Human-Robot Teaming for Search and Rescue

This work establishes an architecture for Urban Search and Rescue and a methodology for mixing real-world and simulation-based testing. A sensor suite and sensor fusion algorithm for robust victim detection permits aggregation of sensor readings from various sensors on multiple robots.

Illah R. Nourbakhsh, Katia Sycara, Mary Koes, and Mark Yong
The Robotics Institute

Michael Lewis
University of Pittsburgh

Steve Burion
Swiss Federal Institute of Technology (EPFL)

Five years ago, Hiroaki Kitano proposed Urban Search and Rescue as a Grand Challenge problem to the robotics community, arguing there were significant challenges to be surmounted in the areas of human-robot interaction and multirobot operation. In response, the US National Institute of Standards and Technologies introduced physical USAR *reference test arenas*—environments "designed to represent, at varying degrees of verisimilitude, challenges associated with collapsed structures."[1] Since then, USAR has emerged as the canonical human-robot interaction (HRI) problem, presenting an obstacle-ridden, unknown environment that can challenge robotic exploration even with the best of human assistance.

NIST originally designed the arenas to advance research in autonomous robotics, but no team has yet succeeded in operating robots autonomously. Recent NIST efforts have turned to identifying interface features and HRI strategies that lead to successful human-robot joint exploration.[2] Beyond the arena wall, little experience exists with robots in actual USAR operations. Most of this comprises reports by Robin Murphy and her students of their experiences at the World Trade Center[3] and in disaster response training exercises.[4] Conclusions drawn from such real-world experiments support the conventional wisdom that robot interface design must remain a critical aspect of USAR robotics research for such systems to function and support real-world disaster settings.

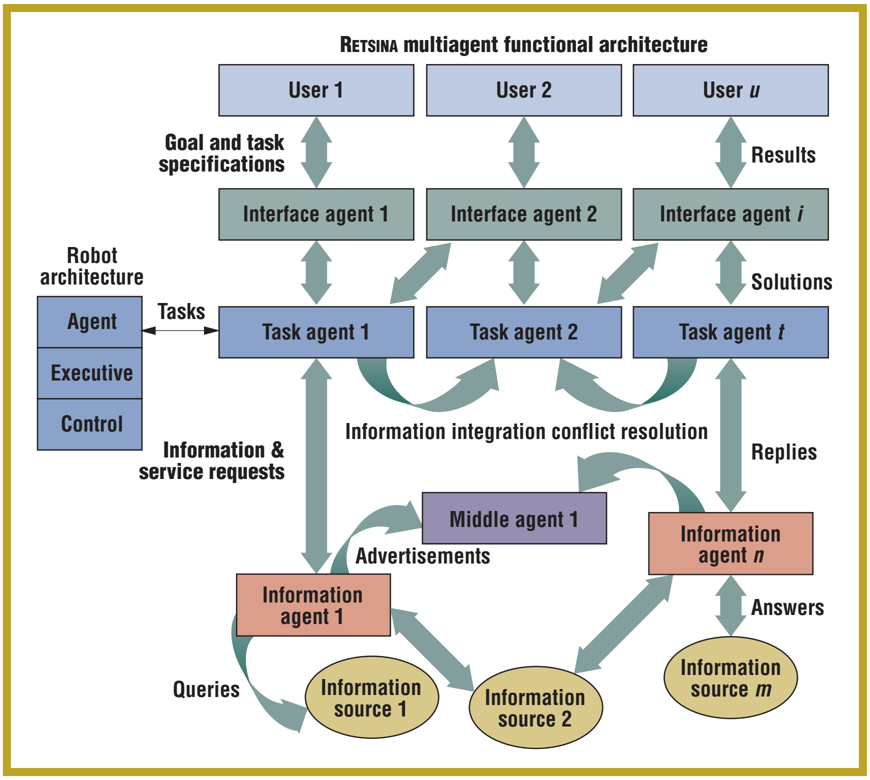
In cooperation with NIST, we have embarked on a research program focusing on the enabling technologies of effective USAR robotic rescue devices. The program is also researching system-level design, evaluation, and refinement of USAR rescue architectures that include teams of sensor-laden robots and human rescuers. Here, we present highlights from our research, which include our *multiagent system* (MAS) infrastructure, our simulation environment, and our approach to sensor fusion and interface design for effective robotic control.

