

Multi-Robot Exploration Controlled by a Market Economy ^{*}

Robert Zlot, Anthony (Tony) Stentz, M. Bernardine Dias, Scott Thayer
{robz, axs, mbdias, sthayer}@ri.cmu.edu
Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

Abstract

This work presents a novel approach to efficient multi-robot mapping and exploration which exploits a market architecture in order to maximize information gain while minimizing incurred costs. This system is reliable and robust in that it can accommodate dynamic introduction and loss of team members in addition to being able to withstand communication interruptions and failures. Results showing the capabilities of our system on a team of exploring autonomous robots are given.

1 Introduction

Inherent to many robotic applications is the need to explore the world in order to effectively reason about future plans and objectives. In order to operate and perform complex tasks in previously unknown, unstructured environments, robots must be able to collect information and understand their surroundings. Many environments are hostile and uncertain, and it is therefore preferable or necessary to use robots in order to avoid risking human lives. In some cases, map-building is the main focus (*e.g.* reconnaissance, planetary exploration, while in others generating a map of the workspace is required for other purposes (*e.g.* navigation and planning). There are situations in which we would like to minimize repeated coverage to expedite the mission, while in the context of dynamic environments some amount of repeated coverage may be desirable. In order to effectively explore an unknown environment, it is necessary for an exploration system to be reliable, robust, and efficient. In this paper, we present an approach to multi-robot exploration

which has these characteristics and has been implemented and demonstrated on a team of autonomous robots. The definition of exploration varies within the literature, but we define it as the acquisition of attainable, relevant information from an unknown or partially known environment (*e.g.* in the form of a map).

Our approach focuses on the use of multiple robots to perform an exploration task. Multi-robot systems have some obvious advantages over single robot systems in the context of exploration. First, several robots are able to cover an area more quickly than a single robot, since coverage can be done in parallel. Second, using a robot team provides robustness by adding redundancy and eliminating single points of failure that may be present in single robot or centralized systems.

Coordination among robots is achieved by using a market-based approach [2]. In this framework, robots continuously negotiate with one another, improving their current plans and sharing information about which regions have and have not already been covered. Our approach does not rely on perfect communication, and is still functional (at reduced efficiency) with zero communication (apart from initial deployment). Furthermore, although a central agent is present, the system does not rely on this agent and will still function if all communication between it and the robots is lost. The role of this agent is simply to act as an interface between the robot team and a human operator. Interface agents can be brought into existence at any time, and in principle several can be active simultaneously. Thus the system is implemented in a completely distributed fashion.

The remainder of the paper is arranged as follows. Section 2 discusses previous work in the area of multi-robot exploration. Section 3 outlines our approach to the problem and section 4 describes the results obtained implementing our approach on real robot teams of different sizes. In section 5, we present our conclusions and discuss future research.

^{*}©2002 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

2 Related Work

There has been a wide variety of approaches to robotic exploration. Despite the obvious benefits of using multiple robots for exploration, only a small fraction of the previous work has focused on the multi-robot domain. Of those, relatively few approaches have been implemented effectively on real robot teams.

Balch and Arkin [1] investigated the role of communication for a set of common multi-robot tasks. For the task of *grazing* (*i.e.* coverage, exploration) they concluded that communication is unnecessary as long as the robots leave a physical record of their passage through the environment (a form of implicit communication). In many cases, it is not clear exactly how this physical trace is left behind and often physically marking the environment is undesirable. In addition, searching for the traces decreases exploration efficiency.

One technique for exploration is to start at a given location and slowly move out towards the unexplored portions of the world while attempting to get full, detailed coverage. Latimer et. al. [4] presented an approach which can provably cover an entire region with minimal repeated coverage, but requires a high degree of coordination between the robots. The robots sweep the space together in a parallel line formation until they reach an obstacle boundary, at which point the team splits up at the obstacle and can opportunistically rejoin at some later point. While guaranteed total coverage is sometimes necessary (*e.g.* land mine detection), in other cases it is preferable to get an initial rough model of the environment and then focus on improving potentially interesting areas or supplement the map with more specific detail (*e.g.* planetary exploration). Their approach is only semi-distributed, and fails if a single team member cannot complete its part of the task.

