

Towards an Understanding of the Impact of Autonomous Path Planning on Victim Search in USAR

Paul Scerri, Prasanna Velagapudi, Katia Sycara
Robotics Institute
Carnegie Mellon University
{pscerri,pkv,katia}@cs.cmu.edu

Huadong Wang, Shih-Yi James Chien and
Michael Lewis
School of Information Sciences
University of Pittsburgh
jwhd82@gmail.com, gsechien@gmail.com,
ml@sis.pitt.edu

Abstract—Technology for multirobot systems has advanced to the point where we can consider their use in a variety of important domains, including urban search and rescue. A key to the practical usefulness of multirobot systems is the ability to have a large number of robots effectively controlled by small numbers of operators. In this paper, two modalities for controlling a team of 24 robots in a foraging task in an urban search and rescue environment are compared. In both modalities, multiple operators must monitor video streams from the robots to detect and mark victims on a map as well as teleoperating robots that cannot get themselves out of difficult situations. In the first modality, the operators must also provide waypoints for the robots to explore, using both video and a partially completed map to choose appropriate waypoints. In the second modality, the robots autonomously plan their paths, allowing operators to focus on monitoring the video, but without being able to interpret video streams to guide exploration. Experimental results show that significantly better overall performance is achieved with autonomous path planning, although the reduction in operator workload is not significant.

I. INTRODUCTION

Applications for multirobot systems (MrS) such as interplanetary construction[15], [19] or cooperating uninhabited aerial vehicles[2] will require close coordination and control between human operator(s) and teams of robots in uncertain environments. Human supervision will be needed because humans must supply the goals that direct MrS activity, change those goals as circumstances change, and deal with situations outside of the sensing, actuation or reasoning capabilities of the robots. Robot autonomy will be needed because the aggregate demands of decision making and control of a MrS are likely to exceed the cognitive capabilities of a human operator. Controlling robots that must act cooperatively, in particular, will likely be difficult because it is these activities that theoretically impose the greatest decision-making load[8].

For many emerging systems, the cost of finding, training and maintaining operators will far exceed the costs of acquiring more robots, hence it will be desirable to have a single operator control more than one robot. When there are multiple operators and many robots, a key question is how to divide up tasks between operators and between operators and autonomy to maximize overall system performance. Earlier work in an urban search and rescue (USAR)

domain suggested that there is substantial cost of concurrence for performing the exploration and perceptual search tasks together particularly as the number of robots increases. Wang et al's[16] operators when controlling 12 simulated robots, for example, found only half as many victims when directing robots through waypoints as they found when their task was limited to finding and marking victims. The navigation task itself appears to have approached human limits at controlling 12 robots through waypoints because the area covered was no greater when this was the operator's sole task than when the operator was required to locate and mark victims as well[16]. These results support the automation of path planning and navigation over efforts to improve automation for target recognition and cueing, provided the technical challenges were comparable.

Questions remain, however, about the suitability of automatically generated paths for human perceptual search. There are distinct qualitative differences between the paths taken by autonomous robots and those laid out by human operators. Humans are able to use camera feedback and an intuitive understanding of the environment to reason about angles and perspectives that will give them the most information while an autonomous planner cannot. In the earlier experiments paths used by operators performing the perceptual search portion of the task were taken from among those generated by previous participants whose task was to explore the environment. These paths, therefore, can be presumed to have possessed qualities of intelligibility and intuition that may have benefited operators in ways that automatically generated paths could not.

The present experiment tests this hypothesis by comparing performance of operators performing the full USAR task with others assisted by an autonomous path planner performing only the perceptual search component. While this comparison cannot rule out potentially higher performance for perceptual search using human generated paths it can address the pragmatic question of whether operator performance can be improved through the use of automated path planning. As part of an extended series of experiments investigating multi-operator as well as multi-robot control, the reported experiment contrasts pairs of operators controlling 24 robots. Because 12 robots have been shown to be at or slightly beyond the limit of human control[16] these conditions

should lead to sharing of navigation and search duties rather than specialization of tasks.