An agent-based architecture

Although the vision of robots working seamlessly with humans and software agents to save lives in an urban disaster is attractive, its realization requires significant scientific advances to address some fundamental challenges. One challenge is coordinating the actions of a set of heterogeneous robots; existing multirobot coordination algorithms and systems are ill suited to this domain. Most systems implemented on robots elicit emergent behavior wherein individual robots follow simple coordination rules, without any explicit teamwork models or goals. This breaks down

 Published by the IEEE CS and IEEE ComSoc ■ 1536-1268/05/$20.00

Figure 1. The robot agent architecture enables robot integration in the RETSINA multiagent system functional architecture by transforming physical robots into embodied task agents.



when a team includes people because the robots can't explain their actions and their role as a team player. Other challenges include facilitating interoperability between existing teams—including robot teams and human teams developed and trained separately—and reasoning about dynamically changing goals across these teams. Efficient solutions to these problems require communication and teamwork models.

Current MAS research addresses these issues. An MAS is a loosely coupled network of agents that interact to solve problems that are beyond a single agent's capacity or knowledge. MASs offer distributed computation; resist individual agent failure; facilitate multiple existing legacy systems' interconnection and interoperation; and can efficiently retrieve, filter, and globally coordinate information from sources that are spatially distributed.

The RETSINA MAS

In our disaster response team instantiation, we used RETSINA,[5] an MAS that categorizes agents into four types based on their function. *Interface agents* facilitate user interaction. *Task agents* seek to accomplish user goals. *Middle agents* provide infrastructure for dynamic runtime discovery of robots and agents that can perform a given task. Such discovery is important when you need to find a replacement for a damaged or failed robot. For example, in real disaster environments, rescue organizations arriving at unpredictable times must be able to dynamically identify robots with needed abilities while coordinating and fielding resources. *Information agents* can access various external information sources, such as disaster site blueprints, hazardous-materials shipping records, and other vital information. Researchers in academia and industry have used RETSINA successfully in aircraft maintenance, demining activities, military logistics planning, and financial portfolio management.[5]

Applying RETSINA to real robots

Extending the MAS infrastructure to teams that include physical agents poses two main challenges. First, each robot must have *social awareness*—knowledge of the MAS infrastructure's existence and how to use the infrastructure to function as a team member. Although robots may vary in morphology and capability, they must have a reasoning layer consistent with task agents in the RETSINA system. We developed a novel robot architecture that transforms a physical robot into a robot agent. The robot agent architecture (see Figure 1) extends the commonly used three-tier architecture, with each higher layer enforcing a functional abstraction on the layer below and each lower layer decreasing the look-ahead horizon while increasing detail. The agent layer contains high-level reasoning and RETSINA communication modules, including the necessary social awareness for interaction. The executive layer is responsible for plan execution and monitoring deviations. The control layer encapsulates the physical robot behavior implementation. This representation combines the functional abstraction of standard three-tier architectures with a high-level semantic abstraction that transforms the robot into a robot agent suitable for conventional software agent coordination and cooperation.

A second challenge in extending MASs to physical agents arises from communication challenges. Physical agents share the same high-level communication requirements as software agents but must also communicate information about their state and the environment state. This low-level communication has high frequency and low latency requirements, yet because the robots are mobile and must communicate wirelessly, the available bandwidth is significantly lower for multirobot systems than multi-software-agent systems. An MAS usually

