ResearchGate

See discussions, stats, and author profiles for this publication at: https://www.researchgate.net/publication/2873269

The SDR Experience: Experiments with a Large-Scale Heterogeneous Mobile Robot Team

Article · June 2004

DOI: 10.1007/11552246_12 · Source: CiteSeer

CITATIONS

51

READS

52

3 authors, including:

Lynne Parker

University of Tennessee

174 PUBLICATIONS **7,286** CITATIONS

SEE PROFILE

Some of the authors of this publication are also working on these related projects:

Project

Cooperative Cognitive Control for Autonomous Underwater Vehicles (Co3-AUVs) View project

All content following this page was uploaded by Lynne Parker on 15 February 2013.

The user has requested enhancement of the downloaded file.

The SDR Experience: Experiments with a Large-Scale Heterogeneous Mobile Robot Team

Andrew Howard[1], Lynne E. Parker[2], and Gaurav S. Sukhatme[1]

[1] Robotics Research Laboratory
Department of Computer Science
University of Southern California
http://robotics.usc.edu/

[2] Distributed Intelligence Laboratory
Department of Computer Science
University of Tennessee
http://www.cs.utk.edu/∼parker/Distributed-Intelligence-Lab/index.html

Abstract. This paper reports on experiments conducted as part of the DARPA SDR (Software for Distributed Robotics) program. The core challenge for this program is to develop a system capable of carrying out 'locate-and-protect' missions: the system must be able to deploy a large number of robots into an unexplored building, map the building interior, locate a valued object, detect and track intruders, and transmit all of the above information to a remote observer/operator. To satisfy these requirements, we have developed a large heterogeneous robot team consisting of approximately 80 robots. This paper sketches the key technical elements of this system, and presents selected results from externally supervised experiments conducted in a 600 m^2 indoor environment.

1 Introduction

This paper describes a multi-robot system designed to meet a very strict set of requirements: to explore and map a single story in a large indoor environment, to detect a valued object, to deploy a sensor network and to use this network to track intruders within the building. This system must also operate autonomously, and employ as *many* robots as possible. These are the specific requirements imposed the DARPA SDR (Software for Distributed Robotics) locate-and-protect mission.

To meet this challenge, we have constructed a large robot team consisting of approximately 80 robots. Cost and power considerations dictate that the bulk of these robots should be relatively simple (i.e., with minimal sensor and computation capabilities). The robots must, however, retain the ability to make maps of the environment, detect valued objects, and navigate safely – tasks that would normally require the use of expensive ranging sensors such as scanning laser range-finders or stereo vision. To solve this apparent conundrum, we have adopted a heterogeneous approach, utilizing three distinct *classes* of robots with varying capabilities. The team therefore consists of: a small number of highly capable *mapping robots* equipped with scanning laser range-finders, cameras and unique fiducials; a small number of

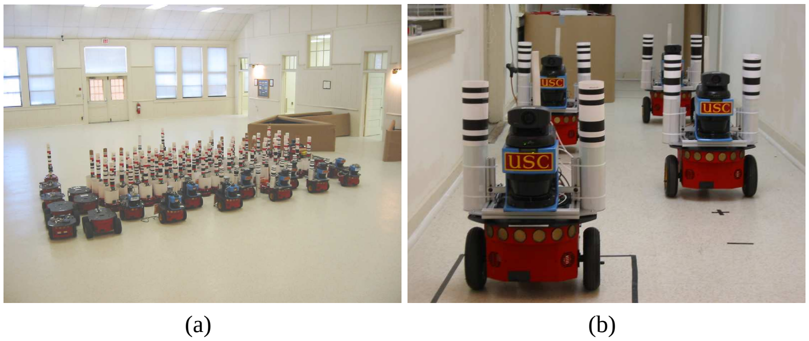

(a) (b)

Fig. 1. (a) The heterogeneous robot team, with three classes of robots (mapper, leader and sensor); the sensor robots are based on the ActivMedia AmigoBot. (b) The mapping sub-team: note each of the four robots carries a unique laser-visual fiducial; these fiducials can be detected and identified at ranges in excess of 8m, and the relative range and bearing of the robots can be determined to within a few centimeters and a few degrees, respectively.

slightly less-capable *leader robots* equipped with scanning laser range-finders and cameras; and a large number (approximately 70) of simple *sensor robots* equipped with a microphone and a crude camera. All of the robots are have 802.11b WiFi, and a modified ad-hoc routing package (AODV) is used to ensure network connectivity.

