

# Improving Human-Robot Interaction through Interface Evolution

Brenden Keyes<sup>1</sup>, Mark Micire<sup>2</sup>, Jill L. Drury<sup>1</sup> and Holly A. Yanco<sup>2</sup>

<sup>1</sup>*The MITRE Corporation, Bedford, MA,*

<sup>2</sup>*University of Massachusetts Lowell, Lowell, MA,  
USA*

## 1. Introduction

In remote robot operations, the human operator(s) and robot(s) are working in different locations that are not within line of sight of each other. In this situation, the human's knowledge of the robot's surroundings, location, activities and status is gathered solely through the interface. Depending on the work context, having a good understanding of the robot's state can be critical. Insufficient knowledge in an urban search and rescue (USAR) situation, for example, may result in the operator driving the robot into a shaky support beam, causing a secondary collapse. While the robot's sensors and autonomy modes should help avoid collisions, in some cases the human must direct the robots' operation. If the operator does not have good awareness of the robot's state, the robot can be more of a detriment to the task than a benefit.

The human's comprehension of the robot's state and environment is known as situation awareness (SA). Endsley developed the most generally accepted definition for SA: "The perception of elements in the environment within a volume of time and space [Level 1 SA], the comprehension of their meaning [Level 2 SA] and the projection of their status in the near future [Level 3 SA]" (Endsley, 1988). Drury, Scholtz, and Yanco (2003) redefined this definition of situation awareness to make it more specific to robot operations, breaking it into five categories: human-robot awareness (the human's understanding of the robot), human-human awareness, robot-human awareness (the robot's information about the human), robot-robot awareness, and the humans' overall mission awareness. In this chapter, we focus on two of the five types of awareness that relate to a case in which one human operator is working with one robot: human-robot awareness and the human's overall mission awareness. Adams (2007) discusses the implications for human-unmanned vehicle SA at each of the three levels of SA (perception, comprehension, and projection).

In Drury, Keyes, and Yanco (2007), human-robot awareness is further decomposed into five types to aid in assessing the operator's understanding of the robot: location awareness, activity awareness, surroundings awareness, status awareness and overall mission awareness (LASSO). The two types that are primarily addressed in this chapter are location awareness and surroundings awareness. Location awareness is the operator's knowledge of where the robot is situated on a larger scale (e.g., knowing where the robot is from where it started or that it is at a certain point on a map). Surroundings awareness is the knowledge the operator has of the robot's circumstances in a local sense, such as when there is an

Source: Human-Robot Interaction, Book edited by: Daisuke Chugo,  
ISBN 978-953-307-051-3, pp. 288, February 2010, INTECH, Croatia, downloaded from SCIYO.COM

obstacle two feet away from the right side of the robot or that the area directly behind the robot is completely clear.

Awareness is arguably the most important factor in completing a remote robot task effectively. Unfortunately, it is a challenge to design interfaces to provide good awareness. For example, in three studies that examined thirteen separate USAR interfaces (Yanco et al., 2002; Scholtz et al., 2004; Yanco & Drury, 2006), there were a number of critical incidents resulting from poor awareness. For instance, an operator moved a video camera off-center to conduct victim identification. After allowing the robot to drive autonomously out of a tight area, the operator forgot that the camera was off-center, resulting in poor robot operation, including collisions and operator confusion (Yanco et al., 2004).

This chapter presents lessons learned from the evolution of our human-robot interaction (HRI) design for improved awareness in remote robot operations, including new design guidelines. This chapter brings together for the first time the four different versions of the interface created during the evolution of our system, along with the motivation for each step in the design evolution. For each version of the interface, we present the results of user testing and discuss how those results influenced our next version of the interface. Note that the final version of the interface discussed in this chapter (Version 4) was designed for multi-touch interaction, and the study we conducted on this version establishes a performance baseline that has not been previously documented.

The next section presents summaries of some of the previous interfaces that have influenced our design approaches, followed by our design and testing methodology in Section 3. Section 4 describes briefly the robot hardware that was controlled by the various interfaces. After a section presenting our general interface design approach, the next four sections describe the four generations of the evolving interface. Finally, we present conclusions and plans for future work.

## **2. Related work**

We did not design in a vacuum: there have been numerous attempts in the past decade to design remote robot interfaces for safety-critical tasks. Remote robot interfaces can be partitioned into two categories: map-centric and video-centric (Yanco et al., 2007). A map-centric interface is an interface in which the map is the most predominant feature in the interface and most of the frequently used information is clustered on or near the map. Similarly, in a video-centric interface, the video window is the most predominant feature with the most important information located on or around the video screen.

Only a few interfaces are discussed here due to space limitations; for a survey of robot interfaces that were used in three years of the AAI Robot Rescue competition, see Yanco and Drury (2006).

### **2.1 Map-centric interfaces**

It can be argued that map-centric interfaces are better suited for operating remote robot teams than video-centric interfaces due to the inherent location awareness that a map-centric interface can provide. The relationship of each robot in the team to other robots as well as its position in the search area can be seen in the map. However, it is less clear that map-centric interfaces are better for use with a single robot. If the robot does not have adequate sensing capabilities, it may not be possible to create maps having sufficient accuracy. Also, due to an emphasis on location awareness at the expense of surroundings

awareness, it can be difficult to effectively provide operators with a good understanding of the area immediately around the robot.

One example of a map-centric interface, developed by the MITRE Corporation, involved using up to three robots to build a global map of the area covered by the robots. Most of the upper portion of the display was a map that gradually updated as ranging information was combined from the robots. The interface also had the ability to switch operator driving controls among the three robots. Small video windows from the robots appeared under the map. The main problems with this interface were the small size of the video screens as well as the slow updates (Drury et al., 2003).

Brigham Young University and the Idaho National Laboratory (INL) also designed a map-centric interface. The INL interface has been tested and modified numerous times, originally starting as a video-centric interface before changing to a map-centric interface (Nielsen et al., 2007; Nielsen & Goodrich, 2006; Nielsen et al., 2004). This interface combines 3D map information using blue blocks to represent walls or obstacles with a red robot avatar indicating its position on the map. The video window is displayed in the current pan-tilt position with respect to the robot avatar, indicating the orientation of the robot with respect to where the camera is currently pointing. If the map is not generated correctly due to moving objects in the environment, faulty sensors or other factors, however, the operator can become confused regarding the true state of the environment. We have witnessed cases in which the INL robot slipped and its map generation from that point on shifted to an offset from reality, with the consequence that the operator became disoriented regarding the robot's position.

Because of these drawbacks for remote robot operations (overreliance on potentially inaccurate maps and a smaller video displays due to larger maps), we found inspiration for our interface designs in video-centric interfaces.<sup>1</sup>

## 2.2 Video-centric interfaces

Video-centric interfaces are by far the most common type of interface used with remote robots. Operators rely heavily on the video feed from the robot and tend to ignore any other sensor reading the interface may provide (e.g., see Yanco & Drury, 2004). Many commercially available robots have video-centric interfaces (e.g., iRobot's Packbot and Foster Miller's Talon).

