

# **Evolving Interface Design for Robot Search Tasks**

***Holly A. Yanco and Brenden Keyes***

Computer Science Department  
University of Massachusetts Lowell  
One University Ave, Olsen Hall  
Lowell, MA, 01854 USA  
{holly, bkeyes}@cs.uml.edu

***Jill L. Drury***

The MITRE Corporation  
Mail Stop K320  
202 Burlington Road  
Bedford, MA, 01730 USA  
jldrury@mitre.org

***Curtis W. Nielsen, Douglas A. Few, and David J. Bruemmer***

Idaho National Laboratory  
P.O. Box 1625  
Idaho Falls, ID 83415 USA  
{Curtis.Nielsen, Doug.Few, David.Bruemmer}@inl.gov

## **Abstract**

This paper describes two steps in the evolution of human-robot interaction designs developed by the University of Massachusetts Lowell (UML) and the Idaho National Laboratory (INL) to support urban search and rescue tasks. We conducted usability tests to compare the two interfaces, one of which emphasized three-dimensional mapping while the other design emphasized the video feed. We found that participants desired a combination of the interface design approaches. As a result, we changed the UML system to augment its heavy emphasis on video with a map view of the area immediately around the robot. We tested the changes in a follow-on user study and the results from that experiment suggest that performance, as measured by the number of collisions with objects in the environment and time on task, is better with the new interaction techniques. Throughout the paper, we describe how we applied human-computer interaction principles and techniques to benefit the evolution of the human-robot interaction designs. While the design work is situated in the urban search and rescue domain, we feel the results can be generalized to domains that involve other search or monitoring tasks using remotely located robots.

## 1 Introduction

Over the past several years, there has been a growing interest in using robots to perform tasks in urban search and rescue (USAR) settings (e.g., Casper and Murphy (2003), Jacoff et al. (2000), and Jacoff et al. (2001)). Even the popular press has noted the role of USAR robots at disaster scenes. In Hurricane Katrina's aftermath, for example, the St. Petersburg Times noted that the Florida Regional Task Force 3 used robots to search two buildings in Biloxi, Mississippi that were too dangerous to enter (Gussow, 2005).

In recognition of the growing role of USAR robotics, there has been much recent work on Human-Robot Interaction (HRI) in general and interaction designs for rescue robots in particular (e.g., Nourbakhsh et al. (2005), Kadous et al. (2006), Yanco and Drury (2007)). Both HRI in general and rescue robotics in particular have benefited from applying human factors and human-computer interaction (HCI) design principles and evaluation techniques. For example, Adams (2005) used Goal Directed Task Analysis to determine the interaction needs of officers from the Nashville Metro Police Bomb Squad. Scholtz et al. (2004) used Endsley's (1988) Situation Awareness Global Assessment Technique (SAGAT) to determine robotic vehicle supervisors' awareness of when vehicles were in trouble and thus required closer monitoring or intervention. Yanco and Drury (2004) employed usability testing to determine (among other things) how well a search and rescue interface supported use by first responders.

Yet HRI design for USAR is by no means a solved problem. There are still open questions on how to best provide to robot operators the necessary awareness to understand what state the robot is in, what part of the space has been searched, and whether there are signs of life. An additional challenge is that the interface needs to facilitate efficient searching because time is important when victims are in need of medical attention or if they are trapped under unstable debris.

While the literature has examples such as those cited above of using specific HCI techniques to design or evaluate human-robot interfaces, we have not seen literature that highlights the process of evolving multiple generations of HRI interaction designs based on systematically applying HCI principles. Failure to use this process is contrary to best practices in designing for human-computer (and human-machine) interaction. Even a quick glance at the bookshelf of HCI practitioners and researchers yields titles such as *The Usability Engineering Lifecycle* (Mayhew, 1999) and *Developing User Interfaces: Ensuring Usability Through Product and Process* (Hix and Hartson, 1993). In our work, we set out to apply best practices of HCI processes in HRI interface design. This paper reports on two steps in that evolutionary process.

Our research groups at the Idaho National Laboratory (INL) and the University of Massachusetts Lowell (UML) have been working on designs for efficient interfaces that promote awareness of the robot's state and surroundings. We have both built robot systems using the same underlying robot and autonomy levels to act as research platforms. The UML researchers have made significant efforts to specifically address the

needs of the USAR community while the INL group has focused on developing general robot behaviors for a variety of conditions, including USAR. Because both groups are developing interfaces for the same basic set of robot behaviors and functionality, comparing these interface approaches prior to evolving to the next step gave us more data on user performance and preferences than would have been the case from simply studying an interface from either group in isolation.

This paper begins with the description of the INL and UML systems and interface designs, then discusses the usability study of the two interfaces, which used trained search and rescue personnel as test participants. Based upon the results of the comparison study, we describe the evolution of the UML interface design. The new design was tested in a second user study with novice participants. For each interface design, we describe the HCI principles and evaluation techniques applied.

Beyond the USAR domain, the results of investigating alternative interface designs can guide the design of remote robot systems intended for other types of search or monitoring tasks. As with USAR, these tasks rely on the system operator understanding the robot's relationship to the objects in the environment, as well as the robot's location within the environment. As another contribution, our work provides a case study of evolving HRI design in accordance with HCI principles.

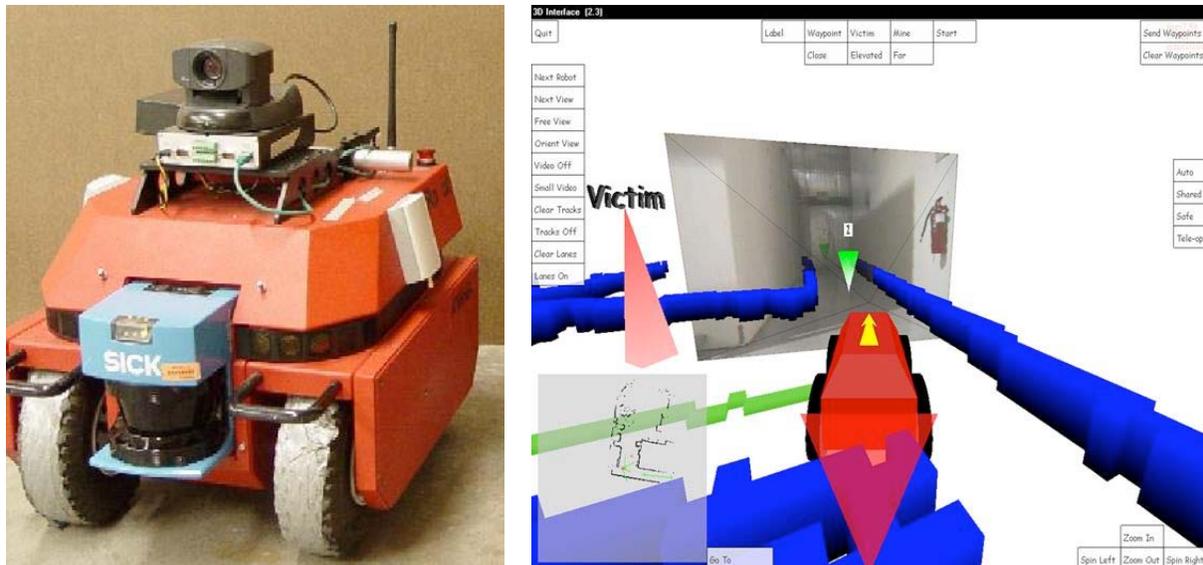
## **2 Robot Systems**

This section describes the robot hardware, autonomy modes and the interfaces for the INL and UML systems.

### **2.1 Idaho National Laboratories (INL)**

The INL control architecture is the product of an iterative development cycle where behaviors have been evaluated in the hands of users (Bruemmer et al., 2005), modified, and tested again. The INL has developed a behavior architecture that can port to a variety of robot geometries and sensor suites. INL's architecture, called the Robot Intelligence Kernel, is being used by several HRI research teams in the community. For the first experiments discussed in this paper, INL used the iRobot ATRV-Mini (shown in Figure 1), which has laser and sonar range finding, wheel encoding, and streaming video.

Using a technique developed at the INL, a guarded motion behavior on the robot uses an event horizon calculation to avoid collisions (Pacis et al., 2004). In response to laser and sonar range sensing of nearby obstacles, the robot scales down its speed using an event horizon calculation, which measures the maximum speed the robot can safely travel in order to come to a stop approximately two inches from the obstacle. By scaling down the speed in many small increments, it is possible to insure that, regardless of the commanded translational or rotational velocity, guarded motion will stop the robot at the same distance from an obstacle. This approach provides predictability and ensures minimal interference with the operator's control of the vehicle. If the robot is being



**Figure 1: The INL robot, an iRobot ATRV-Mini (left), and interface (right).**

driven near an obstacle rather than directly towards it, guarded motion will not stop the robot, but may slow its speed according to the event horizon calculation.

Various modes of operation are available, affording the robot different types of behavior and levels of autonomy. In *teleoperation* mode, the robot takes no initiative to avoid collisions. *Safe* mode is similar to *teleoperation* mode except the robot takes initiative to avoid collisions with the local environment. In *shared* mode, the robot navigates based upon understanding of the environment, yet yields to human joystick input. In *collaborative tasking* mode, the robot autonomously creates an action plan based on the human's high-level planning (e.g. go to a point selected within the map, return to start, go to an entity).

Control of the INL system is actuated through the use of an augmented virtuality interface, defined as displays “in which a virtual environment is enhanced, or augmented, through some addition of real world images or sensations” (Drascic and Milgram, 1996, p. 125). The INL 3D interface combines the map, robot pose, video, and camera orientation into a single perspective view of the environment (Nielsen and Goodrich, 2006; Nielsen et al., 2005). A video window showing real-world information augments the map so that the sum of the two visual elements provides a wider and richer field of view than either could accomplish separately. “When, due to an unnaturally narrow field of view, we are unable to see important parts of the world, such as the floor and our body within the world, we lose a great deal of confidence in our understanding of the world” (Drascic and Milgram, 1996, p. 125). This finding has been confirmed by Keyes et al. (2006), which showed that operators of remotely located robots have fewer collisions when the operator's camera display shows part of the robot in its view of the environment.