With this heterogeneous team, the locate-and-protect mission is divided into two distinct phases: exploration and mapping, and deployment and detection. For the first phase, the mapping sub-team explores the environment, acquires an occupancy grid and locates the valued object. Exploration is coordinated, and mutual observations are used to solve difficult correspondence problems (the mapping robots observe one other, and thereby identify loops in the environment). In the second phase, the acquired map is used to compute a set of deployment locations, and the simple sensor robots are deployed to these locations using an *assistive navigation* technique. That is, since the sensor robots are not capable of navigating by themselves, they are guided into position by the more capable leader robots. Once deployed, the sensor robots collaborate to form a distributed acoustic sensor network that tracks audible targets moving through the environment.

In the sections that follow, we consider these two phases in greater detail. We also describe selected experiments used to validate the system; these experiments are distinguished by the fact that they were carried out under *external supervision*. Specifically, an independent government team selected and prepared the experimental site and supervised the conduct of experiments (monitoring completion times, user interventions, code modifications and so on). At no time during the experiments were operators allowed to enter the site, and prior access was limited to certain prescribed areas. Thus, at the outset, most of the environment was genuinely unknown to the human operators.

2 Exploration and Mapping

For exploration, we employ a fully decentralized frontier-based approach [1,10], using local occupancy grids and minimal communication between the robots. This exploration algorithm is fairly elementary, and will not be treated in detail here; we note only that the algorithm scales linearly with team size, ensures minimal interference between robots, and achieves asymptotic coverage of the environment.

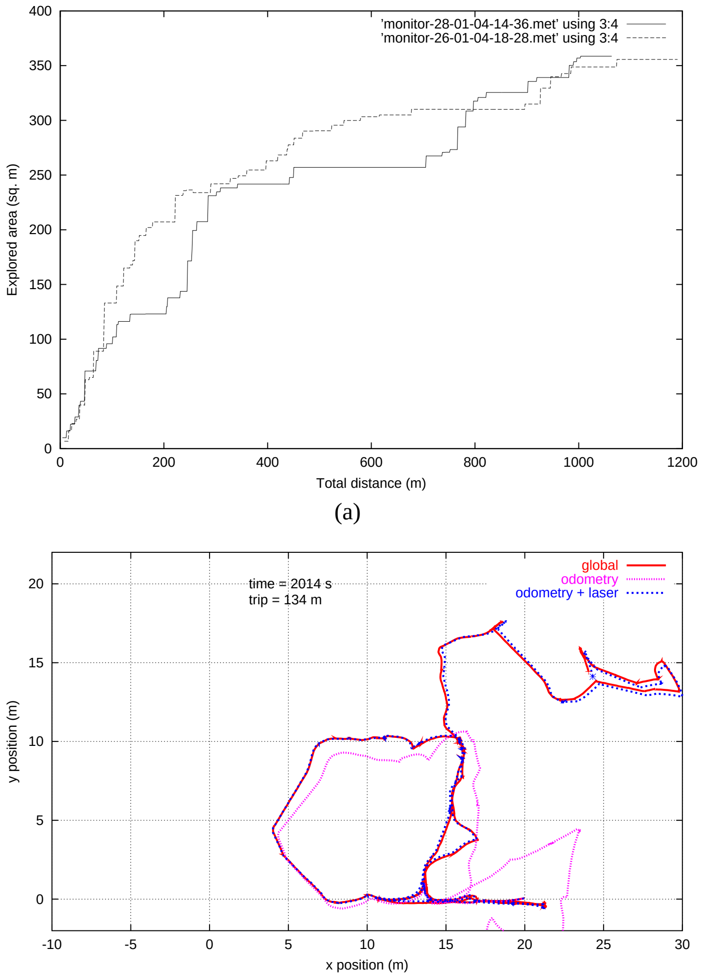
Some results from the final experiments are plotted in Figure 2(a). This plot shows the net area explored by a team of four robots as a function of total distance traveled (i.e., the sum of the distances traveled by all four robots). Note the rapid initial exploration, followed by slow convergence to full coverage; the latter is product of the strictly local nature of the algorithm: robots are not using global knowledge, and are not sharing information about previously explored areas. Faster convergence may be possible with the addition of global knowledge and/or coordination, albeit at the cost of increased system complexity.

In contrast with exploration, the mapping algorithm employs both centralized and decentralized components. Each robot in the team uses an incremental SLAM algorithm to maintain an independent local pose estimate; laser scan data is used to correct most, but not all, of the odometric drift. These independent local pose estimates (along with the raw laser scan data) are transmitted to a remote console, where they are combined and corrected to form single map of the environment. We will consider each of these two components in turn.