ARGOS from Brno University of Technology is an excellent example of a video-centric interface (Zalud, 2006). It provides a full screen video interface with a "heads up" display (HUD) that presents a map, a pan/tilt indicator and also a distance visualization widget that displays the detections from the laser sensor on the front of the robot. What makes this interface unique is its use of virtual reality goggles. These goggles not only display the full interface, but the robot also pans and tilts the camera based on where the operator is looking, making scanning an area as easy as turning your head in the direction you want to look. It also eliminates issues with forgetting that the camera is not centered.

The CASTER interface developed at the University of New South Wales (Kadous et al., 2006) also provides a full screen video interface but incorporates a different arrangement of small sensor feeds and status readouts placed around the edges.

---

<sup>1</sup> Readers who are interested in map-based interfaces in collocated operations may find the guidelines and heuristics in Lee et al. (2007) to be helpful.

Researchers at Swarthmore College (Maxwell et al., 2004) have designed a video-centric interface that includes a main panel showing the view of the video camera. It has the unique feature of overlaying green bars on the video which show 0.5 meter distances projected onto the ground plane. The interface also has pan-tilt-zoom indicators on the top and left of the video screen, and it displays the current sonar and infrared distance data to the right of the video window.

Inspired by these video-centric systems, we have incorporated into our interface a large video feed in the central portion of the interface and a close coupling between pan-tilt indicators and the video presentation.

### 3. Methodology

#### 3.1 Design methodology

We started with an initial design based upon a list of guidelines recommended by Yanco, Drury and Scholtz (2004) and Scholtz et al. (2004). The guidelines state that a USAR interface should include:

- A map of where the robot has been.
- Fused sensor information to lower the cognitive load on the user.
- Support for multiple robots in a single display (in the case of a multi-robot system).
- Minimal use of multiple windows.
- Spatial information about the robot in the environment.
- Help in deciding which level of autonomy is most useful.
- A frame of reference to determine position of the robot relative to its environment.
- Indicators of robot health/state, including which camera is being used, the position(s) of camera(s), traction information and pitch/roll indicators.
- A view of the robots' body so operators can inspect for damage or entangled obstacles.

We also kept in mind the following design heuristics, which we adapted from Nielsen (1993):

- Provide consistency; especially consistency between robot behavior and what the operator has been led to believe based on the interface.
- Provide feedback.
- Use a clear and simple design.
- Ensure the interface helps to prevent, and recover from, errors made by the operator or the robot.
- Follow real-world conventions, e.g., for how error messages are presented in other applications.
- Provide a forgiving interface, allowing for reversible actions on the part of the operator or the robot as much as possible.
- Ensure that the interface makes it obvious what actions are available at any given point.
- Enable efficient operation.

Finally, we designed to support the operator's awareness of the robot in five dimensions:

- Enable an understanding of the robot's location in the environment.
- Facilitate the operator's knowledge of the robot's activities.
- Provide to the operator an understanding of the robot's immediate surroundings.
- Enable the operator to understand the robot's status.
- Facilitate an understanding of the overall mission and the moment-by-moment progress towards completing the mission.

We realized that we were not likely to achieve an optimal design during the first attempt, so we planned for an iterative cycle of design and evaluation.

### 3.2 SA measurement techniques

Because it is important to characterize and quantify awareness as a means to evaluate the interfaces, we discuss SA measurement techniques here. Hjelmfelt and Pokrant (1998) state that experimental methods for measuring SA fall into three categories:

1. Subjective: participants rate their own SA
2. Implicit performance: Experimenters measure task performance, assuming that a participant's performance correlates with SA and that improved SA will lead to improved performance
3. Explicit performance: Experimenters directly probe the participant's SA by asking questions during short suspensions of the task

For these studies, we elected to use mainly implicit measures to associate task outcomes with implied SA; in particular, we focused on task completion time and number of collisions. A faster completion time as well as fewer collisions implies better SA. We also performed an explicit measure at the end of some studies, in which the user was asked to complete a secondary task that required awareness, such as: return the robot to a particular landmark that was previously visited. We used post-task questions that asked for participants' subjective assessment of their performance. We did not place great weight on the subjective assessments, however. Even if participants reported that they had performed well, their assessments were not necessarily accurate. In the past, we had observed many instances in which participants reported that the robot had not collided with obstacles when they had actually experienced collisions that caused damage to the arena (e.g., see Yanco et al., 2004).

### 3.3 General testing methodology

For all of our evaluations, we used similar test arenas that were based upon the National Institute of Standards and Technology (NIST) USAR arena (Jacoff et al., 2000; Jacoff et al., 2001; Jacoff et al., 2002). Each study used multiple arena orientations and robot starting positions, which were permuted to eliminate learning effects. In all the studies, except for the one that was performed on Version 3 of the interface, the users had a set time limit to complete their task. In most cases, participants were told that a disaster had occurred and that the participant had a particular number of minutes to search for and locate as many victims as possible. The time limit was between 15 and 25 minutes depending on the study. We used an "over-the-shoulder" camera that recorded the user's interaction with the interface controls as well as the user's think-aloud comments (Ericsson & Simon, 1980). Think-aloud is a protocol in which the participants verbally express their thoughts while performing the task assigned to them. They are asked to express their thoughts on what they are looking at, what they are thinking, why they are performing certain actions and what they are currently feeling. This allows the experimenters to establish the reasoning behind participants' actions. When all the runs ended, the experimenter interviewed the participant. Participants were asked to rate their own performance, to answer a few questions about their experience, and to provide any additional comments they would like to make.

During the run, a camera operator and a person recording the robot's path on a paper map followed the robot through arenas to create a record of the robot's progress through the test

course. The map creator recorded the time and location on the map of critical incidents such as collisions with obstacles. The map and video data were used for post-test analysis to determine the number of critical incidents and to cross-check data validity.

We analyzed this data to determine performance measures, which are implicit measures of the quality of the user interaction provided to users. As described above, we inferred awareness based on these performance measures. We recorded the number of collisions that occurred with the environment, because an operator with good surroundings awareness should hit fewer obstacles than an operator with poor surroundings awareness. We also analyzed the percentage of the arena covered or the time to complete the task, depending on the study. Operators with good location awareness should not unknowingly backtrack over places they have already been, and thus should be able to cover more area in the same amount of time than an operator with poor awareness, who might unknowingly traverse the same area multiple times. Similarly, we expected study participants with good awareness to complete the task more quickly than users with poor awareness, who may be confused and need additional time to determine a course of action. Participants' think-aloud comments were another important implicit measure of awareness. These comments provided valuable insight into whether or not a participant was confused or correctly recognized a landmark. For example, users would often admit to a loss of location awareness by saying "I am totally lost," or "I don't know if I've been here before," (speaking as a "virtual extension" of the robot).

