Robot-Sensor Networks for Search and Rescue

Joshua Reich
Department of Computer Science
Columbia University
1214 Amsterdam Ave, New York NY 10027 USA
reich@cs.columbia.edu

Elizabeth Sklar
Dept of Computer and Information Science
Brooklyn College, City University of New York
2900 Bedford Ave, Brooklyn NY, 11210 USA
sklar@sci.brooklyn.cuny.edu

***Abstract*— In order to fulfill its mission, a search and rescue system must be able to both quickly and reliably locate victims within the search space. Current search and rescue approaches generally rely on either teleoperated robots, or teams of wireless robots. Since typically the robots used in these systems employ sophisticated hardware components and are few in number, system cost tends to be high and the loss or destruction of even a single robot may seriously compromise mission integrity. We present an alternate approach utilizing *robot-sensor networks* — ad-hoc wireless networks comprised of large numbers of small, simple, and inexpensive wireless sensors and robots. Limiting the use of sophisticated, expensive hardware for rescue system components may be more than compensated for in both cost and performance by the advantages of density and redundancy that smaller, simpler, less costly sensors and robots can provide. In this paper we describe a robot-sensor network for target tracking without reliance on localization services such as GPS or magnetic compass, focusing on simple algorithms for distributed decision-making and information propagation. We demonstrate the efficacy of our system in simulation, providing empirical results and discussion of future work.**

I. Introduction

The past several years have shown great advances in both the capabilities and miniaturization of wireless sensors. These advances herald the development of systems that can gather and harness information in ways previously unexplored. Sensor networks provide a new way of examining environments of interest, delivering numerous small snapshots over time. By fusing these snapshots, a coherent picture of the scene may be produced—rivaling output currently provided by large, complex and expensive remote sensing arrays. Likewise, sensor networks can facilitate propagation of communication through areas unreachable by centralized broadcast due to obstacles and/or irregularities in the connectivity landscape. While traditional non-mobile sensor networks possess tremendous potential, they also face significant challenges. Such networks cannot take an active role in manipulating and interacting with their environment, nor can they physically reconfigure themselves to effect more efficient area coverage, in-depth examination of targets, reliable wireless connectivity, or dynamic protection against inclement environmental developments.

By incorporating intelligent, mobile robots directly into sensor networks, all of these shortcomings may be addressed. Simple, inexpensive, easily programmed, commercial off-the-shelf kits like the Sun SPOT system [13], or LEGO NXT [8] could provide inexpensive test platforms and wireless networking capabilities. Mobile robots provide the means to explore and interact with the environment in a dynamic and decentralized way. In addition to enabling mission capabilities well beyond those provided by sensor networks, these new systems of networked sensors and robots allow for the development of new solutions to classical problems such as localization and navigation [3].

The set of capabilities provided by robot-sensor networks match up well with those needed to build an effective search and rescue system. In order for a search and rescue system to fulfill its mission, the system must be able to both quickly and reliably locate victims within the search space. Moreover a search and rescue system must be able to handle a dynamic and potentially hostile environment. As rescuers move around an uncertain environment, not only do their relative positions change, but also it is not unlikely that their environment will change; collapsed buildings may settle, flood waters may recede or swell, earthquake sites may shift due to aftershock. Current search and rescue approaches generally focus on either teleoperated robots, or teams of wireless robots. Since typically the robots used in these systems employ sophisticated hardware components and are few in number, system cost tends to be high. Consequently such systems can examine only a small portion of the search space at any given moment and the loss or destruction of even a single robot may seriously compromise mission integrity. By leveraging a large number of less sophisticated components, all of which can work in parallel, robot-sensor networks offer an alternate (and potentially complementary) approach that addresses many of the shortcomings of current search and rescue approaches. Specifically, we build a robot-sensor network system that autonomously conducts target tracking in a fully distributed and scalable manner — and does so without any component possessing localization capabilities (e.g., GPS or magnetic compass). The key contribution of our approach is that we provide a solution with minimal hardware assumptions (in terms of sensing, localization, broadcast, memory/processing capabilities), all while subject to a dynamically changing environment.