The video window moves within the interface display as the robot camera is turned. Assuming the human operator's point of view is aligned with the robot's camera direction, the video window movement is intended to simulate the effect of a person turning his or her head to one side or another.

From the 3D control interface, the operator has the ability to place various icons representing objects or places of interest in the environment (e.g. start, victim, or custom label). Once an icon has been placed in the environment, the operator may enter into a collaborative task by right-clicking the icon which commissions the robot to autonomously navigate to the location of interest. The other autonomy modes of the robot are enacted through the menu on the right side of the interface.

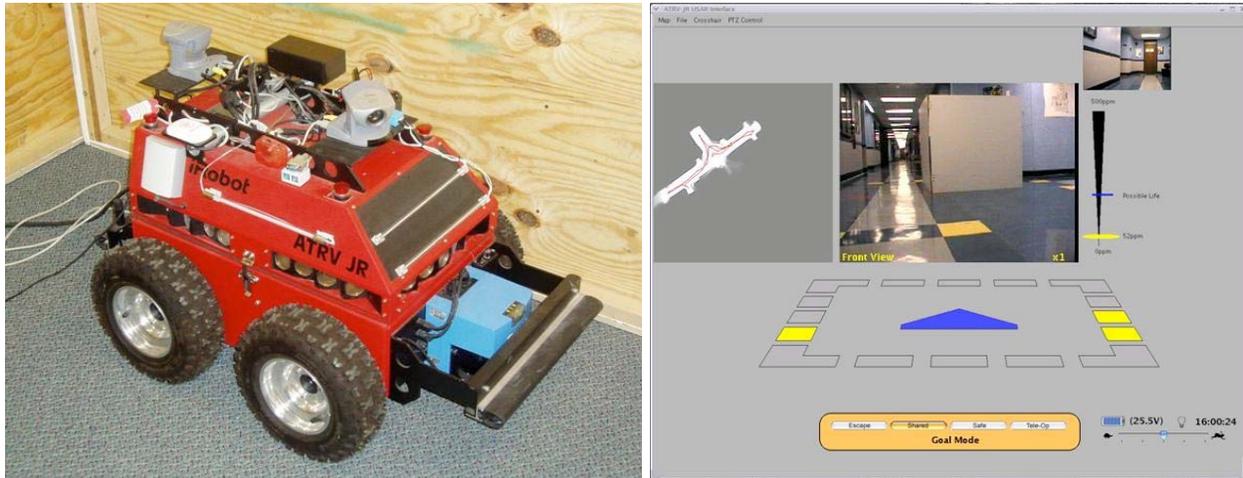
As the robot travels through the remote environment, it builds a map of the area using blue blocks that form columns representing walls or obstacles. The blue blocks are prominent visual features that are given precedence over the video by virtue of appearing in front of the lower part of the video if necessary. The map grows as the robot travels through and discovers the environment. As with all robot-built maps, the fidelity of the map is dependent on the ability of the sensor systems to maintain alignment between the virtual map and the physical map. If the surface traveled by the robot is slippery or the robot slides, physical/virtual alignment may suffer and the user may see such phenomena as the robot traveling through the virtual walls.

The user has the ability to select the perspective through which the virtual environment is viewed by choosing the elevated, far, or close button. The elevated view provides a 2D overview of the entire environment. The far and close buttons result in 3D perspectives showing more (far) or less (close) of the environment and are intended to provide depth cues to help in maneuvering the robot. The default view button returns the perspective to the original robot-centered perspective (also 3D).

Through continuous evaluation of sensor data, the robot attempts to keep track of its position with respect to its map. As shown in Figure 1 on the right, the robot is represented as the red vehicle in the 3D interface. The virtual robot is sized proportionally to demonstrate how it fits into its environment. Red triangles appear if the robot is blocked and unable to move in a particular direction.

## **2.2 UMass Lowell (UML)**

UMass Lowell's platform is an iRobot ATRV-JR robot. This robot came equipped with a SICK laser rangefinder and a ring of 26 sonar sensors. UML added front and rear pan-tilt-zoom cameras, a forward-looking infrared (FLIR) camera, a carbon dioxide (CO<sub>2</sub>) sensor, and a lighting system (see Figure 2). The robot uses autonomy modes similar to the INL system; in fact, the basis for the current autonomy levels is INL's system. Teleoperation, safe, goal (a modified shared mode) and escape modes are available. In escape mode, the robot autonomously maneuvers away from all obstacles and then stops to await further direction from human operators.



**Figure 2: The UML robot, an iRobot ATRV-JR, (left) and interface (right).**

In this version of the interface<sup>1</sup>, there are two video panels, one for each of the two cameras on the robot (see Figure 2). The main video panel is the larger of the two and was designed to be the user’s focus while driving the robot and searching for victims. Because it is important for robot operators to watch the video panel, in addition to the fact that eyes are naturally drawn to movement, having a powerful attention attractant centrally located in the operator’s visual field is likely to be successful. (See Kubey and Csikszentmihalyi (2002) for a discussion of how the movement in television draws attention.) Having two video panels, with one being more prominently displayed than the other, is a variation of the recommendation for future work made by Hughes and Lewis (2004), who said that “further study of the two camera display may find that one of the screens is more dominant, suggesting that a screen-in-screen technique may be appropriate” (p. 517).

The second video panel is smaller, is placed at the top-right of the main video window, and is mirrored to simulate a rear view mirror in a car. The robot operator looks up and to the right to view the second video display just as a driver looks in a car’s rear view mirror. By default, the front camera is in the main video panel, while the rear camera is displayed in the smaller rear view mirror video panel.

The robot operator has the ability to switch camera views, called the Automatic Direction Reversal (ADR) mode. In ADR mode, the rear camera is displayed on the main video panel and the front camera is displayed on the smaller panel. All of the driving commands and the range panel readouts (described below) are reversed. Pressing forward on the joystick in this case will cause the robot to back up, but to the user, the robot will be moving “forward” (i.e., the direction that the camera currently displayed in the main window is looking). This essentially eliminates the front/back of the robot because the user is now very rarely “backing up.”

<sup>1</sup> See Baker et al. (2004) for a description of the earlier interface design or Keyes (2007) for a discussion of the system’s evolution over several years.

UML chose to use the automobile interface metaphor to guide video window placement for two reasons. First, and most obviously, the intended user group (rescue personnel) can be expected to have significant experience driving cars. The ADR mode can take advantage of operators' prior knowledge of using a car's rear-view mirror to enable them to use this mode without training. More important, however, is the fact that this metaphor is congruent with the types of attention focus needed by rescue robot operators. The direction of travel requires the operator's primary focus of attention, with a peripheral focus being 180 degrees from the driving direction (for example, to have awareness of an escape path). These two attention foci align well with the automobile rear-view mirror analogy. Steinfeld (2004) notes that robots that provide a forward video scene are "ripe for mimicking traditional automotive representations" (p. 2753).

The main video panel has textual labels identifying which camera is currently being displayed in it and the current zoom level of the camera (1x - 16x).

The interface, by default, has an option for showing crosshairs, indicating the current pan and tilt of the camera. Crosshairs have been used previously in HRI designs such as that of Zalud (2006). In the case of the UML design, the crosshairs overlay the video as a type of head up display so that critical information about camera orientation can be provided to users without deflecting their attention from the information being provided by the camera. The horizontal part of the cross acts as a kind of horizon line to indicate how far the camera has been tilted up or down and the vertical component of the cross provides a visual reminder of how far from center, and in which direction, the camera has been panned.

The map panel (shown in figure 3) provides a map of the robot's environment, as well as the robot's current position and orientation within that environment. As the robot moves throughout the space, it generates a map using the distance information received by its sensors using a Simultaneous Localization and Mapping (SLAM) algorithm. This type of mapping algorithm uses a statistical model to compare current distance readings and odometry information with other readings it has already seen to try to match up its current location while building a representation environment. SLAM is the most widely used and accepted algorithm for real-time dynamic mapping using robotic platforms (see Thrun (2002) for an overview of robot mapping).

PMap, the mapping package we used (Howard, 2004), also kept track of the robot's path through the environment. This path, displayed as a red line on the map, made it easy for users to know where they had started, where they traveled, as well as the path they took to get to where they currently are. Many users in the first experiment talked about in the next section referred to this trail as being "breadcrumbs" that helped them find their way out of the maze.

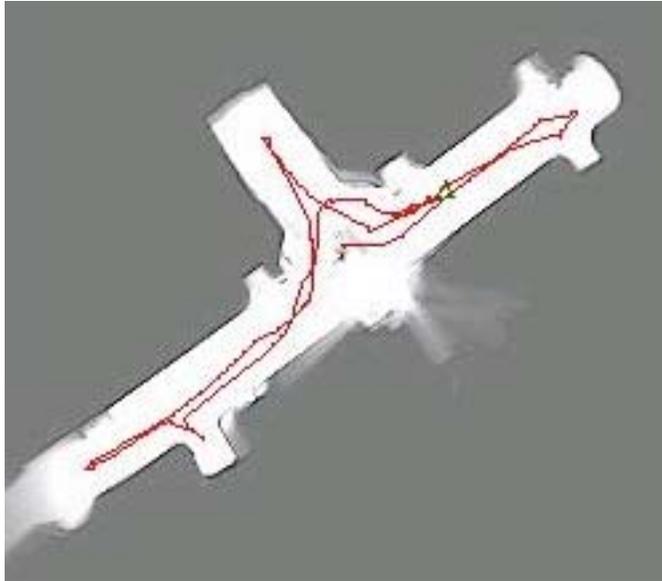


Figure 3. The UML system's map panel, showing the robot's position, the entire environment and the trail of "breadcrumbs."