#### **4. Robot hardware**

Our system's platform is an iRobot ATRV-JR. It is 77cm long, 55cm high and 64cm wide. It is a four-wheeled, all-terrain research platform that can turn in place due to its differential (tank-like) steering. The robot has 26 sonar sensors that encompass the full 360 degrees around the robot as well as a SICK laser range finder that covers the front 180 degrees of the robot. It has two pan/tilt/zoom cameras, one forward-facing and one rear-facing. To help with dark conditions in USAR situations, we added an adjustable lighting system to the robot.

The robot system has four autonomy modes: teleoperation, safe, shared, and escape, based upon Bruemmer et al. (2002). In the teleoperation mode, the operator makes all decisions regarding the robot's movement. In safe mode, the operator still directs the robot, but the robot uses its distance sensors to prevent the operator from driving into obstacles. Shared mode is a semi-autonomous navigation mode that combines the user's commands with sensor inputs to promote safe driving. Escape mode is the only fully autonomous mode on the system and is designed to drive the robot towards the most open space.

#### **5. General interface design**

Our interface was designed to address many of the issues that emerged in previous studies. The interface also presents easily readable distance information close to the main video so that the user is more likely to see and make use of it. The system also provides access to a rear camera and automatic direction reversal as explained below.

The main video panel is the hub of the interface. As Yanco and Drury (2004) state, users rely heavily on the main video screen and rarely notice other important information presented on the interface. Therefore, we located the most important information on or around the

main video screen so that the operator would have a better chance of noticing it. The main video screen was designed to be as large as possible so users can better perceive the visual information provided by the cameras. Further, we overlaid a small cross on the screen to indicate the direction in which the camera is pointing. These crosshairs were inspired by the initial design of the Brno robot system (Zalud, 2006).

In the prior studies discussed by Yanco, Drury and Scholtz (2004), we observed that more than 40% of robot collisions with the environment were on the rear of the robot. We believe a lack of sensing caused many of these rear collisions, so we added a rear-looking camera. Since the rear-looking camera would only be consulted occasionally, we mirrored the video feed and placed it in a similar location to a rear-view mirror in a car.

To further reduce rear collisions, we implemented an Automatic Direction Reversal (ADR) system. When ADR is in use, the interface switches the video displays such that the rear view is expanded in the larger window. In addition, the drive commands automatically remap so that forward becomes reverse and reverse becomes forward. The command remapping allows an operator to spontaneously reverse the direction of the robot in place.

The interface also includes a map panel, which displays a map of the robot's environment and the robot's current position and orientation within the 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. The placement of this panel changed throughout the evolution of the interface, but to ensure it is easily accessible to users, it has always remained at the same horizontal level as the video screen.

Throughout the evolution of our interface, the distance panel has been the main focus of development. It is a key provider of awareness of all locations out of the robot's current camera view. The distance panel displays current distance sensor readings to the user. The presentation of this panel has differed widely during the course of its progression and will be discussed more thoroughly in the next sections.

The autonomy mode panel has remained the same in all of our interface versions; it allows for mode selection and displays the current mode. The status panel provides all status information about the robot, including the battery level, the robot's maximum speed and if the lighting system is on or off.

## **6. Version 1**

### **6.1 Interface description**

The first version of the interface consisted of many of the panels described above in Section 5 and is shown in the top row of Table 1. The large video panel is towards the left center of the screen. The rear-view camera panel is located above the video panel to mimic the placement of a car's rear-view mirror. Bordering the main video screen are color-coded bars indicating the current values returned by the distance sensors. In addition to the color cues, multiple bars were filled in, with more bars meaning a closer object, to aid people with color deficiencies. Directly below the video screen is the mode panel. The illustration in Table 1 indicates that the robot is in the teleoperation mode. Directly to the right of the main video screen is the map panel. On the top-right of the interface is the status panel.

### **6.2 Evaluation description**

We designed a study to determine if adding the rear-facing camera would improve awareness (Keyes et al., 2006). We created three variations of the interface, which we refer

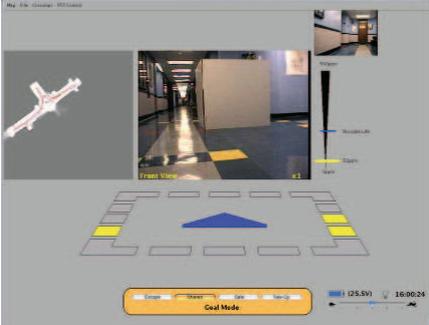
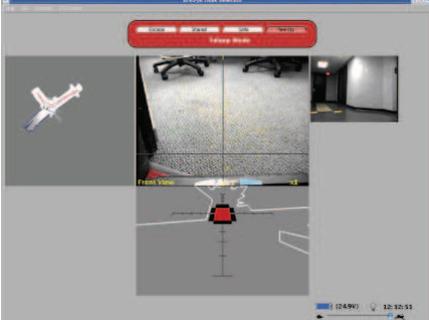
Illustration	Version Description
	<p>Version 1 consisted of the main video screen as well as the rear view camera, map, mode and status panels. The distance panel, placed around the main video screen, displays how close an obstacle is to the robot by filling in the colored bars. The interface was controlled via keyboard and joystick.</p>
	<p>Version 2 moves the video panel to the center of the screen. The distance panel is placed in a perspective view below the video screen, and turns from grey to yellow to red as objects get closer to the robot. It also rotates in response to the panning of the camera. Zoom mode (not shown here, but later in Figure 1), is displayed over the map panel when it is toggled on. This version was controlled via keyboard and joystick.</p>
	<p>Version 3 replaces the distance panel with a zoom mode inspired panel. Instead of colored boxes, lines are drawn around a scale model of the robot based on sensor information. This version was controlled via keyboard and joystick.</p>
	<p>Version 4 keeps the visual presentation the same as Version 3 while replacing the input method with multi-touch gesture activation. The virtual joystick in the lower right-hand corner provided the rotation and translation of the robot along with brake control. The visual feedback from Version 3, such as speed control and mode selection became interactive sliders and buttons in this interface.</p>

Table 1. Summary of the Interface Versions

to as Interfaces A, B, and C. Interface A consisted of the main video panel, distance panel, pan-tilt indicator, mode bar and status panel. For this interface, the participants only had access to the front camera's video stream. Interface B displayed all the same panels as Interface A, but the user could switch the main video panel to display the rear camera's video feed, triggering ADR mode. Interface C added the rear view camera panel and also had ADR mode, providing the users with the full Version 1 interface. Nineteen people participated, ranging in age from 18 to 50, with 11 men and 8 women. Using a within-subjects design, each participant operated the robot through the three different arena configurations using a different interface each time, with the order of the interface use and arena configurations being randomized.