This paper begins with background in robot-sensor networks, highlighting current challenges in the field. Starting with section III, our approach to the problem is described, focusing on the algorithms used for distributed decision-making and information propagation. In section IV, we discuss the implementation-level details, outlining our simulator capabilities and the assumptions our model makes. Experimental results from our simulated USAR environment are provided in section V. We close with a summary and discussion of future work.

II. Background

Robot-sensor networks have evolved from both work in sensor networks and also in mobile robotics, particularly autonomous robot teams. Given this lineage, sensor-robot networks need to address the technical challenges posed in both these fields, along with novel challenges unique to robot-sensor networks.

A. Robot-Sensor Networks

Recently a small group of researchers has begun exploring the synergy between autonomous robots and sensor networks. Kotay et al. [7] have explored several issues, using the synergy between GPS-enabled robots and networked sensors to provide network-wide localization services, path planning, and improved robot navigation. The Centibots project examined the use of large-scale mobile robotic teams for mapping and areas surveillance [6], while a method for the transportation of resources by combining robots with sensor network services was suggested by Gupta et al. [5]. A method by which robot-sensor swarms can effect detection of radioactive materials in a role-based system was suggested in [9]. Work by Sukhatme and colleagues discusses how sensor networks can be used to mediate robot task allocations [1] and algorithms for optimizing sensor placement [15]. Finally, Reich and Sklar [10] considered the use of traditional shortest-path network routing discovery algorithms for guiding robotic searchers to targets detected by an ad-hoc wireless network. These results, produced at the boundary where robotic teams and sensor networks intersect, suggest a large and fascinating problem space open for exploration.

Following is a sampling of the interrelated issues for which techniques, algorithms, and hardware solutions need to be devised:

1) target tracking,
2) localization and mapping,
3) communications and routing,
4) path planning,
5) high-level team formation and mission fulfillment,
6) standardization of hardware services/interfaces, and
7) asymmetric wireless broadcast and network interference/congestion.

While our work touches somewhat on all of these issues, it focuses primarily on first three, exploring how such systems can provide useful and robust base-level behaviors—and do so with minimal hardware requirements or dependence on favorable environmental conditions.

B. Limitations of Previous Work

One commonality amongst much of the works cited above is the reliance on sophisticated hardware and/or friendly or over-simplified environmental conditions. Most work either assumes the existence of basic services such as localization and orientation, or considers only the cases where at least a fraction of the agents possess essential hardware used for global positioning. While these assumptions hold in many situations of interest, they fail to provide techniques that will be effective when such hardware services (e.g., GPS, magnetic compass) fail or are unavailable (e.g., indoor or USAR environments). Currently, wireless sensor sizes range from centimeters to millimeters. The smallest robots are generally one to two orders of magnitude larger, in the centimeter to meter range, although they are quickly shrinking [12]. Such equipment, while small and inexpensive enough for ubiquitous deployment, may also be severely constrained in offering sophisticated hardware services. To allow for the widest range of deployable systems, our work examines systems that make minimal assumptions concerning hardware capabilities. Limiting the use of sophisticated, expensive hardware for network nodes may be more than compensated for in both cost and performance by the advantages of density and redundancy that smaller, simpler, less costly sensors and robots can provide. This approach would be particularly advantageous in harsh operational environments where loss, destruction, or failure of network components becomes likely.

III. Approach and Assumptions

A fully featured search and rescue system should:

1) quickly and accurately locate victims
2) map the search space and locations of victims
3) maintain communication with human responders
4) assess victim status and assist with rescue efforts.

While our immediate goal is to achieve the first step in this program, future work will extend this system to latter stages. Once victims have been located by robotic searchers (in the sense that some robot is directly adjacent to the victim), the searchers can proceed to assess victim condition, initiate rescue, or attempt localization (i.e., using odometry readings, triangulation, or other techniques such as multilateration [11]). We desire that the system be able to autonomously fulfill its mission requirements without any component that has localization capabilities (in a global sense) — and to do so in a distributed manner. Moreover the system should be able to respond automatically to environmental change including target movement and loss/addition of network components. The range of sensors and wireless broadcast are assumed to be significantly restricted with respect to the search space. The only knowledge primitives that the system is assumed to possess are: awareness of neighbors and nearby targets and (for robots) approximate distance from neighbors and approximate direction towards targets. We assume that the targets are beacons (i.e., they generate some detectable signal such as heat, CO_2, or sound). Real-world implementations would likely use as many sensing modalities as possible to gain the highest confidence on these readings; however, for our purposes, we simplify the situation assuming a generic target sensing modality.

