Coordination Strategies for Multi-robot Exploration and Mapping

John G. Rogers III, Carlos Nieto-Granda, and Henrik I. Christensen

Abstract. Situational awareness in rescue operations can be provided by teams of autonomous mobile robots. Human operators are required to teleoperate the current generation of mobile robots for this application; however, teleoperation is increasingly difficult as the number of robots is expanded. As the number of robots is increased, each robot may interfere with one another and eventually decrease mapping performance. Through careful consideration of robot team coordination and exploration strategy, large numbers of mobile robots be allocated to accomplish the mapping task more quickly and accurately.

1 Motivation

Projects like the Army Research Laboratory's Micro-Autonomous Systems Technology (MAST) [1] seek to introduce the application of large numbers of inexpensive and simple mobile robots for situational awareness in urban military and rescue operations. Human operators are required to teleoperate the current generation of mobile robots for this application; however, teleoperation is increasingly difficult as the number of robots is expanded. There is evidence in human factors research which indicates that the cognitive load on a human operator is significantly increased when they are asked to teleoperate more than one robot [18].

Autonomy will make it possible to manage larger numbers of small robots for mapping. There is a continuum of options as to the degree of shared autonomy between robot and human operator [11]. Current robots employed in explosive ordinance disposal (EOD) missions are fully tele-operated. At the other extreme, robots can be given high-level tasks by the operator, while autonomously handling low-level tasks [3] such as obstacle avoidance or balance maintenance. In this paper, our

John G. Rogers III · Carlos Nieto-Granda · Henrik I. Christensen
Center for Robotics and Intelligent Machine
Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA 30332
e-mail: {jgrogers,carlos.nieto,hic}@gatech.edu

J.P. Desai et al. (Eds.): *Experimental Robotics*, STAR 88, pp. 231–243.
DOI: 10.1007/978-3-319-00065-7_17

robot teams occupy the latter end of the spectrum; we imagine that the operator has tasked the robot team to autonomously explore and map an unknown environment while focusing on the high level task of looking for survivors.

In the multi-robot scenario, resources are distributed amongst a team of robots instead of concentrated on one large and expensive machine. This distribution offers a number of advantages and disadvantages over the single robot case. The distributed team is able to continue its mission even if some of the robots are disabled or destroyed. A single robot can only explore or monitor at one location at a time; however, the multi-robot team can provide situational awareness in many locations at once. Unless the single robot is able to move much faster than the multi-robot agents, the lone robot will be slower in performing the exploration and mapping task. These advantages are taken for a multi-robot team at the cost of increased complexity in communication and coordination.

As the number of robots is increased, each robot may interfere with one another and eventually decrease the performance of the mapping task. Careful consideration of exploration strategy and coordination of large numbers of mobile robots can efficiently allocate resources to perform the mapping task more quickly and more accurately.

Mobile robot simultaneous localization and mapping (SLAM) has been thoroughly addressed in the literature, see [2] and [6] for a detailed review of the history and state-of-the-art in SLAM research. The specific techniques used in this paper are based upon the Square Root SAM algorithm [4] [5] which uses the well-known algorithms of linear algebra least-squares system solving to compute the map and robot trajectory based on a set of measurements.

Multi-robot mapping and exploration was addressed in [9] and [17]. These papers build a map using up to 3 robots with a decision-theoretic planner which trades off robot rendezvous operations with frontier exploration. These robots rendezvous to determine their relative pose transforms to provide constraints to recover the final map. In contrast, our approach does not require this rendezvous step because landmarks are globally data associated between each robot on a central map coordinator. The exploration strategy used is similar to our strategy called *Reserve*; however, we will not use a rendezvous step and do not require a decision-theoretic planner.

2 Technical Approach

We use the Robot Operating System (ROS) from [12]. ROS provides interprocess communication as well as coordination of sensor data with pose information. Our robot algorithms are implemented as a distributed set of programs which run in the ROS system. In addition, we make use of several implementations of common mobile robot software components which are provided in the ROS distribution such as motion planning, obstacle avoidance, platform control, and IMU and odometry filtering.