Rekleitis et. al. [5] proposed another method of cooperation in which stationary robots visually track moving robots as they sweep across the camera field of view. Obstacles are detected by obstructions blocking the images of the robots as they progress along the camera image. Since there are always some robots remaining stationary, some of the available resources are always idle. Another drawback is that if one robot fails, others can be rendered useless.

The methods of Rekleitis et. al. [5] and Latimer et. al. [4] have the disadvantage of keeping the robots in close proximity and require close coordination which can increase the time required for exploration if full, detailed coverage is not the primary objective. This also inhibits the reliability of the system in the event of full or partial communication problems or single robot

failures. While these issues are not always drawbacks in some *coverage* applications, for some exploration domains (*e.g.* reconnaissance, mapping of extreme environments), these are typically undesirable traits.

Simmons et. al. [6] presented a multi-robot approach which uses a frontier-based search and a simple bidding protocol. The robots evaluate a set of frontier cells (known cells bordering unknown terrain) and determine the expected travel costs and information gain of the cells (estimated number of unknown map cells visible from the frontier). The robots then submit bids for each frontier cell. A central agent (with a central map) then greedily assigns one task to each robot based on their bids. As with many greedy algorithms, it is possible to get highly suboptimal results since plans only consider what will happen in the very near future. The most significant drawback of this method, however, is the fact that the system relies on communication with a central agent and therefore the entire system will fail if the central agent fails. Also, if some of the robots lose communication with the central agent, they end up doing nothing.

Yamauchi [11] developed a distributed fault-tolerant multi-robot frontier-based exploration strategy. In this system, robots in the team share local sensor information so that all robots produce similar frontier lists. Each robot moves to its closest frontier point, performs a sensor sweep, and broadcasts the resulting updates to the local map. Yamauchi's approach is completely distributed, asynchronous, and tolerant to the failure of a single robot. However, the amount of coordination is quite limited and thus cannot take full advantage of the number of robots available. For example, more than one robot may decide (and is permitted) to go to the same frontier point. Since new frontiers generally originate from old ones, the robot that discovers a new frontier will often be the best suited to go to it (the closest). Another robot moving to the same original frontier will also be close to the newly discovered frontier. This can happen repeatedly; therefore, robots can end up following a leader indefinitely. In addition, a relatively large amount of information must be shared between robots. So, if there is a temporary communications drop, complete information will not be shared possibly resulting in a large amount of repeated coverage. Similar to the work by Simmons et. al. [6], plans are greedy and thus can be inefficient.

3 Approach

The previous examples fall short of presenting a multiple robot exploration system that can reliably and

efficiently explore unknown terrain, is robust to robot failures, and effectively exploits the benefits of using a multi-robot platform. Our approach is designed to meet these criteria by using a market architecture to coordinate the actions of the robots. Exploration is accomplished by each robot visiting a set of goal points in regions about which little information is known. Each robot produces a tour containing several of these points, and subsequently the tours are refined through continuous inter-robot negotiation. By following their improved tours, the robots are able to explore and map out the world in an efficient manner.

3.1 Market architecture

At the core of our approach is a market control architecture [2]. Multiple robots interact in a distributed fashion by participating in a market economy; delivering high global productivity by maximizing their own personal profits. Market economies are generally unencumbered by centralized planning; instead individuals are free to exchange goods and services and enter into contracts as they see fit. The architecture has been successfully implemented on a robot team performing distributed sensing tasks in an environment with known infrastructure [8].

Revenue is paid out to individual robots for information they provide by an agent representing the user’s interests (known as the operator executive, or *OpExec*). Costs are similarly assessed as the amount of resources used by an individual robot in obtaining information.

