

Robot and Sensor Networks for First Responders

A network of distributed mobile sensor systems can help first responders during emergencies. Experiments conducted at a burning building with firefighters show the sensors' potential to localize themselves and acquire and combine information, providing an integrative view of a dynamically changing environment.

The Oklahoma City bombing and attacks on the World Trade Center are unforgettable events in which humans were ill-equipped to respond. The need to collect, integrate, and communicate information effectively in emergency response scenarios exceeds the state of the art in information technology. Commanders can't easily locate their personnel or diagram the inside of affected structures. Sharing information between firefighters is limited to verbal communication over radio, because visibility inside a burning building can be reduced to inches. Furthermore, it's impossible to send humans to some places, either because the access area is too small or the danger too great. This emergency response problem provides an interesting and important test bed for studying networks of distributed mobile robots and sensors.

Here, we describe the component technologies (see the "Related Work" sidebar) required to deploy a networked-robot system that can augment human firefighters and first responders, significantly enhancing their firefighting capabilities. In a burning building at a firefighting training facility, we deployed a network of stationary

Mote sensors, mobile robots with cameras, and stationary radio tags to test their ability to guide firefighters to targets and warn them of potential dangers. Our long-term vision is a physical network that can sense, move, compute, and reason, letting network users (firefighters and first responders) Google for physical information—that is, information about the location and properties of physical objects in the real world.

Our vision

The physical network we're building as part of a collaborative National Science Foundation Information Technology Research project will include several types of nodes (or agents). Some will be small mobile robots (either autonomous or teleoperated), some might be stationary sensors placed in the environment during an operation, and others might be computers embedded in the suits of emergency response personnel.

During an operation, we envision that, along with emergency personnel, tens of agents will enter a building about which potentially little is known (see Figure 1a). If floor plans are available a priori, agents will use them to expedite the search process, acquiring information and providing an integrated view for situational awareness. The agents' small size will let them penetrate nooks and niches, possibly being teleoperated by a human operator.

Vijay Kumar
University of Pennsylvania

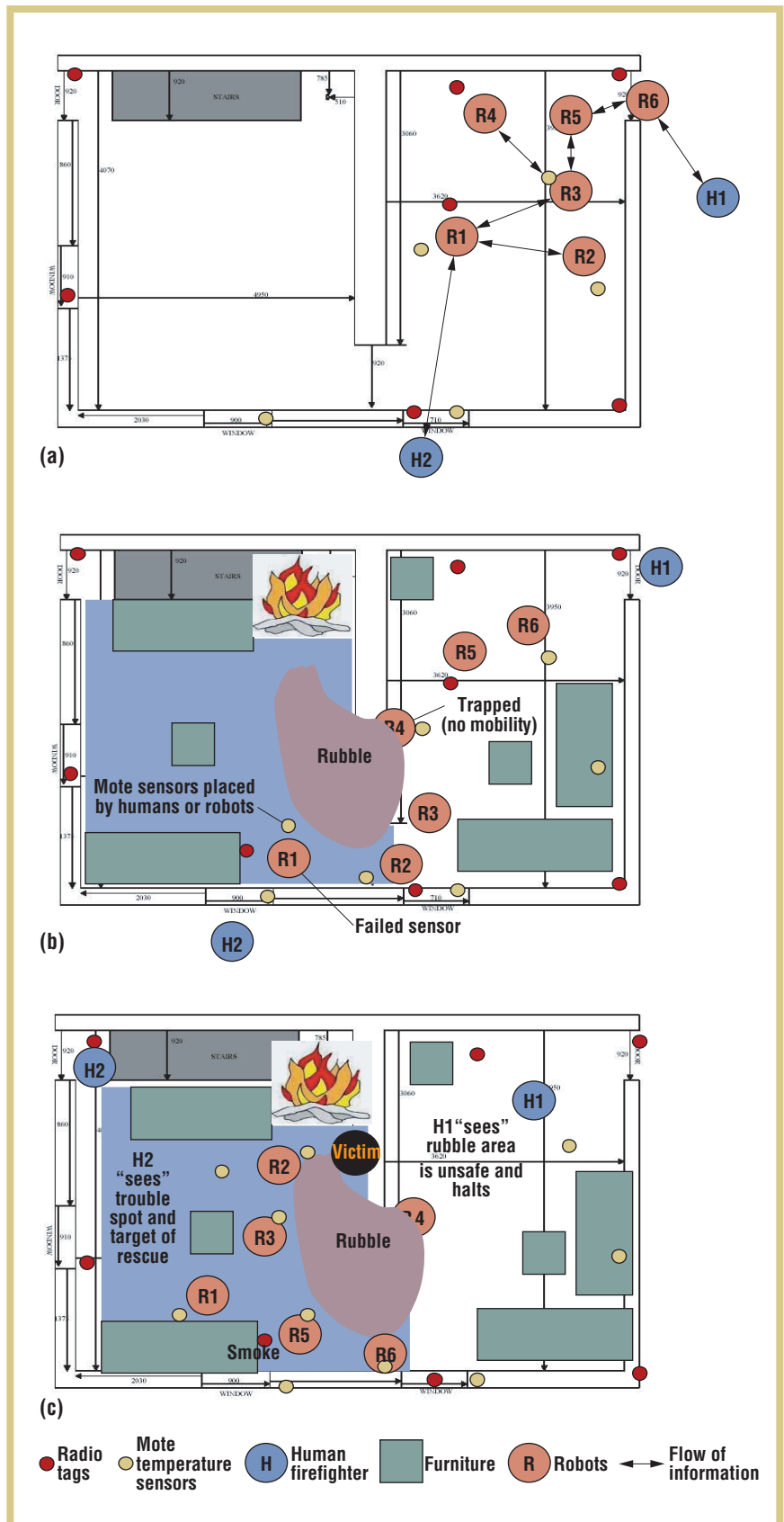
Daniela Rus
Massachusetts Institute of Technology

Sanjiv Singh
Carnegie Mellon University

Figure 1. A motivating scenario:
 (a) A team of six robots enter a burning building, dispersing radio tags and Mote temperature sensors. (b) The team adapts to failures, using robots with failed actuators as static sensors and relays, and guiding robots with failed sensors through smoke and past rubble using information obtained via the network. (c) The network guides human firefighters to potential targets for rescue and to the fire's source while warning them of dangerous areas along the way.