2.1 Mapping System

Our mapping system is based upon the *GTsam* library developed at Georgia Tech. This library extends the Square Root SAM technique in [5] with sparse linear algebra in a nonlinear optimization engine. We have extended the GTsam library with a framework based upon the M-space formulation of Folkesson and Christensen [8] called *OmniMapper*. *OmniMapper* is a map library based upon a system of plug-ins which handle multiple landmark types simultaneously. We have used the *OmniMapper* in the past to build maps using multiple types of landmarks such as walls, doors, and objects [14] [13] [16]. This implementation builds maps of planar regions corresponding to walls and tables from [15].

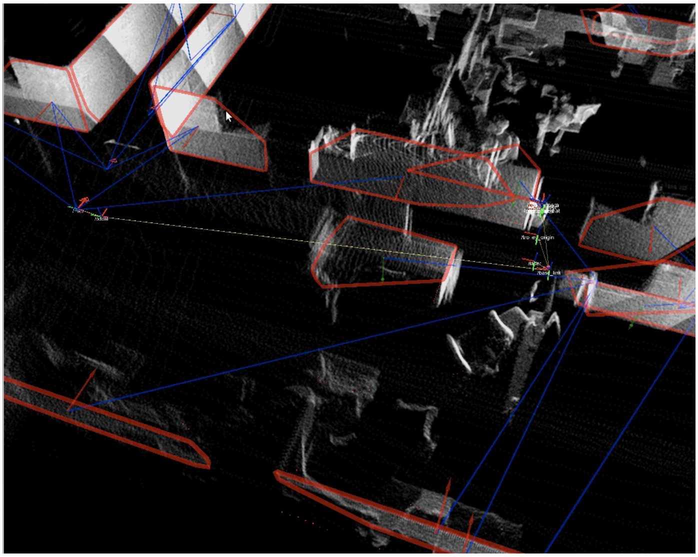


Fig. 1 OmniMapper

Each robot in the team builds a map locally with the *OmniMapper* and sends map data to the map coordinator. Each robot can incorporate new landmark measurements whenever it has moved far enough from the last pose where measurements were made. In the current implementation this is set to 10cm. When a robot finishes optimizing its local map with new landmark measurements, all relevant information needed by the map coordinator is packaged and transmitted.

The information which is needed by the map coordinator to incorporate a new piece of information from a team member consists of many components. First, the sensor measurement data is needed. In the current implementation, this consists of the extracted plane information consisting of a plane equation along with a convex hull of points along the perimeter of the plane. This represents a significant compression over an alternative scheme where all point-cloud data could be transmitted and processed at the master node. Secondly, the team member's integrated odometry is transmitted. This allows the master node to compute the odometric relative pose since the prior landmark measurement data was incorporated; this is used to insert a relative pose factor and also give initial conditions for data association. Finally, the team member's local map pose is transmitted. This is used by the master node to compute a map pose correction. This correction is sent back to the team member so that it knows it's relative pose in the global map frame. This knowledge is needed so that the team member can interpret exploration goals correctly.

The map coordinator maintains trajectories for each of the robots in the team. Measurements from each robot are merged into one global view of the landmarks. This is realized through a simple modification to the standard *OmniMapper* through duplication of data structures tracking indexing data and pose information used for interaction with GTsam into arrays. This implementation potentially allows for an unlimited number of team members to build a map together.

Most modern SLAM approaches use a *pose graph* [10] which is generated via laser scan matching in 2D or point-cloud ICP in 3D. This approach is effective for single robot mapping; however, it has some drawbacks for larger multirobot mapping. Scan matching and ICP algorithms are computationally intensive and matching across many robots would rapidly become intractable. Also, point cloud representations are large and their transport over a wireless link could be prohibitive if the link is limited in capacity due to mesh network routing or environmental interference. To address these limitations, our robots extract relevant, parsimonious features from the environment and transmit them to the master node.

Each turtlebot in these experiments maps planar wall structures using a Microsoft Kinect sensor. Planar segments corresponding to walls are extracted from point clouds via a RANSAC [7] based algorithm [15]. Points are uniformly sampled from the point cloud and any sufficiently large set of points coplanar with these three points are selected as a plane and are removed from the point cloud. This process is repeated until up to four planes are extracted or a fixed number of iterations is reached. To improve the speed of plane extraction, the Kinect point cloud is computed at QQVGA ($frac18$) resolution, which achieves $\tilde{1}$Hz frame rate.