The autonomous path planning was done using a deterministic roadmap planner[10]. We do not claim the path planner as a novel contribution, but describe it to facilitate an understanding of the differences with human planning. As input, the planner used the current occupancy grid representing the joint team knowledge of the environment and some information about the planned paths of other robots. Possible locations are generated and accepted or rejected based on the expected information gain for being at that location. The expected information gain was a function of the uncertainty in the occupancy grid around that point and whether or not another robot was known to be planning to go near that point. If another robot was planning on being near that location, it was assumed that the uncertainty would be resolved. Edges were generated between nodes if the occupancy grid indicated a sufficiently high probability of being able to move between the locations. Finally, a branch-and-bound search was performed across the network of locations and edges for the path that maximized the expected information gain. Plans were allowed to backtrack, but no additional value was received for visiting a location multiple times. When a robot finished planning, it shared its planned path with some of the other members of the team to allow them to both avoid collisions and search distinct areas.

Using a high fidelity simulation environment called USARSim[4], which is based on UnrealEngine2, we compared the foraging performance of two groups of subjects. One set of subjects specified waypoints for the the robots to follow as well as monitored the video streams coming in from the robots for victims. The second set of subjects only had to watch the video streams, while the robots autonomously planned their paths through the environment. Both sets of subjects were allowed and encouraged to teleoperate any robot that got itself stuck in some location. When a victim was seen in the video stream, the subjects were required to mark the location of the victim on the map. The overall performance metric was the number of victims found. The operators worked in teams of two, managing 24 robots searching an office like environment in which victims were usually relatively easy to see provided a person was looking at the video stream when the victim came into view. The results show that, as expected, the autonomously planning robots were able to explore more area. This translated into 25% more victims found and smaller errors in the victim locations. However, somewhat surprisingly, operator workload was not significantly decreased.

II. RELATED WORK

Current estimates of human span of control are severe. Miller, for example, showed that under expected target densities, a controller who is required to authorize weapon release for a target identified by a UCAV could control no more than 13 UAVs even in the absence of other tasks[11]. A similar breakpoint of 12 was found by [7] for retargeting Tomahawk missiles. Smaller numbers (3-9) [6] have been

found for ground robots which typically require more frequent attention.

Controlling multiple robots substantially increases the complexity of the operator's task because attention must constantly be shifted among robots in order to maintain situation awareness (SA) and exert control. Because coordination demands can vary greatly across tasks[17], frequently dominate the demands on operator attention[18], and grow exponentially in the number of robots[8], coordination is an obvious candidate for automation. There remain, however, a persistent subset of tasks such as approving targets or extricating stuck robots that require human attention. When these tasks are independent as in these examples the operator can service robots sequentially and supervise many more than would be possible if their actions were dependent. Luckily many of the tasks for which operators are needed by robot teams are of precisely this type. The foraging task used for this experiment, in which each robot searches its own region, falls into this category. The operator's role in the autonomous condition of monitoring cameras from around the team while teleoperating occasional robots out of impasses is a good match for sequential independent control.

III. AUTONOMOUS PATH PLANNING

In the system, agents use occupancy grids to generate 2D maps from laser scan readings collected both locally and remotely. The implementation is a Java port of the Carnegie Mellon Robot Navigation Toolkit (CARMEN)[12]. The robots are given their ground-truth poses, to minimize the additional complexity of having users correct the robots after collisions and drift. Since this work is not focused on mapping, we argue that this simplification is reasonable.

The autonomous path planning involved some interesting challenges since the aim was to maximize information gain (or equivalently minimize entropy), rather than reach any particular objective. In this paper, the information gain is considered solely with respect to the occupancy grid created by the SLAM process. However, since the ultimate objective is finding of victims by operators using video from the robots, the occupancy grid usually over-estimates the remaining uncertainty since, in most cases, human vision will exceed laser scanners ability to resolve uncertainty about a location. However, the "extra" search performed in an area can give the humans extra time and opportunities to spot hard to see victims.

Using a goal of information gain maximization means that paths leaving a starting location in any direction can be just as good as any other direction and that path features such as backtracking are not necessarily bad. The consequence of this is a very high branching factor on any search expansion and therefore high computational complexity. Previous work simplifies this problem by selecting goal points at the frontier of the explored area and using conventional planning techniques to reach those points[3], [20], [9]. However, this can be arbitrarily inefficient when the robot can collect information along the entire length of its path.

