Toward Automatic Reconfiguration of Robot-Sensor Networks for Urban 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

An urban search and rescue environment is generally explored with two high-level goals: first, to map the space in three dimensions using a local, relative coordinate frame of reference; and second, to identify targets within that space, such as human victims, data recorders, suspected terrorist devices or other valuable or possibly hazardous objects. The work presented here considers a team of heterogeneous agents and examines strategies in which a potentially very large number of small, simple, sensor agents with limited mobility are deployed by a smaller number of larger robotic agents with limited sensing capabilities but enhanced mobility. The key challenge is to reconfigure the network automatically, as robots move around and sensors are deployed within a dynamic, potentially hazardous environment, while focusing on the two high-level goals. Maintaining information flow throughout the robot-sensor network is vital. We describe our early work on this problem, detailing a simulation environment we have built for testing and evaluating various algorithms for automatic network reconfiguration. Preliminary results are presented.

1. INTRODUCTION

This work explores the use of "robot-sensor networks" for urban search and rescue (USAR), where the topography and physical stability of the environment is uncertain and time is of the essence. The goals of such a system are two-fold: first, to map the space in three dimensions using a local, relative coordinate frame of reference; and second, to identify targets within that space, such as human victims, data recorders, suspected terrorist devices or other valuable or possibly hazardous objects. Our approach considers a team of heterogeneous agents and examines strategies in which a potentially very large number of small, simple, sensor agents with limited mobility are deployed by a smaller number of larger robotic agents with limited sensing capabilities but enhanced mobility. While every node in the network need not be directly connected to every other node, it is vital that information be able, eventually, to make its way to designated "contact" nodes which can transmit signals back to a "home base". It is advantageous for the network to possess reliable and complete end-to-end network connectivity; however, even when the network is not fully connected, mobile robots may act as conduits of information — either by positioning themselves tactically to fill connectivity gaps, or by distributing information as they physically travel around the network space. This strategy also enables replacement of failed nodes and dynamic modification of network topology to provide not only greater network connectivity but also improved area coverage. The robotic component of our agent team can leverage its mobility capabilities by allowing dynamic spatial reconfiguration of the robot-sensor network topology, while the sensor components help to improve localization estimates and provide greater situational awareness.

The past several years have shown great advances in both the capabilities and miniaturization of wireless sensors [16]. These advances herald the development of systems that can gather and harness information in ways previously unexplored. Sensor networks may provide broader and more dynamic perspectives if placed strategically around an environment, delivering numerous small snapshots over time. By fusing these snapshots, a coherent picture of an environment may be produced — rivaling output currently provided by large, complex and expensive remote sensing arrays. Likewise, sensor networks can facilitate propagation of communication in 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 for 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 robotics kits like Garcia [7], or even the new LEGO NXT [15], could provide inexpensive test platforms and wireless networking capabilities. Mobile robots provide the ability 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]. Arguably, the development of mixed sensor-robot networks will allow for exploration of and interaction with environments in ways previously infeasible.

One of the biggest challenges in an urban search and rescue environment is the need to maintain consistent and reliable network communication amongst remote rescuers, whether they are human or robot or both. 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. The capability for a team of agents to map their space collaboratively, identify victims and other targets of interest, while maintaining information flow is crucial; and given the dynamic nature of the environments they are exploring, it is also important that such ad-hoc networks be able to reconfigure automatically, not only due to changes in position of the agents but also caused by failure of one or more nodes.

The work presented here, in very early stages of development, examines the issue of automatic reconfiguration of a network of agents under such conditions as described above. The longterm goal of this work is to deploy a physical system in an urban search and rescue test arena [11], but the present stage of work involves development of a simulator in which crucial features are emulated and where design of algorithms for automatic network reconfiguration can be tested and evaluated. This paper begins with background in sensor and robot networks, highlighting current areas of challenge within the field. Starting with section 3, our approach to the problem is described, including detailed discussion of our testbed, the algorithm we are evaluating and preliminary experimental results from testing the algorithm in a simulated USAR environment. We close with discussion of future work.

2. BACKGROUND

The challenges to realizing the potential of sensor-robot networks exist at both hardware and software levels. Open problems include power management, communication, information fusion, message routing, decision-making, role assignment, system robustness, and system security. Current research has begun to address many of these issues. Several methodologies have been tested for target detection and tracking, both with fixed sensors [5] and using large-scale mobile robotic teams [12]. Researchers are actively investigating novel message routing protocols, some of which enable self-organization of networks nodes [17]. As many of these approaches rely on some type of geographic routing scheme, sensor localization has become an area of inquiry [2]. Fundamental issues such as dealing with power supply limitations [6] and ensuring coverage of the area to be sensed [10] are also being explored.