The Kinect sensor on each robot has a narrow field-of-view which is not ideal for detecting exploration frontiers. To alleviate this problem, we incorporated a strategy by which each robot will rotate periodically to get a 360 degree view of its surroundings. This data is synchronized with robot odometry to synthesize a 360 degree laser scan. This synthesized laser scan is sent to the local mapper and forwarded to the global mapper. At the global mapper, it is linked to a trajectory pose element and used to populate an occupancy grid. This occupancy grid is re-computed after every map optimization so that a loop closure will result in a correct occupancy grid map. The frontier based exploration strategies detailed below use this occupancy grid to find the boundary between clear and unknown grid cells.

2.2 Exploration Strategy

Each robot team leader uses a frontier based exploration strategy similar to the one used in [17]. An exploration frontier is defined on a costmap cellular decomposition where each cell has one of three labels: *Clear*, *Obstacle*, and *Unknown*. The costmap is initialized as *Unknown*. Costmap cells are set to *Obstacle* corresponding to locations where the Kinect sensor detects an obstacle in the environment. The cells on a line between the obstacle cell and the robot's current location are set to *Clear*. Exploration frontiers are defined as *Clear* cells which are adjacent to at least one neighbor where the label is *Unknown*.

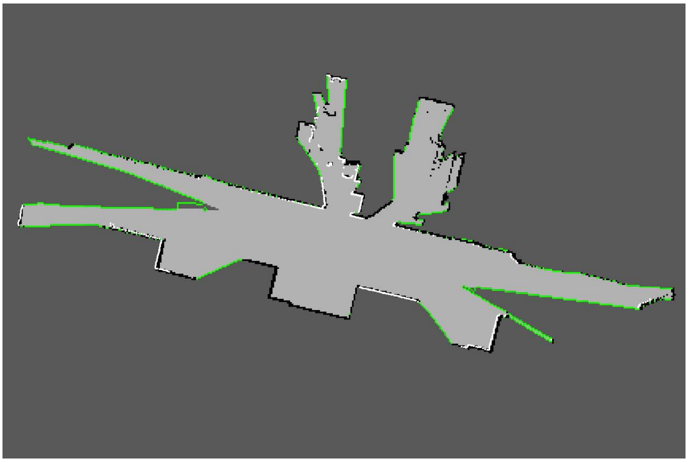


Fig. 2 Global maps using the *Reserve* coordination algorithm described in this paper

The high level robot exploration goal allocation is centrally planned on the same workstation where the global map is constructed. There are many choices which can be made by the exploration planner when choosing which robot or group of robots should move towards an exploration goal. We have chosen to employ a greedy strategy by which the nearest robot or team is allocated to a goal instead of a more sophisticated traveling-salesman type of algorithm. We believe that this is appropriate because the exploration goals will change as the robots move through the environment; re-planning will be required after each robot or team reaches an exploration goal.

2.3 Coordination Strategy

The coordination strategy used between robot agents as well as the number of robots are the independent variables in the experiments performed in this paper. The coordination strategy refers to the proportion of robots which are dispatched to each exploration goal. On one extreme, a single robot can be sent to explore a new goal;

at the other extreme all available robots can be sent to a new goal. Larger robot teams sent to a new exploration goal will improve availability of new agents at the location of new exploration goals are discovered. The larger group has spare robots which can be quickly allocated to explore new goals, such as those discovered when the team moves past a corridor intersection or t-junction. If the group of robots allocated to a navigation goal is too large, then the robots can interfere with each other due to local reactive control of multiple agents with respect to dynamic obstacles and limited space in corridors. The strategies selected for testing trade off *availability* (robots are close and able to explore branching structure quickly) with *non-interference* (robots do not get in each other's way).

The first coordination algorithm is called *Reserve*. In this algorithm, all unallocated robots remain a the starting locations until new exploration goals are uncovered. When a branching point is detected by an active robot, the closest reserve robot will be recruited into active status to explore the other path. This strategy has low availability because all of the reserve robots remain far away at the entrance; however, it has minimal interference because the exploring robots will usually be further away from other robots.