In order to use the market approach as a coordination mechanism, cost and revenue functions must be defined. The cost function, $\mathcal{C} : R \rightarrow \mathbb{R}^+$, is a mapping from the a set of resources R to a positive real number. One can conceivably consider a combination of several relevant resources (time, energy, communication, computation), however here we use a distance-based cost metric – the expected cost incurred by the robot is the estimated distance traveled to reach the goal¹. The item of value in our economy is information. The revenue function, $\mathcal{R} : \mathcal{M} \rightarrow \mathbb{R}^+$, returns a positive real number given map information \mathcal{M} . The world is represented by an occupancy grid where cells may be marked as free space, obstacle space, or unknown. Information gained by visiting a goal point can be calculated by counting the number of unknown cells within a fixed distance from the goal². Profit is then

¹Path costs are estimated using the D* algorithm [7], which is also used for path planning.

²The value we use is actually an overestimate of the information gain in a sensor sweep in order to compensate for the fact that the robot can discover new terrain along its entire path to the goal point.

calculated as the revenue minus the cost. The revenue term is multiplied by a weight converting information to distance. The weight fixes the point where cost incurred for information gained becomes profitable (*i.e.* positive utility). Each robot attempts to maximize the amount of new information it discovers, and minimize its own travel distance. By acting to advance their own self-interests, the individual robots attempt to maximize the information obtained by the entire team and minimize the use of resources.

Within the marketplace, robots make decisions by communicating price information. Prices and bidding act as low bandwidth mechanisms for communicating aggregate information about costs, encoding many factors in a concise fashion. In contrast to other systems which must send large amounts of map data in order to facilitate coordination [6, 11], coordination in our system is for the most part achieved by sharing price information.

3.2 Goal point selection strategies

Tasks (goal points to visit) are the main commodity exchanged in the market. This section describes some example strategies for generating goal points. These strategies are simple heuristics intended to select unexplored regions for the team to visit, with the goal point located at the region’s centre.

Random. The simplest strategy used is random goal point selection. Here goal points are chosen at random, but discarded if the area surrounding the goal point has already been visited. An area is considered visited if the number of known cells visible from the goal is greater than a fixed threshold. Random exploration strategies have been effective in practice, and some theoretical basis for effectiveness of the random approach has been given (*e.g.* [9]).

Greedy exploration. This method simply chooses a goal point centred in the closest unexplored region (of a fixed size) to the robot as a candidate exploration point. As demonstrated previously [3], greedy exploration can be an efficient exploration strategy for a single robot.

Space division by quadtree. In this case, we represent the unknown cells using a quadtree. In order to account for noise, a region is divided into its four children if the fraction of unknown space within the region is above a fixed threshold. Subdivision recursion terminates when the size of a leaf region is smaller than the sensor footprint. Goal points are located at the centres of the quadtree leaf regions.

Because the terrain is not known in advance, it is likely that some goal points are not reachable. When a goal is not reachable, the robot is drawn towards the edge of reachable space while attempting to achieve its goal. This results in more detail in the areas of the map near boundaries and walls, which are usually the most interesting areas. Once the incurred travel cost exceeds the initial expected cost by a fixed margin, the robot decides that the goal is unreachable and moves on to its next goal. This avoids the scenario in which a robot indefinitely tries to reach an unreachable goal point.

Note that the goal generation algorithms are extremely simplistic. The intention is that the market architecture removes the inefficiencies consequent in using relatively simple criteria for goal selection.

3.3 Exploration algorithm

Here we describe the complete exploration algorithm, which implements the ideas discussed in the preceding parts of section 3.

The robots are initially deployed into an unknown space with known relative positions. Each robot begins by generating a list of goal points using one of the strategies described in section 3.2. The robots may uniformly use the same strategies, or the strategy used can vary across robots or even over time on a single robot. If the robot is able to communicate with the *OpExec*, these goals can be transmitted to check if they are new goals to the colony (if the *OpExec* is not reachable, this step is skipped). The robot then inserts all of its remaining goals into its current tour, by greedily placing each one at the cost-minimizing (shortest path) insertion point in the list³. Next, the robot tries to sell each of its tasks to all robots with which it is currently able to communicate, via an auction. The other robots each submit bids, which encapsulate their cost and revenue calculations. The robot offering the task (the auctioneer) waits until all robots have bid (up to a specified amount of time). If any robot bids more than the minimum price set by the auctioneer, the highest bidder is awarded the task in exchange for the price of the bid. Once all of a robot’s auctions close (all goals on the robot’s tour have been sequentially offered), that robot begins its tour by navigating towards its first goal. When a robot reaches a goal, it generates new goal points. The number of goal points generated depends on how many goals are in the current tour –

