lead to an unbalanced assignment of tasks to robots so that the overall exploration time is increased. Ko *et al.* [33] present a variant of our approach that uses the Hungarian Method [36] to compute the assignments of frontier cells to robots. Practical experiments showed that the Hungarian Method yields a similar

performance as our coordination algorithm. Only in the case of small robot teams our approach appeared to be slightly superior since it provides a better distribution of the robots over the environment. A further advantage of our algorithm compared to the Hungarian Method lies in the fact that it can be implemented very easily.

Furthermore, there are approaches which address the problem of coordinating 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 [53] focus on the problem of exploring unknown environments with two robots and present 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.

Several researchers have focused on architectures for multi-robot cooperation. For example, Grabowski *et al.* [25] 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 [32] present a distributed action selection scheme for behavior-based agents which has successfully been applied to a cleaning task. Stroupe *et al.* recently presented the MVERT-approach [59]. Their system uses a greedy approach that selects robot-target pairs based on proximity. The goal of the action selection is to maximize cooperative progress toward mission goals. In contrast to our algorithm, the MVERT system does not discount areas close to the selected goal locations. Matarić and Sukhatme [43] consider different strategies for task allocation in robot teams and analyze the performance of the team in extensive experiments. Recently, Parker [48] described a project in which a large team of heterogeneous robots is used to perform reconnaissance and surveillance task. This work differs from our approach in that it investigates how to jointly accomplish a task with heterogeneous robots that cannot solve it individually.

VI. 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 tradeoff between utility and costs. We also presented an extension of our technique to multi-robot systems that have a limited communication range.

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 reveal that our coordination technique significantly reduces the exploration time compared to exploration approaches that do not explicitly coordinate the robots. Further experiments demonstrate that the performance of our technique nicely scales with the range of the communication link.

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 even if they can communicate with each other. 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. Additionally, we want to investigate scenarios in which the robots may malfunction or break or in which the environment changes over time.

ACKNOWLEDGMENT

The authors would like to thank D. Fox, R. Simmons, and S. Thrun for fruitful discussions. The real-world experiments were hosted by the Research Establishment for Applied Science (FGAN), Germany.