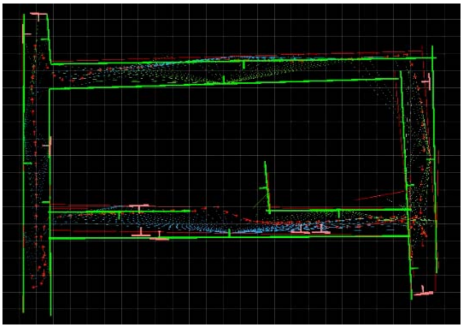
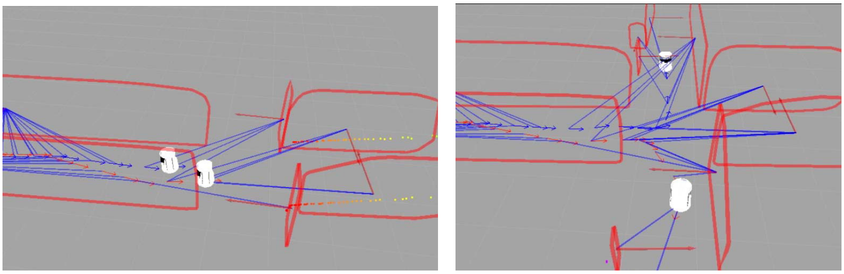


Fig. 3 A map built by three robots using the *Reserve* cooperative mapping strategy

The second coordination algorithm is *Divide and Conquer*. In this strategy, the entire robot group follows the leader until a branching point is detected. The group splits in half, with the first $\frac{n}{2}$ robots following the original leader, robot $\frac{n}{2}+1$ is selected as the leader of the second group, and robots $\frac{n}{2}+2$ through n are now members of its squad. Once there are n squads with one robot, no further divide operations can be made and new exploration goals will only be allocated once a robot has reached a dead-end or looped back into a previously explored area. This algorithm maximizes availability, but potentially causes significant interference between robots.

An example 3D map built by two robots as they approach a branch point can be seen in figure 4(a). At this point, the robot team splits and each team member takes a separate path, as seen in figure 4(b). The map shown is built concurrently with local maps built on each robot. The global map is used to establish a global frame of reference for robot collaboration message coordinates.



(a) Two robots approach the intersection. (b) Two robots split and move past the intersection

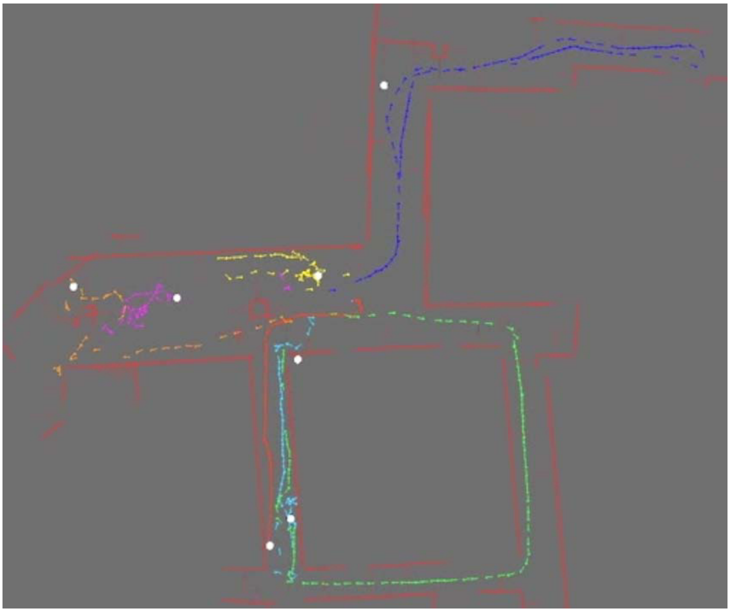
Fig. 4 An illustration of the *Divide and Conquer* exploration strategy. As the robots approach an intersection, the team must split and recruit new partner robots from the reserved units.

3 Experiments

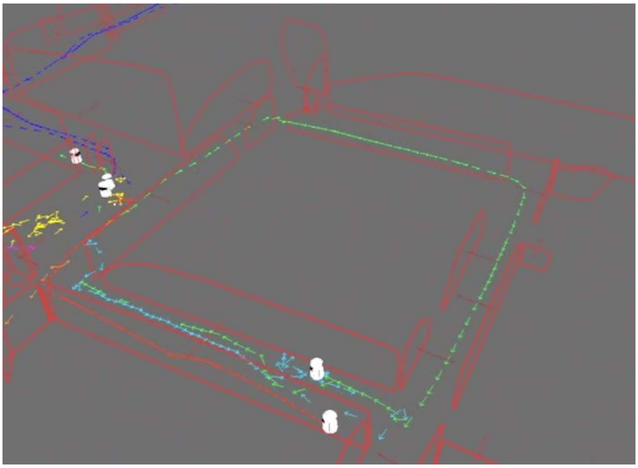
The setting for the multi-robot mapping task for this series of experiments consists of a team of robots being introduced into a single entrance in an unknown environment. Each robot is an inexpensive Willow Garage *TurtleBot*; a team of nine of these robots is shown in figure 3. The *TurtleBot* was chosen for this application due to its low cost and the ease of integrating large numbers of robots through ROS. The *TurtleBot* platform is based on the iRobot *Create* base. The robots make measurements of planes with a Kinect sensor, and use an onboard IMU together with odometry to estimate ego-motion.