The "zoom mode" feature also exists on the map panel. "Zoom mode," which can be seen in figure 4, is essentially a view of the map at a zoomed in level. It takes the raw laser data in front of the robot and draws a line, connecting the sensor readings together. There is also a smaller rectangle on the bottom of the display that represents the robot. As long as the white sensor lines do not touch or cross the robot rectangle, then the robot is not in contact with anything. By using data from a laser sensor with approximately 1 mm accuracy, this panel gives a highly accurate and easy to interpret way to tell if the robot is in close proximity to an object. The disadvantage of placing this view in this location is that it is in the same display location as the map view, forcing users to choose whether they wish to view the map or the zoom mode.

Information from the sonar sensors and the laser rangefinder is displayed in the range data panel located directly under the main video panel. When nothing is near the robot, the color of the boxes is the same gray as the background of the interface to indicate nothing is there. As the robot approaches an obstacle at a 1 meter distance, the box(es) will turn to yellow, and then to red when the robot is very close (less than .5 m). The ring of boxes is a rectangle, drawn in a perspective view, which makes it look like a trapezoid. This perspective view was designed to give the user the sensation that they are sitting directly behind the robot, akin to sitting behind the wheel of a car looking out at the car's hood and the obstacles on the side of the road. If the user pans the camera left or right, this ring will rotate opposite the direction of the pan. If, for instance, the front left corner turns red, the user can pan the camera left to see the obstacle, the ring will then rotate right, so that the red box will line up with the video showing the obstacle sensed by the range sensors.

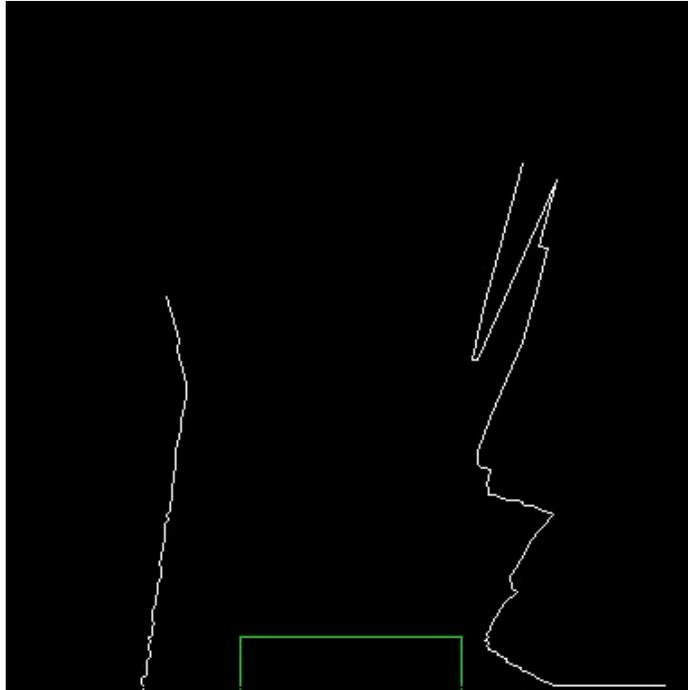


Figure 4. “Zoom mode” showing the front of the robot (the rectangle in the bottom of the figure) and lines indicating obstacles in the area immediately around the robot.

The blue arrow-like triangle, in the middle of the range data panel, indicates the true front of the robot. The system aims to make the robot’s front and back be mirror images, so ADR mode will work the same with both; however, the SICK laser,  $CO_2$  sensor, and FLIR camera only point towards the front of the robot, so this blue arrow helps the user to distinguish front and back if needed. It was difficult to design a simple yet evocative symbol for the front of the robot. The arrow symbol “affords” pointing in the direction of travel (where a “perceived affordance” is an action the user perceives is possible to take (Norman, 1988)). The travel direction is usually (but not always) along the vector through the front of the robot.

The mode indicator panel displays the current mode that the robot is in. The  $CO_2$  indicator, located to the right of the main video, displays the current ambient  $CO_2$  levels in the area. As the levels rise, the yellow marker moves up. If the marker is above the blue line, then there is possible life in the area. By using a thermometer-like display with a clearly marked threshold, operators do not need to perform a mental translation to determine whether the carbon dioxide level is indicative of life: they can see at a glance whether the meter value is above or below the threshold.

The bottom right of the interface shows the status panel. This consists of the battery level, current time, whether the lights are on or off, and the maximum speed level of the robot. The light bulb symbol is either filled with grey or yellow to provide a literal analog of a light being on or off. Similarly, the battery indicator shows a battery symbol filled with dark gray proportionate to the battery power remaining as well as a more precise textual statement of the number of volts remaining. The speed indicator shows

the maximum allowable speed of the robot. The joystick is an analog sensor; therefore the more a user pushes on it, the faster the robot will go, as a linear function of the maximum allowable speed. The tortoise and hare icons at the lowest and highest ends of the scale, respectively: a reference to the widely-known Aesop fable and the fact that most people can recognize these as slow and speedy animals.

The robot is primarily controlled via a joystick. To move the robot, the operator must hold the trigger and then give it a direction. If the user presses the joystick forward, the robot will move forward, left for left, etc. On top of the joystick is a hat sensor that can read eight compass directions. This sensor is used to pan and tilt the camera. By default, pressing up on this sensor will cause the camera to tilt up, likewise pressing left will pan the camera left. An option in the interface makes it so that pressing up will cause the camera to tilt down; pilots find this option to be more natural because they have been trained to push aircraft controls forward to pitch the nose of the aircraft down. The joystick also contains buttons to home the cameras, perform zoom functions, and toggle the brake. The joystick also has a button to toggle Automatic Direction Reversal mode. Finally, a scrollable wheel to set the maximum speed of the robot is also located on the joystick. All of these functions were placed on the joystick for the sake of efficiency so that the operator could spend the majority of time using a single input device rather than having to repeatedly move a hand from the joystick to the keyboard or mouse and back. However, due to the lack of available buttons on the current joystick, the ability to change modes, as well as turning the lighting system on and off, was relegated to keyboard keys.

### **2.3 Comparing the designs against HCI principles**

There are a number of sets of principles and guidelines that have been employed to evaluate how people interact with computer-based systems. Perhaps the most widely used set that is not specific to a single implementation (such as a Macintosh design guide) was developed by Molich and Nielsen (1990; later updated in Nielsen, 1994). Expressed as ten overarching heuristics or “rules of thumb,” this high-level guidance can be tailored for particular systems or used “as is” for a quick evaluation. Drury et al. (2003) adapted Nielsen’s heuristics slightly for use in HRI, and we compared these HRI heuristics to the INL and UML systems to make the observations in table 1. Heuristic evaluations are normally performed by HCI experts who are attempting to keep the needs of “real” users in mind, and so can be thought of as predictions for how the intended users will fare when they use the system. Since heuristic evaluations can surface a large number of minor issues that do not necessarily cause problems for users (Jeffries et al., 1991), it is worthwhile to also perform evaluations involving users or surrogate users before making drastic changes in design. We provide the summary in table 1 as a preview of the difficulties that we expected users might encounter in subsequent experimentation.

**Table 1. Summary of Applying Heuristics**

<b>Nielsen’s Heuristics Adapatad for HRI (Drury et al., 2003)</b>	<b>Heuristics Applied to INL System</b>	<b>Heuristics Applied to UML System</b>
Is the robot’s information presented in a way that makes sense to human controllers?	Yes, except when the robot slips and the virtual walls are no longer aligned with the physical walls. Also, problems with simply seeing the video information can impede task.	Yes, except that it may take some time to get used to the trapezoidal display of proximity information.
Can the human(s) control the robot(s) without having to remember information presented in various parts of the interface?	Mostly, except users must remember joystick controls and facts such as which icons can be right-clicked for further actions.	Users need to remember how to manipulate the joystick in a number of ways to control motion, camera, and brake.
Is the interface consistent? Is the resulting robot behavior consistent with what humans have been led to believe based on the interface?	Mostly. The movement of the video can seem inconsistent to users who would expect the window to remain stationary.	Yes, except that the “animal” or “Aesop’s fables” metaphor for speed control is quite different from the driving metaphor used elsewhere in the interface.
Does the interface provide feedback?	Yes, regarding where the robot is in the environment, the camera orientation, and what the mode the robot is in.	Yes, regarding where the robot is in the environment, the camera orientation, and what the mode the robot is in.
Does the interface have a clear and simple design?	Mostly, but the map can clutter and occlude the video window.	Yes. There is adequate white (actually gray) space and little clutter.
Does the interface help prevent, and recover from, errors made by the human or the robot?	The map can help prevent driving forward into obstacles, but the lack of a rear-facing camera (and display of its information) can lead to hits in the rear.	The dual camera display can prevent driving problems when backing up but the red and yellow blocks do not provide precise enough information regarding obstacles’ proximity.
Does the interface follow real-world conventions, e.g., for how error messages are presented in other applications?	The use of colors is standard, but the few error messages that we saw are not presented in a standard manner.	The use of colors is standard, but the few error messages that we saw are not presented in a standard manner.
Is the interface forgiving; does it allow for reversible actions on the part of the human or the robot?	Yes, except that once the operator drives the robot impacts something in such a manner that a collision occurs, that action is not really “reversible.”	“Escape” mode allows for getting out of tight spots, but once the robot impacts something that action is not really “reversible.”
Does the interface make it obvious what actions are available at any given point?	The interface has many buttons which remain visible, including camera controls that are redundant to the joystick.	The interface does not make visible those actions that can be controlled via the joystick., and keyboard controls are not visible without physical labels.
Does the interface provide shortcuts and accelerators?	The “go to” function acts as a shortcut for directing the robot to a particular point. The “home cameras” function is a shortcut for moving the camera to the front and level. The “elevated” view is a fast way of getting an overview of the environment without a lot of maneuvering and camera movement.	“Home cameras” button is a shortcut for moving the camera to front and level. “Escape” is a shortcut for getting the robot out of a tight spot.