Recently a small group of researchers has begun exploring the synergy between autonomous robots and sensor networks. Kotay et al. [2005] 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. Gupta et al. [2004] have suggested a method for the transportation of resources by combining robots with sensor network services. The Centibots project [12] examines how large numbers of more sophisticated robots may collaborate to create maps and subsequently surveil the area by leveraging ad-hoc wireless networking capabilities. 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. high-level team formation and mission fulfillment,
2. communications and routing,
3. localization and mapping,
4. path planning,
5. target tracking,
6. standardization of hardware services/interfaces, and
7. asymmetric wireless broadcast and network interference.

While our work touches somewhat on all of these issues, it focuses mostly on the fifth, third and second, in that order, 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.

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 localization (e.g., global positioning system or GPS). While these assumptions allow for investigation of important problems, 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. 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, this 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.

3. OUR METHODOLOGY

Our immediate goal is to guide robot searchers effectively to targets by leveraging communications and sensing services provided by a dense network of non-mobile agent-based sensors. Additionally, we desire that the system be able to fulfill its mission requirements without any component that has localization capabilities (in a global sense) — and to do so in a distributed manner. The only knowledge primitives assumed by the simulation are: for all agents, awareness of neighbors and nearby targets, and (for robots) approximate distance from neighbors and approximate direction towards targets.

We employ a network routing scheme to route not just our system's communications, but also the movement of its mobile components. We note that there exist a family of algorithms currently used to do route planning within networks, so as to produce routes with minimal hop distance [8]. In most networks, hop distances are not highly related to the physical distance over which a piece of information is passed. An email to one's next door neighbor might pass over almost as many hops as one sent to a correspondent overseas. However, in high density, short-range sensor networks this tends not to be the case; the correspondence between minimal hop path and physical distance between nodes being fairly strong in many environments. Consequently, knowledge of the minimal hop paths could not only enable efficient message routing in the network but also provide a good approximation of the shortest physical paths from one sensor to another several hops away.

As an example, consider the simple robot-sensor network illustrated in Figure 1. The *robot* arrives at node A, which has been informed that node D, three hops away, has detected a target in its vicinity (the target is the star in the figure, to the northeast of node D). Node A can then inform *robot* that D is detecting a target and that node B is the next hop along the shortest routing path to D. By following some detectable gradient towards B (e.g., signal strength), *robot* will be able to come close enough to B to receive information about and a signal from the next-hop on path to D, namely node C. In this fashion *robot* is able to quickly find its way towards D without any *a priori* localization knowledge. Once *robot* has reached D, it will be close enough to directly detect the target itself.

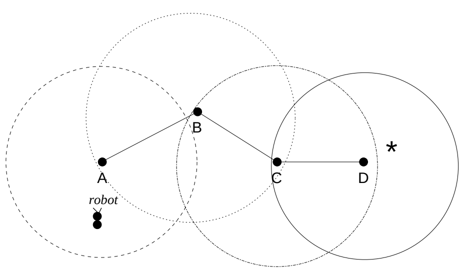

Figure 1: Sample robot-sensor network. Node D detects a target to its northeast. The network can route the *robot* along the nodes from its present location, within range of A, to the node which has detected the target, D.

In order to make the above scheme work several algorithmic questions need to be addressed:

- Where should the network routing information be calculated and stored?
- How should information regarding which sensors are detecting targets be distributed?
- How should robots go about choosing a course of action (e.g. follow path or search for nearby target)?
- What information should be exchanged between network components (both robots and sensors)?

In the remainder of this section, we address these questions and explain the choices we have made in our implementation.

3.1 Network routing and distribution of target information

Our hard requirements for network routing are that any sensor in the system should provide both hop-distance and next-hop to a given destination, if a path exists. Additionally, in the interest of system scalability and responsiveness, we desire path computation and storage to be local to each sensor. A number of options are available, the most straightforward of which is simply to employ a slightly modified version of the popular Distributed Vector (DV) routing algorithm [14], one of the two main Internet routing algorithms. The DV algorithm itself operates in a very straightforward fashion. Each node in the network keeps a routing table containing identifiers of every node to which it knows a path, along with the current hop-distance estimate and next hop along that path. Each asynchronously sends its routing table to all neighboring nodes which, in turn, check their tables to learn of new destinations. Additionally, when node A sends its routing table to node B, B will check its list of known nodes and hop-distances against the table sent by A and choose A as the next hop for any nodes that would be more quickly reached through A. If B does make any additions or adjustments to its table, it will send the revised table to all of its own neighbors to alert them to these new or shorter paths. In this manner, routing information will be diffused throughout the network.