A. *Network-wide Gradient Algorithm*

In order to guide the robot searchers quickly to targets under these considerable constraints, we have adopted a biologically inspired model. Simple creatures can effectively reach desirable locations by simply following some gradient (e.g., heat, light). While no naturally existing gradient (within bounds of the robots' simple sensors) is available, a network-wide gradient can be established by having each of the many sensors scattered in the search space take on a gradient value. By assigning sensors close to the targets "hot" values and those far from targets "cold" values, robotic searchers can make their way towards "hot" spots and thereby reach the targets. Moreover such a gradient will naturally respond to movement, appearance, or disappearance of targets. Our gradient propagation (GP) algorithm is entirely distributed and straightforward to implement. Each sensor independently executes the GP algorithm, broadcasting after some independent, randomly chosen time interval.

Algorithm 1: GP algorithm (running on all nodes)

Data: tg - the gradient value
$maxTG$ - the maximum target gradient value
if *target nearby* **then**
  $tg \leftarrow 0$;
  broadcast tg as update;
else
  listen for neighbor broadcasts;
  $tg \leftarrow \min_{neighbors}(tg_i) + 1$;
  if $tg < maxTG$ **then**
    broadcast tg as update;

The GP algorithm guarantees (under certain conditions, proof omitted) that each sensor's individual target-gradient value will converge to the minimum number of hops between that sensor and one detecting a target. By only updating when $tg <= maxTG$ we avoid an infinite count-up when no targets are visible to the network. Furthermore when $tg > maxTG$, sensors can broadcast a message informing robots to look elsewhere as no targets are in the vicinity.

The algorithm is not only fully distributed, allowing for dynamic and automatic addition and deletion of sensors to the network; but will also dynamically respond to movement, appearance or disappearance of targets—the speed of the response being controlled by expected size and variance of the random update intervals. Moreover, this update algorithm provides for some measure of protection against broadcast collisions as the updates occur asynchronously. Future implementations could responsively decrease the update likelihood to provide for a back-off mechanism analogous to the kind used in standard wireless networking protocols such as the 802.11 family.

The theoretical performance of this mechanism is quite good: the number of broadcast operations scales linearly with the total number of sensors, and the number of listen operations scales linearly with the average number of neighbors. Each sensor broadcasts on the average once per time interval and needs to store and choose the minimum target-gradient value from the broadcasts made by its neighbors.

Bandwidth (and therefore power) requirements for each broadcast can be made very low. With 6 bits of information (ignoring CRC codes and header information), sensors up to 64 hops away from targets can acquire active target-gradient values (notably this update mechanism does not require sensors to broadcast their ID, or even necessarily possess an ID).