### 6.3 Evaluation results

As expected, participants who had access to the rear camera had greater awareness than participants who did not. Using a two-tailed paired *t*-test, we found a significant difference in the number of collisions that occurred between the different interfaces. Participants made significantly more collisions when using Interface A (no rear-looking camera) than Interface C (both front- and rear-looking cameras displayed simultaneously) ( $M_A = 5.4$  collisions,  $SD_A = 3.2$ ,  $M_C = 3.6$ ,  $SD_C = 2.7$ ,  $p < 0.02$ ).

Participants also made significantly more collisions when using Interface A than Interface B (front and rear cameras both available but not displayed simultaneously) ( $M_A = 5.4$  collisions,  $SD_A = 3.2$ ,  $M_B = 3.9$ ,  $SD_B = 2.7$ ,  $p < 0.04$ ). These results indicate that awareness regarding the rear of the robot is improved by having access to the rear camera, even if the rear camera is not constantly being displayed. We did not find any significant difference in the time it took to complete the task.

There was only one user in this study who did not use the rear camera at all. The other eighteen participants made at least one camera switch when using Interface B. For Interface C, three of eighteen participants did not switch camera modes. One user stated that it was not necessary to switch camera modes because both cameras were being displayed already. Another user discussed being reluctant to switch views because it caused confusion when trying to keep track of the robot's current environment.

Five of the nineteen participants stated that they preferred to use only the front camera because they were able to pan the camera down to see the front bumper of the robot. The front of the robot has a larger bumper than the back of the robot, so the front camera is the only camera that can see the robot chassis. We found that the five users who had the strategy of looking at the bumper to localize the robot in the environment had fewer collisions ( $M = 8.0$  collisions,  $SD = 4.1$ ) than the other fourteen participants ( $M = 14.7$  collisions,  $SD = 6.6$ ).

We found that most of the collisions between the robot and the arena occurred on the robot's tires. Seventy-five percent of all the front collisions that occurred with the robot involved the robot's tires. These tires lie just outside the visible area and widen the robot by about five inches on each side. Despite warnings by the experimenter, users acted under the assumption that the boundaries of the video reflected the boundaries of the robot. It is also important to note that 71% of the total collisions in the study occurred on the tires. Because the tires make up almost the entire left and right sides of the robot, this result is unsurprising. The use of two cameras helped to improve situation awareness with respect to

the front and rear of the robot, but users still lacked SA with respect to the sides of the robot.

Fifteen of the nineteen participants (79%) preferred the interface with two camera displays. Three of the participants preferred the interface with two cameras that could be switched in a single video window. Two of these participants had little computer experience, which suggests that they might have been overwhelmed by the two video windows. The final participant expressed no preference between the two interfaces with two cameras but did prefer these two to the single camera case. No participant preferred the single camera case.

Two of the users in this study found the distance panel to be unintuitive. They thought the bars on top of the video window corresponded to distance sensors pointing directly up from the robot and the bars on the bottom represented distance sensors that were pointing down from the bottom of the robot. We also noted that due to the number of colors displayed by the bars, as well as the fact that different numbers of bars were filled, it was difficult for users to keep track of what was important. Often the display panel appeared to be blinking due to the high frequency with which distance values were changing. This resulted in the undesirable situation in which users started to ignore the panel altogether. While the addition of the rear camera helped improve SA significantly, the distance panel was not particularly helpful to prevent collisions on the side of the robot.

## 7. Version 2

Based upon the results of the previous study, particularly with respect to the lack of surroundings awareness relating to the sides of the robot, the focus of this design iteration was to improve the distance panel. Version 2 of the interface is the second image from the top in Table 1.

### 7.1 Interface description

The range data was moved from around the video window to directly below it. We altered the look and feel of the distance panel by changing from the colored bars to simple colored boxes that used only three colors (gray, yellow and red) to prevent the distance panel from constantly blinking and changing colors. In general, when remotely operating the robot, users only care about obstacles in close proximity, so using many additional colors to represent faraway objects was not helpful. Thus, in the new distance panel, a box would turn yellow if there was an obstacle within one meter of the robot and turn red if an obstacle was within 0.5 meters of the robot.

The last major change to the distance panel was the use of a 3D, or perspective, view. This 3D view allows the operator to easily tell that the “top” boxes represent forward-facing sensors on the robot. We believe this view also helps create a better mental model of the space due to the depth the 3D view provides, thus improving awareness around the sides of the robot. Also, because this panel was in 3D, it was possible to rotate the view as the user panned the camera. This rotation allows the distance boxes to line up with the objects the user is currently seeing in the video window. The 3D view also doubles as a pan indicator to let the user know if the robot’s camera is panned to the left or right.

This version of the interface also included new mapping software, PMap from USC, which added additional functionality, such as the ability to display the robot’s path through the environment (Howard, 2009).

One feature that resulted from the use of PMap was a panel that we termed “zoom mode.” This feature, which can be seen in Figure 1 on the left, represents a zoomed-in view of the map. It takes the raw laser data obtained in front of the robot and draws a line connecting the sensor readings together. The smaller rectangle on the bottom of this panel represents the robot. As long as the sensor’s lines do not touch or cross the robot rectangle, the robot is not in contact with anything. This sensor view gives highly accurate, readily visible cues regarding whether the robot is close to an object or not. Our goal was to develop an approach to make it easier to visualize the environment than the information from Version 1’s colored boxes by requiring the operator make fewer mental translations. However, due to the PMap implementation, the zoom mode and the map display panel were mutually exclusive (only one could be used at a time).

The video screen was moved from the left side to the center of the screen. This shift was mainly due to the fact that the new distance panel was larger, and, with the rotation feature, was not fully visible on the screen. Placing it in the center allowed for the full 3D view to be displayed at all times. The map was moved to the right side of the video.

## 7.2 Evaluation description

During Version 2’s evaluation, we studied the differences between our video-centric interface and INL’s map-centric interface (Yanco et al., 2007). Similar to the evaluation for Version 1, we designed a within-subjects study, counterbalancing whether participants began with the Version 2 interface or INL’s interface. We also varied the robot’s starting point into the arena to avoid introducing a learning effect. Seven men and one woman participated, ranging in age from 25 to 60. All had experience in search and rescue.

## 7.3 Evaluation results

Here we concentrate on the lessons learned about the Version 2 interface rather than the comparison between the Version 2 and INL interfaces. We found that users liked the new distance panel as well as the zoom mode capability. Although the colored boxes on the new distance panel resulted in better performance than the previous colored bar approach, the new distance panel design did not result in a large performance improvement. The main problem was that the use of only two colors, yellow and red, was too simple. When in a tight area, which is often the case in a USAR environment, the robot may not have 0.5 meters on either side of it; this was the case during the experiment. If a distance box was red, the user knew that an obstacle was within 0.5 meters but did not know exactly how close it was. When the robot is in a very confined area, 0.5 meters is a large distance. While the interface could have been tuned so that the boxes only turned red at 0.1 meters or even 0.05 meters, the basic problem would remain. The colored boxes are not informative enough, and that uncertainty causes the user to distrust the system.