Our initial approach was to apply a modified RRT based cooperative information gain approach developed for UAVs[13]. However, it turned out to be difficult to tune to a complex, indoor environment. Thus, we implemented a roadmap planner to do the autonomous planning[10]. The roadmap planner did not have the flexibility of the RRT to find the highest quality paths for information gain, but it focused the robot on moving to locations where there was the most uncertainty to be resolved. However, by planning through a sequence of locations, all with some value and potentially reasonably close together, the planner should have been more efficient than a pure frontier based planner.

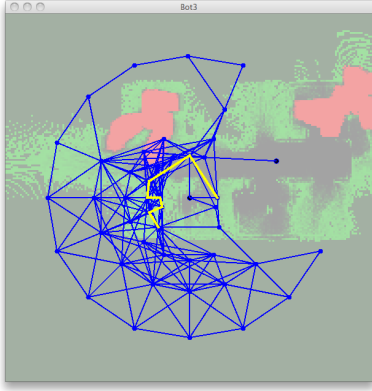


Fig. 1. A screenshot of the path planning debugging interface. Green background shows unexplored areas, which brighter green representing higher uncertainty. Red background is the proposed path of another robot. Blue lines connecting blue circles show the possible locations and edges. The yellow line shows the planned path from the center of the window.

A. LOCATION SELECTION

An initial set of locations was selected by sampling concentric circles around the initial location of the robot. The circles are exponentially bigger as they get further from the robot. This balanced the need to have more locations near the robot, to allow it to plan around local obstacles, but still create long plans to get from one part of the environment to another and keep the number of locations down to keep the search time down. Locations were sampled at 15 degree increments around the circle.

The locations were then filtered based on two factors. First, the location needed to have no cells with probability greater than 0.8 of being occupied within a fixed, small distance. This helped keep the robots away from obstacles in the environment. If the location was accessible, the planner would look at the value of being in that location, which was a weighted sum of the amount entropy at that location and the probability of another robot being near that location. A threshold was set such that robots would avoid locations where there was significant probability of any of their team mates being, unless there was very high entropy in the area. The information gain for an area was computed as the inverse of the current uncertainty, assuming that uncertainty would be completely resolved if the robot visited that location.

As discussed below, planned paths were shared with other robots to allow cooperation. A robot does two things with a path shared from by robot. First, it assumes that all uncertainty near the planned path of that robot will be resolved, therefore there is no value in visiting the same locations. Second, it treats the path of the other robot as a low probability obstacle, to discourage interactions that will slow the progress of both robots. In some cases of high congestion, some robots will not move at all due to the number of planned paths in their area. Better solutions to cases where many robots are in a small area are required, since this algorithm is sometimes very inefficient.

B. COMPUTING TRAVERSABILITY

Once a useful set of locations were found, edges were added between locations within a fixed distance of one another. The occupancy grid was used to remove any edges where there was not a reasonable probability of being able to traverse that link. Although more sophisticated computations may improve performance, we found that simply integrating the probability of being occluded on a straight line between the locations provided a good balance between speed and performance.

In open areas, this approach resulted in a very large number of edges, increasing the complexity of the search. To keep the search time in check, edges were removed so that each location had a maximum of eight edges. The edges with the lowest probability of being successfully traversed were removed. This pruning was predominantly used between the relatively dense locations near the robot, when the robot was in an open area. Hence the pruning typically had no material impact on the eventual path.

C. GRAPH SEARCH

A branch-and-bound search is used to find a plan from the graph produced in the previous two steps. The search maximizes the expected value of the path, taking into account both the value of the locations visited and the probability of successful traversal of the edges joining them. The best path will have high probability of reaching locations that will result in significant information gain. Since there is no goal point, the heuristic estimate of remaining possible utility assumes that the path could be expanded from the current point to a series of high value locations with high probability of reaching those locations. Nodes are not expanded if the probability of reaching that point becomes too low, such that an existing best path exceeds the expected value of any future path from that node.

The aim of the search is to maximize information gain, not reach any particular point. Therefore, back-tracking over a path may have some value, however we assume that visiting the same location a second time provides no value. Visiting a location near another location reduces the value accumulated at the second location. This primarily happens near the starting point of the robot, where locations are close to one another.