if there are a large number of goals in the current tour, fewer goals are generated since introducing many new tasks into the system could limit performance by increasing computation and negotiation time. The robot then starts off towards its next goal, and offers all of its remaining goals to the other robots.

The selling of tasks is done using single-item first-price sealed-bid auctions [10]. A robot may announce an auction for any task in its tour, with the interpretation that it currently owns the right to execute the task in exchange for payment from the *OpExec*. Given a task under consideration, a robot’s valuation of the task is computed as the profit expected if the task were added to the current tour (expected revenue minus expected cost). The auctioneer announces a reservation price for the auction, P_r . P_r is the seller’s valuation of the task with a fixed mark-up, and represents the lowest possible bid that the seller will accept. The remaining robots act as buyers, negotiating to receive the right to execute the task, and therefore payment from the *OpExec*. Each buyer calculates its valuation for the goal, v_i , by finding the expected profit in adding that goal to its current tour. The bidding strategy is defined by each buyer i submitting a bid of

$$B_i = P_r + \alpha * (v_i - P_r) \quad (1)$$

where α is between 0 and 1. We use $\alpha = 0.9$, which gives seller some incentive to sell the task to a better-suited robot, while at the same time allowing the buyer to reap a larger fraction of the additional revenue the task generates (as a reward for actually executing the task).

If the bidder expects to make a profit greater than the reservation price, then B_i from equation (1) will be greater than P_r , and the bidder will be awarded the task if no other robot has submitted an even higher bid. If the bidder expects to make a profit which is less than the reservation price, then B_i will be smaller than P_r , and so no bid is submitted (or equivalently, the bid is lower than the reservation price so it cannot win the auction). If none of the bidding robots offer more than the reservation price, then the seller will make more profit by keeping the goal, and so there is no winner. Given this mechanism, the robot that owns the task after the auction is in most cases the robot that can perform the task most efficiently, and is therefore best-suited for the task.

Since communication is completely asynchronous, a robot must be prepared to handle a message regardless of current state. In order to achieve system robustness, it is important to ensure that some communications issues inherent to the problem domain are addressed. No agent ever assumes that it is connected to or able to communicate with any of the other agents. Many of

³The problem encountered here is an example of the traveling salesman problem (TSP), which is known to be \mathcal{NP} -hard. The optimal tour cannot be found in polynomial time and goals arrive in an online fashion, so a greedy insertion heuristic is used to approximate.

the robots' actions are driven by events which are triggered upon the receipt of messages. If for some reason a robot does not receive a message it is expecting (*e.g.* the other party has had a failure, or there are communication problems) it must be able to continue rather than wait indefinitely. Therefore, timeouts are invoked whenever an agent is expecting a response from any other agent. If a timeout expires, the agent is able to carry on and is also prepared to ignore the response if it does arrive eventually.

Although a single robot can offer only one task at a time, there can be multiple tasks simultaneously up for bids by multiple robots. Therefore, it is possible for a robot to win two tasks from simultaneous auctions which may have been wise investments individually, but owning one may devalue the other (*e.g.* two tasks which may be equally far from the robot, but far away from each other). In this situation the robot has no choice but to accept both tasks, but can offload the less desirable task at its next opportunity to call an auction (*e.g.* when it reaches its next goal point). In this way, robots have constantly occurring opportunities to exchange the less desirable tasks that they may have obtained through auction or goal generation. If two instances of the *same* goal are simultaneously auctioned off and won by different robots, one robot will eventually own both as it is highly unlikely that these two goals will be auctioned off at the same time more than once. The solutions will still be local minima in terms of optimality because we are only allowing single task exchanges.