The theoretical performance of DV is quite good and its wide adoption attests to its reliability, simplicity, and scalability. However, in our simulation we found a significant time lag once network density increased past an average of 10 neighbors — reflecting the high number of messages being sent before the nodes converged. Additionally, the size of the routing table held at each node scales linearly with the network size — possibly making this approach infeasible for very dense networks, at least not without modification. Lastly, while DV provides a sophisticated means for passing unicast messages, it may not provide competitive advantage justifying its cost in applications where much information may be expressed in the form of a network-wide gradient. In our current work, we are comparing the performance of DV to a network gradient, where nodes learn only hop-distance from the nearest sensor detecting a target, supplemented by a more expensive direct message-passing service.

3.2 Robot behavior

Our goal for robot behavior is for each robot to make an independent decision (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, each robot is given a simple hierarchy of behaviors, using a simple subsumption architecture [1], 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) the robot discovers another robot has gotten there first, or (c) the robot loses the target signal. In the first case the robot settles near the target and broadcasts a signal of ownership. 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. 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).

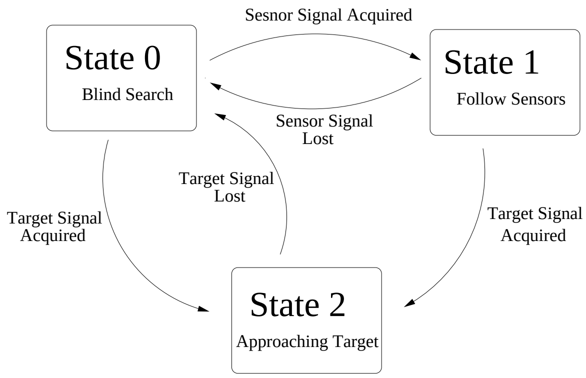

Figure 2: Robot behavior hierarchy.

3.3 Information exchange

In our initial implementation, agents only provide each other with path information to sensors' nearby targets. Our current work involves expanding the information exchange capabilities of the system so that additional data may be passed between nodes in an efficient manner. We are looking for this to improve system performance in several ways. First, once a target has been found and its surroundings explored (for any additional targets), the sensors close enough to receive the target signal should be marked by the network accordingly. This information should then be propagated throughout the network, preventing these sensors from being continually revisited by curious robots. Second, sensors may mark the passage of robots with a time-stamp and/or visit counter. By doing so, robots may decide to avoid sensors visited very often or very recently, choosing to explore paths less traveled or even areas entirely out of the network coverage. Third, robots may leave "trails" [4], in order to facilitate quick transference of information back to the home base.

4. IMPLEMENTATION

We have used the NetLogo (version 3.0.2) multiagent programming environment [18] for constructing our initial simulation. All results presented here are based on experiments designed and executed in this simulator. Figures 3, 4 and 5 illustrate the environment. The gray regions represent obstacles, both for physical travel by the robot and wireless connectivity of the network. 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 make this assumption, but current work is exploring situations in which the two types of obstructions are handled separately. In the white areas on the figures, the robots (and the signal) are free to travel. The dark circles represent agent sensors which are immobile, and the lines between them show the connectivity of the network. The bug-like symbols represent the mobile, robotic agents. Section 3.2 describes the hierarchical control algorithm we have implemented for the robots. The sensor agent behavior is even more simplistic. In our current implementation, these agents do not possess any decision-making capabilities; as described below, they merely broadcast any target information as well as beacon signals for mobile agents.

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. Current work is exploring this aspect in detail, incorporating models of trust in the existing system and endowing the sensor agents with decision-making abilities such that broadcast becomes non-deterministic. The sensing model (similarly non-probabilistic) is also spherical, while the robots are assumed to possess directional sensing arrays. The simulation allows for the investigation of areas with obstacles to robot movement and can adjust both percentage of area covered by obstacles as well as their clustering tendency.

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 is controlled by a global variable and can be adjusted to consider different robot platforms or surfaces. The robots have the ability to move around the environment and disperse a potentially large number of non-mobile sensor agents. Currently two types of sensor dispersal algorithms have been compared: random distribution radially from the center of the robot start location, and uniform random distribution throughout the environment.

5. PRELIMINARY EXPERIMENTS

The primary issue we aimed to assess with our initial implementation was whether at system's current level of development, a performance difference could be ascertained between our sensor-robot network and a system employing robots alone. In order to evaluate the problem space, we conducted 1152 runs, sampling over the following six additional variables: *obstacle density*, *number of robots*, *number of non-mobile sensors*, *dispersal method*, *broadcast radius* and *spread of communication*. The metric used for all experiments was the number of time steps taken until 90% of the targets had been discovered.