Once inside, the agents will autonomously organize themselves to communicate effectively, integrate information efficiently, and obtain relative position information quickly. They will record temperature gradients, measure concentrations of toxins and relevant gases, track sources of danger, and look for human victims. They will then cordon off areas of threat (for example, areas where the temperature is greater than 300°F) and convey to remote human operators information about the environment and about emergency response personnel inside the building (see Figure 1b). Information broadcast from each group will be integrated into an immersive environment that rescue workers and firefighters can visualize on remote workstations or helmet-mounted displays (see the “Panoramic Display” sidebar). Additionally, a microphone on the agent will let a trapped human call for help, and the agent’s ability to localize (identify its spatial location) with assistance from its neighbors will help rescuers find the trapped victim (see Figure 1c).

Losing inexpensive robots because of fire or falling debris is much more acceptable than human injuries or deaths. For example, in the event of contamination from hazardous materials arising from a spill or transportation accident, we could deploy the network to determine the extent of the contamination and the speed with which it’s propagating. We could integrate simul-



Related Work

Current research in pervasive computing,^{1,2} sensor networks,³ and active perception⁴ is relevant to our work. In addition, several other projects are investigating search and rescue applications. Perhaps the two most relevant projects are the Scout project at the University of Minnesota, where small robots are used for mapping and exploration, and the University of South Florida effort to effectively deploy remotely operated, radio-controlled robots in real disaster areas, including a successful attempt at the World Trade Center.⁵ There is even a competition for robot search and rescue.⁶

In contrast to these efforts, our focus is on deriving useful functionality from the network and autonomously deploying the network. So, the results we describe in this article are first steps toward developing autonomous networked solutions to control and perception problems that arise in emergency response scenarios.

From an application viewpoint, our work differs from the work at the University of South Florida in that our project concerns automating support for first responders as opposed to search and rescue. First responders need real-time deployment of communication and computation infrastructure. They need adaptive, active, and proactive processing for information flow, and their world is dynamic and dominated by the difficulties of the rapidly changing threat (such as fire) and uncertainties associated with the information. In search and rescue, the automation is post hoc and the real-time adaptive and

proactive aspect of the computation isn't crucial. The two problem domains (support for first responders and search and rescue) are synergistic and solutions to both problems are needed to develop a comprehensive program that supports national infrastructure security.

REFERENCES

1. M. Satyanarayanan, "A Catalyst for Mobile and Ubiquitous Computing," *IEEE Pervasive Computing*, vol. 1, no. 1, 2002, pp. 2–5.
2. D. Estrin et al., "Connecting the Physical World with Pervasive Networks," *IEEE Pervasive Computing*, vol. 1, no. 1, 2002, pp. 59–69.
3. J. Hill et al., "System Architecture Directions for Network Sensors," *Proc. 9th Int'l. Conf. Architectural Support for Programming Languages and Operating Systems (ASPLOS IX)*, ACM Press, 2000, pp. 93–104.
4. M. Pollefeys, R. Koch, and L.J. Van Gool, "Self-Calibration and Metric Reconstruction in Spite of Varying and Unknown Internal Camera Parameters," *Proc. IEEE Int'l Conf. Computer Vision (ICCV 98)*, IEEE CS Press, 1998, pp. 90–95.
5. J. Casper and R. Murphy, "Workflow Study on Human-Robot Interaction in Usar," *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA 02)*, IEEE Press, 2002, pp. 1997–2003.
6. R. Murphy, J. Blitch, and J. Casper, "AAAI/Robocup-2001 Urban Search and Rescue Events Reality and Competition," *AI Magazine*, vol. 23, no. 1, 2002, pp. 37–42.

taneous measurements of chemical concentration over a widespread, distributed area, letting the network follow the source's movement—for example, following a chemical plume as it spreads in the air, a moving object on the ground, or the source of a fire.

As a first step toward realizing this vision, we conducted experiments in a burning building at the Allegheny County firefighting training facility in Pittsburgh. Based on these experiments, we highlight three key challenges to realizing an intelligent robot network: localizing robots, maintaining the flow of relevant information, and controlling the robot network.

Localization

Localization in the dynamic environments that search and rescue operations pose is difficult because we can't pre-

sume an infrastructure or even make simple assumptions—for example, we can't assume responders will be able to see known features. The system thus must be able to localize each node in the network, including those attached to firefighters.

Our method

We propose combining small (12×9 cm), low-cost (approximately US\$40) radio tags (or beacons) with conventional line-of-sight optical sensors, because rescue personnel and robots can easily disperse the radio tags, and optical sensors can operate in the visible or infrared (IR) spectrum. A robot can query these radio tags, and any tags within range will reply. The robot can then estimate the distance to each responding tag by determining the time elapsed between sending the query and receiving the response.

Because this method doesn't require a line of sight between the tags and the mobile robot, it's useful in many environmental conditions where optical methods fail. Because each tag transmits a unique ID number, robots can automatically associate distance readings with the appropriate tags, easily solving the data association problem—a difficult issue in environments that can be visually obscured. Robots that first responders can't locate using the RF system might be able to sense their position with respect to other RF-located robots, using the robots' optical sensors.