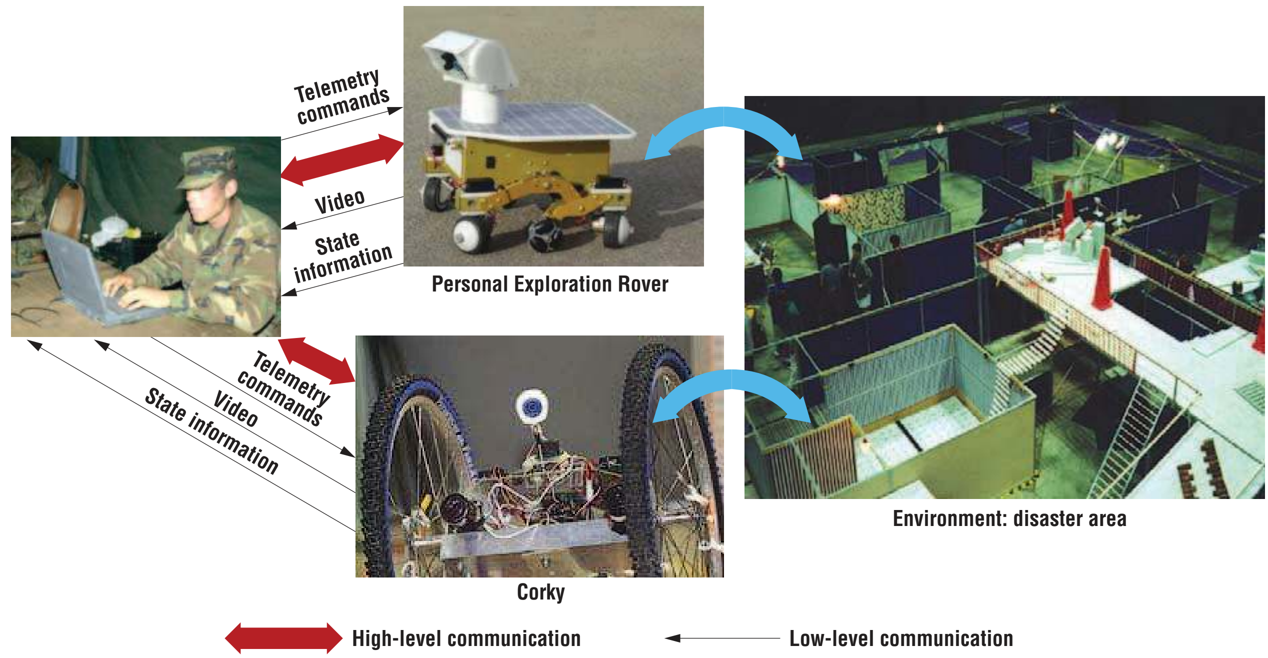


Figure 2. Interactions in our agent-based urban search and rescue system, involving the Personal Exploration Rover and Corky, two robots we developed and deployed.

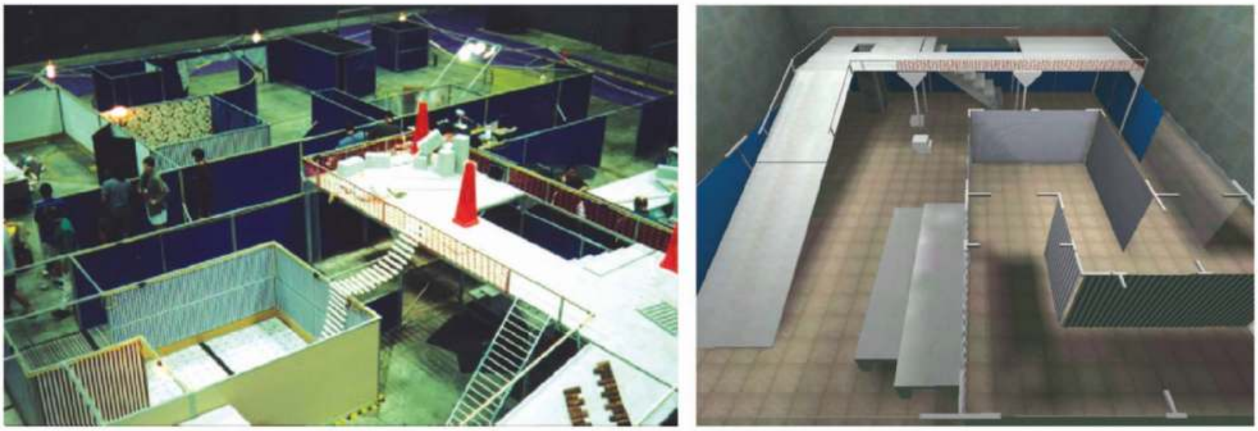
has one standard for communication between all agents: an *agent communication language*. ACLs incur overhead that makes them impractical or infeasible for the transfer of low-level data such as video, audio, sensory, or telemetry data. In response to these opposing needs for high- and low-level communication, we developed a *two-tiered communication hierarchy*, allowing additional, more efficient lines for low-level communication. Integrating the two-tiered communication architecture into the RETSINA MAS significantly improved the system's performance[6] on physical robots for USAR (see Figure 2).