The variable with the clearest effect was obstacle density. Spaces with few obstacles, like Figure 5, were easily solved by both sensor-robot teams and robot-only teams. Spaces with many obstacles (like Figures 3 and 4) proved significantly more difficult, often taking upwards of 5 times longer to find 90% of the targets. Consequently, we chose to focus our set of experiments on environments with 25-30% of the area occupied by obstacles. Sensors were distributed according to a uniform random distribution, as were targets. We used 30 robots and 90 sensors for the trials and a broadcast radius varying between $1/8^{th}$ and $1/12^{th}$ of the area's width.

The results of our experiments so far are statistically inconclusive; as yet, we are unable to show a comparative advantage between the sensor-robot and robot-only teams under the parameterization chosen. However, by viewing several simulations and examining system performance, we are

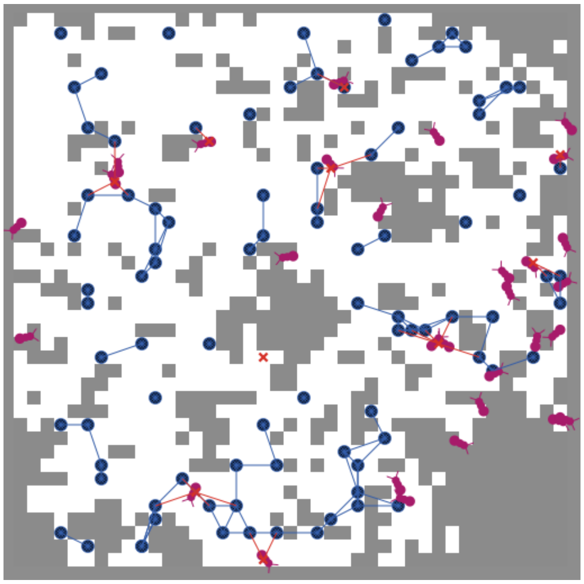

Figure 3: Many Obstacles: Open.

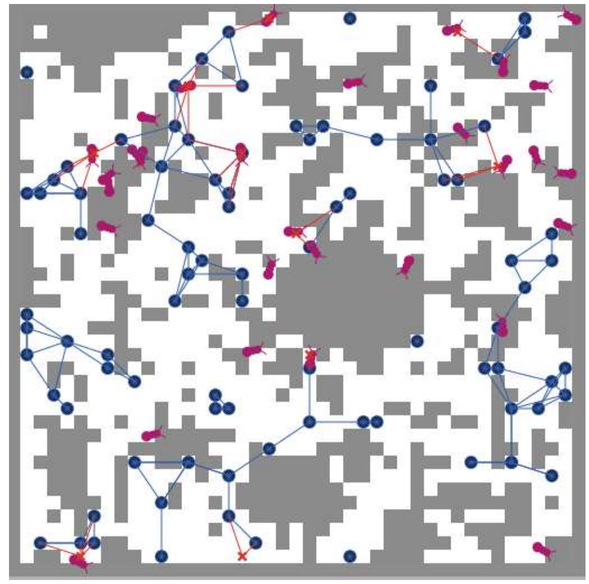

Figure 4: Many Obstacles: Segmented.

key (applies to figures 3, 4 and 5):

robot	obstacle
target	sensors

able to generate some qualitative observations that encourage us to continue with this line of inquiry. On individual trials, the sensor-robot teams often significantly outperform the robot-only teams, but these are offset by occasions in which the sensor-robot teams becomes bogged down in parts of the network already explored. The sensor-robot teams do very well in situations where the environment is highly segmented and both sensor and targets are fairly well spread out (e.g., Figure 4). The robots are able to follow the network paths successfully through small crevices to reach new compartments and thereby find targets effectively; in contrast, with only random guessing about where to move next, the robot-only teams tend to do rather poorly in such spaces. In the space shown in Figure 4, for example, the robot-only team took 1405 time steps to complete the search, while the sensor-robot team managed it in only 728.

In relatively open spaces, like (Figure 3), the robot-only teams have much less trouble (in this case the two approaches both took around 450 time steps). The sensor-robot systems perform badly when some of the targets have several sensors nearby, while others have few or no nearby sensors. In these cases, the robots continually revisit the sensors near targets already discovered, keeping too many robots from exploring other areas. The robot-only teams ignore the network in these situations and perform considerably better.