We evaluated the performance of various robot coordination strategies in the multi-robot exploration and mapping task. An example scenario for the *Divide and Conquer* cooperative mapping strategy can be seen in the panorama image in figure 3.

We performed a series of experiments to demonstrate the performance of our two cooperative mapping strategies. A total of 6 runs were performed for each cooperation strategy, team size, and starting location. For each experiment run, the *TurtleBot* team explored the environment from a wedge-shaped starting configuration, which can be seen in figure 3. These experiments were performed in an office environment. In order to measure the exploration and mapping performance in each location, we chose specific starting locations which are labeled *Base1* and *Base2* in figure 3. These starting locations were chosen because the area around the robot teams could be blocked off so there is only one initial exploration frontier, directly in front of the lead robot. This initial configuration was chosen to represent a breaching behavior which would be needed for implementation of collaborative mapping in a hostile environment.



(a) A map built by seven robots in an experiment using the *Reserve* cooperative mapping strategy.



(b) The same map shown from a different angle to demonstrate 3D plane features which are used for map landmarks.

Fig. 5 Global maps gathered by a team of seven mobile robots

Fig. 6 Our nine TurtleBots used in these experiments

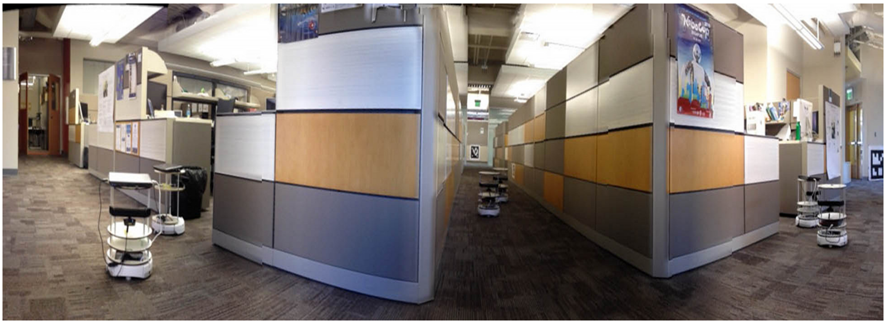


Fig. 7 An example scenario for the experiments described in this paper. Three teams of two robots are exploring the branching hallway structure in an office environment. In this illustration, the robots are using the *Divide and Conquer* cooperative mapping strategy.

4 Results

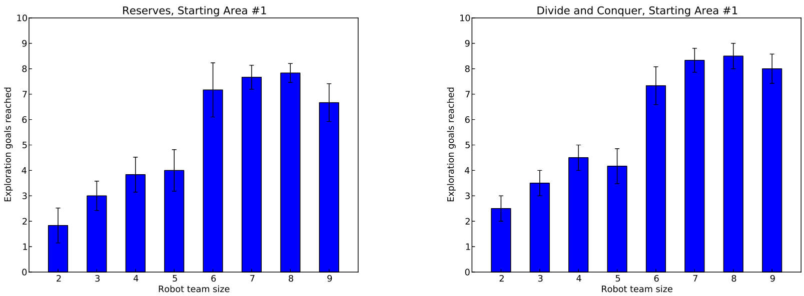
We performed a series of experiments for this paper which demonstrate team performance based upon coverage in a mapping task on an unknown office environment. Robot team sizes were varied from 2 to 9 robots. An map built with 7 robots at TurtleBots using the *Reserve* strategy is seen in Figure 5(a). An image showing the same final global map from a side view demonstrates the 3D plane features in figure 2.3.