Because we don't initially know where the tags are located, and because their positions can change during operation, robots must localize both the receiver and the tags simultaneously. This problem is often known as Simultaneous Localization and Mapping (SLAM).¹ Although

Panoramic Display

When robots or people interact with a sensor network, it becomes an extension of their capabilities. We've thus developed software that allows an intuitive, immersive display of environments. Using panoramic imaging sensors that small robots can carry into the heart of a damaged structure, the display can be coupled to head-mounted sensors. These sensors then let a remote operator look around in the environment without the delay associated with mechanical pan and tilt mechanisms (see Figure A). Distributed protocols collect data from the geographically dispersed sensor network and integrate this data into a global map (such as

a temperature gradient) that can also be displayed on a wearable computer to the user.

Even if practical limitations on size and packaging make it difficult to display this information to firefighters, such information would be invaluable to commanders outside the building, who typically have limited information about where firefighters are located and the local information accessible to each individual. According to the National Institute for Occupational Safety and Health, such information can help commanders better coordinate and guide firefighters.



Figure A. The 360° panorama of a room in the burning building taken by a catadioptric system.

it's generally assumed that a receiver can measure both range and bearing to "features," we can determine only range—and even that measurement might be very noisy. To localize robots using range-only measurements, we adapted the well-known estimation techniques of Kalman filtering, Markov methods, and Monte Carlo localization.² All three methods estimate robot position as a distribution of probabilities over the space of possible robot positions. We also have an algorithm that can solve SLAM in cases where approximate a priori estimates of robot and landmark locations exist.²

The primary difficulty stems from the annular distribution of potential relative locations, resulting from a range-only measurement. Because the distribution is highly non-Gaussian, SLAM solutions derived from Kalman filters are ineffective. In theory, Markov methods (prob-

ability grids) and Monte Carlo methods (particle filtering) have the flexibility to handle annular distributions. In fact, they have more flexibility than we need—they can represent arbitrary distributions, but we need only deal with well-structured annular distributions. Unfortunately, the scaling properties of these methods severely limit the number of landmarks robots can map.

We've implemented a compact way to represent annular distributions together with a computationally efficient way of combining annular distributions with each other and with Gaussian distributions. In most cases, we expect the results of these combinations to be well approximated by mixtures of Gaussians so that we can apply standard techniques such as Kalman filtering or multiple hypothesis tracking to solve the remaining estimation problem.

We've also extended these results to deal with the case in which the tag locations are unknown using a simple bootstrapping process. The idea is to use the best estimate of robot localization along with measured ranges to an unknown tag to coarsely estimate the tag's location. Once we have an estimate, it's added to the Kalman filter, which improves the estimate along with estimates of other (previously seen) tags. Because this method takes advantage of the problem's special structure (in that it's an easier mapping problem than a simultaneous location and mapping problem), it's less computationally cumbersome in that it reasonably estimates the tag's location, avoiding local minimas.

We can also combine range-only measurements with range and bearing or bearing-only measurements available from robot cameras. Our robots are

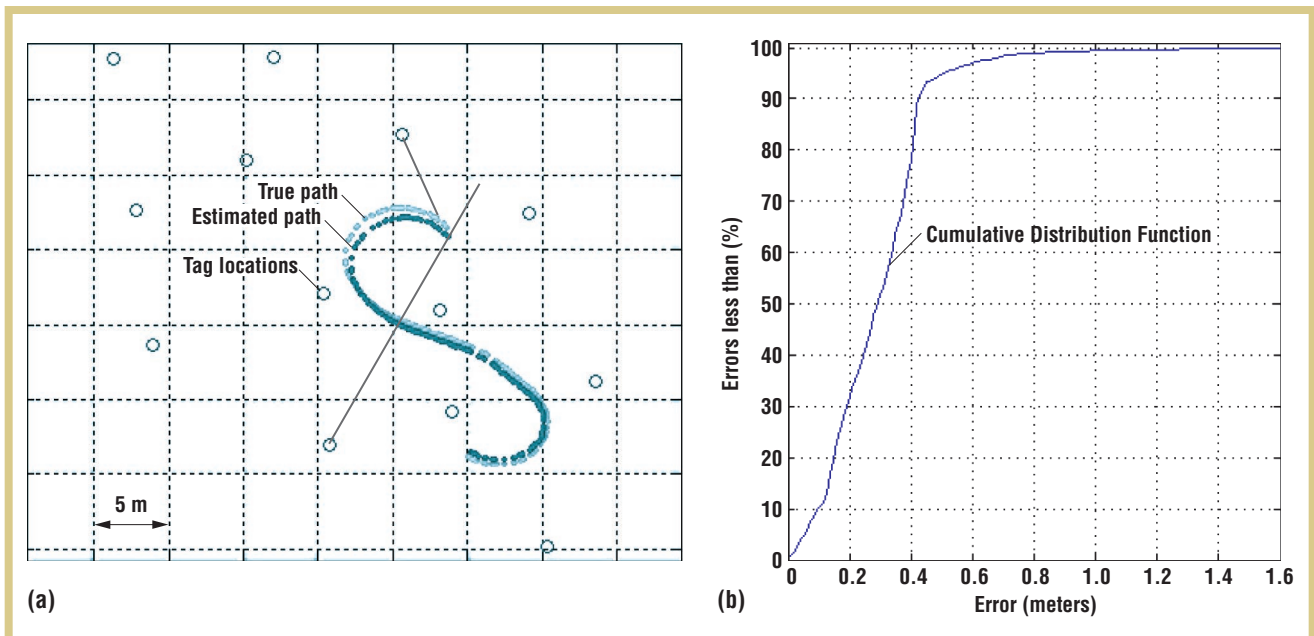


Figure 2. (a) 13 RF tags in known locations, in conjunction with a wheel encoder and gyro, localize a robot moving in an open area. The lines between the paths and tags show two range measurements (one noisy). (b) A cumulative distribution function of errors in localization from an extended run, showing the percentage of errors for each range (in meters). During this run, the vehicle traveled 1,401 m. Over the course of the run, 14,633 range readings were received, of which 163 were rejected. The mean Cartesian error was 0.296 m with a standard deviation of 0.1627 m.

equipped with omnidirectional cameras that currently work in the visible and near-IR range. IR omnidirectional cameras are feasible but are not yet off-the-shelf.