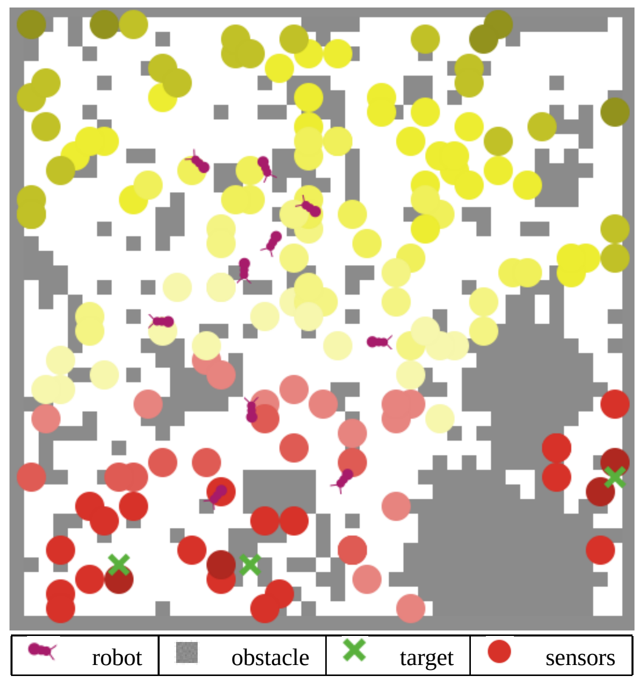

Fig. 1. Simulation environment, gradient runs from low (red) to high (green) target-gradient values.

B. *Robot Architecture*

Our goal for robot behavior is for each robot to make independent decisions (as opposed to receiving orders from a centralized node in the network), but at the same time to avoid the computational costs associated with sophisticated decision-making. Consequently, we imbue each robot with a simple hierarchy of behaviors, using a simple subsumption architecture [2], along with state transitions, as illustrated in Figure 2.

The hierarchy contains three states, numbered in increasing order of precedence. The most dominant state is state 2 in which a target has been detected. The robot's behavior in state 2 is to search for the target until (a) the robot finds the target, (b) another robot finds the target, or (c) the robot loses the target signal. In the first case the robot initiates further action (e.g., target inspection or localization inference). In the two latter cases, the robot returns to behavior state 0 (from which it may immediately jump to state 1). State 1 is reached from state 0; when no target signal is present but some sensor is in range, the robot's behavior is to traverse the network towards a target some hops away. To traverse the network, all the robot need do is move towards the neighboring sensor with the lowest target-gradient value. To find these values the robot can either actively query sensors within broadcast range, or passively listen for sensors declaring their target-gradient values during gradient update as described in III-A. In practice this could be done either with a local gradient search on the broadcast (or other) signal produced by this sensor (e.g., RSSI), or via use of a directional antenna/sensor. Finally, in state 0 the robot conducts a blind search, looking first for target signals (transition to state 2) and second for sensor signals (transition to state 1).

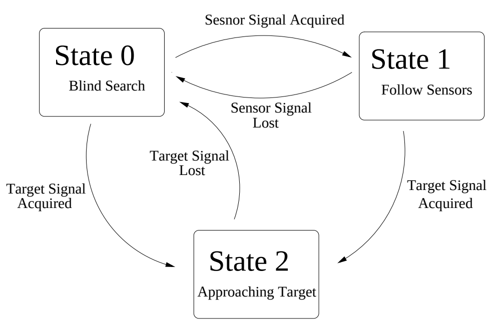

Fig. 2. Robot behavior hierarchy.

IV. Implementation

A. Software Platform

We have used the NetLogo (version 3.0.2) multiagent programming environment [14] for constructing our initial simulation. All results presented here are based on experiments executed in this simulator (figure 1).

B. Robot Movement and Obstacles

Robot movement is modeled probabilistically. When a robot moves forward, it turns randomly a bit to one side or the other. The degree to which the movement of robots is skewed can be adjusted to consider different robot platforms or surfaces.

The gray regions in figure 1 represent obstacles to physical movement but not sensing or wireless broadcast. In the white areas on the figures, the robots are free to travel. We note that in the real world, some physical obstructions may not interfere with wireless connectivity and vice versa; for ease in constructing our initial implementation, we chose to consider obstructions that only block movement. Future work will explore more complex and realistic situations. The simulation allows for the investigation of areas with obstacles to robot movement and can adjust both obstacle density (the percentage of area covered by obstacles) and clustering tendency.

C. Broadcast & Sensing Models

For the present, we have adopted a simplified non-probabilistic model of wireless broadcast. We assume a spherical broadcast model, and, for the moment, consider neither broadcast collisions nor other types of signal propagation effects. The sensing model (similarly non-probabilistic) is also spherical, while the robots are assumed to possess directional sensing arrays. While these assumptions are admittedly simplified, they do provide a reasonable first-approximation for the very short range broadcast a system such as ours would use.