Each of the collaboration strategy and robot team size experiments were performed from two starting locations. These starting locations are labeled *Base1* and *Base2* in figure 3. A series of interesting locations was determined in advance by examining the building floor-plan; these points of interest are also marked in figure 3. Each experiment run gets a score based on how many of these points of interest are

Fig. 8 Our office environment where the experiments were performed. The areas labeled Base1 and Base2 are the initial position of the robots. Red lines indicate artificial barricades to restrict the initial exploration of the robot teams to simulate a breach entrance into a hostile environment. Blue squares indicate the position of points-of-interest. Results are reported on the number of these points-of-interest visited by the robot team.

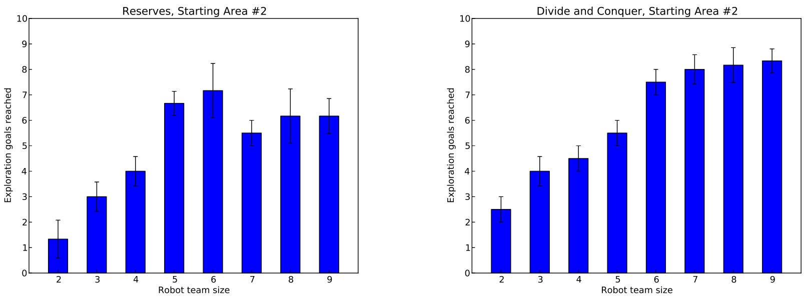
visited and mapped before a time limit is reached. This score represents the effectiveness of that algorithm and team size at providing coverage while exploring an unknown map.

In the first experiment series from *Base1* in figure 3, both strategies achieve reduced exploration coverage per robot as the team size is increased, as can be seen in the graphs in figure 9. In this starting location, there is limited space to maneuver, so both strategies generate significant interference between robots trying to move to their goals. In several instances, pairs of robots even crashed into each other due to the limited field-of-view of their sensors. We believe that the *Divide and Conquer* strategy results in figure 9(b) indicate that the team was slightly more effective than the *Reserves* strategy in figure 9(a). At the largest team size of 9 robots, the *Divide and Conquer* strategy usually visited one additional point-of-interest more than the *Reserves* strategy. Additional qualitative impressions are that the *Divide and Conquer* strategy explored the points-of-interest that it reached more quickly than with the *Reserves* strategy. For both strategies, the best team size appears to be 6 robots in this starting location.



(a) Reserves (b) Divide and Conquer

Fig. 9 Results from the first starting area



(a) Reserves (b) Divide and Conquer

Fig. 10 Results from the second starting area

In the second set of experiments, the robot teams were placed in the starting area labeled *Base2* in figure 3. As in the first experiment, the per-robot performance of both strategies decreased as the number of robots were increased. This series of experiments demonstrates a marked improvement of the *Divide and Conquer* strategy over the *Reserves* strategy as can be seen in figure 10. The *Divide and Conquer* strategy causes more robots to be making observations of exploration frontiers due to the fact that groups contain more than one robot. These additional observations of the frontier allow the *Divide and Conquer* strategy to find exploration frontiers faster than the *Reserves* strategy, and therefore explore more points-of-interest. The second experiment started from an area where there is more room to maneuver. This allowed the *Divide and Conquer* strategy to have less interference since the entire team moved together out of the starting area into the larger area before any divide operations were performed. The *Reserves* strategy still had to initially maneuver from the cramped starting location. As in the first experiment, the *Divide and Conquer* strategy qualitatively explored the environment faster than the

Reserves strategy. The best value for the number of robots is 6, which is the same value found in the first experiment.

5 Discussion

We have presented experiments which evaluate two collaboration strategies which can be used by teams of mobile robots to map and explore an unknown environment. We have also evaluated the impact of the number of robots on coverage in the exploration and mapping task.

The first collaboration strategy, called *Reserves* keeps a pool of unallocated robots at the starting location. A new robot is activated when there are more exploration frontiers than currently active robots. This strategy was intended to minimize the amount of interference between robot agents since robots would be far away from each other during exploration. The results from our experiments do not indicate that this strategy results in less interference than other strategies since performance decreases more when more robots are added in some environments. The *Reserves* strategy is significantly slower at exploring the environment than other strategies.

The second collaboration strategy, called *Divide and Conquer* has all available robots proceed in one large group. Once there are two exploration frontiers, at a corridor t-junction for example, the team will divide in half and each sub-team will follow one of the exploration frontiers. This process will be repeated with teams dividing in half each time they see branching structure in the environment. It was anticipated that this strategy would result in higher interference since robots would be maneuvering close together; however, the increased availability of robots near new exploration frontiers offsets this phenomenon.