Experiment and results

To benchmark the radio tags' performance, we conducted localization experiments in a 30 m × 40 m flat, open, outdoor environment with an autonomous robot where we could use highly accurate (2 cm) GPS receivers, a fiber-optic gyro, and wheel encoders for ground truth. We equipped the robot with an RF ranging system (Pinpoint from RF Technologies) that has four antennae pointing in four directions and a computer to control the tag queries and process responses.

For each tag response, the system produces a time-stamped distance estimate to the responding tag, along with the unique ID number for that tag. The distance estimate is simply an integer estimate of the distance between the robot and tag. A calibration step conducted

ahead of time produces a probability density function (PDF) for each measurement. The calibration step takes many measurements from candidate environments, using them to determine an offset of the mean and standard deviation for each reading. That is, for each range reading we could measure, we calculated the offset of the experimental mean from the reading and the standard deviation of the reading. We expect that this calibration needs to be done only once to get noise characteristics based on various environments.

We distributed 13 RF tags throughout the area and then programmed the robot to drive in a repeating path among the tags. With this setup, we collected three kinds of data: the robot's ground truth path (from GPS and inertial sensors), the robot's dead-reckoning estimated path (from inertial sensors only), and the range measurements to the RF tags. Figures 2 and 3 show results from these experiments (greater details of the data set and results obtained appear else-

where³). Although the error for individual range readings has a standard deviation of as much as 1.3 m, the tags localized the robot to within 0.3 m of its location.

Although our results are preliminary because they were obtained outdoors, they're still promising. It's important to recognize that multipath can severely affect the time-of-flight measurement in populated indoor environments. However, our preliminary experiments in office buildings show that we can reject multipath by estimating position uncertainty to accept or reject range readings using a mechanism such as a chi-squared filter. That is, the more uncertain the position estimate, the more likely the algorithm is to incorporate a large difference between measured and expected range.

Additionally, because we don't assume we'll get range readings from all or any particular set of tags, the fact that only some might be visible at any given time isn't a problem, because we use dead reckoning between range readings. If the

Figure 3. When we don't initially know a tag's location, we can approximate it using a batch scheme first and then add it to a Kalman filter to continuously update the tag's position along with the robot's position.

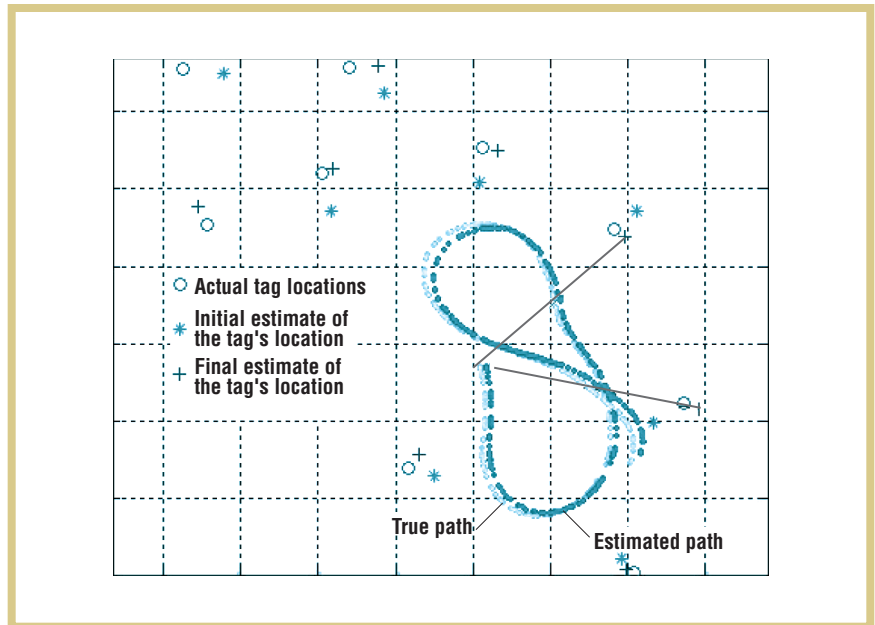
range readings are obtained in a fashion such that little overlap exists between the tags that are heard (as in a long linear traverse), then odometric errors come into play and the map is topologically correct rather than metrically correct. (The reference frame for the map and localization is arbitrary and can be either specified by an axis formed by two of the tags or by the robot's initial starting position, such as at a building's entrance.)

In addition to localizing robots, which have the potential to interact with radio tags and other robots, it's also necessary to localize other sensors. A robot deploying the sensors can localize them using techniques such as the robot-assisted localization method.⁴

Information flow

Sensors can locally store information about the area they cover or forward it to a base station for further analysis and use. They can also use communication to integrate their sensed values with the rest of the sensor landscape. With localized information, we can build global maps of sensory information (such as a temperature map) and use them to navigate humans or robots to a target, avoiding potentially dangerous areas. Network users (robots or people) can use this information as they traverse the network.

In the Allegheny experiments, we used Mote sensors to measure environmental conditions such as temperature, humidity, and chemical concentrations of toxins. Each Mote sensor (www.xbow.com/Products/Wireless_Sensor_Networks.htm) consisted of an Atmel ATmega128 microcontroller (with a 4 MHz, 8-bit CPU, a 128-Kbyte flash program space, 4-Kbyte RAM, and 4-Kbyte EEPROM), a 916 MHz RF transceiver (50 Kbps,



100 ft. range), a Universal Asynchronous Receiver-Transmitter, and a 4-Mbit serial flash. A Mote runs for approximately one month on two AA batteries. It includes light, sound, and temperature sensors, but you can add other types of sensors. Each Mote runs the TinyOS operating system. Because of the limited sensing and computational power, Mote sensors must be deployed either manually or by robots at locations that are registered on the map.