The zoom mode feature helped to address the uncertainty caused by not knowing how far the robot is situated from an obstacle. Using the lines provides a concrete idea of how close an obstacle is to the robot without overwhelming the user. It is also extremely accurate, which can help produce a better mental model of the environment. The operator does not have to extrapolate what the area might look like based on colored boxes, thus reducing cognitive load. The user can also see the flow of the obstacles with respect to the robot’s movements. The zoom mode feature also helps to give the user a more accurate idea regarding the layout of obstacles.

## 8. Version 3

The results of the Version 2 user study demonstrated the utility of the zoom mode feature (shown on the left in Figure 1). The distance panel was still problematic and so became the focus of our next iteration.

### 8.1 Interface description

The Version 2 distance panel was removed and replaced by a distance panel based on the zoom mode feature (the Version 3 interface is the third image down in Table 1). The zoom mode was extended to encompass the entire circumference of the robot. The view of the front part of the robot would be based on laser data, whereas the views of the left, right, and rear of the robot would use sonar data. We also added tick marks at 0.25 meter increments to the lines to indicate distance. This panel was again placed directly under the main video display. Unlike the previous top-down zoom mode, this new panel also had the ability to provide a perspective view (the top down and perspective view distance panels are shown in the center and right of Figure 1). Results from the previous study indicated that users liked having the ability to go from a 2D map to a 3D map, so we felt users would appreciate this toggle ability here as well. Also, as with the previous distance panel, this panel also rotated with as the user panned the camera.

### 8.2 Evaluation description

We conducted an evaluation of this version of the interface. This new study consisted of 18 users, 12 men and 6 women. They varied in age from 26 to 39, with varying professions. None of them were USAR experts. The main purpose of this study was to compare the Version 2 distance panel (referred to as Interface D here) with the new Version 3 distance panel (referred to as Interface F here). For experimentation purposes, we also included a modified version of the distance panel from Interface D that overlaid the distance values in meters on the colored boxes (referred to as Interface E here) to give users exact distance information.

This test differed slightly from our previous studies. For this study, the user was only tasked to go through an arena and back again. Unlike all of the previous arenas, there was only one path for the user to take. When the user reached the end of the path, he was asked to turn around and come back out the way he had come in. Participants were not searching for victims: they were only asked to maneuver through the course. The courses in this study were much narrower than ones from previous studies. In some cases, there were only three centimeters of clearance on either side of the robot. This was done to fully exploit the weaknesses of the distance panels on each interface to determine which facilitated the best performance. If the arenas were wide open and easy to navigate, it would have been more difficult to discern differences between the interfaces.

We also forced the participants to use only teleoperation mode, which was not done in the previous studies, to remove the confounder of autonomy. Because we were studying the effects of the distance panel, if we allowed the robot to take some initiative, such as stopping itself, it may have prevented many of the collisions from happening and thus skewed the results.

For this study, we hypothesized that Interface D would perform the worst due to the lack of information that it provides. We believed that Interface D would result in the most collisions

due to the lack of specificity of the distance information. However, we believed that Interface D would result in relatively fast time-on-task since the user would eventually come to ignore this unhelpful distance information, and it always takes time to perceive information in an interface.

We hypothesized that Interface E would result in fewer collisions than Interface D due to the exactness of the data presented. However, because the user would have to interpret the numerical data, we felt Interface E would lead to longer run times than both Interfaces D and F. We felt users would perform the best using Interface F due to how easy it is to visually process the information. We hypothesized that Interface F would yield fewer collisions and faster run times than either of the other interfaces. It is very easy to interpret, thus it is extremely easy to recognize if an obstacle is close to the robot without having to expend mental effort calculating distances. Due to the constantly changing numerical values on Interface E, we believed that users might experience cognitive overload and misinterpret the values. With Interface F, we felt this would not be an issue. Even though Interface F's data presentation is still technically not as precise as Interface E's, the fact that the user can instantly know if obstacles are close or not would provide much better surroundings awareness.

We did not want the user to get lost in the arena, so the courses were designed to have exactly one possible path. Because we were primarily interested in which interface resulted in the fastest run times, we did not want the results to be skewed by the users becoming lost in the arena. If a user was confused as to the direction in which to proceed, the test administrator indicated verbally the correct direction. (This was the only information the test administrator gave the operators while the runs were in progress.)

Once again, participants used all three versions of the interface in a within-subjects study design and the order of the interface use was randomized. We collected and analyzed the time it took each participant to finish the task.

### 8.3 Evaluation results

When comparing time on task, our initial hypotheses held true for most of the test cases. Using two-tailed paired *t*-tests ( $df = 17$ ), we found significant differences between the interfaces in the amount of time on task. Interface D (the distance panel with colored boxes) was significantly faster than Interface E (the distance panel with colored boxes and distance values) ( $M_A = 508$  seconds,  $SD_A = 283.6$ ,  $M_B = 635$ ,  $SD_B = 409.1$ ,  $p = 0.02$ ). Interface F (the "Zoom mode"-inspired panel) was also significantly faster than Interface E ( $M_B = 635$  seconds,  $SD_B = 409.1$ ,  $M_C = 495$ ,  $SD_C = 217.8$ ,  $p = 0.031$ ). These results indicate that Interface F was the fastest while Interface E was by far the slowest. We believe this difference is due to likely cognitive loads induced by the two distance panels. Interface E requires many mental calculations to yield important results, whereas no mental number calculations are needed to use Interfaces D or F. We suspect that Interface D performed similarly to Interface F in part because there were no calculations to be done and in part because it provided only vague information. For most of the run, the boxes displayed on Interface D were all red, so users tended to ignore them.

When comparing the number of collisions that occurred, the data supported our initial hypotheses. The number of collisions experienced using Interface D versus E was not significant ( $M_A = 8.78$  collisions,  $SD_A = 3.72$ ,  $M_B = 7.61$ ,  $SD_B = 3.11$ ,  $p = 0.14$ ). However, both of these interfaces resulted in significantly more collisions than Interface F ( $M_C = 6.00$  collisions,  $SD_C = 3.07$ ;  $p = 0.007$  and  $p = 0.041$  for Interfaces D and E, respectively).

Overall, this study provided very conclusive results. We found the data closely matched our initial hypotheses. Interface F had significantly fewer collisions and yielded significantly faster run times than both Interfaces D and E. The total number of collisions that stemmed from this experiment was much larger than the number of collisions seen in our previous studies. As was previously stated, the arenas in this experiment were extremely narrow and operators were only allowed to be in teleoperation mode, so a larger number of total collisions were expected.