For estimating local pose, each robot employs an incremental maximum likelihood filter [9] that is similar in spirit – if not in detail – to that described in [3]. The state vector for this filter consists of two components: a local pose estimate and a local map. Roughly speaking, the local map is simply a finite-length queue containing recent laser scan data; each incoming scan is fitted against this local map (using a dense scan-matcher), thereby inducing an updated pose estimate for the robot. This estimate will, of course, drift over time (an unavoidable feature of every incremental method); the drift *rate*, however, is one or two orders of magnitude slower than that seen with odometry alone. See, for example, the results plotted in Figure 2(b): while the odometry estimates quickly diverge, the error in the local pose estimate is less than less 0.1 m after 100 m of travel.

There is no attempt by individual robots to close loops or merge data from other robots; this is the role of the remote console, which aggregates local pose estimates and range scans from each of the mapping robots into a central repository. The overall map is assembled from this data using a global SLAM algorithm that is built around three key technologies: maximum likelihood estimation, manifold representations and loop closure using mutual observation.

Maximum likelihood estimation (MLE) is used to generate globally consistent maps; put simply, MLE determines the set robot trajectories that minimizes the inconsistency between overlapping laser scans [5]. In practice, this is a high-dimensional optimization problem that must be solved using sparse graph-based representations and numerical optimization algorithms (modern algorithms allow simultaneous optimization over hundreds of variables and thousands of constraints). In this context,



(a)

(b)

Fig. 2. (a) Coverage plot for four robots: the plot shows the area explored by the robots as a function of the total distance traveled; coverage is determined by a post-hoc analysis of the data, once a complete map is available. (b) Comparison of robot pose estimates using odometry and incremental SLAM; the incremental pose error is less that 0.1 m after 100 m of travel. Note that the ground-truth trajectory is determined post-hoc using a global localization algorithm and a complete map of the environment.

it should be noted that the use of local pose estimates in the place raw odometry greatly simplifies the optimization problem, as the local and global estimates diverge comparatively slowly.

For this project, we have made one important extension to the basic MLE formalism: instead of treating the map as a two-dimensional planar structure, we represent

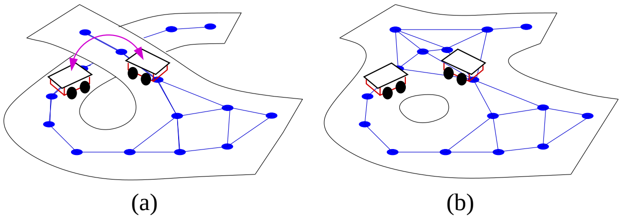


Fig. 3. A manifold map representation, before and after loop closure. The manifold is a self-consistent – but redundant – representation, in which the same point in the world may appear more than once on the manifold. (a) Robots at remote locations in the manifold observe that they are proximal in the world (i.e., the robots detect one another using their sensors). (b) This information is propagated through the manifold, allowing the loop to be closed.

it using a two-dimensional *manifold* [4]. Unlike planar representations, the manifold representation is always *self-consistent* irrespective of whether or not loops have been closed. To achieve this self-consistency, the manifold representation must sacrifice *uniqueness*; i.e., a single location in the world may be represented more than once in the manifold (see Figure 3). Thus, in this representation, loop closure is reduced to the problem of identifying and bringing together those points on the manifold that represent the same point in the world.

To recognize such points, we make use of mutual observations: if two robots are far apart on the manifold, but are proximal in the world (i.e., they can see one another), we can infer a new set of correspondence points and close the loop. It should be noted that this active, multi-robot approach to loop closure entirely side-steps the hard correspondence problems that bedevil global SLAM algorithms.

Exploration and Mapping Results

Five exploration and mapping trials were performed under government supervision, all of which resulted in successful map generation. The first three trials were performed with identical software and initial conditions; the final two trials were performed under 'challenge' conditions designed to test the system envelope.