Our work in this area has led to two novel developments. First, we've developed distributed protocols for navigation tasks in which a distributed sensor field guides a user across the field.⁴ Second, we've developed the Sensory Flashlight, a handheld device that lets a network user (human or robot) interact with the network as a whole or talk to individual nodes in the network. An alternative approach would be to tag the Motes with pinpoint devices.

The navigation guidance application is an example of how simple nodes distributed over a large geographical area can assist with global tasks. This application relies on the user's ability to interact with the network as a whole and with specific nodes. This interaction is directed at retrieving data from the net-

work (such as collecting local information from individual nodes and collecting global maps from the network) and injecting data into the network (such as configuring the network with a new task or reprogramming its nodes).

In our experiments,⁵ we started a fire in a room and obtained local temperature data from each sensor. A communication protocol running over the entire system uses the local data to compute a global temperature map in this space, ultimately producing a temperature gradient for the space, computing the map and the gradients during the fire. For this experiment, we manually deployed and localized the Mote sensors at the locations marked in Figure 4a. The Mote sensors established multihop communication paths to a base station placed at the door. The sensors sent data to the base station over a period of 30 minutes, from which we were able to create a temperature gradient (see Figure 4b).

Figure 5a shows our prototype Sensory Flashlight (the name is derived from the optical flashlight metaphor), which lets a mobile node interact with a sensor field. The prototype comprises an analog compass, an alert LED, a pager vibrator, a three-position mode switch, a power switch, a range potentiometer,

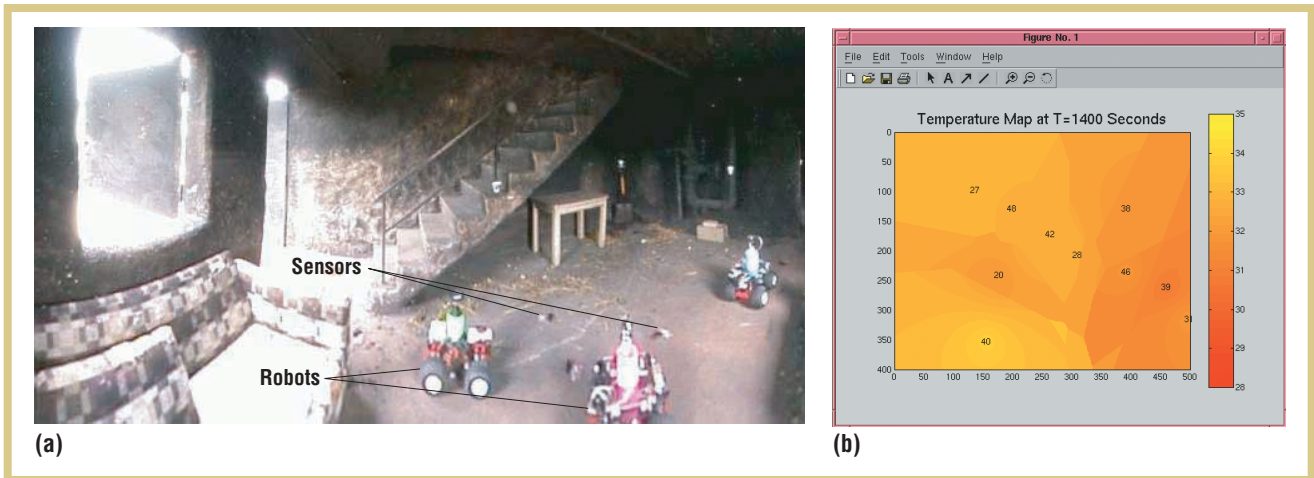


Figure 4. (a) An ad hoc network of robots and Mote sensors deployed in a burning building at the Allegheny Fire Academy, 23 August 2002 (from an experimental exercise involving Carnegie Mellon University, Dartmouth, and the University of Pennsylvania). (b) The temperature gradient graph collected using an ad hoc network of Mote sensors.

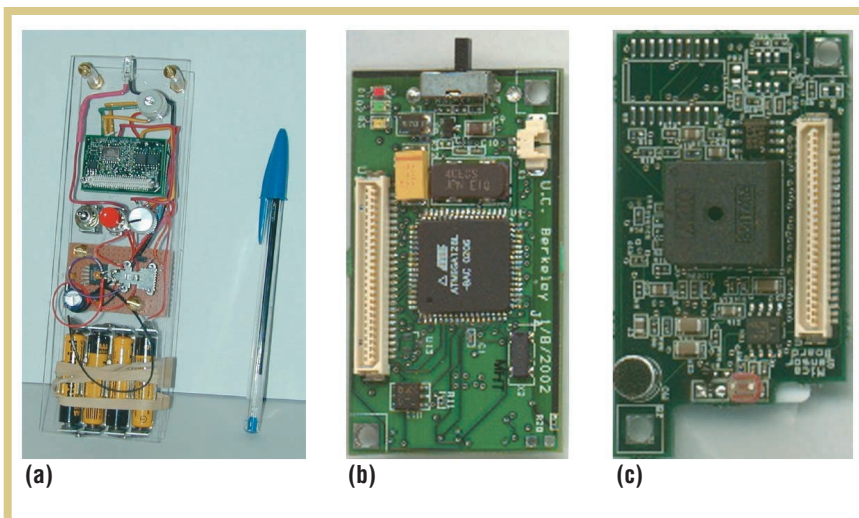


Figure 5. (a) The Flashlight prototype, (b) a Mote board, which has the computer and communication system, and (c) a Mote sensor board, which holds the sensors.

some power conditioning circuitry, and a microcontroller-based CPU/RF transceiver. The processing and RF communication components and the sensor network are Berkeley Motes (Figure 5b and 5c show the Mote board and Mote sensor board).