One factor that may limit generalization is that this experiment differed more than the studies that were carried out with the other interface versions. In this study, participants traversed a path and then returned along the same path. They did not have to search for victims or traverse a maze, which are challenges that the previous studies presented. As a result, participants may have been more apt to concentrate on the distance panel more than they would have otherwise because there was no threat of missing a victim or important landmark in the video. However, this study still shows that the zoom mode inspired distance panel of Interface F is superior to the previous one. As future work, we would like to conduct a study on Version 3 utilizing tasks more similar to those in the studies of Versions 1 and 2.

With respect to the new distance panel, the majority of the users (11 of 18) preferred Interface F, while six of the eighteen participants preferred Interface E. 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 F, they would like it better. Only one user selected Interface D as being the best, stating that he/she liked how it was less cluttered, but also that he/she was more used to the system by the third run (during which he/she used Interface D). We note that had this participant used a different interface on the final run, he/she may have chosen that interface as the favorite instead.

Three users did, however, say they liked Interface F the least. All three commented that the lines kept changing their distance, which made it hard to track. The sides and rear of the robot use sonar sensors to detect 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 because Interface F is easy to interpret, every shift is noticed. With Interface D, the box will most likely stay the same color, or in Interface E, users may not notice fluctuating numbers as much if they are not looking directly at the panel. We believe this result is related to the quality of the sensor data, rather than the quality of the interface, because if there were laser sensors on the sides and rear of the robot, instead of the sonar sensors, 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 D the most and one user disliked Interface E the most.

About half the users preferred having Interface F in its perspective view, while the other half preferred it in the top-down view, which suggests that the ability to switch between views should be preserved in future versions of the interface. Most users generally chose which view they preferred at the beginning of the run and continued to use it throughout the study. 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 the front of the robot was very close to an obstacle.

## 9. Version 4

### 9.1 Interface description

For this version of the interface, we investigated the impact of a multi-touch interaction device on robot control.

The last few years have shown a great deal of interest in multi-touch tabletop and screen display research. Hardware solutions such as the Mitsubishi DiamondTouch (Dietz & Leigh, 2001), Microsoft Surface, and Touchtable by TouchTable, Inc., have been in low volume production for some time now. Increases in processor and graphics co-processor speeds have allowed for innovative software solutions that now rival the responsiveness of exclusively hardware solutions (Han, 2005).

By removing the joystick, mouse, or keyboard from the interaction, we increase the degree of direct manipulation by removing a layer of interface abstraction (Shneiderman, 1983). In the case of human-robot interaction, this should allow users to more directly interact with the robot and affect its behavior. To our knowledge, this study represents the first use of a multi-touch table with a physical agent. Many unexpected events occur when a system contains a moving, semi-autonomous physical object that is affecting the world. As such, we must determine if multi-touch interaction decreases the performance of systems in the real, dynamic, and noisy world.

To enable a baseline comparison that isolated the effect of the different interaction modality (traditional display device and joystick versus multi-touch device), we needed to ensure that the multi-touch interface was visually identical to Version 3 of the interface. The goal was to duplicate all of the functionality without creating any confounding issues in presentation or arrangement of display elements. Each of the discrete interaction elements is shown in Figure 2 and was previously described in Section 5. Besides the drive control panel, we made no visible changes to the interface.

Despite this attention to visual similarity between Versions 3 and 4, the DiamondTouch was immediately able to provide more functionality than the joystick could alone. For example,

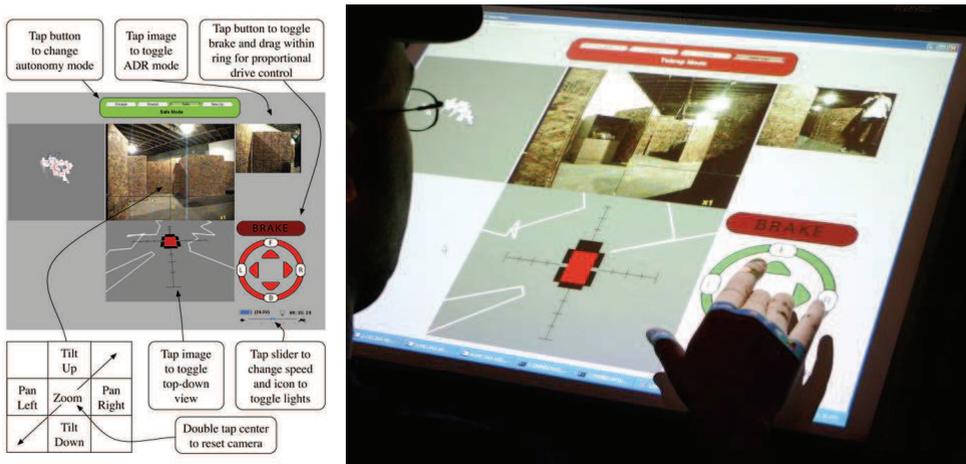


Fig. 2. Screenshot and guide to gestures (left) allowed the user (right) to activate interface features and autonomy modes.

the autonomy mode selection was offloaded to the keyboard in the joystick interface due to a limited number of buttons. In the case of the DiamondTouch, the buttons that were already displayed were used for this purpose. This “free functionality” was also true for the distance panel, speed control, and light control.

## 9.2 Evaluation description

Since the two interfaces make use of the same graphical elements and provide the same functionality, we hypothesized that performance using the two interfaces would be comparable. As in the previous studies, we used a within-subject design with all participants using both the Version 3 and Version 4 interfaces. Participants consisted of six trained search and rescue personnel (four men and two women).

We assessed the positive, or constructive, aspects of performance based on measuring the number of victims found and the amount of new or unique territory the robot covered while traversing the arena. These measurements are related because it is difficult to find additional victims if the operator is not successful in maneuvering the robot into previously unexplored areas. Collisions constituted a measure of destructive performance.

## 9.3 Evaluation results

Participants explored an average of 376 square feet and found an average of 5 victims when using the joystick-based interface ( $SD = 90.4$  and  $SD = 2.3$ , respectively). The DiamondTouch interface shows remarkably similar results: participants directed robots to 373.3 square feet of territory and found 5.7 victims ( $SD = 107.4$ ,  $SD = 2.9$ , respectively). Thus, there is no significant difference in the constructive performance of the two interfaces.

Paired, two-tailed  $t$ -tests ( $df = 5$ ) indicated that there were no significant differences with respect to the numbers of collisions participants made using each interface ( $M_{\text{Joystick}} = 1.54$  collisions,  $SD_{\text{Joystick}} = 4.22$ ,  $M_{\text{Touch}} = 1.92$ ,  $SD_{\text{Touch}} = 2.8$ ). Thus we confirmed that there was no difference in constructive or destructive performance when using the two interfaces.

