2004-08-31

An Ecological Display for Robot Teleoperation

Robert W. Ricks
Brigham Young University - Provo

AN ECOLOGICAL DISPLAY FOR ROBOT TELEOPERATION

by

Robert W. Ricks

A thesis submitted to the faculty of

Brigham Young University

in partial fulfillment of the requirements for the degree of

Master of Science

Department of Computer Science

Brigham Young University

August 2004

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

Robert W. Ricks

This thesis has been read by each member of the following graduate committee and by majority vote has been found to be satisfactory.

Date	Michael Goodrich, Chair
Date	Bryan Morse
Date	Irene Langkilde-Geary

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of Robert W. Ricks in its final form and have found that (1) its format, citations, and bibliographical style are consistent and acceptable and fulfill university and department style requirements; (2) its illustrative materials including figures, tables, and charts are in place; and (3) the final manuscript is satisfactory to the graduate committee and is ready for submission to the university library.

Date	Michael Goodrich Chair, Graduate Committee

Accepted for the Department

David Embley
Graduate Coordinator

Accepted for the College

G. Rex Bryce
Associate Dean, College of Physical and Mathematical Sciences

ABSTRACT

AN ECOLOGICAL DISPLAY FOR ROBOT TELEOPERATION

Robert W. Ricks

Department of Computer Science

Master of Science

This thesis presents an interface for effectively teleoperating robots that combines an ecological display of range and video information with quickening based on dead-reckoning prediction. This display is built by viewing range and video information from a virtual camera trailing the robot. This is rendered in 3-D by using standard hardware acceleration and 3-D graphics software. Our studies demonstrate that this interface improves performance for most people, including those that do not have much previous experience with robotics. These studies involved 32 test subjects in a simulated environment and 8 in the real world. Subjects were required to drive the robot through several mazes while remembering a sequence of items. People took less time using the ecological interface and experienced fewer collisions, with a much lower workload as measured by joystick entropy. People preferred the interface over a standard interface with side-by-side range and video information and no prediction. Participants tended to rate the interface as more learn-able and more intuitive; participants also felt more confident in the robot's expected behavior.

ACKNOWLEDGMENTS

I would like to thank my wife, our little daughter, Jaide, and my parents for their support. I would also like to thank my advisor, Dr. Michael Goodrich for his advice and input on my research. I would like to thank DARPA for partially funding my research. I would also like to acknowledge my fellow research assistants for their input, help, encouragement and fellowship. Finally, I would like to once again thank my wife, Meg, for making it all worth-while.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

There are many times when robots could be used to do work that would be impractical or unsafe for a human. Examples of this include nuclear waste disposal, interplanetary exploration, and military or search and rescue operations. Teleoperation, literally operating at a distance [44], is the operation of these vehicles or systems from a remote location. This operation can be difficult for a variety of reasons, most of which stem from either (a) the lack of ordinary visual and vestibular[1] cues that help us navigate and locate things or (b) delay. These factors contribute to a loss of situation awareness, which we define as perception of the environment and how a user's actions affect the user and what will happen in the near future [13].

The first factor that impedes situation awareness in teleoperation is the lack of the ordinary visual cues used in navigation. Human vision has an effective field-of-vision of approximately 200 degrees [1]. We use peripheral vision to better sense where we are in our environment and how we are moving in it. Studies have shown that restricting a human's field of view significantly reduces one's ability to navigate in the real world [1, 23]. Other senses are used as well. For instance, we also use gravity to feel which direction is

[1] Vestibular cues relate to a human's sense of equilibrium and balance from the vestibule of the inner ear.

up. When guiding a robot through a cluttered environment with a lot of obstacles to climb over or around, limited field of vision, lack of proprioception, and absence of vestibular cues severely hamper the ability of a human operator to stay oriented in the remote world.

Another hindrance to obtaining proper situation awareness in teleoperation is the delay inherent in a remote link to the robot. Because of communication delays, limited bandwidth, and sensor update times, the human may not see the results of commands sent to the robot for some time. This can be exacerbated by physical distance between the human and the robot or by communication delays inherent in a communication medium such as the Internet. These problems manifest themselves in reduced ability to maintain self-orientation and by misjudging distances to objects.

Additionally, delay can cause instability in the control loop [8], causing the human to overcorrect for errors [55, Chapter 10]. Delay can also dramatically increase the cognitive load on human operators because they have to remember the commands they gave the robot since the last update and extrapolate a new robot position and orientation. This requires the human to mentally connect the new images and sensor data with the last ones. Often users resort to a move-and-wait strategy in order to keep track of where they are and to avoid overcorrection [12].