### 3 Experiment 1

The purpose of this first experiment was to learn which interface design elements are most useful to USAR personnel while performing a search task. To do this we conducted usability studies of the two robot systems at the USAR test arena at the National Institute of Standards and Technology (NIST) in Gaithersburg, MD. Some of the results from this experiment were reported in the 2006 SSRR conference (Yanco et al., 2006).

#### 3.1 Experiment design

Because we wished to see differences in preferences and performance between the UML interface and the INL interface, we designed a within-subjects experiment with the independent variable being the human-robot system used. Eight people (7 men, 1 woman) ranging in age from 25 to 60 with search and rescue experience agreed to participate.

We asked participants to fill out a pre-experiment questionnaire to understand their relevant experience prior to training them on how to control one of the robots. We allowed participants time to practice using the robot in a location outside the test arena and not within their line of sight so they could become comfortable with remotely moving the robot and the camera(s) as well as the different autonomy modes. Subsequently, we moved the robot to the arena and asked them to maneuver through the area to find victims. We allowed 25 minutes to find as many victims as possible, followed by a 5-minute task aimed primarily at ascertaining whether the robot operator understood where the robot was located with respect to prominent features in the environment (or victims). After that, we took a short break during which an experimenter asked several Likert scale questions. Finally, we repeated these steps using the other robot system, ending with a final short questionnaire and debriefing. The entire procedure took approximately 2 1/2 hours per person.

The specific task given to the participants during their 25-minute runs was to “fully explore the approximately 2000 square foot space and find any victims that may be there, keeping in mind that, if this was a real USAR situation, you’d need to be able to direct people to where the victims were located.” Additionally, we asked participants to “think aloud” (Ericsson and Simon, 1980) during the task. After the exploration portion of the experiment, participants were asked to maneuver the robot towards a previously seen point with a five minute time limit. Participants were not informed ahead of time that they would need to remember how to get back to any particular point.

We counterbalanced the experiment in two ways to minimize confounding factors. Five of the eight participants started with the UML system and the other three participants began with the INL system. (Due to battery charging considerations, a robot that went first at the start of the day had to alternate with the other system for the remainder of that day. UML started first in testing on days one (2 participants) and three (3 participants). INL started first on day two (3 participants).) Additionally, two different starting positions were identified in the arena so that knowledge of the arena gained from using

the first system would not transfer to the use of the second system. The two counterbalancing techniques led to four different combinations of initial arena entrance and initial interface.

The tests were conducted in the Reference Test Arenas for Autonomous Mobile Robots developed by the National Institute of Standards and Technology (NIST) (Jacoff et al., 2000; Jacoff et al., 2001). During these tests, the arena consisted of a maze of wooden partitions and stacked cardboard boxes. The first half of the arena had wider corridors than the second half.

### **3.2 Analysis Method**

Analysis consisted of two main thrusts: understanding how well participants performed with each of the two systems, and interpreting their comments on post-run questionnaires. Performance measures are valuable as implicit measures of the quality of the human-robot interaction. Under ordinary circumstances, users who are given better interfaces could be expected to perform better at their tasks than those who are given poor interfaces. Accordingly, we measured the percentage of the arena explored, the number of times the participants bumped the robot against obstacles, and the number of victims found. We investigated two hypotheses.

*H1: Using INL's interface will result in exploring more area than using UML's interface.*

We predicted that INL's 3D map display would ease navigation and thus support operators exploring a greater percentage of the arena per unit time.

*H2: Using UML's interface will result in finding more victims than using INL's interface.*

The prediction leading to H2 is based on the greater emphasis on video and sensor displays such as FLIR and CO<sub>2</sub> in the UML interface.

After each run, participants were asked to name the features of the robot system that they found "most useful" and "least useful." We inferred that the "useful" features were considered by participants to be positive aspects of the interface and the "least useful" features were, at least in some sense, negative. After reviewing all of the comments from the post-run questionnaires, we determined that they fell into five categories: video, mapping, other sensors, input devices, and autonomy modes. Results are discussed next.

### **3.3 Results**

In this section, we present both quantitative and qualitative comparisons of the INL and UML interfaces.

### 3.3.1 Area Coverage

We hypothesized that the three-dimensional visualization of the mapping system on INL's interface would provide users with an easier exploration phase. Table 2 presents the results of arena coverage for each participant with each of the robot systems. There is a significant difference ( $p < .022$ , using a two-tailed paired t-test with  $dof=7$ ) between the amount of area covered by the INL robot and the amount covered by the UML robot, seeming to confirm the first hypothesis.

**Table 2: Comparison of the Percentage of the Arena Covered**

Participant	% Area Covered	
	INL	UML
1	8.7	12.6
2	37.9	25.2
3	34.8	34.8
4	37.9	19.7
5	30.3	27.3
6	33.3	22.7
7	53.0	31.8
8	30.3	19.7
Average	33.3 (7.8)	24.2 (5.8)

One possible confounding variable for this difference was the size of the two robots. The ATRV-Mini (INL's robot) is smaller than the ATRV-JR (UML's robot) and thus was able to fit in smaller areas. However, the first half of the arena, which was the primary area of coverage, had the widest areas, allowing both robots to fit comfortably.

### 3.3.2 Number of Bumps

One implicit measure of the operator's awareness of the robot's proximity to obstacles is the number of times that the robot bumps into something in the environment. However, in this experiment there were several confounding issues in this measure. First, the INL robot experienced a sensor failure in its right rear sensors during the testing. Second, the INL robot has a similar length and width, meaning that it can turn in place without hitting obstacles, whereas the UML robot is longer than it is wide, creating the possibility of hitting obstacles on the sides of the robot when the robot is rotated in place. Finally, when using the INL system participants were instructed not to use the teleoperation mode (which has no robot initiative to protect itself), while, on the other hand, participants were allowed to use teleoperation mode with the UML system.

Despite these confounding factors, we found no significant difference in the number of hits that occurred on the front of the robot ( $\bar{\chi}_{INL} = 4.0$ ,  $\bar{\chi}_{UML} = 4.9$ ,  $p = 0.77$ ). Both robots are equipped with similar cameras on the front and both interfaces present some sort of ranging data to the user. As such, the user's awareness level of obstacles in front of the robot seems to be similar between systems.

When hits occurring in the back right of the robot were eliminated from total hits on the backs of the robots, we did find a significant difference in the number of hits occurring in the rear of the robot ( $\bar{\chi}_{INL} = 2.5$ ,  $\bar{\chi}_{UML} = 1.2$ ,  $p < 0.05$ ). The UML system has a camera on the rear of the robot, adding additional sensing capability that the INL robot does not have. While both robot systems present ranging information from the back of the robot on the interface, the addition of a rear camera appears to improve awareness of obstacles behind the robot. This result correlates well to the results found in Keyes et al. (2006), which studied the numbers of cameras on a robot and their placement for a remote robot task.

The systems also had a significant difference in the number of hits on the side of the robot ( $\bar{\chi}_{INL} = 0$ ,  $\bar{\chi}_{UML} = 0.5$ ,  $p < 0.05$ ). As the two robots had equivalent ranging data on their sides, the difference in hits could be due to the differences in the robots' sizes and geometry and/or the differences in how data was presented in the interface. The UML robot was bigger and thus had more difficulty getting through narrow corridors, but also the colored boxes used in the UML interface provided imprecise guidance regarding the distance to objects in the environment.

### **3.3.3 Victims Found**

We had hypothesized (H2) that the emphasis on the video window and other sensor displays such as the FLIR and  $CO_2$  sensor of the UML interface would allow for users to find more victims in the arena. However, H2 was not supported by the data because there was an insignificant difference ( $p=0.35$ ) in the number of victims found. Using the INL system, participants found an average of 0.63 victims whereas with the UML system, participants found an average of 1.0 victim. In general, victim placement in the arena was sparse, and the victims that were in the arena were well hidden. Using the number of victims found as an awareness measure might have been improved by having a larger number of victims, with some easier to find than others.

### **3.3.4 User Surveys**

At the end of each run, users were asked to rank the ease of use of each interface, with 1 being extremely difficult to use and 5 being very easy to use. In this subjective evaluation, operators rated the UML interface easier to use ( $\bar{\chi}_{INL} = 2.6$ ,  $\bar{\chi}_{UML} = 3.6$ ,  $p < 0.05$ ). Users were also asked to rank how the controls helped or hindered them in performing their task, with 1 being "hindered me" and 5 being "helped me tremendously." Operators felt that the UML controls helped them more ( $\bar{\chi}_{INL} = 3.2$ ,  $\bar{\chi}_{UML} = 4.0$ ,  $p = 0.0547$ ).

### **3.3.5 Interface Features**

Users were also asked what features on the robots helped them and which features did not. We performed an analysis of these positive and negative statements, clustering them into the following groups: video, mapping, sensors, input devices and autonomy. The

statements revealed insights into the features of the systems that the users felt were most important.

In the mapping category, there were a total of 10 positive mapping comments and one negative for the INL system and 2 negative mapping comments overall for the UML system. We believe that the number of comments shows that the participants recognized the emphasis on mapping within the INL interface and shows that the three-dimensional maps were preferred to the two-dimensional map of the UML interface. Furthermore, the preference of the INL mapping display and the improved average percentage of the environment covered by the INL robot suggests that the user preferences were in line with features that improved performance. Interestingly, two of the positive comments for INL identified the ability to have both a three-dimensional and two-dimensional map. Operators also liked the waypoint marking capabilities of the INL interface as a means to move the robot through the map (as predicted by the heuristic “provide shortcuts and accelerators”).

There were a similar number of comments made on video about the two systems (13 for UML and 16 for INL). This seems to suggest that video is very important in this task, and most participants were interested in having the best video possible. There were more positive comments for UML (10 positive and 3 negative) and more negative comments for INL (3 positive and 13 negative). In the INL interface, when the camera is panned or tilted the robot stays in a fixed position within the map while the video information moves around the robot. This video movement caused occlusion and distortion of the video when the camera was panned and tilted, making it difficult to use the window to identify victims or places in the environment. We noted participants turning their heads and craning their necks in what looked like uncomfortable positions when attempting to see the video; so the negative comments are consistent with ergonomic principles that dictate the need for an interface that is not physically uncomfortable to use.