D. Sensor Dispersal

We currently have examined only uniform random distribution of the sensors throughout the environment (this could correspond to release of sensors from above the rescue space). We are planning to develop, experiment with and evaluate three additional distributions in the near future:

1) release of bouncing sensors — encasing the sensors in rubberized spherical cases will allow them to bounce off obstacles and hopefully penetrate further into the environment before coming to rest
2) robot driven distribution — in which the robots would each carry a store of sensors that could be dropped or spread out from desired locations.
3) sensor mobility — sensors that can periodically move themselves (and potentially interchange roles with robotic searchers) for periods of time.

V. Experiments & Results

In previous work we have shown that areas with 25% obstacle density prove significantly more difficult than lower density (i.e., $< 15\%$) environments, but not so difficult as to prevent comparison of the robot-sensor network (RS) with an unaided robot-only (RO) system [10]. Therefore we have selected an obstacle density of 25% for the experiments done in the paper. The environment was randomly generated on a 2-dimensional grid ($43X43$ patches). In all experiments, a trial consisted of 1000 time-steps, during which targets were presented one at a time. Our primary fitness metric was percentage of targets found. We also used average time to find a target as a secondary metric. The target starting positions were selected according to uniform random distribution, as was the point selected at which all robots started. Half of the trials examined randomly moving targets (MT) and half examined stationary targets (ST). Any target not found within 300 time-steps disappeared and was replaced. For consistency, each generated environment was used for 4 trials: (RS:MT), (RS:ST), (RO:MT), (RO:ST).

It worth noting that systems excelling on the our primary metric may be more likely to locate difficult to find targets (which will tend to require longer than average search times). Since undiscovered targets did not contribute to the average time to find a target, improved performance of a system with respect to our primary fitness metric could conceivably have a negative impact its performance on our secondary metric.

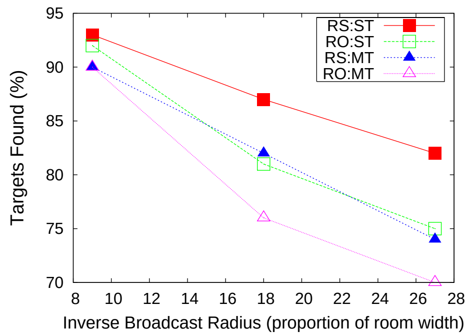

Fig. 3. Percentage Targets Found vs. Inverse Broadcast Radius

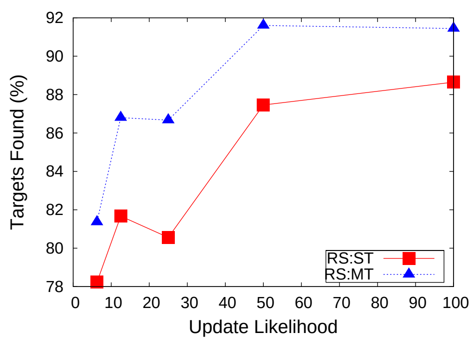

Fig. 4. Percentage Targets Found vs. Update Likelihood

Before attempting to assess the quality of our system, we wanted to develop our understanding of the various parameters involved. Consequently the first set of experiments consisted of 1200 trials comprising 1,200,000 ticks worth of data. The trials were taken over 2 system types (RS, RO), 2 target types (MT, ST) and three variables: *update likelihood* (the percent likelihood of update for a given sensor on any time-step), *inverse sensing range*, and *inverse broadcast range*. We tabulated the results for each variable over all values over the other two variables for each case (RS:MT, RO:MT, RS:ST, RO:ST). 10 robots and 150 sensors were used in each trial, both robots and sensor sensing and broadcast ranges were held identical.

Unsurprisingly, as sensing range decreased, target location became markedly more difficult — as can be seen from figure 3. More interestingly, one can clearly see a marked divergence in performance between RS and RO systems by inverse broadcast range of 18 (a divergence which seems to stay fairly consistent thereafter). Based on this trial, we selected a sensing range between $1/20^{th}$ and $1/21^{st}$ of the search space width for subsequent experiments in order to provide for a task difficult enough to easily separate the performance of RS and RO.