One common method for dealing with the delay problem is the use of prediction [42]. For example, airplanes use a “tunnel-in-the-sky” to help them stay on their flight plan [35]. This is especially useful on certain vehicles, such as commercial aircraft and big ships [53], because they have higher-order control and significant inertia. For teleoperators with longer communication delay and lower-order control, drawing lines in the last image to represent the new position is usually more helpful. For example, the Mars rover could use this scheme to draw lines representing where the rover has moved and what direction it is now facing [32].

Another type of predictive display, known as a quickened display, has also been used for navigation [55]. The difference between quickening and prediction is that prediction shows the current state of the system and a prediction of what will be happening in the future. Quickened displays, by contrast, only show the predicted future state or error without any representation of what the current state of the system is. The most notable use of a quickened display is in the flight director of many modern commercial airlines, which tells the pilot where to head to stay on the flight plan. The reasoning behind leaving out the current state of the system is that "current error contains no information that is useful for correction" [55, Pg. 409]. It is better to correct for things that we predict will turn into errors than to wait for those problems to occur because this way we can keep the errors from compounding and we can keep out of bad situations.

The most common method for dealing with the narrow field-of-view of most video images is to have range sensors give approximate positions of objects in the the area around the robot. This is typically in a separate display, which the user must integrate with the video for localization purposes [46]. This requires users to divide attention between multiple displays, which increases cognitive load and takes time to learn. Color-coding images to show depth information [49] has also been tried.

We approach the problem of teleoperation from another perspective. Instead of focusing on adding sensors or otherwise giving the user more information, we would like to improve the way information is presented. Most interfaces focus on displaying "what is there." We focus on displaying the information in a manner that may not look exactly like the remote environment, but which affords the same information to the operator.

This thesis will improve the current state-of-the-art in teleoperation by creating an ecological display that helps users visualize affordances in the environment using a camera, laser range sensor and sonar a sonar range sensor. Our interface is based on a first-person perspective of the scene, like what one would see from a camera. However, to improve the visual field-of-view, we pull a virtual camera back from the robot to a fixed position above and behind the robot. Obstacles detected by range sensors are displayed as barrels and the most recent camera image from the robot is displayed in front of the robot. This allows the operator to see the robot and its immediate surroundings as well as what the robot is seeing.

This thesis is organized into three main sections. First, we will discuss previous work and the terms we will use to describe our interface. Second, we will discuss what we have done to create the interface, what makes it novel and why we believe it works better than other interfaces. Finally, we will show how we have validated the interface through user studies and discuss our conclusions, limitations and ideas for future work on the interface.

Chapter 2

Previous Work

2.1 Robot Autonomy

Many methods have been developed to make robots easier to teleoperate. Supervisory control, which involves a human operator supervising somewhat intelligent robots, is one such method. Sheridan [44] is the seminal reference on supervisory control. Many others have worked on supervisory control, safeguarded control [26, 18] and adjustable autonomy [20]. These methods work by adding intelligence to the robot so that the human needs only to give high-level commands such as "go straight" or "turn at the next intersection." The robot uses the high-level commands and sensor data from the environment to steer around obstacles and accomplish objectives given by the operator.

One limitation of these approaches is that final control of the robot is taken away from the human. This limits the robot to the behaviors and intelligence that have been hard-coded into it. There are situations where the operator may know more about the situation than the robot's algorithm does. An example of this would be broken glass or other hazards that may not be detected by the robot's sensors, but which could prove disastrous to the robot. Clearly these situations could be addressed in future algorithms, but developing algorithms can get more difficult with added sensors and situations [22]. A related issue is that adding intelligence to the robot makes it harder to model the robot and study the effects of the interface on its performance. Due to these limitations we are limiting our research to simple robots without autonomous behaviors.

2.2 Enhanced Displays

Some effort has also gone into improving the visual experience afforded human operators. One method is to use a panospheric camera, which gives a distorted view of the entire region around the robot [50], but can be dewarped to look more natural. This has many advantages, including giving the human the ability to visually find and track landmarks. To use such a camera, a high-bandwidth communication channel that allows frequent image updates is usually necessary for users to maintain continuity between images. Additionally, demonstrations of the panospheric camera have been limited to high-level navigation in the middle of the desert, so some intelligence or other control scheme may be still be required for cluttered environments. An alternative to panospheric cameras is to use multiple cameras, which has been done by David Woods[1] and Stephen Hughes et al. [25]. This may help operators better understand what is all around the robot, but it still requires fast communications to overcome delay. We are restricting attention to robots with a single forward-looking camera.