Robot failure (loss) is handled completely transparently. The lost robot no longer participates in the negotiations and thus is not awarded any further tasks. The lost robot's tasks are not completed, but other robots eventually generate goal points in the same areas, since those unexplored regions are worth a large amount of revenue. New robots can also be introduced into the colony if position and orientation relative to another robot (or equivalently some landmark if available) at some instant of time is known.

3.4 Information Sharing

Information sharing is helpful in ensuring that the robots coordinate the exploration in a sensible manner. We would like the robots to cover the environment as completely and efficiently as possible with minimal repeated coverage. This is achieved in several ways, most of which emerge naturally from the negotiation protocol. Information sharing mechanisms are not crucial to the completion of the task, but can increase the efficiency of the system. Any communication disruptions or failures do not disable the team, but can reduce the

efficiency of the exploration.

First, the robots are usually kept a reasonable distance apart from one another, since this is the most cost-effective strategy. If one robot has a goal point that lies close to a region that is covered by some other robot, the other robot wins this task when it is auctioned off (this robot has lower costs and thus makes more profit). The effect is that the robots tend to stay far apart and map different regions of the workspace, thereby minimizing repeated coverage.

Second, if one (auctioneer) robot offers a goal that is in a region already covered by another (bidder) robot, the bidder sends a message informing the auctioneer of this fact. The auctioneer then cancels the auction and removes that goal from its own tour. Here the bidder robot is giving the auctioneer robot a better estimate of the profit that can be gained from the task, and prevents the seller from covering or selling space which has already been seen. In view of this new information, the auctioneer now realizes that it will not be profitable for any of the robots to go to this waypoint.

Third, there is also *explicit* map sharing which is done at regular intervals. A robot can periodically send out a small explored section of its own map to any other robot with which it can communicate in exchange for revenue (based on the amount of new information, *i.e.* the number of new known map cells, which is being transmitted). This information can conceivably be exchanged on the marketplace, where each robot can evaluate the expected utility of the map segments and then offer an appropriate price to the seller, who may sell if the cost of exchange (in time and communication required to send the information) is small compared to the offered price. This type of information exchange can improve the efficiency of the negotiation process in that robots are able to estimate profits more accurately and are less likely to generate goals which are in regions already covered by other team members. In the case of a contradiction between a robot's map and the map section being received, the robot always chooses to believe its own map.

Map information from the robots is gathered upon request from an *OpExec* on behalf of a human operator. The *OpExec* sends a request for map data to all reachable robots, and then assembles the received maps assuming the relative orientations of the robots are known. The maps are combined by simply summing the values of the individual occupancy grid cells where an *occupied* reading is counted as a +1 and a *free* reading is counted as a -1. By superpositioning the maps in this way, conflicting beliefs destructively interfere resulting in a 0 value (*unknown*), and similar beliefs constructively interfere resulting in larger positive or negative values which represent the confidence

in the reading (there is an upper limit to the absolute value a combined reading can have in order to allow for noise or changes in the environment).

4 Results

4.1 Experimental setup

The experiments were run on a team of ActivMedia PioneerII-DX robots (Figure 1). Each robot is equipped with a ring of 16 ultrasonic sensors, which are used to construct occupancy grids of the environment as the robot navigates. Each robot is also equipped with a KVH E•CORETM 1000 fiber optic gyroscope used to track heading information. Due to the high accuracy of the gyroscopes ($2 - 4^\circ$ drift/hr), we use the gyro-corrected odometry at all times rather than employing a localization scheme. Using purely encoder-based dead reckoning the positional error can be as high as 10% to 25% of the distance traveled for path lengths on the order of 50-100m, while using gyro-corrected odometry reduces the error to the order of 1% of the distance traveled. However, an accurate localization algorithm may improve the results, especially if the experimental runs extend over a much longer period of time (a typical run takes 5 to 10 minutes to map several hundred square metres).