Simulation environment

The scarcity and expense of USAR-capable robots has severely restricted USAR robotics research. Field research shows that mobility is only one problem hindering effective use of robots for search and rescue.[3] Testing human perception, situational awareness, and teamwork depends on combining sensed data, human interaction, and automation in experiments. Expense, unreliability, and difficulties in running participants in parallel, especially in multirobot experiments, make physical robotics impractical for the large samples, repeated trials, and varied conditions needed for HRI research. To support HRI, a robotic simulation must accurately render the user interface (particularly, camera video), represent robot automation and behavior, and represent the remote environment that links the operator's awareness with the robot's behaviors. By anchoring the simulation to actual platforms and sensors, we hope to extend our experiments to novel, large, and hazardous environments with control of greater numbers of robots.

To meet these requirements, we developed USARsim[7,8] a high-fidelity, extensible simulation of the NIST USAR arenas using the Unreal game engine (see Figure 3). The USAR arenas provide a controlled environment for comparing the effectiveness of different robotic designs, control and mapping algorithms, and team regimes. Each arena (yellow, orange, or red) can contain multiple "victims"—mannequins outfitted with thermal signatures, carbon dioxide emitters, and both noise (for example, screams for help) and motion (for example, waving of hands and fingers) as multimodal clues to the victims' vital signs. The quantitative challenge is discover as many victims as possible quickly and convey sufficient information for human rescuers to navigate the disaster and approach the victims. The arenas pose search tasks with varying difficulty on different dimensions. Challenges to mobility progress from the office-like environment of the yellow arena to the nearly impassable rubble of the red arena. Perceptual difficulties vary, from visually confusing patterns, glass panels, mirrors, and sonar-absorbing padding in the yellow arena to the few perceptual difficulties in the red arena.

Our simulated environments include these three arenas and the larger, fixed USAR reference site in an abandoned Nike silo on the NIST Gaithersburg campus. Because the Unreal engine uses stan-



(a) (b)

Figure 3. Overhead views of (a) the US National Institute of Standards and Technologies reference orange arena and (b) an Unreal simulation model.

dard terrain representations reachable through translation chains[9] from most 3D-modeling and GIS (geographic information system) formats, we can add new environments with relative ease. The Unreal client-server architecture for multiplayer games lets multiple, independent robots participate in a single simulation.

We added GameBots, a modification to the game engine that let us control *bots*, synthetic players running simple reactive programs, through a normal TCP/IP socket. This gives us direct access to range data needed to simulate sensors and the ability to directly control parameters, such as wheel velocity, to simulate robot dynamics. Because synthetic characters don't get a rendered view of the scene, we must provide camera views another way. Unreal's client options offers a "spectate" mode. As a spectator, a client can attach its viewpoint (camera location and orientation) to any other player, including a bot. By combining a bot controlled by GameBots with a spectator client, we can simulate a robot with access to the bot's simulated sensor data and the spectating client's simulated video feed.

We develop physical robots and their simulations in parallel, which provides complementary advantages. Simulations let us conduct studies involving many participants and trials, and building robotic platforms lets us validate these findings and identify aspects the simulation missed. Robots and simulation follow the same architecture with the user interacting with the robot through RETSINA agents. We constructed the detailed models of the simluation's robots from the `vehicle` class of the Karma physics engine, a component of the Unreal engine. Because we can't access Unreal's rendering engine directly, we must acquire images from video memory and store them compressed on an image server. This lets us vary frame rates to match observed camera feeds and provide images for visual processing. We can simulate high-quality analog video by presenting the renderer's raw output. By combining accurate models of robot dynamics and controls, camera field of view and frame rate, and the environment, we create HRI tasks. These might allow the robot to become entangled with unseen debris, and visual clutter and unaccustomed perspective could thwart victim recognition. Consequently, operators can easily become confused and lost just as in the actual tasks. We developed two platforms in this manner: Corky, an experimental two-wheeled robot designed for USAR, and the Personal Exploration Rover (PER), an educational robot modified for USAR tasks. Human-in-the-loop simulation showed robot frame rates were too low for effective teleoperation and that more support for the operator's situation awareness was needed. This led to our three-frame panoramic display and the mediated and incremental teleoperation control modes.