The Flashlight's "beam" comprises sensor-to-sensor, multihop-routed RF messages that send or return information. A switch selects the sensor type (such as light, sound, or temperature). When pointed in a specific direction, the sensor

reports information received from any sensors in that direction. The human-operated version includes a silent vibrating alarm, with the vibration amplitude encoding the distance (either actual distance estimated or the number of hops) to the triggered sensor. A potentiometer sets the detection range while an electronic compass supplies heading data indicating the direction in which the device is pointed. Thus the flashlight collects information from all the sensors located in that direction and provides its

user with sensory feedback. The device can also issue commands to the sensors in that direction.

We envision using the Flashlight as a handheld or robot-held device that can help guide robots or human responders along relevant paths that might dynamically change over time, because the Flashlight can reconfigure the wireless sensor network in a patterned way. Furthermore, it can facilitate interaction with the network, both consuming and dispersing the network's stored information and changing and reacting to its topology. Finally, it lets the user "mark" geographic regions with information, enabling effective management of sensors and efficiency in message routing.

We deployed 12 Mote sensors along corridors in our building and used the Flashlight and the communication infrastructure presented here to guide a human user out of the building (see Figure 6). The Flashlight was used by a human in the building to interact with sensors to compute the next direction of movement toward the exit. For each interaction, the user did a rotation scan until the Flashlight was pointed in the direction computed from the sensor data. The user then walked in that direction to the next sensor. Each time, we recorded the correct direction and the

direction the Flashlight detected. Its directional error was eight percent (or 30 degrees) on average. However, because the corridors and office doorways are wide, and the sensors sufficiently dense, it successfully identified the exit and never directed the user toward a blocked or wrong configuration.

Certainly, a device such as the Flashlight raises many practical and theoretical questions: How dense should the sensor network be for such applications, given the feedback accuracy? How can the Flashlight's functionality be integrated into mobile robots? It would also be useful to integrate multiple sensors and multiple directions into the Flashlight. These are some directions of ongoing and future work.

Network control

Robots augment sensor networks' surveillance capabilities using mobility. We envision large, networked groups of robots and sensors operating in dynamic, resource-constrained, adversarial environments. The constraint in resources comes from the fact that each node might have to operate for relatively long periods without recharging and without human attention, while the adversarial conditions are typical of emergency response. Because there could be numerous agents and because the system must be robust enough to add and delete new sensors or vehicles, each agent must be anonymous. Furthermore, each node must operate asynchronously and be agnostic to who its neighbor is (in other words, the neighbors are interchangeable). There will be variations in dynamics from agent to agent. However, while individual agents might not exhibit performance guarantees, groups offer guarantees at the population level that algorithm designers must model and understand. Groups of this type will typically operate with little or no direct human supervision, and it will be diffi-

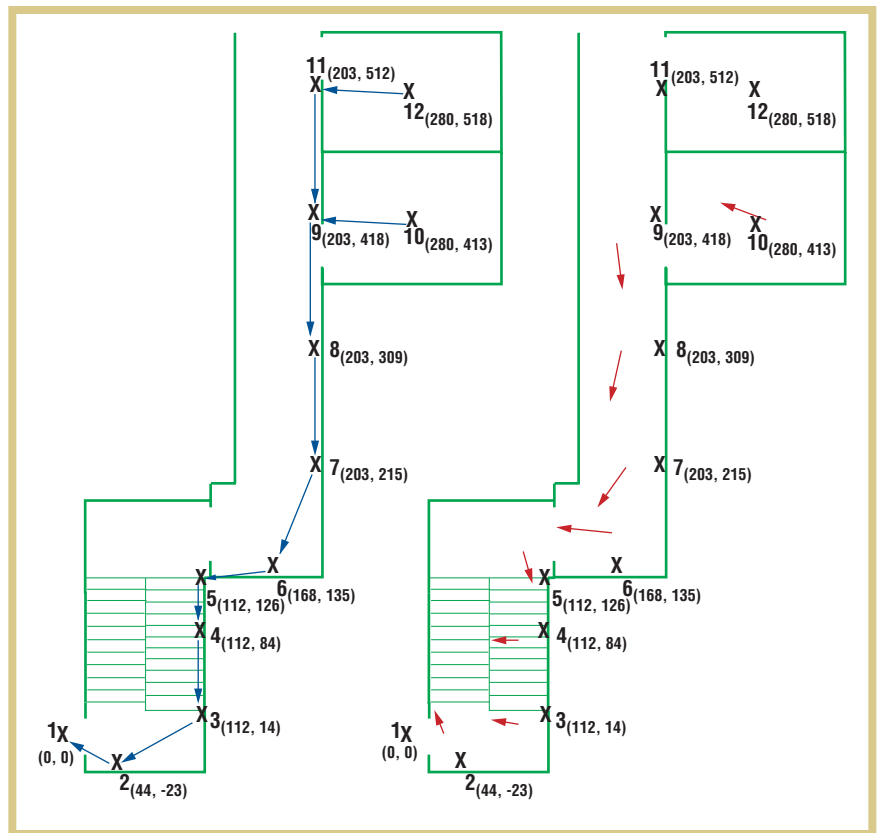


Figure 6. The floor map for the directional guidance experiment. Blue arrows indicate the correct direction to be detected by the Flashlight; red arrows show Flashlight-computed feedback directions.

cult, if not impossible, to efficiently manage or control such groups through programming or teleoperation. Thus, managing such large groups will be extremely challenging and will require the application of new, yet-to-be-developed methods of communication, control, computation, and sensing specifically tailored to the control of autonomously functioning robot networks.