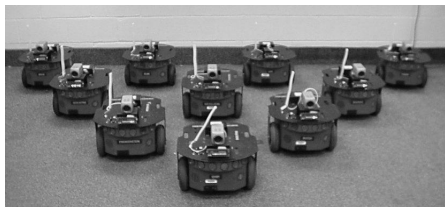


Figure 1: Robot team used in experiments.

Test runs were performed in three different environments. The first is in the Field Robotics Center (FRC) highbay at Carnegie Mellon University. The highbay is nominally a large open space (approximately 45m x 30m), although it is cluttered with many obstacles (such as walls, cabinets, other large robots, and equipment from other projects – see Figure 2). Figures 3 and 4 show the constructed maps from two separate highbay explorations. The second environment is an outdoor run in a patio containing open areas as well as some walls and tables (size is approximately 30m x 30m). Figure 5(a) shows the resulting map created by a team of five robots in this environment. The third environment is a hotel conference room during a demonstration in which approximately 25 tables were set up and in excess of 100 people were wandering



Figure 2: Two different views of the FRC highbay environment used in testing.

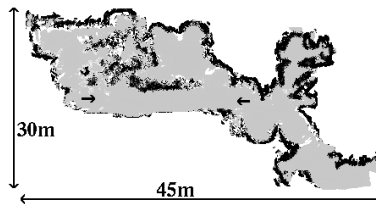


Figure 3: Five robot map of FRC highbay. Approximate size of mapped region is $550m^2$. The arrows in the figure show where the photographs in Figure 2 were taken.

about the rooms and lobbies (size is approximately 40m x 30m). A map created by five robots is shown in Figure 5(b). The results for the environments shown in Figure 5 were not quantified, but were provided as examples of wide applicability.

4.2 Experimental Results

In order to quantify the results, we use a metric which is directly proportional to the amount of information retrieved from the environment, and is inversely proportional to the costs incurred by the team. The amount of information retrieved is the area covered, and the cost is the combined distance traveled by each robot. Thus, the quality of exploration is measured as:

$$Q = \frac{A}{\sum_{i=1}^n d_i} \quad (2)$$

where d_i is the distance traveled by robot i , A is the total area covered, and n is the number of robots in the team. The sensor range utilized by each robot is a $4m \times 4m$ square (containing local sonar data as an occupancy grid), and so a robot can view a maximum previously uncovered area of $4m^2$ for every one metre it travels ($Q_{max} = 4m^2/m$). This is a considerable overestimate for almost any real environment, as it assumes that there is zero repeated coverage and that robots always travel in straight lines (no turning) and

never encounter obstacles. Nevertheless, it can serve as a rough upper bound on exploration efficiency.

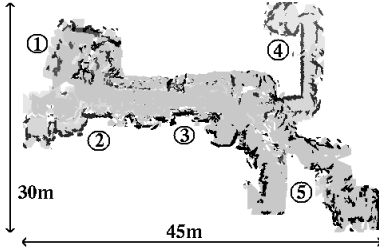


Figure 4: Four robot map of FRC highbay. Approximate size of mapped region is $500m^2$. (The map differs from the one in Figure 3, as a different set of doors were open and other objects in the environment had been moved.) The numbered areas in the figure represent the five areas that the robots were required to visit in order to reach the stopping criteria.

Table 1 shows a comparison of the results obtained in running our exploration algorithm using the three different goal selection strategies outlined in section 3.2, plus one run in which no communication was permitted between the robots. In each case, the run was carried out in the FRC highbay using four robots which were initially deployed in a line formation. Exploration was terminated when the robots had mapped out a rough outline of the complete floor plan of the highbay, which required them to visit and map the five main areas labeled in Figure 4. Each value in Table 1 is an average obtained over 10 runs with the best and worst Q values discarded. During these experiments, robots in the team were sporadically disabled in order to demonstrate the system’s robustness to the loss of individual robots.