We also find simulation to be a valuable tool for investigating general HRI issues. In a series of experiments involving larger, more extreme terrains, we examined an operator's ability to use multiple cameras under a variety of control regimes[10,11] and investigated the advantages of gravity-referenced views over separated displays of attitude.[12] The next series of experiments prompted by field experience at USAR competitions will examine strategies for combining egocentric and exocentric views to improve the human operator's situational awareness and performance in controlling multirobot teams.

Although we developed USARsim to assist our own research into robotic teams, the interest it aroused prompted us to make it available to the wider research community. (Download the simulation at http://usl.sis.pitt.edu/ulab/usarsim_download_page.htm.) A Robocup rescue demonstration league will use

$$p_f = \frac{c_1 f_1(x_1) + c_2 f_2(x_2) + \ldots + c_n f_n(x_n)}{c_1 \max(f_1(x_1)) + c_2 \max(f_2(x_2)) + \ldots + c_n \max(f_n(x_n))} = \frac{\sum_{i=0}^{i=n} c_i p_i}{\sum_{i=0}^{i=n} c_i \max(p_i)}$$

Figure 4. The probability that a particular location contains a victim. x_i is the measurement from sensor i, and p_i is the probability that a victim is present. Each sensor i has its own confidence c_i.

USARsim in 2005, adding ActivMedia Pioneer robots (P2AT and P2DX) and the iRobot ATVR Jr. to the robot platform. The simulated robots can be controlled using the Player, Pyro, or native Gamebot interfaces.

Real-world environment and robot platforms

Urban disaster sites' mobility and sensing challenges (reflected in the red, orange, and yellow NIST arenas) vary widely. A disaster site with small confined voids and narrow passages might require serpentine robots that can crawl through such spaces. Conversely, you might more efficiently search a relatively intact building using a larger, faster robot carrying more sensing and processing power. Clearly, USAR robot teams need heterogeneous platforms to successfully tackle the myriad challenges faced.

Our research focuses on enabling heterogeneous robots to work semiautonomously or in conjunction with humans to explore this challenging environment. Although our research emphasis is in designing algorithms and user interfaces rather than robot platforms, the lack of commercially available, robust, and inexpensive robots prompted the design of two different mobile platforms, which we displayed in the 2004 RoboCup US Open Urban Search and Rescue league competition. In this competition, we entered two- and three-robot teams with heterogeneous sensors and mobility in each of seven rounds, logging over five total robot hours of operation and locating eight victims, to place third overall. Our physical robot team currently consists of several PERs and Corky (see Figure 2). Here, we address two interesting challenges—namely, sensor fusion and interface design for effective robot control.

Sensor fusion

USAR provides an excellent test bed for the positive and negative identification of victims. The range of heat, noise, rubble, and lighting conditions and victim status confound any available sensor. For example, infrared sensors can detect victims entombed in rubble but are easily confused by excess ambient heat or alternative heat sources such as fire. Vision systems provide data intuitive for humans but fail in situations with insufficient lighting or excess dust and debris. We thus combine data from multiple sensors using sensor fusion techniques.[13]

Each sensor has an associated probability density function, $p_i = f_i(x_i)$, where x_i is the measurement from sensor i and p_i is the probability that a victim is present. To account for accuracy discrepancies of various sensors in different situations, we use confidence values. Each sensor i has its own confidence c_i, where higher values of c_i indicate more reliable sensor results. Given a set of n sensors and associated measurements of some location, the equation in Figure 4 represents the probability that this location contains a victim.

This problem formulation is sufficiently broad to represent the variety of sensors available, the reliability of sensors in different situations, and the fusion of information from different robots or information sources. We developed a sensor suite that you can place on an arbitrary mobile platform or distribute among multiple robots and characterized the probability density function, p_i, for these sensors. This sensor suite consists of a USB Web camera, microphone, pyroelectric sensor, and infrared camera used for detecting motion, sound, heat waves in the human emission spectrum, and heat and motion based on infrared images, respectively. We employed a series of experiments to develop the probability density function and confidence values for each sensor in the USAR environment.[13] You can use results of the sensor fusion algorithm to identify and direct a human operator's attention to possible victims (see Figure 5).

Control interface

Search and rescue robots are designed to have adjustable levels of autonomy. This requires not only the development of control algorithms to increase autonomy but also a system to enable an external agent to exert varying degrees of control over the robot. We desig ned an interface agent that provides the operator the necessary feedback and enables the operator to control the robot as desired (see Figure 6).