The robots used in our experiments are car-like robots equipped with omnidirectional cameras as their primary sensors.⁵ The communication among the robots relies on IEEE 802.11b networking. By using information from its camera system, each robot can estimate only its distance and bearing from its teammates. However, if two robots exchange information about their bearings to each other, they can also estimate their relative orientations. We use this idea to com-

bine the information of a group of two or more robots to improve the group's knowledge about its relative position. We can combine this information with radio tag information to get local or global position and orientation information.

We decompose the team's motion into a gross position (and orientation) descriptor g and its shape or distribution ρ . We can model g as an element of a Lie group (a differentiable manifold obeying the group properties and satisfying the additional condition that the group operations are differentiable), while ρ depends on the representation used for the shape space. Previously, we've established the mathematical framework for decentralized controllers that let a team of robots achieve a desired position and shape trajectory.⁶ This work also showed how to compose such deliberative motion controllers with reactive controllers that are

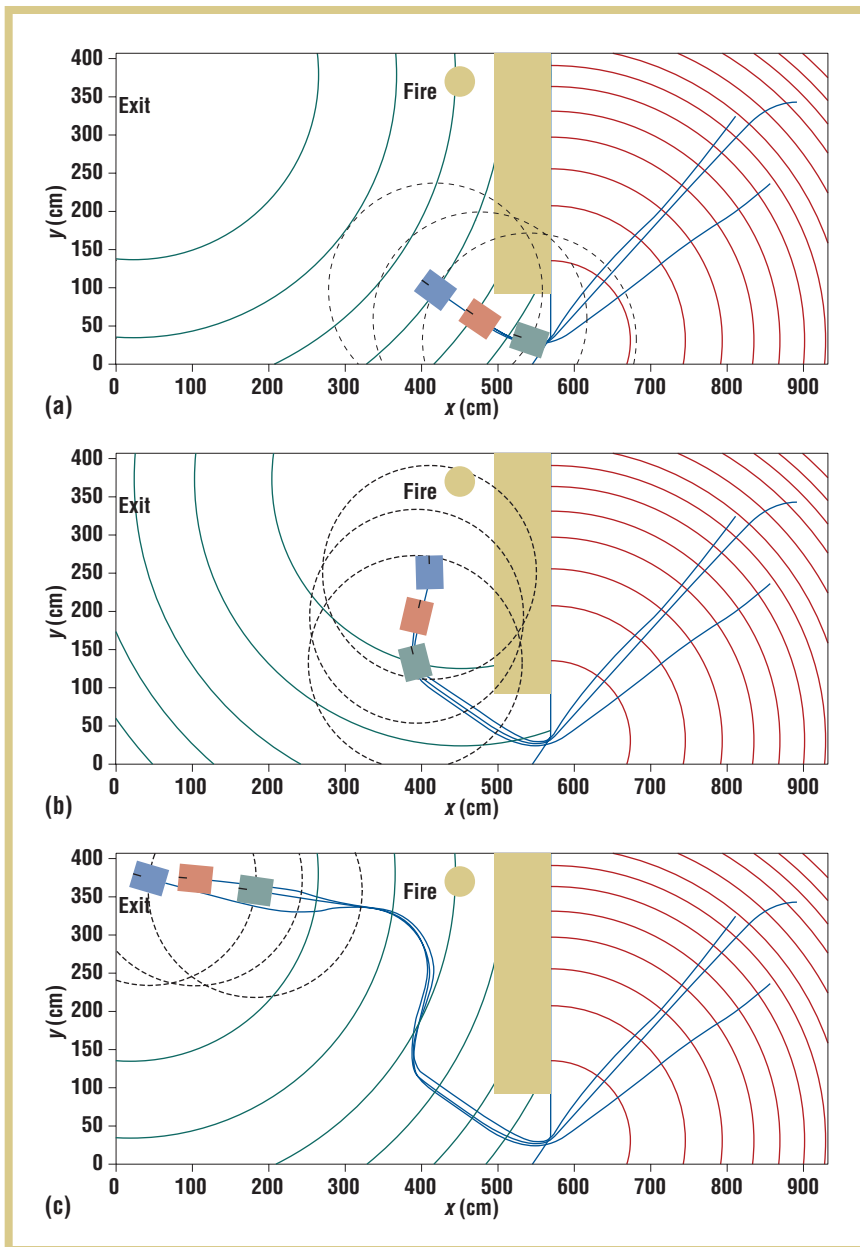


Figure 7. Three robots switching motion plans in real time to get information from the building's hottest spot: (a) In the absence of any significant temperature gradient detected by the sensors, the robots can use a potential-field controller to navigate the building, traversing specified waypoints while maintaining formation. (b) A network of Mote sensors distributed on the ground obtains a gradient of temperature. (c) At any point, when commanded by a human, the robots can switch to a new goal.

establish gradient information. When this gradient is significant, the robots switch controllers and follow the temperature gradient toward the fire's source (see Figure 7b). Omnidirectional visible and IR cameras provide information back to one or more remotely located human responders. At any point, when commanded by a human, the robots can switch back to a designated goal, ignoring the temperature gradient (see Figure 7c).

We can easily tailor these algorithms to adapt to constraints that sensors impose (such as a limited field of view) and radios (such as a weak signal). This lets robots stay within reasonable distances of each other so that they can function as a team. They're also scalable to large numbers of robots¹ and lend themselves to a "swarm" paradigm for navigation and autonomous exploration.

necessary to circumvent unexpected obstacles. Other work discusses abstractions and control laws that allow partial state information to be used effectively and in a scalable manner for numerous units.⁷

In this example, we assume that a nominal map of the building is available and that a network of Mote temperature sensors has been deployed in the building. The nominal map lets robots develop potential-field-based planners to navigate while achieving a desired

shape. The ability to communicate with each other and with Mote temperature sensors lets the team incrementally build a global temperature map.