Other methods which have been used to improve teleoperation include multisensor and adjustable autonomy interfaces [15]. Multisensor [49], or sensor fusion [33] displays, present information from multiple sensors in a single, integrated view [16]. Adjustable autonomy [20] systems[2] are another way to approach the shortcomings of teleoperation yet

[1] Personal communication with David Woods, a Professor at Ohio State University who studies human factors in technology [56, 57], in May 2004.

[2] Multimodal interfaces have referred to both (a) interfaces that present information through multiple

retain its benefits. The basic idea is to choose the autonomy mode based on the situation at hand; it may be beneficial to control the robot with direct teleoperation at times and have the robot fully autonomous at other times [6]. Other control schemes have been introduced to give a robot basic teleoperation instructions and receive feedback from the robot [15], but these tend to focus more on using other devices to give instructions to the robot rather than focusing on making it easier to control the robot, perceive the remote environment or improve situational awareness. Examples of these control modes include gestures, web-based controls and buttons on a PDA.

An example of an adjustable autonomy-based interface is VEVI which was used for volcano exploration [17] with the Dante II robot. This robot could be controlled by moving each leg individually or giving it a path to follow. The Dante II mission also made use of a virtual reality environment that indicated crucial parameters such as the pressure on each leg and the height of the ground around it. VEVI has also been used to control indoor robots, and an underwater vehicle [23].

Yet another way to enhance teleoperation is to use VR to create a sense of presence. For example, Nguyen et al. [39] use a virtual reality interface for robot control. They used 3D terrain models from stereo images to display a terrain map of the surrounding landscape. VR-based interfaces can use a virtual environment to display information about robots in an intuitive way. For example, a virtual terrain map could be shown beneath a virtual robot or colors in the virtual model of the robot could indicate stress levels on the robot's legs.

2.3 Predictive Methods

Prediction has been used to give the operator an idea of where the robot will move based on current and previous operator input [26, 32]. Prediction usually involves drawing lines

sensory channels [21] and (b) interfaces that allow multiple interaction schemes [15].

in the previous image to show a new position and heading. Studies by Mica Endsley at SA Technologies have shown that predictive displays can increase performance and situation awareness in the Air Traffic Control domain [14].

Distributed simulations, such as NPSNET [29, 4] and computer games, make use of prediction to present a visually appealing environment. The technique they use, referred to as dead-reckoning, allows client computers to simulate a client's view of the world locally in the absence of "true" information from the server [5]. This is done either through extrapolating only from user input [27], local simulation or information sent from the server [48]. Network traffic can thus be decreased while giving users a smoother experience. While this does not give the user a completely accurate picture of what is actually happening, they give a "reasonable approximation of a shared reality" [48]. Another benefit is that users are able to see results of their actions immediately, even though they will not affect the "real" world for some small amount of time. Users have found this system to be far superior in terms of susceptibility to latency and intuitive feel of the virtual environment. The virtual world should be more accurate at the current time than simply displaying the last "true" positions from the server.

Computer simulations such as video games have also worked to eliminate jerkiness associated with receiving only periodic updates from a simulation server or other participants. Computer games have had similar prediction techniques [5, 27] for some time and the gaming industry has been pushing the limitations of Internet technology and real-time 3D [48]. Some of these ideas have also been used on simulations of spacecraft docking [12], and in augmented reality displays [47] for virtual presence. It appears that very little research in prediction has been applied to simple land robots with ordinary hardware and displays.

2.4 Teleoperation and Telepresence

Many of the terms used to define robotic interfaces are defined in different ways by different people. For example, we define *teleoperation* to be control of a robot or actuator in a real environment, which may be at some distance from the operator, through some sort of communications medium. This means that one can teleoperate a wheeled vehicle, a robotic arm, or a human-like robot with arms and legs. This definition is much less restrictive than what some people refer to as teleoperation which is having a remote mechanism mimic the actions of the human operator. Similarly, teleoperation can be based partially on how similar the individual's virtual body is in appearance or functionality to the individual's own body [52]. While this latter approach may contribute to one's sense of physically being in a remote location, we believe it is far more important that the connection between an individual's actions and the effects of those actions be simple and intuitive [52]. Since many of the terms we use have been defined so differently by different people, we will use definitions that tend to be generally applicable to computers and robotics.