The most difficult of the challenge conditions involved a team of four mapping robots deployed from two different locations (i.e., two robots from each of two entry points). Since the relative pose of the two entry points was unknown, each pair of robots was required to explore and map independently, giving rise to two unconnected maps or 'islands'. After approximately 10 minutes, however, the two teams encountered one another, and as a result of this mutual observation, the two maps were merged into one. The combined robot team then proceeded to complete the exploration task, yielding the final occupancy grid map shown in Figure 4. It should be emphasized that this exploration and mapping process was entirely autonomous, with the exception of a single user intervention to direct a robot into the otherwise unexplored room at the bottom left of the map (this intervention was

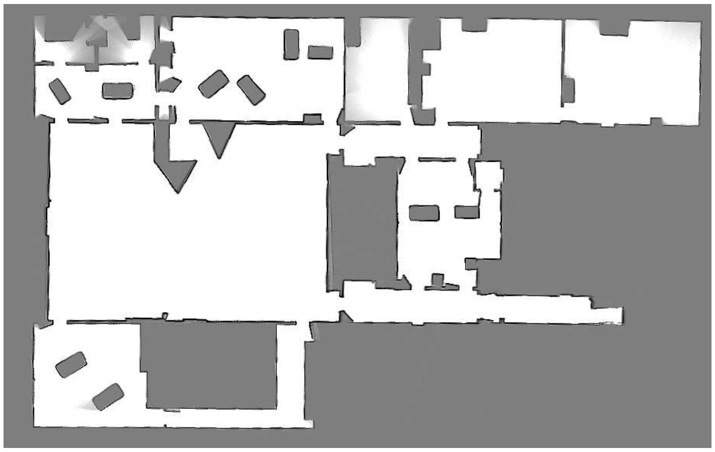


Fig. 4. Occupancy grid map produced during a multiple-robot, multiple-entry trial: two robots entered from the door at the right top, another two robots entered from the door at the right bottom (the relative pose of the two doorways was unknown). The environment is approximately 45 by 25 meters in size, with an internal area of 600 m^2.

not strictly necessary, was performed to achieve faster map completion). Maps were generated in real time, for an environment approximately 600 m^2 in size. Map accuracy is comparable to or better than that achieved by a human survey team using a tape-measure, pencil and paper.

3 Deployment and Detection

The intruder detection and tracking task requires the deployment of a large number of simple sensor robots. These robots have low-fidelity microphones to serve as a distributed acoustic sensor network, but no sensors for localization or obstacle avoidance. Because they cannot navigate safely on their own, our approach provides cooperative assistive navigation to the sensor robots through the use of more capable leader robots. Using the map generated in the first phase, these leader robots are able localize themselves and guide the sensor robots to their deployment positions. The basic method is as follows: sensor robots are assembled into chains behind a leader using simple color blob tracking (multi-robot follow-the-leader); once a deployment destination is reached, a single robot in the chain is autonomously ‘tele-operated’ by the leader to the correct position, using the camera mounted on the leader; the leader and the remaining chain then proceed to the next deployment position. Thus, the leader visits a series of locations in turn, and deploys a single sensor robot at each. Figure 5 shows a series of snapshots of our navigational assistance system in operation. Details of this deployment strategy are reported in [7].

With this heterogeneous cooperative approach, it is important to develop a thoughtful deployment strategy to ensure that the sensor robots will be positioned in the desired distribution. An especially challenging problem is for the leader robots to follow paths that the simple robots can follow, to ensure that the sensor robots reach their desired positions while moving in a follow-the-leader formation. Obviously, the navigational challenges grow if the leader robots move to random sensor node deployment positions without taking into account the formation of robots that is following behind. Our approach begins with four planning steps. Step 1 generates the planned sensor deployment positions to meet several criteria, including minimizing pathway obstruction, achieving a minimum distance between sensor robots, and maximizing visibility coverage. Step 2 then generates the waypoints of the path that the leader robot must follow to guide the sensor robots to the vicinity of the sensor deployment positions. Figure 6 shows a set of autonomously planned sensor and leader waypoint positions for the earlier map. Since each leader robot can only deploy a few sensor robots at a time, and since several leader robots are available to operate in parallel, Step 3 divides the sensor deployment positions into groups to facilitate the deployment operation. Each group of positions is assigned to a team (consisting of one leader robot and several sensor robots) for deployment. The team assignments are generated to achieve the objectives of: (1) minimizing travel by the deployment teams, (2) minimizing interference between teams, and (3) minimizing the amount of turning a team must perform as it travels to all of its assigned deployment positions. The traffic management of Step 4 incorporates deployment delays to coordinate multiple deployment teams operating in parallel, similar to [2]. Details of Steps 1-3 of the planning process are reported in [8].