REFERENCES

- [1] S. Albers and M. Henzinger, "Exploring unknown environments," *SIAM J. Comput.*, vol. 29, pp. 1164–1188, 2000.
- [2] S. Albers, K. Kursawe, and S. Schuierer, "Exploring unknown environments with obstacles," *Algorithmica*, vol. 32, pp. 123–143, 2002.
- [3] D. Apostolopoulos *et al.*, "Robotic antarctic meteorite search: outcomes," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2001, pp. 4174–4179.
- [4] T. Balch and R. Arkin, "Communication in reactive multiagent robotic systems," *J. Autonomous Robot.*, vol. 1, no. 1, pp. 27–52, 1994.
- [5] R. Bellman, *Dynamic Programming*. Princeton, NJ: Princeton Univ. Press, 1957.
- [6] M. Bender and D. Slonim, "The power of team exploration: two robots can learn unlabeled directed graphs," in *Proc. 35th Annu. Symp. Foundations Comput. Sci.*, 1994, pp. 75–85.
- [7] A. Billard *et al.*, "A multi-robot system for adaptive exploration of a fast changing environment: probabilistic modeling and experimental study," *Connection Science*, vol. 11, no. 3/4, pp. 357–377, 2000.
- [8] W. Burgard *et al.*, "Collaborative multi-robot exploration," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2000, pp. 476–481.
- [9] Y. Cao *et al.*, "Cooperative mobile robotics: antecedents and directions," *Autonomous Robots*, vol. 4, no. 1, pp. 7–27, 1997.
- [10] J. Castellanos *et al.*, "The SPMAP: a probabilistic framework for simultaneous localization and map building," *IEEE Trans. Robot. Autom.*, vol. 15, no. 5, pp. 948–953, Oct. 1999.
- [11] H. Choset, "Topological simultaneous localization and mapping (SLAM): toward exact localization without explicit localization," *IEEE Trans. Robot. Autom.*, vol. 17, no. 2, pp. 125–137, Apr. 2001.
- [12] W. Cohen, "Adaptive mapping and navigation by teams of simple robots," *J. Robot. Autonomous Syst.*, vol. 18, pp. 411–434, 1996.
- [13] X. Deng *et al.*, "How to learn in an unknown environment," in *Proc. 32nd Symp. Foundations Comput. Sci.*, Los Alamitos, CA, 1991, pp. 298–303.
- [14] X. Deng and C. Papadimitriou, "How to learn in an unknown environment: the rectilinear case," *J. ACM*, vol. 45, no. 2, pp. 215–245, 1998.
- [15] G. Dissanayake *et al.*, "A computationally efficient solution to the simultaneous localisation and map building (SLAM) problem," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2000, pp. 1009–1014.
- [16] G. Dudek *et al.*, "Robotic exploration as graph construction," *IEEE Trans. Robot. Autom.*, vol. 7, no. 6, pp. 859–865, Dec. 1991.
- [17] —, "A taxonomy for multi-agent robotics," *Autonomous Robots*, vol. 3, no. 4, pp. 375–397, 1996.
- [18] T. Edlinger and E. von Puttkamer, "Exploration of an indoor environment by an autonomous mobile robot," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 1994, pp. 1278–1284.
- [19] H. Endres *et al.*, "Field test of a navigation system: autonomous cleaning in supermarkets," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1998, pp. 1779–1781.
- [20] M. Schneider-Fontan and M. Mataric, "Territorial multi-robot task division," *IEEE Trans. Robot. Autom.*, vol. 14, no. 5, pp. 815–822, 1998.
- [21] D. Fox *et al.*, "Collaborative multi-robot localization," in *Proc. 23rd German Conf. Artif. Intell.*, New York, 1999.
- [22] D. Fox *et al.*, "A probabilistic approach to collaborative multi-robot localization," *Autonomous Robots*, vol. 8, no. 3, pp. 325–344, 2000.
- [23] D. Goldberg and M. Mataric, "Interference as a tool for designing and evaluating multi-robot controllers," *J. Robot. Autonomous Syst.*, vol. 8, pp. 637–642, 1997.
- [24] H. González-Baños *et al.*, "Planning robot motion strategies for efficient model construction," in *Proc. Int. Symp. Robot. Res. (ISRR)*, 2000, pp. 345–352.
- [25] R. Grabowski *et al.*, "Heterogeneous teams of modular robots for mapping and exploration," *J. Autonomous Robot.*, vol. 8, no. 3, pp. 293–308, 2000.
- [26] D. Guzzoni *et al.*, "Many robots make short work," *AI Mag.*, vol. 18, no. 1, pp. 55–64, 1997.
- [27] D. Hougen *et al.*, "A miniature robotic system for reconnaissance and surveillance," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2000, pp. 501–507.
- [28] A. Howard, M. Mataric, and S. Sukhatme, "An incremental deployment algorithm for mobile robot teams," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2002, pp. 2849–2854.
- [29] R. Howard, *Dynamic Programming and Markov Processes*. New York: MIT Press/Wiley, 1960.
- [30] Y. Huang *et al.*, "Automatic operation for a robot lawn mower," in *SPIE Conf. Mobile Robots*, vol. 727, 1986, pp. 344–354.
- [31] M. Jäger and B. Nebel, "Dynamic decentralized area partitioning for cooperating cleaning robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2002, pp. 3577–3582.
- [32] D. Jung and A. Zelinsky, "An architecture for distributed cooperative planning in a behavior-based multi-robot system," *J. Robot. Autonomous Syst.*, vol. 26, no. 2–3, pp. 149–174, 1999.
- [33] J. Ko *et al.*, "A practical, decision-theoretic approach to multi-robot mapping and exploration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2003, pp. 3232–3238.
- [34] S. Koenig, B. Szymanski, and Y. Liu *et al.*, "Efficient and inefficient ant coverage methods," *Ann. Math. Artif. Intell.*, vol. 31, pp. 41–76, 2001.
- [35] S. Koenig *et al.*, "Greedy mapping of terrain," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 4, 2001, pp. 3594–3599.
- [36] H. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist. Quart.*, vol. 2, no. 1, pp. 83–97, 1955.
- [37] B. Kuipers and Y.-T. Byun, "A robot exploration and mapping strategy based on a semantic hierarchy of spatial representations," *J. Robot. Autonomous Syst.*, vol. 8, pp. 47–63, 1991.
- [38] D. Kurabayashi *et al.*, "Cooperative sweeping by multiple mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1996, pp. 1744–1749.
- [39] R. Kurazume and N. Shigemori, "Cooperative positioning with multiple robots," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 1994, pp. 1250–1257.
- [40] D. Lee and M. Recce, "Quantitative evaluation of the exploration strategies of a mobile robot," *Int. J. Robot. Res.*, vol. 16, no. 4, pp. 413–447, 1997.
- [41] J. Leonard and H. Feder, "A computationally efficient method for large-scale concurrent mapping and localization," in *Proc. 9th Int. Symp. Robot. Res.*, 2000, pp. 169–176.
- [42] S. Lumelsky *et al.*, "Dynamic path planning in sensor-based terrain acquisition," *IEEE Trans. Robot. Autom.*, vol. 6, no. 4, pp. 462–472, Aug. 1990.