The search is cut off after at most 100,000 expansions, simply to ensure robot reactivity. Even when a full 100,000 expansions were required, the search took well under 0.5s.

D. COOPERATION

Since the aim of this work is to find victims via humans monitoring video feeds from robots, the requirements on robot coordination are more relaxed than for some other applications. Some overlapping of paths and revisiting of locations is reasonable, since human operators can easily miss details or require a second look to ascertain whether a victim is present at some location. Thus, if the robots mostly spread out and search unexplored areas, good performance should be expected.

To facilitate some level of cooperation, the robots are required to share their planned paths with other members of the team. Paths are only shared with robots in easy communication reach and no effort is made to ensure complete propagation of the path information. In practice, we expect this to work well, since it is generally more important to know the paths of robots within coordination range than those far away. If the robots are so close that they interfere with one another, search times can be dramatically impacted. The shared planned paths help limit this direct interaction.

IV. MRCS

The experiment was performed using the USARSim simulation environment with 24 simulated P2ATs performing urban search and rescue (USAR) foraging tasks. USARSim is a high-fidelity simulation of urban search and rescue robots and environments, developed as a research tool for the study of HRI and multi-robot coordination. USARSim supports HRI by accurately rendering user interface elements (particularly camera video), accurately representing robot automation and behavior, and accurately representing the remote environment that links operator awareness with robot behaviors. It can be downloaded from <http://www.sourceforge.net/projects/usarsim> and serves as the basis for the Virtual Robots Competition of the RoboCup Rescue League. USARSim uses Epic Games' UnrealEngine2 to provide high fidelity rendering at low cost, while using MathEngine's Karma physics engine to support high fidelity rigid body simulation. Validation studies showing close agreement in behavior between USARSim models and real robots being modeled are reported in [1], [4]. Further validation showing agreement for a variety of feature extraction techniques between USARSim images and camera video are described in [4] and detection of walls and associated Hough transforms for a simulated Hokuyo laser range finder are described in [5]. Other sensors including sonar and audio are also accurately modeled.

MrCS (Multi-robot Control System), a multirobot communications and control infrastructure with accompanying user interface developed for experiments in multirobot control and RoboCup competition, was used in these experiments. MrCS provides facilities for starting and controlling robots in the simulation, displaying camera and laser output, and

supporting inter-robot communication through Machinetta, a distributed multiagent system[14]. Figure 2 shows the elements of MrCS. The operator selects the robot to be controlled from the colored thumbnails on the left of the screen. To view more of the selected scene shown in the large video window the operator uses pan/tilt sliders to control the camera. Robots are tasked by assigning waypoints on a heading-up map on the Map Viewer (bottom right) or through a teleoperation widget (upper right). The current locations and paths of the robots are shown on the Map Data Viewer (bottom right).

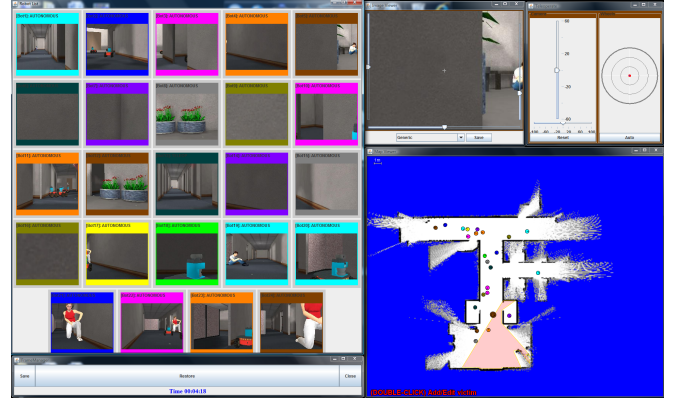


Fig. 2. The MrCS user interface for 24 robots. Thumbnails of robot camera feeds are shown on the left, a video feed of interest in the top, middle. A GUI element in the top right allows teleoperation and camera pan and tilt. The bottom right shows the current map and allows operators to mark victims.