The interface agent has three separate components. The *communication module* handles all communication with the robot and other agents. The *feedback module* continuously polls the robot for information on the environment (for example, video) and robot (for example, battery level) states and displays this data to the user. In the feedback module, we significantly improved operator performance by adding the ability to request panoramas of the environment to expand the robot's effective field of view. Experience in the simulator demonstrated that operators needed a wider field of view to maintain situational awareness and correctly locate the robot in its environment. The *control module* lets the operator control the robot's attention using the pan-tilt head and the robot position using one of four control paradigms:

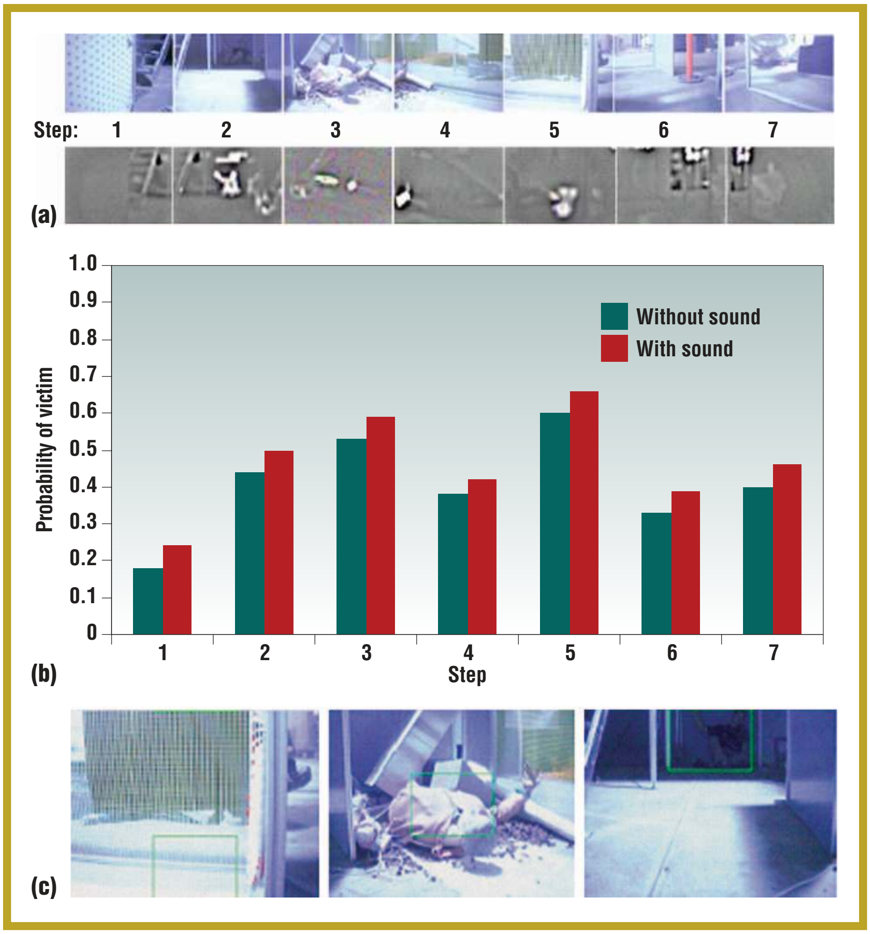


Figure 5. A sensor suite mounted on a mobile platform (a) surveys the environment and (b) applies sensor fusion to determine (c) the three most likely victim locations. In this case, the victims found included a person standing behind a curtain, a mannequin with an electric blanket, and a person hidden in shadow (steps 5, 3, and 2 in the histogram, respectively).

- *Direct teleoperation* gives the operator fine-grain control of the robot by translating joystick commands directly to motor velocities, overriding obstacle avoidance safeguards.
- *Incremental teleoperation* is useful when frame rates and frequencies are too slow or lag is too long for direct teleoperation. The operator commands the robot to turn or drive a short distance, stop, and sense. Although slower than direct control, incremental control imposes no performance requirements on network throughput and latency.
- *Command mode* lets the operator direct the robot to drive or turn a certain distance. The robot performs this action or, if prevented by an obstacle, alerts the operator. A status bar shows the operator the robot's progress. The operator can stop the robot at any time.
- *Mediated mode* lets the operator select a point of interest in the field of view and sends the robot to that position. If the robot can't reach the position, it alerts the operator. Like command mode, driving in mediated control mode is safe in that the robot is actively scanning for obstacles and avoiding them.