Once the acoustic sensor robots have been deployed, they activate their distributed acoustic sensor net to detect targets that are moving through the environment. Figure 7 shows these robots deployed in the planned sensor net positions and acting as this sensor network. Since our sensor robots are equipped with only a very simple microphone, we assume that the target moving through the environment is making some detectable noise, and that the target is the only source of sound. Our approach involves each robot filtering its acoustic data and then communicating its volume heard to its local neighbors. The robot that detects the highest volume in its neighborhood (above a threshold) reports its own position as the current target position estimate to the operator control unit. Details of this distributed acoustic sensor network process are reported in [6].

Deployment and Detection Results

In three separate trials supervised by the government team, our distributed acoustic sensor network achieved 100% detection of targets in the environment (all targets we localized to the correct room or corridor) with no false positives; these trials involved 3 leaders and up to 35 sensor robots. Clearly, these results show that the planned sensor positions and the distributed acoustic sensor processing worked well for the objectives of these experiments.

Another metric of evaluation is the percentage of sensor robots successfully deployed (i.e., the ratio of successful deployments to attempted deployments). Our experimental data shows an overall deployment success rate of 60% - 90%, depending upon the environmental characteristics. In other words, for each attempt at deploying a simple robot, 60% - 90% of those robots successfully reached their planned deployment position. The reason for the low end of this success rate is the complexity of our heterogeneous robot system. Our system for simple robot deployment is composed of several non-trivial modules, including localization, path planning, navigation, leader following, visual marker detection, and inter-robot communication. The successful completion of the entire deployment process depends upon the successful completion of all of the system modules while the robots are operating in cluttered environments along complex paths. Additionally, the independent experimentation reported here was especially challenging because we forced the robot team to deal with the consequences of prior deployment failures. Thus, subsequent robot team deployments had to deal with situations such as partially blocked doorways if a prior deployment resulted in a simple robot being caught on the doorway. If all the test runs had been independent, the overall system success rate would certainly have been higher.

Clearly, there are many potential failure modes in such a complex heterogeneous system involving such a large number of robots. The most common failure modes of the system were caused by variable lighting conditions (which could cause the sensor node robots to lose the color blobs, or the leader robots to lose the visual marker for autonomous tele-operation), cluttered environments (which could cause the follower sensor node robots to lose the leader robot amidst many navigational twists and turns), and communications failures (due to delays in multi-hops in the wireless ad-hoc network). To account for these potential subsystem failures, we built extensive fault tolerance into the behavior of the leader robot. Table 1 shows the set of base failure states identified for this system and the implemented recovery action. Using these methods of behavior fault tolerance, the success rate of the leader robots making it back home autonomously in these rigorous experiments was 91% (over 45 trials).

Table 1. Failure states detected by the leader robot and implemented recovery actions.

Failure Type	Fault Recovery Action
Can’t reach waypoint	Re-plan path.
Lost simple robot	Leave lost robot in wait state and move on to next robot in chain.
Leader robot camera failure	Leave simple robot(s) in wait state, send camera failure feedback to human operator and return home.
Simple robot motor failure	Check if simple robot is close to goal; if so, change simple robot state to sensor detection and proceed; else, leave simple robot in wait state and proceed.
Localization drift	Check if simple robot is close enough to goal; if so, change simple robot state to sensor detection and proceed; else, leave simple robot in wait state and proceed;
Lost marker	Leave simple robot in wait state and move on to next robot in chain.
Communication failure	Return back home.

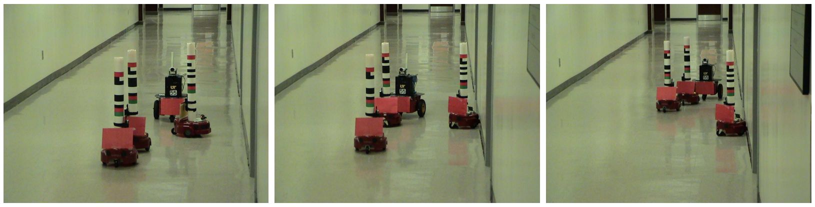


Fig. 5. Deployment of a sensor robot using assistive navigation: the lead robot first guides and then directs the sensor robot into position.