Interestingly, most of the positive video comments for UML did not address a fixed position window (only 1 comment). Four users commented that they liked the ability to home the camera (INL had two positive comments about this feature as well). Three users commented that they liked having two cameras.

All comments on input devices were negative for both robots, suggesting that people just expect that things will work well for input devices and will complain only if they aren't working. The heuristic evaluation predicted that there would be problems remembering how the joystick controls worked. On the UML system, 2 of the 5 negative comments about input devices were about the joystick; on the INL system, 2 out of 6 were about the joystick. Users commented negatively on the latency between the controls and the robots on both systems (2 out of 5 comments for UML; 3 out of 6 for INL). The remaining negative comment for each system was about selections to be made on the interface.

There were a similar number of positive comments for autonomy, suggesting that users may have noticed when the robot had behaviors that helped. It is possible that the users

didn't know what to expect with a robot and thus were just happy with the exhibited behaviors and accepted things that they may not have liked.

We saw many more comments on UML's sensors (non-video), which is congruent with the emphasis on adding sensors on the UML system. INL had two negative comments for not having lighting available on their robot. As a result, the video was darker than of the UML system; participants literally could not see the environment very well in many cases. UML had 10 positive comments (1 each for lights, FLIR and  $CO_2$ , 4 for the laser ranging display and 3 for the sonar ring display) and 3 negative comments (2 for the sonar ring display blocks not being definitive and 1 for the FLIR camera). Our analysis suggests that there are a few categories of great importance to operators: video, labeling of maps, ability to change perspective between 3D and 2D maps, additional sensors, and autonomy. In fact, in their suggested ideal interface, operators focus on these categories.

### 3.3.6 Designing the User-Preferred Interface

After using both interfaces, users were asked which features they would include if they could combine features of the two interfaces to make one that works better for them. Every user had his or her opinion, as follows:

- Participant 1 wanted to combine map features (breadcrumbs on the UML interface and labeling available on the INL interface).
- Participant 2 wanted to keep both types of map view (3D INL view, 2D UML view), have lights and add other camera views (although this user also remarked that he didn't use UML's rear view camera much).
- Participant 3 wanted to add the ability to mark waypoints to the UML system.
- Participant 4 liked the blue blocks on INL (3D map walls), the crosshairs on UML (pan and tilt indicators on the video), the stationary camera window on the UML interface, marking entities and going to waypoints on the INL interface, the breadcrumbs in the UML map, and the bigger camera view that the UML interface had.
- Participant 5 liked the video on the UML interface and preferred the other features on the UML interface, as well. He would not combine any features from the INL interface into the UML interface.
- Participant 6 wanted a fixed camera window (like UML), a 2D map in the left hand corner of the 3D interface, the ability to mark waypoints on the map, roll and pitch indicators, and lights on the robot.
- Participant 7 wanted to take UML as a baseline interface, but wanted a miniaturized blue block map (3D map) instead of the 2D map, since it provided more scale information.

- Participant 8 wanted to start with the UML interface, with the waypoint marking feature and shared mode capability of the INL system.

### 3.4 Discussion

When asked to design their ideal interface, most participants commented on the maps, preferring the 3D map view to the 2D view; we feel this preference may be due to the fact that the 3D map view provides more information about the robot's orientation with respect to the world. Features of the two maps could be combined, either with a map view that could swing between 3D and 2D or by putting both types of maps on the screen. However, operators did comment that they did not like the way that the current implementation of the blue blocks on the INL system obscured the video window when it was tilted down or panned over a wall (we predicted in the heuristic evaluation that INL users would not like the clutter that resulted from the blue blocks overprinting the video). This problem could be remedied by giving precedence to video information when both map and video occupy the same place in the interface.

Most participants also expressed a desire to have an awareness of where they had been, with the ability to make annotations to the map. They wanted to have the breadcrumbs present on the UML interface, which showed the path that the robot had taken through the arena. This feature was available on the INL interface but not turned on for the experiments. Participants also wanted to be able to mark waypoints on the map, which was a feature in the INL system, but not the UML system.

In general, the participants did not like the moving video window present on the INL interface, preferring a fixed camera window instead. We believe that in a USAR task, a fixed window of constant size allows for the operator to more effectively judge the current situation. While this hypothesis seems to be borne out by the comments discussed above, it was not verified by measures such as number of victims found and number of hits in the front of the robot, neither of which were statistically different between the two systems.

Interestingly, when designing their interface, no participants commented on the additional sensors for finding victims that were present on the UML system: the FLIR camera and the  $CO_2$  sensor. It seemed that their focus fell on being able to understand where they were in the environment, where they had been, and what they could see in the video.

## 4 Redesign of the UML Robot System

As can be seen in table 1, the application of the adapted heuristics noted that the UML interface did not provide precise enough distance information using the red and yellow blocks, a fact that was borne out in the usability tests, both qualitatively and quantitatively. We found that there were more hits on the sides of the UML robot than the INL robot, indicating that the operators did not have sufficient awareness, with the information provided, of the robot's relationship with respect to obstacles. Users themselves identified the lack of definite distance information as one of their dislikes.

Due to the testing results, the UML designers scrapped the range panel used in experiment 1. In its place, UML inserted the zoom mode feature as the main distance panel. However, instead of only showing the front of the robot, the zoom mode encompassed the entire circumference of the robot. The display for the front part of the robot uses the laser data, whereas the left, back, and right sides use the sonar data. This panel also includes tick marks to indicate the distance that the lines portrayed, spaced in 0.25 meter increments. This panel was again placed directly under the main video display (see figure 5). As with the previous distance panel, this new panel also rotated in concert with the user panning the camera.

Because it was slightly bigger than the previous range panel, in order to keep the main video display centered in the display for the reasons discussed earlier, the mode panel was moved to the top of the display, slightly displacing the secondary video panel. Moving the mode panel is also in accordance with the HCI principle that states the interface should support a top to bottom, left-to-right flow consistent with the information needed while performing a sequence of tasks<sup>2</sup>; and one of the first things that users must do when working with a robot is decide which autonomy mode to select. Although the secondary video display was displaced slightly downward, UML designers judged that its new position was still evocative enough of the “rear view mirror” to take advantage of the users’ understanding of the automobile metaphor. For the comparison experiments described below, the video panels were in identical positions (see figure 7).

Unlike the previous zoom mode, this new panel also had the ability to not only give a top-down view (figure 6, left), but also a perspective view (figure 6, right). We saw in the previous study that users liked having the ability to go from a 2D map to a 3D map in the INL system, and since the zoom mode is technically a local space map, UML conjectured that users might wish to toggle between 2D and 3D in this map as well. While not tested in our experiment, it is possible that the 2D view would nevertheless provide better support for navigation in tight spaces than the 3D view despite users’ subjective preferences. This supposition is based on the work of Alexander and Wickens (2005), which directly compared 2D and 3D approaches for navigation display in aviation. They found better performance in locating aircraft traffic with the 2D version due to spatial awareness biases negatively affecting accuracy when using the 3D version.

## **5 Experiment 2**

The purpose of this next experiment was to determine if the changes made to the UML robot system would improve the performance of the human-robot team when maneuvering through terrain filled with obstacles. Similar user studies have been performed previously by Ricks, Nielsen, and Goodrich at Brigham Young University wherein they compared a traditional interface design with a 3D interface design in a

---

<sup>2</sup> There are many references for this HCI principle. For a readable discussion of this and other principles, see Galitz (2007).

variety of navigation tasks (Ricks et al., 2004, Nielsen and Goodrich, 2006a, Nielsen and Goodrich, 2006b).

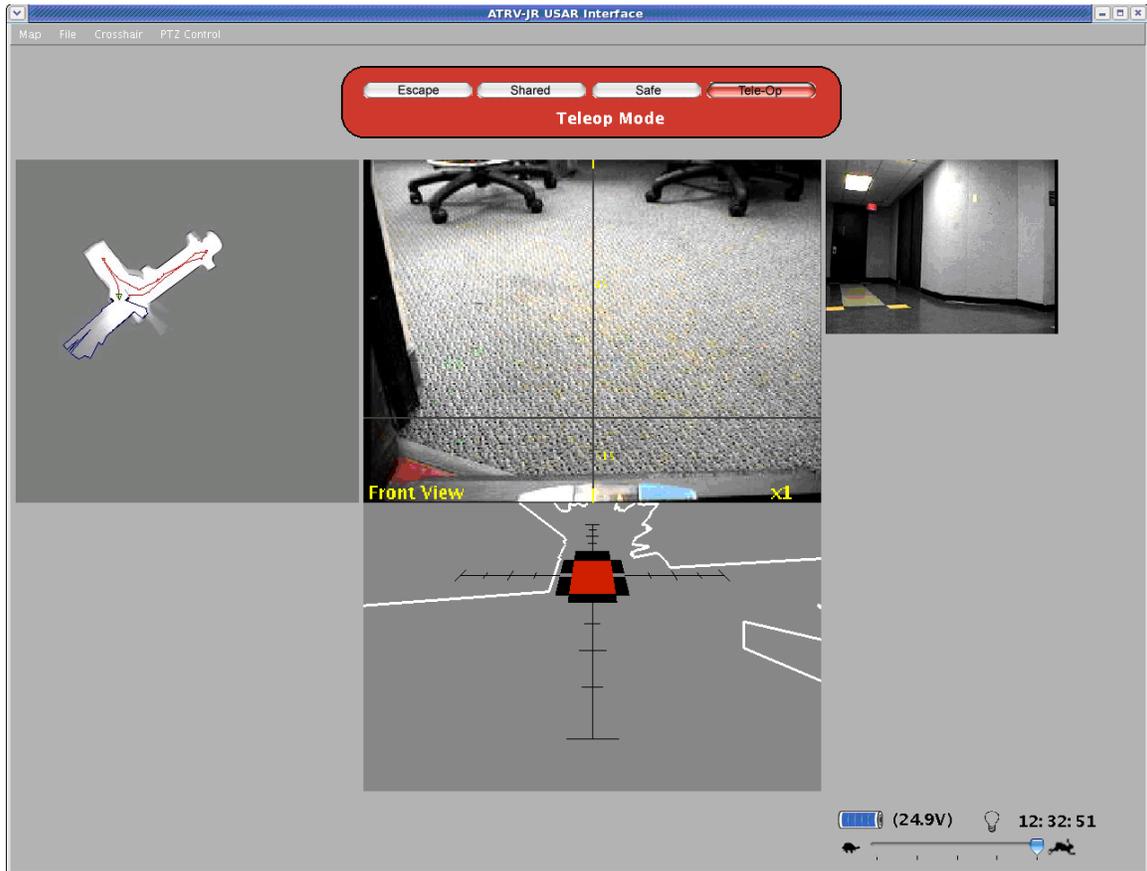


Figure 5. UML's new design. It has a larger video window, a new distance panel, and a relocated mode bar.