Many open problems remain in USAR, including optimally searching a space with distributed, heterogeneous robots with spatiotemporal coordination, applying learning techniques to improve sensor fusion, developing better control interfaces for multiple robots, and improving individual robots' autonomy, mobility, and sensors. Future work by this and other hybrid (simulation and real-world) research teams holds great promise for developing field-worthy solutions that will assist firefighters in USAR conditions.

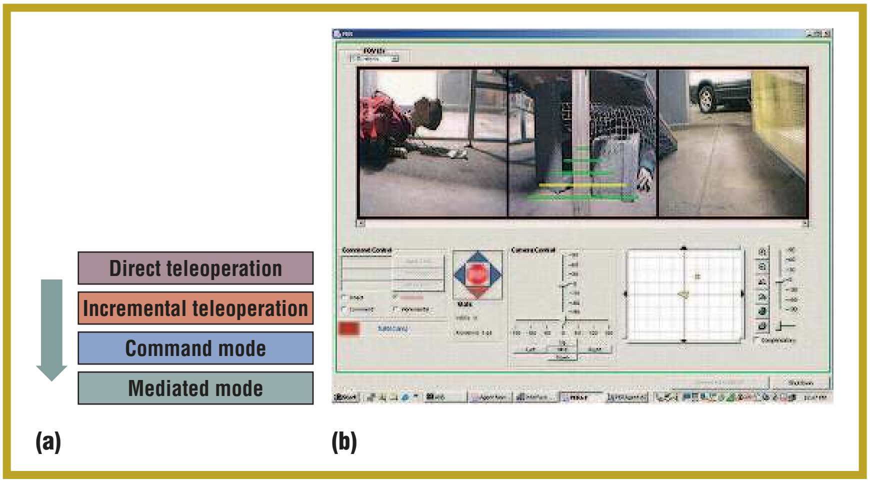
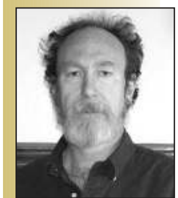


Figure 6. The user interface's (a) four control modules and (b) a sample screen shot using the mediated control mode to control the robot. The operator can, for example, click on one of the two victims in the robot's field of view to direct the robot explore that victim.

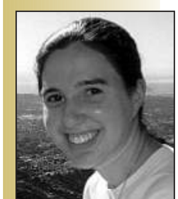
the AUTHORS

Illah R. Nourbakhsh is an associate professor of robotics in the Robotics Institute at Carnegie Mellon University and is cofounder of the Toy Robots Initiative. He was on leave for 2004, serving as the Robotics Group Lead at the NASA Ames Research Center. His research projects include educational and social robotics, electric wheelchair sensing devices, believable robot personality, visual navigation, and robot locomotion. He received his PhD in computer science from Stanford University. Contact him at 364 Banks St., San Francisco, CA 94110; illah@ri.cmu.edu.



Michael Lewis is an associate professor in the Department of Information Science and Telecommunications in the School of Information Sciences at the University of Pittsburgh. His research interests include investigating automation, human-robot interaction, information fusion, and human-agent interaction. His current research focuses on representing information and human control of mixed-initiative systems. Contact him at the Dept. of Information Science and Telecommunications, School of Information Sciences, Univ. of Pittsburgh, 135 N. Bellefield Ave., Pittsburgh, PA 15260; ml@sis.pitt.edu.

Katia Sycara is a professor in the School of Computer Science at Carnegie Mellon University and is director of the Laboratory for Agents Technology and Semantic Web Technologies. Her research interests include multiagent systems, software agents and agent teams, Web services and the Semantic Web, human-agent interaction, case-based reasoning, and the application of these techniques to crisis action planning, scheduling, manufacturing, and financial planning and e-commerce. She received her PhD in computer science from the Georgia Institute of Technology. Contact her at the Robotics Inst., Newell-Simon Hall, Carnegie Mellon Univ., Pittsburgh, PA 15213; sycara@ri.cmu.edu.