We define *telepresence* as the sense of being in an environment in which one is not physically present. Note that this environment does not necessarily have to be at the remote site or even an actual physical location. This is less restrictive than Sheridan's definition [44] because one does not have to feel present at the remote site. This makes our definition of telepresence more similar to Steuer's definition [45] which allows telepresence to refer to a "real" environment or a "non-existent virtual world". For more information about the various definitions of telepresence, see [30, 31]. Telepresence is important because many believe that increased telepresence will increase performance on various tasks. Schloerb takes the opposite approach and defines objective telepresence as the probability of successfully completing a specified task [41]. Other methods of determing presence usually involve questionnaires, which can be misleading [52].

2.5 Virtual Reality

Virtual environments immerse the user in a synthetic environment [2] which is composed completely of artificial or computer-generated elements. Our interface uses elements of virtual environments to allow them to view information from a 3-D virtual perspective Other ways to refer to virtual environments are virtual reality (VR), artificial reality, and virtual worlds. Virtual reality is often seen as being more restrictive than virtual environments. Some argue that virtual reality must involve immersive 3D graphics and data gloves. Others give much broader definitions of virtual reality [45] which include things ranging from telephone calls to video games. Our interface could be categorized as a virtual reality display in this sense because it creates a virtual world that affords the same behaviors as the real world. However, since virtual reality and virtual environments usually refer to interfaces that control simulated entities that have no direct real-world counterparts, we do not classify our display as a virtual reality interface.

We are not using a strictly virtual environment, so this makes it a *mixed reality* interface [34]. There are two generally accepted sides of mixed reality: augmented reality and augmented virtuality. The difference is that an augmented reality (AR) display presents images which are primarily composed of real environments, but which have been enhanced or augmented by computers [11]. Additionally, augmented reality usually refers to either specific technologies or computer-generated images registered in 3-D with a real scene [2]. Since our interface does not fit the latter half of the definition, we would classify it as an augmented virtuality display. Augmented virtuality refers to virtual environments which have been enhanced or augmented by inclusion of real world images or sensations [11].

2.6 Ecological Definitions

The real problem with all of these definitions is that they focus on the accuracy with which an environment is presented instead of focusing on communicating effective environmental cues. Our approach is based more on J. J. Gibson's *ecological* approach to realism [19, 30, 59]. In the ecological approach there is no need to distinguish between real and virtual worlds because valid perception is that which makes possible successful action in the environment [19]. Similarly, in this ontology, "presence is tantamount to successfully supported action in the environment" [59]. While this sounds similar to Schloerb's definition of telepresence [41], successfully supported action is achieved when actions are perceived as lawful or commensurate with the response we would expect in the real world [59] whether or not this action satisfies objective performance criteria.

One approach to applying the ecological approach to interface design is known as *Ecological Interface Design* (EID). This methodology has been used to make complex processes such as thermal-hydraulics [54], network management [7], medicine [43] and nuclear reactors [58] easier to use. The goals of this approach are to exploit operators' perception and action capabilities while supporting problem solving activities. The basis of EID is to represent the work domain as an *abstraction hierarchy* [40]. This would include layers such as purpose, function and physical form. For example, the purpose of a nuclear power plant is to safely generate power. Its function involves fuel, water, steam, turbines, etc. to fulfill its purposes. The physical components that perform these functions make up the physical form.

Each level of abstraction provides more detail about the makeup of the individual system, yet information can be provided at each level of the abstraction. So, for example, an operator could get information about how much electricity the plant is generated, the temperature of cooling water, or whether particular components are working properly.

Domain-specific information is integrated into this hierarchy to help operators understand the relationships between system variables. Additionally, “information needed to cope with events which are unfamiliar to operators and which have not been anticipated by designers” [54] must be identified and included. For complex domains, where an abstraction hierarchy is probably necessary, EID is very useful.

Another ecological approach to interface design is to build *ecological displays*,- which simply focus on communicating effective environmental cues. This kind of ecological display has been used to help aircraft taxi safely and effectively [24]. Examples of environmental cues used to help aircraft taxi are markers to identify taxiway centerlines and edges. This can be enhanced with global awareness in the form of map displays and traffic information. These types of displays allow users to navigate effectively when some of the normal environmental cues may be missing, for example during low-visibility conditions or when doing teleoperation. Additionally, enhancing normal ecological cues with global information gives the operator higher situational awareness to improve performance in good conditions. Our interface certainly qualifies as an ecological display since it focuses on communicating effective environmental cues instead of depicting what can currently be observed through various sensors.