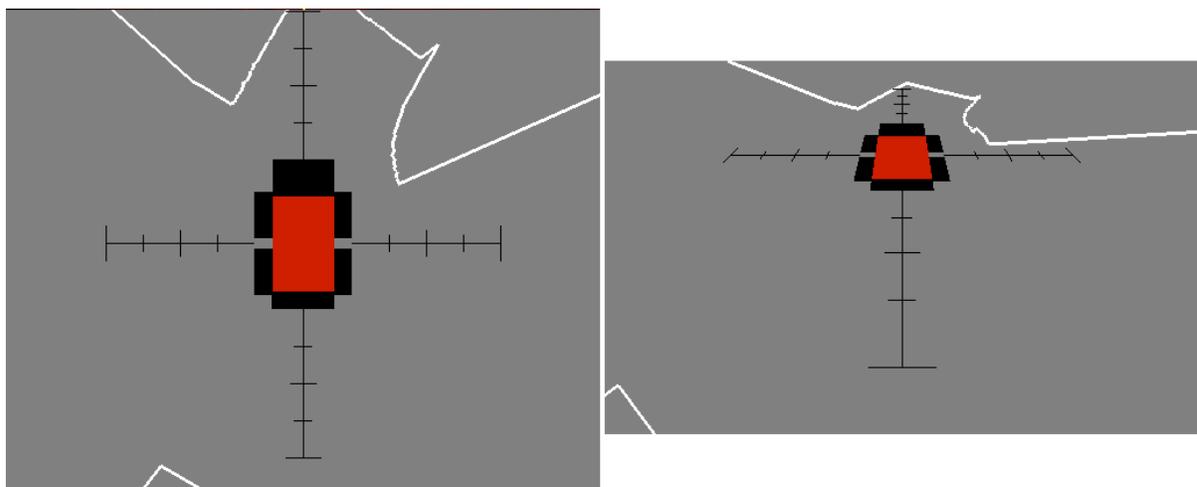
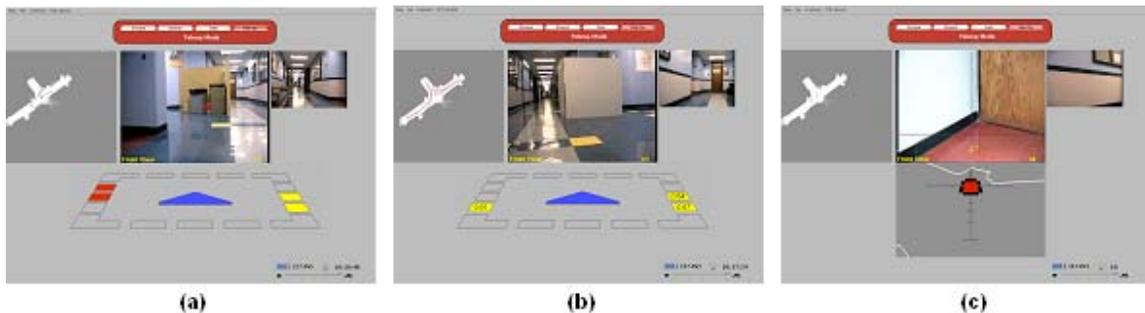


Figure 6. Close-up of new distance panel. The left-hand figure shows the top-down (2D) view, and the right-hand figure shows the same view but in a 3D perspective display.

## 5.1 Experiment Design

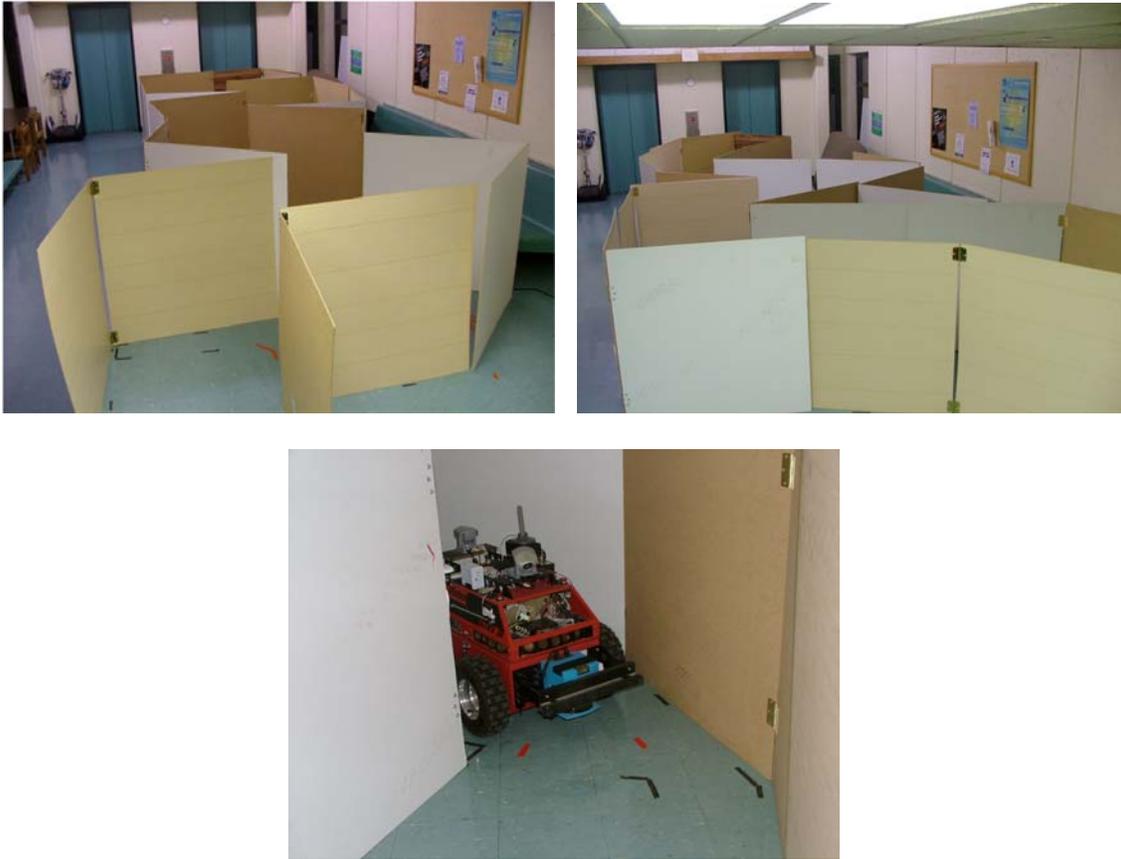
To see whether the new distance panel was more effective, we designed an experiment that compared the new distance panel described above in section 4 with the previous UML distance panel described in section 2.2. Further, we included one other variant: the previous UML distance panel with numbers written in the colored boxes that represented the distances in meters between the robot and the obstacles. This additional variant was included because the previous version did not provide an exact measure of the distance between the robot and obstacles. While we felt the new design could provide substantial benefits, we needed to make sure that the new design would be more effective than amending the previous design slightly. These interface layouts are shown in figure 7, with the previous interface (termed Interface A for ease of discussion), the previous interface amended with distances in meters (Interface B), and the revised interface (Interface C).

This new study followed a within-subjects design with 18 participants: 12 men and 6 women. They ranged in age from 26 to 39, with varying professions. None of them were USAR experts. Similar to experiment 1, we introduced the study to participants, requested that they sign an informed consent form, had participants answer questions from a pre-experiment questionnaire, then provided training and a practice period on an interface before starting a run. The order of presentation of the interfaces was counterbalanced to eliminate confounding from a learning effect. Finally, participants answered questions from a post-experiment questionnaire that explored their subjective opinions of the interfaces.



**Figure 7: a) Interface A: Distance panel with just the colored boxes. b) Interface B: Colored boxes with distance values displayed in the boxes. c) Interface C: Mini-map-inspired panel with crosshair-like lines indicating the distance values.**

Each participant was tasked to go through an arena and back again (the way they came in) along a single path. Because we wished to concentrate on the navigation task, the participant was not searching for victims, just traversing a course. There were three variants of the course to eliminate the confounder of participants learning a course in one run and having that learning help them in the subsequent two runs. The courses in this study, two of which can be seen in figure 8, were of equal length and difficulty and were extremely narrow. In some cases, there was only 3 centimeters of clearance on either side of the robot. This was done to fully exercise the distance panels on each interface to



**Figure 8: Examples of the testing arenas used for Experiment 2.**

find which one was truly the best. If the arenas were wide open and easy, then there may have been no significant difference found between the distance panels.

We did not want the participants to get lost in the arena, so the courses were deliberately made to have only one possible way to go. We wanted to know which interface yielded the fastest results, and if a participant was lost in a maze, the results could get skewed, yielding an incorrect result for which is the best distance panel. Each operator was given all the time they needed to go from the start to the end of the arena and back to the start. In a few rare cases, after turning the robot, the subject got confused as to which way to go. This mostly occurred if they were so close to a wall so the video camera only showed the wall, or if they turned too much, and the path they saw ahead was the way they just came from. If a participant did become confused regarding the robot's location, the test administrator told them the correct way to go, which is the only information the test administrator gave the operators while the actual runs were in progress.