Mary Koes is a doctoral graduate student at Carnegie Mellon's Robotics Institute. Her research interests include all aspects of robotics with a special focus in multirobot coordination. She received her BS in mechanical engineering and computer science from Carnegie Mellon University. She is a member of the AAAI and IEEE. Contact her at the Robotics Inst., Newell-Simon Hall, Carnegie Mellon Univ., Pittsburgh, PA 15213; mberna@andrew.cmu.edu.

Mark Yong is currently fulfilling his military service obligations in the Singapore Armed Forces. His research interests lie in the design of locomotive, sensing, and mission-planning systems that will allow robots to operate autonomously in uncertain and unstructured environments. He received his MS in electrical and computer engineering from Carnegie Mellon University. Contact him at markyong@cmu.edu.

Steve Burion is a research assistant at the Robotic Systems Lab at EPFL, the Swiss Federal Institute of Technology (Lausanne, Switzerland). His current research focus is medical robotic devices. He earned an EPFL Masters focusing on search and rescue robotics at the Robotics Institute at Carnegie Mellon University. Contact him at VRAI – Group, EPFL – LSRO, PPH 327, Station 9, CH - 1015 Lausanne, Switzerland; steve.burion@epfl.ch.

REFERENCES

1. A. Jacoff, E. Messina, and J. Evans, "Experiences in Deploying Test Arenas for Autonomous Mobile Robots," *Proc. 2001 Performance Metrics for Intelligent Systems Workshop* (PerMIS 01), National Inst. of Standards and Technologies, 2001.

2. H. Yanco, J. Drury, and J. Scholtz, "Beyond Usability Evaluation: Analysis of Human-Robot Interaction at a Major Robotics Competition," *Human-Computer Interaction*, vol. 19, no. 2, 2004, pp. 117–150.

3. R. Murphy, "Human-Robot Interaction in Rescue Robotics," *IEEE Trans. Systems, Man, and Cybernetics*, vol. 34, no. 2, 2004, 138–153.

4. J. Burke, R. Murphy, and M. Coovert, "Moonlight in Miami: An Ethnographic Study of Human-Robot Interaction in the Context of an Urban Search and Rescue Disaster Response Training Exercise," *J. Human-Computer Interaction*, vol. 19, 2004, pp. 85–116.

5. K. Sycara et al., "The RETSINA MAS, a Case Study," *Software Engineering for Large-Scale Multi-Agent Systems: Research Issues and Practical Applications*, LNCS 2603, Garcia et al., ed., Springer-Verlag, 2003, pp. 232–250.

6. M. Berna-Koes, I. Nourbakhsh, and K. Sycara, "Communication Efficiency in Multiagent Systems," *Proc. Int'l Conf. Robotics and Automation* (ICRA 04), IEEE Press, 2004.

7. J. Wang, M. Lewis, and J. Gennari, "USAR: A Game-Based Simulation for Teleoperation," *Proc. 47th Ann. Meeting Human Factors and Ergonomics Soc.*, 2003.

8. J. Wang, M. Lewis, and J. Gennari, "A Game Engine Based Simulation of the NIST USAR Arenas," *Proc. 2003 Winter Simulation Conf.*, Winter Simulation Board, 2003.

9. J. Manojlovich et al., "UTSAF: A Multi-Agent-Based Framework for Supporting Military-Based Distributed Interactive Simulations in 3D Virtual Environments," *Proc. 2003 Winter Simulation Conf.*, Winter Simulation Board, 2003.

10. S. Hughes and M. Lewis, "Robotic Camera Control for Remote Exploration," *Proc. 2004 Conf. Human Factors in Computing Systems* (CHI 2004), ACM Press, 2004, pp. 511–517.

11. S. Hughes et al., "Camera Control and Decoupled Motion for Teleoperation," *Proc. 2003 IEEE Int'l Conf. Systems, Man, and Cybernetics*, IEEE Press, 2003, pp. 1339–1344.

12. M. Lewis et al., "Experiments with Attitude: Attitude Displays for Teleoperation," *Proc. 2003 IEEE Int'l Conf. Systems, Man, and Cybernetics*, IEEE Press, 2003, pp. 1345–1349.

13. S. Burion, "Human Detection for Robotic Urban Search and Rescue," master's thesis, Robotics Inst., Carnegie Mellon Univ., 2004; www.cs.cmu.edu/afs/cs/project/retsina-31/www/Report/Final%20Report.pdf.