The results on broadcast range indicated a slight decline in performance as the range decreased. Given that it is reasonable to assume a broadcast radius significantly larger than the sensing radius for most modalities, we chose to use a broadcast radius of $1/7^{th}$ the search space for future experiments (which yields approximately a 3:1 ratio between broadcast and sensing radii).

Lastly, we examined the effect of update likelihood and found that for both MT and ST, as the likelihood of an update decreased from 100% to 6.25%, the system performance dropped about 10%, with the seeming appearance of a logarithmic curve (figure 4)—for every halving of likelihood, the system performance dropped about 2.5%. (Note that RO measurements do not appear in figure 4 as RO is unaffected by update likelihood.) This is an exciting result insofar as it indicates the potential robustness of our gradient propagation algorithm and low performance cost of doubling bandwidth and power savings. Since our primary goal in the following experiments was to get a clear idea of the performance benefits a robot-sensor network might provide as opposed to maximizing power efficiency, we choose to use update likelihoods between 33–50% for subsequent experiments.

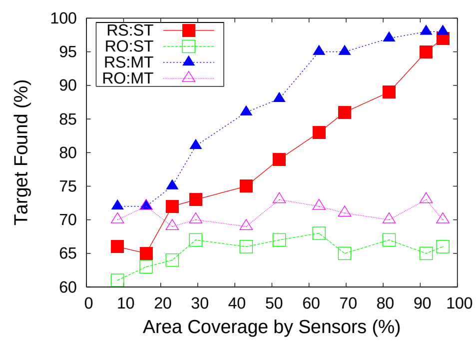

Fig. 5. Percentage Targets Found vs. Sensor Coverage

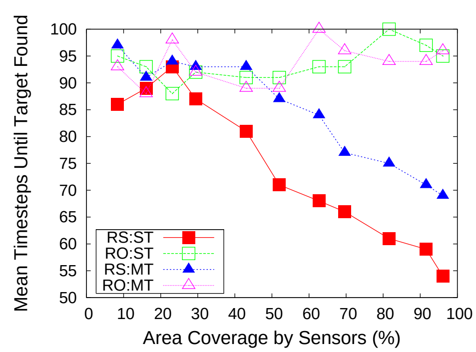

Fig. 6. Average Time to Find Targets vs. Sensor Coverage

For the next series of experiments, we examined the effect of sensor coverage on system effectiveness. As our world was not overly large, we reduced the number of robots used by half to 5 robots, examining the results used from 4400 trials over 11 sensor densities. As we examined the same 4 combinations of system and target types as previously, each point in figures 5 and 6 represents the averages over 100 trials or 100,000 time-steps. The results of our experiment were quite encouraging. From 40% area coverage and on RS systems clearly dominate the performance of RO systems with respect to the first metric and by 60% with respect to both metrics. Our robot-sensor network was able to find targets both more quickly and with significantly better success than a robot only system. Moreover this result is even more striking in light of our earlier argument that high target detection rates should, if anything, diminish average speed.

From 60% coverage and up, search and rescue successfully found an additional 24% of the stationary targets on average and 25% additional moving targets. In terms of the total targets, the RS found, respectively, 36% and 35% more targets than RO. Concurrently, the RS system took an average of 34 fewer time-steps than RO to find stationary targets—RO took more than half-again as much time as RS. For moving targets this difference declined, RO took only 28% longer, with RS's average advantage being 21 time-steps.

VI. Summary & Future Work

We have presented an algorithmic framework for utilizing an ad-hoc wireless network comprised of large numbers of small, simple, and inexpensive wireless sensors and robots to conduct efficient and robust target tracking - and do so in absence of the hardware services required by previous approaches. Moreover we have shown this framework to adjust dynamically to both target movement and addition/deletion of network components. The network gradient algorithm provides an advantageous trade-off between power consumption and performance and requires relatively low bandwidth. Our system functions well in simulation, locating 70-99% of the targets (dependent on network density and vagaries of search space and target placement and movment) in the time allotted, outperforming robot-only systems significantly in both accuracy and speed.