We also forced experiment participants to only use the teleoperation mode. The chief way in which autonomy is used, in this scenario, is to sense proximity to obstacles and stop the robot prior to collisions. Allowing the robot to take this kind of initiative would skew the results, so we disallowed the use of autonomy entirely.

## 5.2 Analysis Method

Analysis focused on task performance. We calculated time to complete the course for each interface and the number of collisions (which we sometimes call “hits”) during each run. We analyzed the results in light of six hypotheses.

*H1: Using Interface A will result in more collisions than the other interfaces.*

*H2: Run times using Interface A will be fastest.*

We predicted that Interface A would yield the most collisions due to the uncertainty of the distance information it provides. However, we thought that it would yield relatively fast run times because the operator would start to ignore this uncertain information, and so would not take the time to glance at it and mentally decode it.

*H3: Using Interface B will result in many fewer collisions than using Interface A.*

*H4: Run times using Interface B will be the slowest of the three interfaces.*

Interface B is the same as Interface A except that it provides exact data on the distances between the robot and obstacles in the form of numeric values written in the color-coded blocks. We predicted that the precision of the information will aid users in avoiding collisions. However, because the operator would have to take the time to mentally process the numerical data, we felt Interface B would lead to longer run times than both Interfaces A and C. In particular, the numbers in the boxes would constantly change values as the robot moved, drawing operators’ eyes to them and causing high cognitive loading as the operator frequently reassessed their meanings.

*H5: Using Interface C will result in the lowest number of collisions of the three interfaces.*

We predicted that experiment participants would expend less time perceiving and mentally processing the information provided by Interface C compared to the other interfaces, leading to faster run times. We also predicted that Interface C would yield fewer collisions than the other interfaces. The map-like representation with the true-to-scale robot icon allows for a more direct comparison of the relationship between the robot and its immediate surroundings than can be obtained with the relatively abstract blocks presented in the control interfaces. Based on human-computer interaction principles, we conjectured that Interface C requires less mental translation and rotation, which results in a more accurate understanding of the relationship between the robot and obstacles.

*H6: Participants will prefer Interface C to the other two interfaces.*

Because we believed that experiment participants would expend less cognitive effort understanding the robot’s relationship to its immediate environment, we predicted that users would like Interface C better than the other interfaces.

## 5.3 Results

In this section, we report the results of the experiment with the redesigned UML system. Our quantitative measures of task performance were completion time and the number of collisions with obstacles in the environment. We also present qualitative results garnered through questions answered by participants.

### 5.3.1 Task Completion Time

On average, participants took 508 seconds to complete the runs using Interface A, 635 seconds using Interface B, and 495 seconds using Interface C. We feel that run times using Interface A were similar to that of Interface C because there were no mental calculations to be done in either case. Interface B required the user to translate the numbers indicating distance into an assessment of whether the robot was getting too close to an obstacle or was still at a safe distance; this type of translation adds to mental workload (Norman, 1988). For most of the runs, the boxes in Interfaces A and B remained red due to the narrowness of the passages in much of the test courses.

**Table 3: Time and Collision Results**

Interface	Time (s)	$\sigma_t$	Collisions	$\sigma_c$
A	507.9	283.6	8.8	3.7
B	634.6	409.1	7.6	3.1
C	495.4	217.8	6.0	3.1

Using a two-tailed paired t-test (dof=17), we found that Interface A was significantly faster than Interface B ( $p=0.02$ ). Interface C was also significantly faster than Interface B ( $p=0.031$ ). There was not a significant difference between the times taken with Interface A versus Interface C, so H2 could not be proved nor disproved. Hypothesis H4, however, was shown to be true.

As with many robot driving tasks, we observed a learning effect with respect to the time it took to complete the task; regardless of the ordering of the interfaces, completion times for later runs were, on average, faster than previous runs. However, because the order of using the interfaces and test courses was counterbalanced, one interface was not unfairly given the edge for time. No course was significantly easier than the others with respect to time to complete when using the same interface.

### 5.3.2 Number of Collisions

When comparing the number of collisions that occurred, hypothesis H5 was proven to be correct. Interface A had the most collisions with an average of 8.8 collisions per run, Interface B had an average of 7.6 collisions per run, and Interface C averaged 6.0 collisions per run. While the number of collisions experienced using Interface A versus Interface B was not significant ( $p=0.14$ ; thus neither H1 nor H3 could be proven), the number of collisions was significant when comparing both the other interfaces to

Interface C: Interface A compared to Interface C resulted in  $p = 0.007$ , and compared to Interface B,  $p = 0.041$ . These results show that Interface C was clearly the winner when it came to the amount of collisions that occurred.

### **5.3.3 Subjective Assessment**

As for the new distance panel, the majority of the participants (11 of 18) preferred Interface C, lending support to H6, while six of the eighteen participants preferred Interface B. Some commented that having the exact numbers were a huge benefit and indicated that if somehow the numbers could be shown along with the lines from Interface C, they would like it better. Only one participant selected Interface A as being the best, stating that they liked how it was less cluttered, but also that they were more used to the system by the third run. Because Interface A was the last interface they used, perhaps this participant would have chosen a different favorite if they happened to have another interface last.

Three participants, however, did say they liked Interface C the least. All three commented that the lines kept changing their distance, which made it hard to track. The sides and back of the robot use sonar sensors to detect the distance. Sonar sensors are inherently unstable and fluctuate a great deal. There is an averaging algorithm being performed as the robot collects the distance readings, to try to minimize this fluctuation, but since Interface C is easy to interpret, every shift is noticed. With Interface A, the box will most likely stay the same color, or, in Interface B, the robot operator won't notice the changing numbers as much if they aren't directly looking at it. We believe this is more of a poor sensor issue, rather than a poor display issue, because if there were laser sensors on the sides and back of the robot, instead of the sonars, these fluttering lines would not occur: the movement of the lines as the robot moves through the environment would be much more fluid. Fourteen of the eighteen users disliked Interface A the most, and one user disliked Interface B the most.

Half of the participants preferred having Interface C in its perspective view, while the other half preferred it in the top-down view. Therefore the ability to be able to switch views is a feature that will be kept. Most participants generally chose the view they liked at the beginning of the run and stuck with it. Several participants, however, did change the panel's view at various times during the run. Generally these users would put the panel in the top-down view when they were very close to an obstacle.

## **5.4 Discussion**

We found the data closely matched hypothesis H1. Interface C far out-performed Interface A in number of collisions and Interface C also had significantly less collisions and yielded significantly faster run times than Interface B.

The results from this experiment are similar to the results from the experiments by Ricks, Nielsen, and Goodrich, in that time to complete a navigation task and the number of collisions can be reduced by changing the manner in which the information is presented

to the operator (Ricks et al., 2004, Nielsen and Goodrich, 2006a, Nielsen and Goodrich, 2006b). The current experiment offers the additional insight that adding small amounts of textual information appeared to impede the operator's ability to complete the task in a timely fashion, as evidenced by the differences in the time to completion and the number of collisions between Interface A and Interface B. Future work should explore the contexts in which text is used by the interface as a means to convey information to the operator from the robot as it may actually distract the operator and hinder performance.

The total number of collisions that occurred in this experiment is much larger than the number we've seen in previous studies. This is not a sign that the newer interface is in some way inferior. The test courses in this experiment were extremely narrow, so a larger number of total collisions was expected.

On average, participants took longer to drive with Interface B (634.6 seconds, as shown in table 3) than with Interface A (507.9 seconds), but had fewer collisions (7.6 for Interface B compared to 8.8 for Interface A). Recall that these two interfaces were identical, except for the inclusion of numbers giving the absolute distance to an obstacle being shown in Interface B. As would be expected, and was predicted by H4, Interface B was the slowest of the three. Its design required participants to interpret the meaning of the numbers in order to judge how close obstacles were to the robot. Without these numbers, in Interface A, participants drove more quickly, but the lack of specific distance information resulted in more collisions, as predicted in H1 and H3.

We found that Interface C, which presented the sensor data showing obstacles around the robot in a graphical fashion showing the absolute distance as a line drawn around a robot icon, performed the best, both in terms of average course traversal time (495.5 seconds) and average number of collisions (6.0). So although both Interface B and Interface C provided the same information (absolute distance from the robot), the presentation of this data in Interface C allowed for more effective driving. In interface design, efficient operation depends upon the effective presentation of information; merely presenting additional sensor information on the interface screen can hinder the task, if data is not presented well.

One confounder that may need to be studied in the future regards the effect of the simplification of the participants' tasks. In this study, the robot operators started down a path and then came back without having to look for victims or make choices about how to get through a maze of corridors. Therefore, a participant may have been more apt to concentrate on the distance panel more than otherwise would have been the case because there was no threat of missing a victim or important landmark in the video.

## **6 Conclusions and Future Work**

Despite the multiple years of work reported in this paper, additional work remains to be done. We've shown that the new distance panel in the UML interface is more effective for driving in narrow spaces in the absence of confounding issues as confirmed by significantly fewer collisions for the new distance panel in Interface C over the indefinite

Interface A and definite, but text based, Interface B, and a significantly faster driving time than Interface B. (Although the driving time was not significantly different for Interfaces A and C, the greater number of collisions with Interface A indicate that Interface C is preferable in a task where both time and collisions are important.) The next step is to perform usability tests on the newly designed interface that include a search task. These tests will allow for a comparison with the results reported for Experiment 1.