- [43] M. Mataric and G. Sukhatme, "Task-allocation and coordination of multiple robots for planetary exploration," in *Proc. Int. Conf. Adv. Robot.*, 2001, pp. 61–70.
- [44] S. Moorehead, R. Simmons, and W. Whittaker, "Autonomous exploration using multiple sources of information," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 3, 2001, pp. 3098–3103.
- [45] M. Moors, "Koordinierte Multi-Robot Exploration," Master's thesis (in German), Dept. Comput. Sci., Univ. Bonn, Germany, 2000.
- [46] H. Moravec, "Sensor fusion in certainty grids for mobile robots," *AI Mag.*, pp. 61–74, Summer 1988.
- [47] R. Murphy, "Human-robot interaction in rescue robotics," *IEEE Syst., Man, Cybern., C. Appl. Rev.*, vol. 34, no. 2, pp. 138–153, May 2004.
- [48] L. Parker, "The effect of heterogeneity in teams of 100+ mobile robots," in *Multi-Robot Systems Volume II: From Swarms to Intelligent Automata*. Boston, MA: Kluwer, 2003.
- [49] N. Rao *et al.*, "Robot Navigation in Unknown Terrains: Introductory Survey of Non-Heuristic Algorithms," Oak Ridge National Lab., Oak Ridge, TN, Tech. Rep. ORNL/TM-12410, 1993.
- [50] I. Rekleitis *et al.*, "Multi-robot exploration of an unknown environment, efficiently reducing the odometry error," in *Proc. Int. Joint Conf. Artif. Intell. (IJCAI)*, 1997, pp. 1340–1345.
- [51] —, "Accurate mapping of an unknown world and online landmark positioning," in *Proc. Vision Interface (VI)*, 1998, pp. 455–461.
- [52] I. Rekleitis *et al.*, "Collaborative exploration for the construction of visual maps," in *Proc. IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, vol. 3, 2001, pp. 1269–1274.
- [53] N. Roy and G. Dudek, "Collaborative robot exploration and rendezvous: algorithms, performance bounds and observations," *J. Autonomous Robot.*, vol. 11, no. 2, pp. 117–136, 2001.
- [54] R. Simmons and D. Apfelbaum *et al.*, "Coordination for multi-robot exploration and mapping," in *Proc. Nat. Conf. Artif. Intell. (AAAI)*, 2000, pp. 852–858.
- [55] M. Simoncelli *et al.*, "Autonomous pool cleaning: self localization and autonomous navigation for cleaning," *J. Autonomous Robot.*, vol. 9, no. 3, pp. 261–270, 2000.
- [56] K. Singh and K. Fujimura, "Map making by cooperating mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1993, pp. 254–259.
- [57] C. Stachniss and W. Burgard, "Exploring unknown environments with mobile robots using coverage maps," in *Proc. Int. Conf. Artif. Intell. (IJCAI)*, 2003, pp. 1127–1132.
- [58] A. Stentz, "Optimal and efficient path planning for partially-known environments," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 4, May 1994, pp. 3310–3317.
- [59] A. Stroupe *et al.*, "Value-based action selection for exploration and mapping with robot teams," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2004, pp. 4190–4197.
- [60] C. Taylor and D. Kriegman, "Exploration strategies for mobile robots," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 1993, pp. 248–253.
- [61] S. Thrun, "A probabilistic online mapping algorithm for teams of mobile robots," *Int. J. Robot. Res.*, vol. 20, no. 5, pp. 335–363, 2001.
- [62] S. Thrun *et al.*, "A system for volumetric robotic mapping of abandoned mines," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, 2003, pp. 4270–4275.
- [63] B. Yamauchi, "Frontier-based exploration using multiple robots," in *Proc. 2nd Int. Conf. Autonomous Agents*, 1998, pp. 47–53.
- [64] B. Yamauchi, A. Schultz, and W. Adams, "Integrating exploration and localization for mobile robots," *Adap. Beh.*, vol. 7, no. 2, pp. 217–229, 1999.
- [65] A. Zelinsky *et al.*, "Planning paths of complete coverage of an unstructured environment by a mobile robots," in *Proc. Int. Conf. Adv. Robot. (ICAR)*, 1993, pp. 533–538.
- [66] R. Zlot *et al.*, "Multi-robot exploration controlled by a market economy," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, vol. 3, May 2002, pp. 3016–3023.



Wolfram Burgard (M'97) studied computer science at the University of Dortmund, Germany, and received the Ph.D. degree from the Department of Computer Science of the University of Bonn, Bonn, Germany, in 1991.

Since 1999, he has been an Associate Professor at the Department of Computer Science of the University of Freiburg, Germany, where he heads the research laboratory for Autonomous Intelligent Systems. His areas of interest lie in artificial intelligence and robotics, covering mobile robot navigation, multi-robot systems, state estimation, human robot interaction, activity monitoring, and networked robots.



Mark Moors was born in Oberhausen, Germany, in 1973. He received the M.Sc. degree in computer science from the University of Bonn, Bonn, Germany, in 2000.

He is currently working towards the Ph.D. degree at the Institute for Computer Science III, University of Bonn, Germany. His interests include multi-robot planning, tracking, and exploration.



Cyrill Stachniss (S'03) studied computer science at the University of Freiburg, Germany, and received the M.Sc. degree in 2002. Currently, he is working towards the Ph.D. degree at the research laboratory for Autonomous Intelligent Systems at the University of Freiburg.

His research interests lie in the areas of mobile robot exploration, SLAM, and collision avoidance.



Frank E. Schneider was born in 1967 in Cologne, Germany. He received the M.Sc. degree in computer science from the University of Bonn, Bonn, Germany, in 1995. Currently, he is working towards the Ph.D. degree in the Department of Computer Science III, University of Bonn, on the subject of multi-robot systems (MRS).

He is also affiliated with the Research Establishment for Applied Science (FGAN) in Bonn. His research interests include MRS navigation, relative localization, tracking, and active sensing.