V. RESULTS

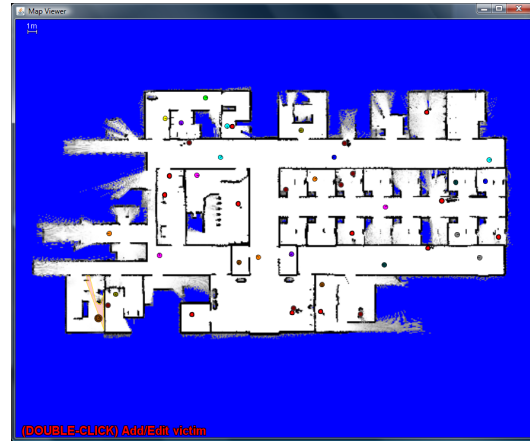


Fig. 3. Map at completion of run with 24 robots. Victims are marked with red circles.

A. EXPERIMENTAL CONDITIONS

A large USAR environment previously used in the 2006 RoboCup Rescue Virtual Robots competition was selected for use in the experiment. The environment was an office environment with many rooms and obstacles, such as chairs, desks, cabinets, and bricks. Victims were evenly distributed throughout the environment. A second simpler environment

was used for training. The experiment followed a *between groups* design. Each task was performed by a team of 2 operators. The teams in the autonomous condition were assigned 24 robots with autonomous path planning capability but could also control one robot at a time via teleoperation. Participants were instructed to use teleoperation only for helping stuck robots, not for exploration. The teams in the manual control condition were assigned 24 robots, for which participants could issue sequences of waypoints as well as teleoperate any robot.

The users were seated at separate interfaces, were able to control the same robots and watch the same video and were able to communicate freely with one another. They were not given any specific instructions on how to coordinate, although they were told it was a cooperative task.

60 paid participants (30 teams) were recruited from the University of Pittsburgh community. None had prior experience with robot control although most were frequent computer users.

B. PROCEDURE

After collecting demographic data, each participant read standard instructions on how to control robots via MrCS. In the following 25 minute training session, participants in all conditions practiced control operations. Participants were encouraged to find and mark at least one victim in the training environment under the guidance of the experimenter. After the training session, participants began the experimental session (25 minute) in which they performed the search task using 24 robots. Upon completion the participants were asked to complete a NASA-TLX workload survey.

C. EXPERIMENTAL DATA

Overall participants were successful in searching the environment in both conditions finding as many as 21 victims per team on a trial. The average number of victims found was 15.86 in the autonomous condition but only 12.33 for the manual control condition. A T-test (Figure 4) showed that there was a significant difference for victims found in the two conditions ($t(27) = 3.270, p = .003$). The region explored also showed a significant advantage ($t(27) = 6.799, p < .001$) for the autonomous condition (Figure 5). The extra exploration was due to the autonomous robots moving almost constantly, while in the manual case, an average of 7.66 robots were left after being given a single set of waypoints, while an average of 4.26 received no waypoints at all. It is clear from this result that taking the cognitively and time demanding task of exploration away from the operator and automating it helped overall system performance.

The participants' basic task was to mark victim locations as accurately as possible on the map being made by the robots. Comparing accuracy in marking victims directly, RMS Error (Figure 6) shows participants in the autonomous condition were significantly more precise ($t(27) = -3.424, p = .002$) than manual control participants, although accuracy in both groups varied from 0.27 to 0.83 m. This is surprising, since it was expected that with automated

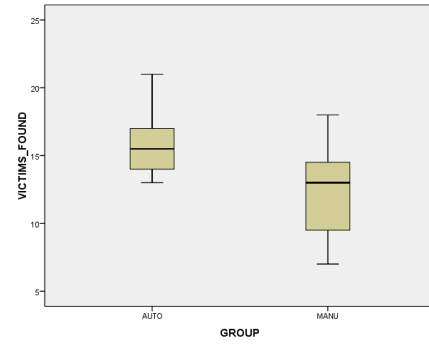


Fig. 4. Number of victims found in manual and automated modalities. Significantly more victims were found when robot path planning was automated.