Although the new distance panel provides information about the area immediately surrounding the robot, it does not serve to improve the operator's awareness of the whole environment. One way to help the operator better understand the environment is to provide representations illustrating to the operator where the robot has and has explored within the remote environment, which both robots did in all versions of the interfaces. In particular, participants called out the breadcrumb trail on the UML interface as useful to indicate where the robot had been; however, it does not illustrate where, in the environment, the camera was pointing while the robot was traversing the environment. To increase this knowledge of the environment, INL is investigating the use of a representation that presents information about where the camera was pointing as the robot was moved through the environment. This should enable operators to quickly recognize what parts of the environment have been "seen" by the robot and continue on to unseen areas.

To help the operator remember the environment better and perform larger area searches efficiently, we are also investigating new ways to illustrate both ego-and exo-centric information about the environment such that the operator can better direct the robot in a search task. We anticipate that these approaches will improve the usefulness of remote robots in urban search and rescue tasks as well as other remote robot tasks that require the use of video information in conjunction with navigational information.

As INL and UML continue the development of their interfaces, we are planning another set of experiments similar to Experiment 1. The results of our years of collaborative testing are two interfaces with design elements that are becoming more similar to one another. UML's new distance panel can be seen as a local version of the 3D map that the INL interface provides.

The iterative and interactive design process that we have undertaken has resulted in robot systems that can be navigated in remote environments more effectively. We have shown that effective fusing of sensor data results in improvements in driving performance. While the HCI domain has long advocated fusing of information, robot developers have not always effectively applied this principle. Our designs and experiments show the importance of learning from the HCI domain while adapting that knowledge to human-robot interaction.

## Acknowledgments

This work is sponsored in part at UMass Lowell by the National Science Foundation (IIS-0415224, IIS-0308186) and the National Institute of Standards and Technology (70NANB3H1116). At the Idaho National Laboratory, it is sponsored by the Idaho National Laboratory's Intelligent Systems Initiative.

## References

Adams, J. A. (2005). "Human-robot interaction design: understanding user needs and requirements." *Proceedings of the 2005 Human Factors and Ergonomics Society 49<sup>th</sup> Annual Meeting*, 2005, Orlando, FL.

Alexander, A. L. and Wickens, C. D. (2005). 3D navigation and integrated hazard display in advanced avionics: Performance, situation awareness, and workload. Technical Report AHFD-05-10/NASA-05-2. Urbana-Champaign, Illinois: Aviation Human Factors Division, Institute of Aviation, University of Illinois at Urbana-Champaign. Available at: [www.humanfacors.uiuc.edu/ Reports&PapersPDFs/ TechReport/05-10.pdf](http://www.humanfacors.uiuc.edu/Reports&PapersPDFs/TechReport/05-10.pdf)

Baker, M., Casey, R., Keyes, B., and Yanco, H. A. (2004). "Improved interfaces for human-robot interaction in urban search and rescue." *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, pp. 2960–2965, The Hague, The Netherlands.

Bruemmer, D. J., Marble, J. L., Few, D. A., Boring, R. L., Walton, M. C., and Nielsen, C. W. (2004). "Shared understanding for collaborative control." *IEEE Transactions on Systems, Man, and Cybernetics - Part A*, 35(4):494–504.

Casper, J. and Murphy, R. R. (2003). "Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center." *IEEE Transactions on Systems, Man, and Cybernetics - Part B*, 33(3):367–385, June.

Drascic, D. and Milgram, P. (1996). "Perceptual issues in augmented reality." *SPIE: Stereoscopic Displays and Virtual Reality Systems III*, p. 2653.

Drury, J., Riek, L. D., Christiansen, A. D., Eyler-Walker, Z. T., Maggi, A. J. and Smith, D. B. (2003). "Command and control of robot teams." *Proceedings of the Association of Unmanned Vehicle System International 2003 Conference*.

Endsley, M. R. (1988). "Design and evaluation for situation awareness enhancement." *Proceedings of the Human Factors Society 32nd Annual Meeting*, Santa Monica, CA, October.

Ericsson, K. A. and Simon, H. A. (1980). "Verbal reports as data." *Psychological Review*, 87:215–251.

Galitz, W. O. (2007). The essential guide to user interface design, 3 Ed. New York: John Wiley and Sons.

Gussow, D. (2005). "Robot rescue: these guys go where human searchers can't." *St. Petersburg Times*, St. Petersburg, Florida, 19 September. Available at [http://www.sptimes.com/2005/09/19/Technology/Robot\\_rescue\\_\\_These\\_g.shtml](http://www.sptimes.com/2005/09/19/Technology/Robot_rescue__These_g.shtml).

Hix, D. and Hartson, H. R. (1993). Developing User Interfaces: Ensuring Usability Through Product and Process. New York, New York: John Wiley and Sons.

Howard, A. (2004). Simple mapping utilities (pmap) v. 0.1.0. Los Angeles: University of Southern California. Available at <http://robotics.usc.edu/~ahoward/pmap/index.html>.

Hughes, S. and Lewis, M. (2004). "Robotic camera control for remote exploration." *Proceedings of the CHI 2004 Conference on Human Factors in Computing Systems*, Vienna, Austria.

Jacoff, A., Messina, E., and Evans, J. (2000). "A standard test course for urban search and rescue robots." *Proceedings of the Performance Metrics for Intelligent Systems Workshop*.

Jacoff, A., Messina, E., and Evans, J. (2001). "A reference test course for autonomous mobile robots." *Proceedings of the SPIE-AeroSense Conference*, Orlando, FL.

Jeffries, R. J., Miller, J. R., Wharton, C., and Uyeda, K. M. (1991). "User interface evaluation in the real world: A comparison of four techniques" *Proceedings of the CHI'91 Conference on Human Factors in Computing Systems*, New Orleans, LA.

Kadous, M. W., Sheh, R. K., and Sammut, C. (2006). "Effective user interface design for rescue robotics." *Proceedings of the 2006 ACM Conference on Human Robot Interaction*, Salt Lake City, March 2006.

Keyes, B. (2007). "Evolution of a telepresence robot interface." Master's Thesis, Computer Science Department, University of Massachusetts Lowell, May.

Keyes, B., Casey, R., Yanco, H. A., Maxwell, B.A. and Georgiev, Y. (2006). "Camera placement and multi-camera fusion for remote robot operation." *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, National Institute of Standards and Technology (NIST), Gaithersburg, MD, August 22-24.

Kubey, R. and Csikszentmihalyi, M. (2002). "Television addiction is no mere metaphor." *Scientific American*, 286(2):62-68.

Mayhew, D. J. (1999). The Usability Engineering Lifecycle: A Practitioner's Handbook for User Interface Design. San Francisco: Morgan Kaufmann Publishers.

- Molich, R. and Nielsen, J. (1990). "Improving a human-computer dialogue." *Communications of the ACM*, 33(3):338-348, March.
- Nielsen, C. W. and Goodrich, M. A. (2006a). "Comparing the usefulness of video and map information in navigation tasks." *Proceedings of the 2006 Human-Robot Interaction Conference*, Salt Lake City, UT.
- Nielsen, C. W. and Goodrich, M. A. (2006b). "Testing the usefulness of a pan-tilt-zoom (PTZ) camera in human-robot interactions." *Proceedings of the Human Factors and Ergonomics Society 50<sup>th</sup> Annual Meeting*. San Francisco, CA.
- Nielsen, C. W., Goodrich, M. A., and Rupper, R. J. (2005). "Towards facilitating the use of a pan-tilt camera on a mobile robot." *Proceedings of the 14th IEEE International Workshop on Robot and Human Interactive Communication (RO-MAN)*, Nashville, TN.
- Nielsen, J. (1994). "Enhancing the explanatory power of usability heuristics." *Proceedings of the CHI '94 Conference on Human Factors in Computing Systems*, Seattle, WA, April.
- Norman, D. A. (1988). The Psychology of Everyday Things. New York: Basic Books.
- Nourbakhsh, I. R., Sycara, K., Koes, M., Yong, M., Lewis, M., and Burlon, S. (2005). "Human-robot teaming for search and rescue." *IEEE Pervasive Computing*, 4(1):72-78, Jan-March.
- Pacis, E. B., Everett, H. R., Farrington, N., and Bruemmer, D. J. (2004). "Enhancing functionality and autonomy in man-portable robots." *Proceedings of the SPIE Defense and Security Symposium*, pp. 13-15.
- Ricks, B. W., Nielsen, C. W., Goodrich, M. A. (2004). "Ecological displays for robot interaction: A new perspective." *Proceedings of the 2004 International Conference on Intelligent Robots and Systems IEEE/RSJ*, Sendai, Japan.
- Scholtz, J., Antonishek, B., and Young, J. (2004). "Evaluation of a human-robot interface: development of a situational awareness methodology." *Proceedings of the 2004 Hawaii International Conference on System Sciences*, Waikoloa, Hawaii, January.
- Steinfeld, A. (2004). Interface lessons for fully and semi-autonomous mobile robots. *Proceedings of the 2004 IEEE International Conference on Robotics and Automation*, New Orleans, LA, April.
- Thrun, S. (2002). "Robotic mapping: a survey." Exploring Artificial Intelligence in the New Millennium, Morgan Kaufmann.

Yanco, H. A., Baker, M., Casey, R., Keyes, B., Thoren, P., Drury, J. L., Few, D., Nielsen, C. and Bruemmer, D. (2006). “Analysis of human-robot interaction for urban search and rescue.” *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, National Institute of Standards and Technology (NIST), Gaithersburg, MD, August 22-24.

Yanco, H. A. and Drury, J. L. (2004). ““Where am I?": acquiring situation awareness using a remote robot platform.” *Proceedings of the 2004 IEEE Conference on Systems, Man, and Cybernetics*, The Hague, Netherlands, October.

Yanco, H. A. and Drury, J. L. (2007). “Rescuing interfaces: a multi-year study of human-robot interaction at the AAI Robot Rescue Competition.” *Autonomous Robots*, 22(4):333-352, May.

Zalud, L. (2006). “Argos – system for heterogeneous mobile robot teleoperation. *Proceedings of the 2006 IEEE RSJ International Conference on Intelligent Robots and Systems*, October 9 – 16, Beijing, China.