The quadtree and random strategies performed equally well, covering on average $1.4m^2$ per metre traveled. The greedy strategy performed relatively poorly,

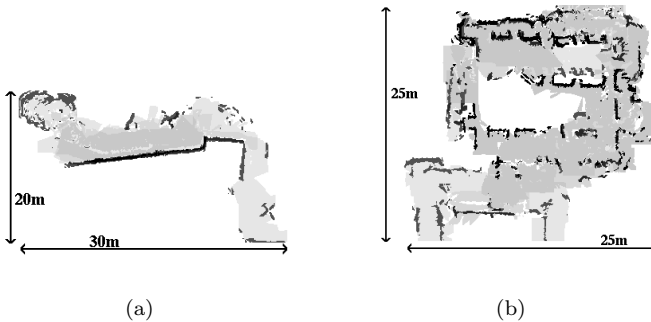


Figure 5: (a) Four robot map of exterior environment. Approximate size of mapped region is $50m^2$. The ‘X’ shaped objects are the bases of outdoor tables. (b) Five robot map of hotel conference room. Approximate size of mapped region is $250m^2$. The rectangular-shaped objects are tables which were covered on three of their four sides.

Strategy	Area covered / distance traveled [m^2/m]
Random	1.4
Quadtree	1.4
Greedy	0.85
No comm	0.41

Table 1: Comparison of goal selection strategy results

covering an average of $0.9m^2$ per metre traveled. The main advantage of the quadtree and random strategies is the fact that many goal points are selected which are spread out over the *entire* exploration space, irrespective of current robot positions. Through negotiation, the robots are able to come up with plans which allow them to spread out and partition the space efficiently.

The greedy approach has a number of drawbacks which limit the exploration efficiency. By design, the goal points generated by a robot are always close to the current position, so the robot generating a goal is usually best suited to visit that goal. Thus, very few tasks are exchanged between robots, and so the efficiency benefits of negotiating are not fully exploited by the team. This also means that the plans that the robots are using do not in general have the effect of globally dividing up the space and spreading out the paths of the robots.

The final entry in Table 1 shows the effect of removing all negotiation and information sharing from the system. This effectively leaves the robots exploring concurrently, but without any communications they cannot efficiently cover the environment. Robots used the random goal generation strategy. Without the ability to negotiate, robots did not have the opportunity to fully improve their tours by exchanging tasks, and to divide up the space requiring coverage. The resulting coverage efficiency of $0.41m^2/m$ is only 29% of the coverage efficiency achieved when coordinating the robot team using the market architecture. Without communication, the worst possible case for coverage occurs when all of the robots cover all of the space individually before the combined coverage is complete (*i.e.* termination occurs when $\bigcup A_i = \bigcap A_i = A$, where A_i is the area covered by robot i and A is the complete area being mapped). Assuming n robots are used and there is no repeated coverage, if the robots are allowed to communicate then efficiency can *at best* be improved by a factor of n . In our results we have come close to this upper bound by adding negotiation, improving the efficiency by a factor of 3.4 when using $n = 4$ robots.

Figure 6 shows a trace of the paths followed by the

robots in one of the experimental runs using random goal generation. Here we can see the beneficial effect that the negotiation process had on the plans produced by the robots. Although the initial goal points were randomly placed, the resulting behaviour is that the robots spread out to different areas and covered the space efficiently.

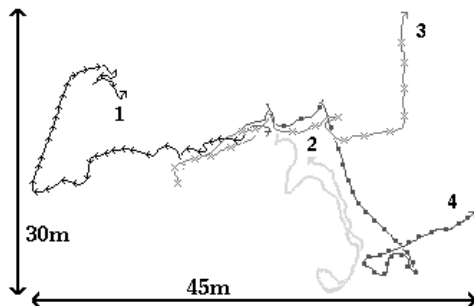


Figure 6: Paths taken by four exploring robots in FRC highbay. The robots initially were in a line formation near the centre of the image and dispersed in different directions to explore the highbay. The small amount of repeated coverage near the centre of the map is unavoidable, as there is only a narrow lane joining the left and right areas of the environment (compare with photos shown in Figure 2 and map shown in Figure 4 for reference).