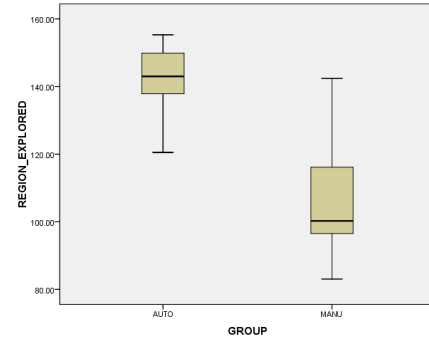


Fig. 5. The amount of area explored in manual and automated modalities. Significantly more area was explored in autonomous mode, primarily because all robots were exploring at all times.

planning, the usually inferior camera angles would lead operators to make larger errors. It is possible that the extra time available from the operator not thinking about exploration allowed them to be more precise with their marking of victim locations.

A one-way ANOVA was used to test the difference in the Victim/Region ratio among the autonomous and manual control conditions. The victims found per square meter (Figure 7) had no significant difference across the two conditions ($F(1, 27) = 0.185, p = .671$). This suggests that the difference in the number of victims found may have been due to the larger area searched.

No significant difference ($t(56) = -1.055, p = .296$) in mental workload was found in the T-test between the condi-

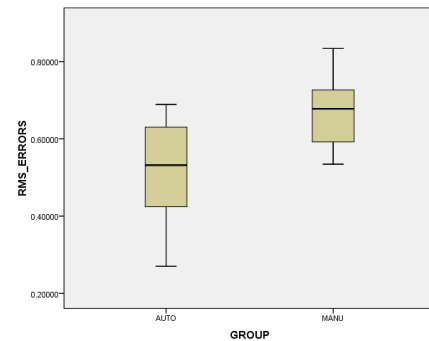


Fig. 6. RMS Error in distance between actual victim location and marked victim location. The RMS is smaller in the autonomous mode.

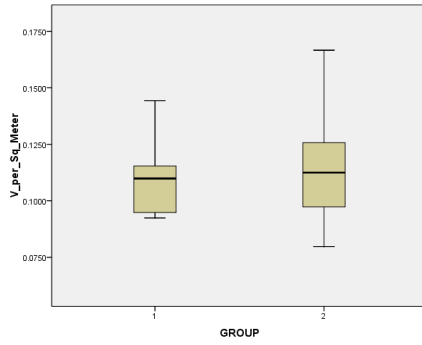


Fig. 7. Victims found per square meter of explored space. The difference is not significant.

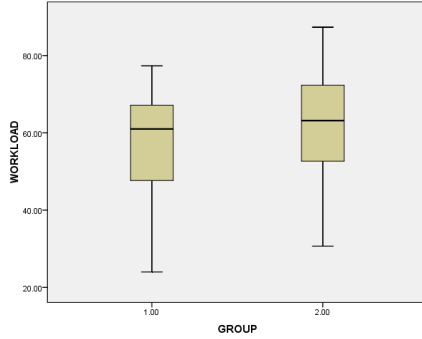


Fig. 8. NASA-TLX Measurement of mental workload. The difference is not significant.

tions, although the graph suggests a small advantage for the autonomous condition (Figure 8). When operators controlled 12 robots individually [16], by contrast, workload was substantially lower when they were relieved of path planning. We believe this is because in teams operator organization was qualitatively different between manual and automated path planning conditions. Because manual operators could only control 12 or fewer robots they were forced to divide responsibility over the robots in order to control them. In the automated condition, by contrast, both operators undertook to monitor the full team of 24 robots leading to a high level of workload comparable to that of the manual operators. More detailed cognitive modeling will be required to understand how the operators cognitive processes have changed.

VI. CONCLUSIONS AND FUTURE WORK

This paper has shown initial evidence that the benefits of autonomous, cooperative path planning for USAR outweigh the drawbacks. The increase in the number of victims found was primarily due to the larger area searched within the time limit. The increased exploration was in turn due to the robots not having to wait for operators to provide additional waypoints. Surprisingly, the operators marked locations more accurately in the autonomous mode, perhaps because they had more attention to focus on the task. However, the mental workload of the operators was not substantially decreased by taking away this part of their task. The results suggest that autonomous path planning is useful for overall system efficiency, even if the operator's attention must move to other things and their workload, therefore, does not drop.