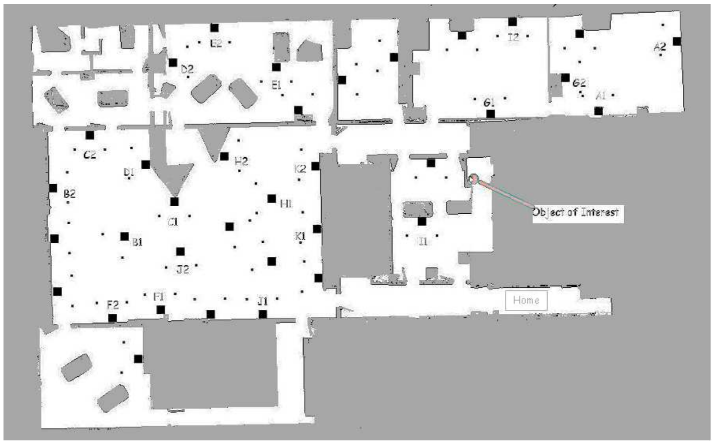


Fig. 6. Autonomously planned sensor net positions (black squares) and planned leader waypoints associated with each sensor position (small dots).

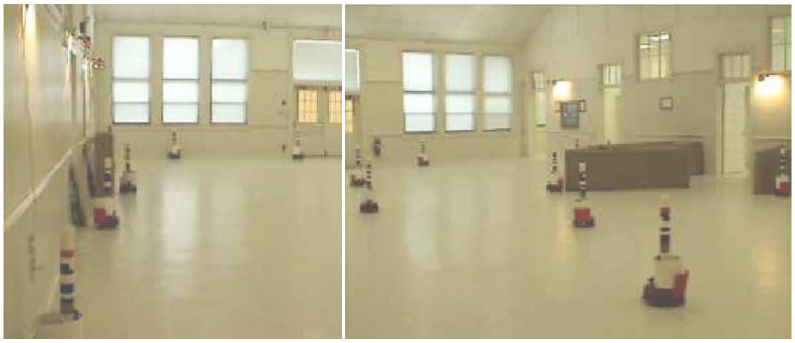


Fig. 7. Physical robots deployed according to the results of the autonomous planning process for the environment shown in Figure 6.

4 Conclusion

In our view, the SDR project has established two new benchmarks in the field of multi-robot systems: this is by far the most complex task ever demonstrated by a large heterogeneous robot team, and by far the most rigorous set of experimental conditions. It should also be noted that while much of our work on this project was necessarily focused on robust engineering, we are confident that the products of this engineering (such as algorithms for multi-robot mapping) will quickly find broader application within the robotics community.

Acknowledgments

This work is sponsored in part by DARPA grant 4400057784/SDR (Software for Distributed Robotics) in conjunction with SAIC, Telcordia, the University of Southern California and the University of Tennessee Knoxville.

Particular thanks go to John Spofford for his patience and understanding, and to Eric Krotkov and Doug Hackett for their extraordinary efforts in the preparation and supervision of the Fort AP Hill experiments.

References

1. W. Burgard, M. Moors, D. Fox, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proc. of IEEE International Conferenceon Robotics and Automation (ICRA)*, volume 1, pages 476–81, 2000.
2. Y. Guo and L. E. Parker. A distributed and optimal motion planning approach for multiple mobile robots. In *Proceedings of IEEE International Conference on Robotics and Automation*, 2002.
3. J. Gutmann and K. Konolige. Incremental mapping of large cyclic environments. In *Proceedingsof the IEEE International Symposium on Computational Intelligence in Robotics and Automation (CIRA)*, 2000.
4. A. Howard. Multi-robot mapping using manifold representations. In *IEEE International Conference on Robotics and Automation*, pages 4198–4203, New Orleans, Louisiana, Apr 2004.
5. F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4:333–349, 1997.
6. L. E. Parker, B. Birch, and C. Reardon. Indoor target intercept using an acoustic sensor network and dual wavefront path planning. In *Proceedings of IEEE International Symposium on Intelligent Robots and Systems (IROS ’03)*, October 2003.
7. L. E. Parker, B. Kannan, F. Tang, and M. Bailey. Tightly-coupled navigation assistance in heterogeneous multi-robot teams. In *submitted to IROS ’04*, 2004.
8. Y. Tang, B. Birch, and L. E. Parker. Mobile sensor net deployment planning using ray sweeping and wavefront path planning. In *IEEE International Conference on Robotics and Automation (ICRA ’04)*, 2004.
9. S. Thrun. Robotic mapping: A survey. In G. Lakemeyer and B. Nebel, editors, *Exploring Artificial Intelligence in the New Millenium*. Morgan Kaufmann, 2002.
10. B. Yamauchi. Frontier-based approach for autonomous exploration. In *Proceedings of the IEEE International Symposium on Computational Intelligence, Robotics and Automation*, pages 146–151, 1997.

View publication stats