In the absence of any significant temperature gradient detected by the sensors, the robots can use a potential-field controller to navigate the building, traversing specified waypoints while maintaining formation (see Figure 7a). Because each robot can communicate with Mote sensors in its neighborhood, they each sense a different temperature and thus

Although this set of experiments didn't provide a complete demonstration, it did reveal component solutions that can enhance the first responders' capabilities. The firefighters were quick to see the potential but also sensitized us to the extreme environments they must function in.

For example, we can't expect firefighters to carry anything substantial or deploy devices carefully. Similarly, because firefighters often must crawl to

avoid the smoke and heat, this has implications for designing displays and developing the sensors they wear. Furthermore, even though floor plans of buildings in many US cities have been computerized and are available over the Internet, firefighters have no way of integrating this information for emergency response. Finally, firefighters and emergency response personnel who have used robotic devices claim that robots are most useful as enablers for sensing. However, the need to remotely control each robot means that significant time and manpower is spent during an operation that necessitates speed of response and efficiency. This points to the need for autonomy with robot and sensor networks, and the infrastructure to support network deployment.

Addressing these needs will require surmounting many challenges in mechanical design, control, sensing and estimation, and human-robot interaction—which is why first response is such a compelling and unifying application for pervasive computing. In our current work, we're developing scalable algorithms for large numbers of networked heterogeneous agents (some simple and specialized, some static, some mobile, some controllable, some human) that will have to coordinate in an environment that might be poorly known, dynamic, and obscured. Because each sensor, by itself, might not provide useful information—and might not even survive—the functionality stems from the network and its ability to adapt on-the-fly.

As part of our collaborative NSF ITR project, we're also examining the synergy between communication, control, and perception in support of such applications and developing new algorithms for communication-assisted control, communication-assisted sensing, and control and perception-assisted communication. We'll explore in greater depth how to use a communication network to control the



Room 111, Towne Bldg., 220 S. 33rd St., Philadelphia, PA 19104-6315; kumar@central.cis.upenn.edu.

Vijay Kumar is a professor and the deputy dean for research in the School of Engineering and Applied Science at the University of Pennsylvania. He also directs the GRASP Laboratory, a multidisciplinary robotics and perception laboratory. His research interests include robotics, dynamical systems, and multiagent coordination. He received his PhD in mechanical engineering from Ohio State University. He is a fellow of the American Society of Mechanical Engineers and a senior member of the IEEE and of Robotics International, Society of Manufacturing Engineers. Contact him at the Dept. of Mechanical Engineering & Applied Mechanics, Univ. of Pennsylvania, Room 111, Towne Bldg., 220 S. 33rd St., Philadelphia, PA 19104-6315; kumar@central.cis.upenn.edu.



Daniela Rus is an associate professor in the Electrical Engineering and Computer Science department at MIT. Her research interests include distributed robotics, mobile computing, and self-organization. She received her PhD in computer science from Cornell University. Contact her at MIT Computer Science and Artificial Intelligence Laboratory, The Stata Center, Bldg. 32-374, 32 Vassar St., Cambridge, MA 02139; rus@csail.mit.edu.



Sanjiv Singh is an associate research professor at the Robotics Institute, Carnegie Mellon University. His recent work includes perception in natural and dynamic environments (developing autonomous ground and air vehicles to navigate with little or no a priori information using onboard sensors) and multiagent coordination (developing teams of agents made of humans, embedded sensors, and robots to perform tasks more efficiently and reliably). He received his PhD in robotics from Carnegie Mellon. Contact him at Robotics Inst., Carnegie Mellon Univ., Pittsburgh, PA 15213; ssingh+@cmu.edu; www.frc.ri.cmu.edu/~ssingh.

mobility of the agents as a group and how each agent can use the network to infer its relative location and develop an integrated model of its environment. We're also exploring how the agents' ability to control their position and orientation can help sustain the communication network. ■

ACKNOWLEDGMENTS

We are grateful to George Kantor, Ron Peterson, Aveek Das, Guilherme Pereira, and John Spletzer for their contributions to developing and implementing this system. We thank the staff at the Allegheny Fire Academy for their help and most valuable advice. Support for this work has been provided in part by NSF grants IIS-0426838, CCR02-05336, and 0225446, ISTS ODP 2000-DT-CX-K001, Intel, Boeing, and ISTS at Dartmouth. Finally, we thank the reviewers for their careful reading and thoughtful suggestions.

REFERENCES

1. S. Thrun, W. Burgard, and D. Fox, "A Probabilistic Approach to Concurrent Mapping and Localization by Mobile Robots," *Machine Learning*, Apr. 1998, pp. 29–53.
2. G. Kantor, S. Singh, and D. Strelow, "Recent Results in Extensions to Simultaneous Localization and Mapping," *Proc. 8th Int'l Symp. Experimental Robotics*, Springer Tracts in Advanced Robotics, vol. 5, Springer-Verlag, 2003, pp. 210–221.
3. D. Kurth, G. Kantor, and S. Singh, "Experimental Results in Range-Only Localization with Radio," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, IEEE CS Press, 2003, pp. 974–979.
4. P. Corke, R. Peterson, and D. Rus, "Networked Robotics: Flying Robot Navigation with a Sensor Network," *Proc. 2003 Int'l Symp. Robotics Research*, Springer Tracts in Advanced Robotics, Springer-Verlag, 2003.
5. A. Das et al., "Distributed Search and Rescue with Robot and Sensor Teams," *Proc. Field and Service Robotics (FSR 03)*, to be published, Springer-Verlag.
6. A. Das et al., "Vision Based Formation Control of Multiple Robots," *IEEE Trans. Robotics and Automation*, vol. 18, no. 5, 2002, pp. 813–825.
7. C. Belta and V. Kumar, "Abstractions and Control Policies for a Swarm of Robots," *IEEE Trans. Robotics and Automation*, vol. 20, no. 5, 2004, pp. 865–875.

For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/publications/dlib.