Our future goals are to introduce new system capabilities, improve our simulator and models (including a comparison with GPS-based systems), and begin implementation of a prototype system in hardware. Likely the most significant addition we will be making to our system in the near future is the introduction of what we have termed *network-based stigmergy*. *Stigmergy* can be defined as the use of environmental elements to provide a means of indirect communication between agents. The classic examples of stigmergy involve creatures such as termites or ants that can effectively create complex buildings and find efficient paths towards food sources by leaving chemical messages at appropriate spatial locations. By storing spatially related messages at actual locations to which the messages relate, very simple creatures manage to coordinate large scale and complex behavior [4]. Practical attempts at building robotic stigmergic systems have had minimal success due to the obvious difficulty of encoding messages in a physical environment. However with the availability of a large number of wireless nodes located around the environment, it becomes feasible to encoding spatially related messages at or near the spatial locations to which the message relate simply by writing them to the nearest sensor. We are conducting work to incorporate this idea of network-based stigmergy into our robot-sensor network towards the end of providing estimates of obstacles density near sensors, assessing reliability of network components, and helping guide the robots actively towards unexplored or under-explored areas.

Concurrently, we will continue to improve our simulator and introduce new modeling aspects along the lines described in section IV and begin implementing a hardware prototype of our system. We are particularly enthused to examine robot-assisted and sensor-mobility distribution models as these could provide for significant additional system capabilities (e.g., increased sensor coverage, active environmental mapping) and fit well with our focus on testing these methods in hardware.

References

[1] M. Batalin and G. S. Sukhatme. Sensor network-mediated multi-robot task allocation. In *The Third International Naval Research Laboratory Multi-Robot Systems Workshop*, pages 27–38, Naval Research Laboratory, Washington, DC, Mar 2005.

[2] R. Brooks. A robust layered control system for a mobile robot. *IEEE Transactions on Robotics and Automation*, 2:14–23, 1986.

[3] P. Corke, R. Peterson, and D. Rus. Localization and navigation assisted by cooperating networked sensors and robots. *International Journal of Robotics Research*, 24(9), 2005.

[4] M. Dorigo, E. Bonabeau, and G. Theraulaz. Ant algorithms and stigmergy. *Future Generation Computer Systems*, 16:851–871, 2000.

[5] A. K. Gupta, S. Sekhar, and D. P. Agrawal. Efficient event detection by collaborative sensors and mobile robots. In *First Annual Ohio Graduate Student Symposium on Computer and Information Science and Engineering*, 2004.

[6] K. Konolige, C. Ortiz, and R. Vincent. Centibots large scale robot teams. In *AAMAS*, 2003.

[7] K. Kotay, R. Peterson, and D. Rus. Experiments with robots and sensor networks for mapping and navigation. In *International Conference on Field and Service Robotics*, 2005.

[8] LEGO. `http://mindstorms.lego.com/`.

[9] H. V. D. Parunak, S. A. Brueckner, and J. Odell. Swarming pattern detection in sensor and robot networks. In *American Nuclear Society (ANS) 10th International Conference on Robotics and Remote Systems for Hazardous Environments*, 2004.

[10] J. Reich and E. Sklar. Toward automatic reconfiguration of robot-sensor networks for urban search and rescue. In *First International Workshop on Agent Technology for Disaster Management (ATDM): Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, Hakodate, Japan, May 2006. ACM.

[11] A. SAVVIDES, H. PARK, and M. B. SRIVASTAVA. The n-hop multilateration primitive for node localization problems. In *Mobile Networks and Applications*, 2003.

[12] G. T. Sibley and M. H. a. Rahimi. Robomote: A tiny mobile robot platform for large-scale ad-hoc sensor networks. In *IEEE International Conference on Robotics and Automation*, 2002.

[13] Sun. `http://sunspotworld.com/`.

[14] U. Wilensky. NetLogo. `http://ccl.northwestern.edu/netlogo`, 1999.

[15] B. Zhang and G. S. Sukhatme. Controlling sensor density using mobility. In *The Second IEEE Workshop on Embedded Networked Sensors*, pages 141 – 149, May 2005.