Divide and Conquer appears to be a more effective strategy than *Reserves* for exploring and mapping an unknown environment. There are additional hybrid strategies which could now be considered such as the *Buddy System*, which modifies the *Reserves* strategy with teams of 2 robots instead of 1. We believe that this strategy will mitigate much of the slowness of the *Reserves* strategy while still minimizing interference.

Acknowledgements. This work was made possible through the Army Research Lab (ARL) MAST CTA project, and the Boeing corporation.

References

1. ARL: Army Research Lab Micro Autonomous Systems and Technology Collaborative Technology Alliance MAST CTA (2006), `http://www.arl.army.mil/www/default.cfm?page=332`
2. Bailey, T., Durrant-Whyte, H.: Simultaneous localisation and mapping (SLAM): Part II state of the art. Robotics and Automation Magazine (September 2006)

3. Chipalkatty, R., Daepp, H., Egerstedt, M., Book, W.: Human-in-the-loop: MPC for shared control of a quadruped rescue robot. In: 2011 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 4556–4561. IEEE (2011)
4. Dellaert, F.: Square root SAM: Simultaneous localization and mapping via square root information smoothing. Robotics: Science and Systems (2005)
5. Dellaert, F., Kaess, M.: Square root SAM: Simultaneous localization and mapping via square root information smoothing. International Journal of Robotics Research (2006)
6. Durrant-Whyte, H., Bailey, T.: Simultaneous localisation and mapping (SLAM): Part I the essential algorithms. Robotics and Automation Magazine (June 2006)
7. Fischler, M.A., Bolles, R.C.: Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. Communications of the ACM 24(6), 381–395 (1981)
8. Folkesson, J., Christensen, H.: Graphical SLAM - a self-correcting map. In: IEEE International Conference on Robotics and Automation (2004)
9. Fox, D., Ko, J., Konolige, K., Limketkai, B., Schulz, D., Stewart, B.: Distributed multi-robot exploration and mapping. Proceedings of the IEEE 94(7), 1325–1339 (2006)
10. Grisetti, G., Grzonka, S., Stachniss, C., Pfaff, P., Burgard, W.: Efficient estimation of accurate maximum likelihood maps in 3d. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007, pp. 3472–3478. IEEE (2007)
11. Heger, F., Singh, S.: Sliding autonomy for complex coordinated multi-robot tasks: Analysis and experiments. In: Proceedings of Robotics: Science and Systems, Philadelphia, USA (August 2006)
12. Quigley, M., Gerkey, B., Conley, K., Faust, J., Foote, T., Leibs, J., Berger, E., Wheeler, R., Ng, A.: ROS: an open-source robot operating system. In: ICRA Workshop on Open Source Software (2009)
13. Rogers, J., Trevor, A., Nieto, C., Cunningham, A., Paluri, M., Michael, N., Dellaert, F., Christensen, H., Kumar, V.: Effects of sensory perception on mobile robot localization and mapping. In: International Symposium on Experimental Robotics, ISER (2010)
14. Rogers III, J.G., Trevor, A.J.B., Nieto-Granda, C., Christensen, H.I.: Simultaneous localization and mapping with learned object recognition and semantic data association. In: IEEE International Conference on Intelligent RObots and Systems, IROS (2011)
15. Rusu, R.B., Cousins, S.: 3D is here: Point Cloud Library (PCL). In: IEEE International Conference on Robotics and Automation (ICRA), Shanghai, China (2011)
16. Trevor, A.J.B., Rogers III, J.G., Nieto-Granda, C., Christensen, H.I.: Tables, counters, and shelves: Semantic mapping of surfaces in 3D. In: IROS Workshop on Semantic Mapping and Autonomous Knowledge Acquisition (2010)
17. Vincent, R., Fox, D., Ko, J., Konolige, K., Limketkai, B., Morisset, B., Ortiz, C., Schulz, D., Stewart, B.: Distributed multirobot exploration, mapping, and task allocation. Annals of Mathematics and Artificial Intelligence 52(2), 229–255 (2008)
18. Zheng, K., Glas, D., Kanda, T., Ishiguro, H., Hagita, N.: How many social robots can one operator control? In: Proceedings of the 6th International Conference on Human-Robot Interaction, pp. 379–386. ACM (2011)