Now that we know that performance is not degraded by the act of porting the interface to the DiamondTouch table, we can begin optimizing the design for use with multi-touch interaction based on incorporating what we learned from participants' subjective feedback and a detailed understanding of how they interacted with the interface.

To capture participants' preferences, we asked them six Semantic Differential-scale questions. Using a scale of one to five, we asked how they would rate each interface along six dimensions: hindered in performing the task/helped in performing the task, difficult to learn/easy to learn, difficult to use/easy to use, irritating to use/pleasant to use, uncomfortable to use/comfortable to use, and inefficient to use/efficient to use.

Prior to the experiment we conjectured that participants would find the DiamondTouch interface easier to learn and use and to be more efficient. The rationale for the ease of learning is that the controls are more dispersed over the table and incorporated into the areas that they relate to, as opposed to being clustered on the joystick where users must remember what motions and buttons are used for what functions. The predictions for ease of use and efficiency result from our postulation that an interface with a higher degree of direct manipulation will be easier and faster to use.

The DiamondTouch interface scored the same or higher on average in all categories, although four of these categories evidenced no statistically significant difference. We found weak significance using a paired, 1-tailed  $t$ -test ( $df = 5$ ) for ease of learning ( $M_{\text{Joystick}} = 4.7$ ,

$SD_{\text{Joystick}} = 0.5$ ,  $M_{\text{Touch}} = 5.0$ ,  $SD_{\text{Touch}} = 0.0$ ,  $p = 0.088$ ), and efficiency ( $M_{\text{Joystick}} = 3.3$ ,  $SD_{\text{Joystick}} = 1.2$ ,  $M_{\text{Touch}} = 4.33$ ,  $SD_{\text{Touch}} = 0.8$ ,  $p = 0.055$ ) and assert that it is likely we would have attained true significance with a greater number of participants.

We believe that the scores given the DiamondTouch interface with respect to its ease of use suffered because of several implementation problems. Sometimes the robot did not receive the “recenter camera” command despite the fact that the participants were using the correct gesture to send that command, requiring the participants to frequently repeat the recentering gesture. At other times, the participants attempted to send that command by tapping on the very edge of the region in which that command could be activated, so sometimes the gesture was effective and at other times it failed, and it was difficult and frustrating for the participants to understand why the failures occurred. Also, it was not always clear to participants how to form the optimal gestures to direct the robot’s movement.

Because differences in Semantic Differential-scale scores for ease of learning and pleasantness to use were on the edge of significance, we looked for other supporting or disconfirming evidence. We noted that participants asked questions about how to activate functions during the runs, which we interpreted as indication that the participants were still learning the interface controls despite having been given standardized training. Accordingly, we investigated the number of questions they asked about each system during the runs as well as the number of times they showed uncertainty in finding a particular function such as a different autonomy mode. We found that five of the six participants asked a total of eight questions about the joystick interface and one participant asked two questions about the DiamondTouch interface ( $p = 0.072$ ,  $df = 5$  for paired, 1-tailed  $t$ -test). This result, while again being on the edge of significance due to the small sample size, tends to support the contention that the DiamondTouch interface is easier to learn than the joystick interface.<sup>2</sup>

## 10. Conclusions and future work

Through our iterative design and testing process, we succeeded in providing a useful surroundings awareness panel that displays accurate data to the user in an easy-to-interpret manner. In the testing for Version 3, the current distance panel was proven to provide faster run times, with fewer collisions than the previous two versions.

Our results support the usefulness of the guidelines we followed in the creation of the interface. For example, we fused sensor information to lower the cognitive load on the user. Having the laser and sonar sensor values being displayed in the same distance panel provided users a single interface through which to access distance information. Through an iterative process, we gradually improved the distance panel. This panel rotates when the operator pans the camera, which allows the user to line up the obstacle they see in the video with where it is represented in the distance panel, to help reduce cognitive load.

The distance panel also functions as a camera pan indicator. To provide redundant cueing in a location where operators will be naturally focused much of the time, crosshairs are overlaid on the video screen to show the current pan/tilt position of the main camera.

---

<sup>2</sup> For an analysis of the gestures participants used to direct the robot using the Version 4 multi-touch interface, see Micire et al. (2009).

Additionally, we provide indicators of robot health and state, as well as include information on which camera is currently in the main display. Finally, we have shown that the ability to see the robot's chassis improves surroundings awareness. This finding provides strong support for the guideline that states the operator should have the ability to inspect the robot's body for damage or entangled obstacles.

Through this iterative design and testing process, we have also added the following guidelines to enhance the list of previously-reported guidelines:

- Important information should be presented on or very close to the video screen. Users primarily pay attention to the video screen, so keeping important information on or near it makes it more noticeable.
- If the robot system has more than one camera, a second camera should be mounted facing the rear of the robot to provide enhanced awareness of the robot's surroundings.
- If the robot system has more than one camera, the system should include an ADR mode to improve awareness and reduce the number of collisions that occur while the robot is backing up.

Once we completed three iterations of the interface design, we investigated alternative interaction methods. A joystick interface limits the user to a relatively small set of interaction possibilities. The multi-touch surface is quite different, allowing for numerous interaction methods using a large set of gestures on a 2D plane. However, the flexibility of the interface also presents a problem for the designer, who must carefully choose control methods that give clear affordances and appropriate feedback to the user. Users are accustomed to haptic feedback, such as spring-loaded buttons and gimbals, and auditory feedback, such as clicks, even from a non-force-feedback joystick controller.

Nevertheless, the results show promise for the multi-touch interface since little optimization was actually performed during the porting process. In fact, we know that several of the interaction methods that survived the porting process are sub-optimal, and yet performance was not degraded. This research thus provides a good baseline. We are confident that more can be done to enrich the user experience because we no longer are limited to the constraints of the number of degrees of freedom of a joystick. Because this is a software system, it is easier to iteratively tailor the interaction approach using a multi-touch table than when using a joystick. This feature strikes a beneficial middle ground between a software and hardware solution for interaction functionality.

Our future work will focus on the lessons learned from this experiment, particularly designing new versions of the interface that are optimized for the multi-touch display. We will explore direct map manipulation and "point to send the robot here" commands that can provide easier navigation.

## 11. Acknowledgments

Michael Baker, Robert Casey, Andrew Chanler, and Philip Thoren contributed to the development of the robot system. The authors wish to thank the experiment participants. We would also like to thank Harold Bufford, Munjal Desai and Kate Tsui for their data gathering assistance and Kristen Stubbs and Kate Tsui for their editing assistance. Thanks to Mitsubishi Electric Research Laboratories for the extended use of a DiamondTouch tabletop. And finally, thanks to Adam Jacoff, Elena Messina, and Ann Virts at NIST for the use of their facilities for and support throughout the testing. This research was supported in part

by the National Science Foundation (IIS-0415224, IIS-0308186, IIS-0546309) and the National Institute of Standards and Technology (70NANB3H1116).