The main problem the sensor-robot teams experience is that each robot keeps its own list of target-detecting sensors that it has visited. Since robots choose the sensors they will visit randomly from the list of unvisited target-detecting sensors, every robot can end up visiting a multiply-detected target several times for each time it looks for a singly-detected target. Moreover robots try to visit *every* detectable target before looking for targets un-sensed by the network! Consequently, in certain trials, the network effectively traps the robots in one portion of the environment for a significant time-span. We believe that once additional information sharing facilities outlined in section 3.3 have been implemented, the sensor-robot system will statistically outperform robot-only systems when repeating the experiments outlined above.

6. SUMMARY AND FUTURE WORK

We have presented early work in the development of strategies for controlling teams of heterogeneous agents, possessing a mixture of sensing and mobility characteristics. Taking advantage of recent advances in sensor networks and routing schemes, we are interested in exploring situations in which a potentially very large number of small, simple, sensor agents with limited mobility are deployed by a smaller number of larger robotic agents with limited sensing capabilities but enhanced mobility. Our longterm goal is to apply techniques developed to urban search and rescue problems.

In the short term, our work is focusing primarily on continued development of simulation platform. The immediate steps involve: (a) introduction of gradient-based routing, (b) incorporation of enhanced information sharing facilities, and (c) improvement of robot behavior to incorporate new information. The next steps entail producing comprehensive empirical results, evaluating hardware platforms and building prototype hardware systems for testing our strategies. Our plan is to contrast simulated results with those from our physical prototype, using data collected in the physical world to seed learning algorithms for building error models in the simulator, which can then be used to improve performance in the physical setting.

7. REFERENCES

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

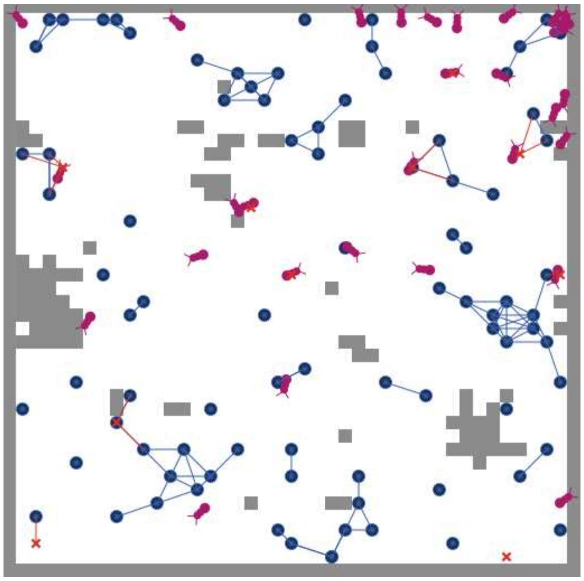

Figure 5: Screen-shot of simulation with few obstacles.

[2] A. Caruso, S. Chessa, S. De, and A. Urp. Gps free coordinate assignment and routing in wireless sensor networks. In *IEEE INFOCOM*, 2005.

[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, V. Maniezzo, and A. Colorni. The Ant System: Optimization by a colony of cooperating agents. *IEEE Transactions on Systems, Man and Cybernetics-Part B*, 26(1):1–13, 1996.

[5] P. Dutta, M. Grimmer, A. Arora, S. Bibyk, and D. Culler. Design of a wireless sensor network platform for detecting rare, random, and ephemeral events. In *he Fourth International Conference on Information Processing in Sensor Networks (IPSN '05)*, pages 497–502. IEEE, 2005.

[6] P. K. Dutta and D. E. Culler. System software techniques for low-power operation in wireless sensor networks. In *Proceedings of the 2005 International Conference on Computer-Aided Design*, 2005.

[7] Garcia. `http://www.acroname.com/garcia/garcia.html`.

[8] J. Gross and J. Yellen. *Graph Theory and Its Applications, Second Edition*. Chapman & Hall/CRC Press, 2005.

[9] 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.

[10] N. Heo and P. K. Varshney. A distributed self spreading algorithm for mobile wireless sensor networks. In *Wireless Communications and Networking*. IEEE, 2003.

[11] A. Jacoff, E. Messina, B. A. Weiss, S. Tadokoro, and Y. Nakagawa. Test Arenas and Performance Metrics for Urban Search and Rescue Robots. In *Proceedings of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2003.

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

[13] 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.

[14] J. Kurose and K. Ross. *Computer Networking: A Top-Down Approach Featuring the Internet*. Pearson Education, 2005.

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

[16] K. Pister. `http://robotics.eecs.berkeley.edu/ pister/SmartDust/`.

[17] A. Rogers, E. David, and N. R. Jennings. Self-organized routing for wireless microsensor networks. *IEEE Transactions on Systems, Man, and Cybernetics—Part A: Systems and Humans*, 35(3), 2005.

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