5 Conclusions

In this paper we present a reliable, robust, and efficient approach to distributed multi-robot exploration. The key to our technique is utilizing a market approach to coordinate the team of robots. The market architecture seeks to maximize benefit (information gained) while minimizing costs (in terms of the collective travel distance), thus aiming to maximize utility. The system is robust in that exploration is completely distributed and can still be carried out if some of the colony members lose communications or fail completely. The effectiveness of our approach was demonstrated through results obtained with a team of robots. We found that by allowing the robots to negotiate using the market architecture, exploration efficiency was improved by a factor of 3.4 for a four-robot team.

To build on the promising results seen so far, future work will look at several possible ways to improve the overall performance of the system. Currently, the algorithm is designed to minimize distance traveled while exploring. Instead of distance based-costs, using a time-based cost scale will lead to rapid exploration. This will also facilitate a more straightforward way to prioritize some types of tasks over others in the market framework, for example if there are other mission objectives in addition to exploration. A more com-

plex cost scheme could be implemented which combines several cost factors in order to efficiently use a *set* of resources. It may also be worthwhile to include some simple learning which may increase the effectiveness of the negotiation protocol. Characterizing the dependence of exploration efficiency on the number of robots in the team may also provide interesting results. In addition, testing different goal generation strategies (*e.g.* frontier-based strategies) may lead to performance improvements. Finally, robot loss can be handled more explicitly which may lead to a faster response in covering the goals of the lost team member.

Acknowledgments

The authors would like to thank the Cognitive Colonies group⁴ at Carnegie Mellon University for their valuable contribution. This research was sponsored in part by DARPA under contract “Cognitive Colonies” (contract number N66001-99-1-8921, monitored by SPAWAR).

References

- [1] T. Balch and R. C. Arkin. Communication in reactive multi-agent robotic systems. In *Autonomous Robots*, volume 1(1), pages 27–52, 1994.
- [2] M. B. Dias and A. Stentz. A free market architecture for distributed control of a multirobot system. In *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, pages 115–122, 2000.
- [3] S. Koenig, C. Tovey, and W. Halliburton. Greedy mapping of terrain. In *Proceedings of the International Conference on Robotics and Automation*, pages 3594–3599. IEEE, 2001.
- [4] D. Latimer IV, S. Srinivasa, A. Hurst, H. Choset, and V. Lee-Shue, Jr. Towards sensor based coverage with robot teams. In *Proceedings of the International Conference on Robotics and Automation*. IEEE, 2002.
- [5] I. M. Rekleitis, G. Dudek, and E. E. Miliotis. Multi-robot collaboration for robust exploration. In *Proceedings of the International Conference on Robotics and Automation*. IEEE, 2000.
- [6] R. Simmons, D. Apfelbaum, W. Burgard, D. Fox, S. Thrun, and H. Younes. Coordination for multi-robot exploration and mapping. In *Proceedings of the National Conference on Artificial Intelligence*. AAAI, 2000.
- [7] A. Stentz. Optimal and efficient path planning for partially-known environments. In *Proceedings of the International Conference on Robotics and Automation*, volume 4, pages 3310–3317. IEEE, May 1994.
- [8] S. Thayer, B. Digney, M. B. Dias, A. Stentz, B. Nabbe, and M. Hebert. Distributed robotic mapping of extreme environments. In *Proceedings of SPIE: Mobile Robots XV and Telemanipulator and Telepresence Technologies VII*, 2000.

⁴<http://www.frc.ri.cmu.edu/projects/colony/>

- [9] I. A. Wagner, M. Lindenbaum, and A. M. Bruckstein. Robotic exploration, brownian motion and electrical resistance. In *RANDOM98 - 2nd International workshop on Randomization and Approximation Techniques in Computer Science*, October 1998.
- [10] E. Wolfstetter. Auctions: An introduction. *Journal of Economic Surveys*, 10(4):367-420, 1996.
- [11] B. Yamauchi. Frontier-based exploration using multiple robots. In *Second International Conference on Autonomous Agents*, pages 47-53, 1998.