## 12. References

- J. A. Adams. Unmanned vehicle situation awareness: a path forward, *Proceedings of the 2007 Human Systems Integration Symposium*, 2007.
- D. Bruemmer, D. Dudenhoeffer, and J. Marble. Dynamic autonomy for urban search and rescue. In *Proceedings of the 2002 AAAI Mobile Robot Workshop*, Edmonton, Canada, August 2002.
- P. Dietz and D. Leigh. DiamondTouch: a multi-user touch technology. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 219–226, November 2001.
- J. Drury, B. Keyes, and H. Yanco. LASSOing HRI: analyzing situation awareness in map-centric and video-centric interfaces. In *Proceedings of the ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pages 279–286, 2007.
- J. Drury, L. Riek, A. Christiansen, Z. Eyler-Walker, A. Maggi, and D. Smith. Evaluating Human-Robot Interaction in a Search-and-Rescue Context. In *Proceedings of the Performance Metrics for Intelligent System (PerMIS) Workshop*, 2003.
- J. Drury, J. Scholtz, and H. Yanco. Awareness in human-robot interactions. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 1, 2003.
- M. R. Endsley. Design and evaluation for situation awareness enhancement. In *Proceedings of Human Factors Society 32nd Annual Meeting*, Santa Monica, CA, 1988.
- K. A. Ericsson and H. A. Simon. Verbal reports as data. *Psychological Review*, 87:215–251, 1980.
- J. Y. Han. Low-cost multi-touch sensing through frustrated total internal reflection. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, 2005.
- A. Hjelmfelt and M. Pokrant. Coherent Tactical Picture. CNA RM, pages 97–129, 1998.
- A. Howard. Simple Mapping Utilities, <http://www-robotics.usc.edu/~ahoward/pmap/index.html> Accessed on Oct. 15, 2009.
- A. Jacoff, E. Messina, and J. Evans. A standard test course for urban search and rescue robots. In *Proceedings of the Performance Metrics for Intelligent Systems (PerMIS) Workshop*. National Institute of Science and Technology, 2000.
- A. Jacoff, E. Messina, and J. Evans. A reference test course for autonomous mobile robots. In *Proceedings of the SPIE-AeroSense Conference*. International Society for Optical Engineering, 2001.
- A. Jacoff, E. Messina, and J. Evans. Performance evaluation of autonomous mobile robots. *Industrial Robot: An International Journal*, 29(3):259–267, 2002.
- M. W. Kadous, R. Ka-Man Sheh, and C. Sammut. Effective user interface design for rescue robotics. In *Proceedings of the ACM/IEEE Conference on Human-Robot Interaction*, 2006.
- B. Keyes, R. Casey, H. Yanco, B. Maxwell, and Y. Georgiev. Camera placement and multi-camera fusion for remote robot operation. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, 2006.
- W. Lee, H. Ryu, G. Yang, H. Kim, Y. Park, and S. Bang. Design guidelines for map-based human-robot interfaces: A collocated workspace perspective. *International Journal Of Industrial Ergonomics*, 37(7), 589–604, July 2007.

- B. Maxwell, N. Ward, and F. Heckel. A configurable interface and architecture for robot rescue. In *Proceedings of the 2004 AAAI Mobile Robotics Workshop*, San Jose, CA, 2004.
- M. Micire, J. L. Drury, B. Keyes, and H. A. Yanco. Multi-touch interaction for robot control. In *Proceedings of the Intelligent User Interfaces 2009 Conference*, Sanibel Island, FL, February 2009.
- C. Nielsen and M. Goodrich. Comparing the usefulness of video and map information in navigation tasks. In *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pages 95-101. ACM New York, NY, USA, 2006.
- C. W. Nielsen, M. A. Goodrich, and B. Ricks. Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics and Automation*. Vol 23, No 5, pp. 927-941, October 2007.
- C. Nielsen, B. Ricks, M. Goodrich, D. Bruemmer, D. Few, and M. Few. Snapshots for semantic maps. In *Proceedings of the IEEE International Conference on Systems, Man and Cybernetics*, volume 3, 2004.
- J. Nielsen. *Usability Engineering*. San Diego: Academic Press, 1993.
- J. Scholtz, J. Young, J. L. Drury, and H. A. Yanco. Evaluation of human-robot interaction awareness in search and rescue. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, New Orleans, April 2004.
- B. Shneiderman. Direct manipulation: a step beyond programming languages. *IEEE Computer*, 16:57-69, 1983.
- H. A. Yanco and J. L. Drury. A taxonomy for human-robot interaction. In *Proceedings of the AAAI Fall Symposium on Human-Robot Interaction*, 2002.
- H. A. Yanco and J. L. Drury. Rescuing interfaces: a multi-year study of human-robot interaction at the AAAI Robot Rescue Competition. *Autonomous Robots Journal*, Special Issue on the AAAI Mobile Robot Competition and Exhibition, Dec. 2006.
- H. A. Yanco, J. L. Drury, and J. Scholtz. Beyond usability evaluation: analysis of human-robot interaction at a major robotics competition. *Human-Computer Interaction*, 19(1&2):117-149, 2004.
- H. A. Yanco and J. L. Drury. "Where am I?": acquiring situation awareness using a remote robot platform. In *Proceedings of the IEEE Conference on Systems, Man and Cybernetics*, October 2004.
- H. A. Yanco, B. Keyes, J. L. Drury, C. W. Nielsen, D. A. Few, and D. J. Bruemmer. Evolving interface design for robot search tasks. *Journal of Field Robotics*, 24:779-799, August/September 2007.
- L. Zalud. ARGOS - System for heterogeneous mobile robot teleoperation. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 211-216, 2006.



## **Human-Robot Interaction**

Edited by Daisuke Chugo

ISBN 978-953-307-051-3

Hard cover, 288 pages

**Publisher** InTech

**Published online** 01, February, 2010

**Published in print edition** February, 2010

Human-robot interaction (HRI) is the study of interactions between people (users) and robots. HRI is multidisciplinary with contributions from the fields of human-computer interaction, artificial intelligence, robotics, speech recognition, and social sciences (psychology, cognitive science, anthropology, and human factors). There has been a great deal of work done in the area of human-robot interaction to understand how a human interacts with a computer. However, there has been very little work done in understanding how people interact with robots. For robots becoming our friends, these studies will be required more and more.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Brenden Keyes, Mark Micire, Jill L. Drury and Holly A. Yanco (2010). Improving Human-Robot Interaction through Interface Evolution, Human-Robot Interaction, Daisuke Chugo (Ed.), ISBN: 978-953-307-051-3, InTech, Available from: <http://www.intechopen.com/books/human-robot-interaction/improving-human-robot-interaction-through-interface-evolution>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.