While this work provides an initial, important data point, much follow up work is required. We are currently performing experiments that use different numbers of robots to determine whether these benefits disappear when the overall workload is lower. We are also looking to replicate the experiments with exactly the same interface and a team of physical iRobot Create robots (using fiducials for navigation instead of a laser scanner) to determine whether users deal with real video differently to simulated video. Finally, we are working on updates to the path planner that use the laser scan data to infer how difficult it might be for the operator to interpret video and adjust paths accordingly.

REFERENCES

- [1] S. Balakirsky, S. Carpin, A. Kleiner, M. Lewis, A. Visser, J. Wang, and V. Zipara. Toward heterogeneous robot teams for disaster mitigation: Results and performance metrics from robocup rescue. *Journal of Field Robotics*, 2007.
- [2] R. Beard, D. Kingston, M. Quigley, D. Snyder, R. Christiansen, W. Johnson, T. McLain, and M. Goodrich. Autonomous vehicle technologies for small fixed-wing uavs. *Journal of Aerospace Computing, Information, and Communication*, 2(1), 2005.
- [3] W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proc. of ICRA'00*, 2000.
- [4] S. Carpin, T. Stoyanov, Y. Nevatia, M. Lewis, and J. Wang. Quantitative assessments of usarsim accuracy. In *Proc. of PerMIS*, 2006.
- [5] S. Carpin, J. Wang, M. Lewis, A. Birk, and A. Jacoff. High fidelity tools for rescue robotics: Results and perspectives. In *Robocup 2005 Symposium*, 2005.
- [6] J. W. Crandall, M. A. Goodrich, D. R. Olsen, and C. W. Nielsen. Validating human-robot interaction schemes in multitasking environments. *IEEE Transactions on Systems, Man, and Cybernetics, Part A*, 35(4):438449, 2005.
- [7] M. Cummings and S. Guerlain. An interactive decision support tool for real-time in-flight replanning of autonomous vehicles. In *AIAA Unmanned Unlimited Systems, Technologies, and Operations*, 2004.
- [8] B. Gerkey and M. Mataric. A formal framework for the study of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939954, 2004.
- [9] R. Grabowski, P. Khosla, and H. Choset. Autonomous exploration via regions of interest. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS)*, 2003.
- [10] J.C. Latombe. *Robot motion planning*. Springer, 1991.
- [11] C. Miller. Modeling human workload limitations on multiple uav control. In *Proc. of the Human Factors and Ergonomics Society 47th Annual Meeting*, 2004.
- [12] M. Montemerlo, N. Roy, and S. Thrun. Perspectives on standardization in mobile robot programming: the carnegie mellon navigation (carmen) toolkit. In *Proc. of IROS'03*, 2003.
- [13] P. Scerri, R. Grinton, S. Owens, D. Scerri, and K. Sycara. Geolocation of rf emitters by many uavs. In *AIAA Infotech@Aerospace 2007 Conference and Exhibit*, 2007.
- [14] P. Scerri, D. V. Pynadath, L. Johnson, P. Rosenbloom, N. Schurr, M. Si, and M. Tambe. A prototype infrastructure for distributed robot-agent-person teams. In *Proc. of AAMAS'03*, 2003.
- [15] L. Schooley, B. Zeigler, F. Cellier, and F. Wang. High-autonomy control of space resource processing plants. *IEEE Control Systems Magazine*, 13(3):29–39, 1993.
- [16] H. Wang, M. Lewis, P. Velagapudi, P. Scerri, and K. Sycara. How search and its subtasks scale in n robots. In *Proc. of HRI*, 2009.
- [17] J. Wang and M. Lewis. Assessing coordination overhead in control of robot teams. In *Proc. of 2007 IEEE International Conference on Systems, Man, and Cybernetics*, 2007.
- [18] J. Wang, H. Wang, and M. Lewis. Assessing cooperation in human control of heterogeneous robots. In *Proc. of the Third International Conference on Human-Robot Interaction*, 2008.
- [19] S. Yun and D. Rus. Optimal distributed planning of multi-robot placement on a 3d truss. In *Proc. of IROS'07*, 2007.
- [20] R. Zlot, A. Stentz, MB Dias, and S. Thayer. Multi-robot exploration controlled by a market economy. In *Proc. of ICRA'02*, 2002.