Chapter 3

Innovations

Our interface consists of an ecological display that helps users teleoperate a robot. Teleoperation is difficult because of delay and the lack of normal ecological cues, such as perspective. Our interface helps people combat these problems by (a) using 3-D symbols to represent known obstacles, in 3-D along with a representation of the robot, (b) including camera information, and (c) quickening the position of obstacles and image data through the use of simple prediction. This ecological approach integrates many of the ideas people have used in teleoperation into one display.

Our interface uses augmented virtuality to help users feel telepresence even if the virtual environment is not exactly like the remote environment where the robot is located. The first element we have added to our interface is 3-D representation of where obstacles are in relation to the robot by placing virtual barrels in a 3-D display. This is combined with a simple 3-D representation of the robot so it is easy to tell how big the robot is in relation to gaps between the obstacles visible in the display. Obviously the robot will rarely be situated in a room full of barrels, but the relationship between the barrels representing objects and the robot facilitate successful action because people are used to this sort of spatial navigation. Another way in which successful action is supported by the interface is that it allows the user to interact with the environment in real-time by use of prediction. The only element of the display that is made up of real-world elements is the camera image sent from the robot. This information is included because it helps users match up the virtual elements of the display with corresponding elements of the remote environment such as walls and doors.

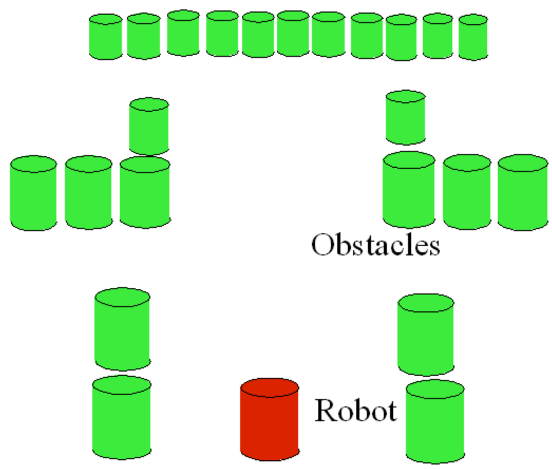


Figure 3.1: Barrels show what is to the sides of the robot.

One ecological cue that teleoperation interfaces often lack is the sense of knowing what is directly to the sides of the robot. Related to that is scale ambiguity: knowing how big the robot is in relation to the objects detected by the various sensors. For example, robots operating in the World Trade Center complex after September 11, 2001 were hindered by operators' lack of perceptual cues as to whether the robot would fit between obstacles [57]. This is somewhat mediated in our display by the bird's-eye perspective a little above and behind the robot seen in Figure 3.1. This perspective is possible because the sonars and lasers are integrated into the display in 3-D as barrels allowing the operator to see the robot in its environment, or at least what the sensors say.

Another ecological cue that is often missing from teleoperation interfaces is the sense of *immediacy* in the interface due to delay between actions and their effects on the world

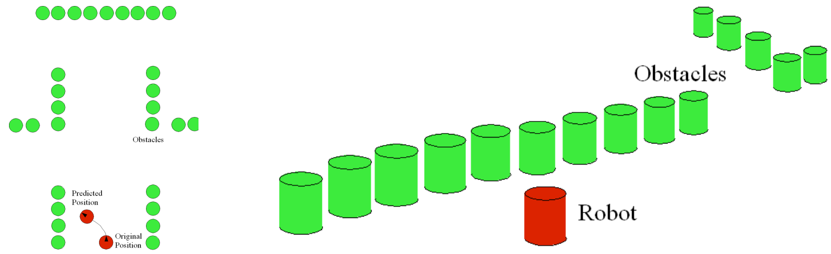


Figure 3.2: The display shows where the robot will be in the ecological display. The current control input will make the robot move forward and turn left. The left image shows what this would look like from the top. The right image shows what this would look like in a quickened perspective display.

around them. Delay between action and perception has been found to have a profound effect on performance [59]. We allow a closer coupling between action and perception by quickening information received from the robot. This prediction is achieved by using simple dead-reckoning using the position of the robot at the last sensor update and the control inputs sent from the interface to the robot(Figure 3.2). Since control inputs should have a deterministic and predictable effect on a teleoperated robot, modeling this predicted movement is a feasible and effective way of increasing ecological presence in the interface.

3.1 Driving in 3-D Barrel World

Teleoperation interfaces often severely limit the visual information to the operator. This has been described as looking at the world through a 'soda straw' [57]. Because of this, streaming video is often not enough for navigation [10]. According to Wickens, for navigation tasks such as driving, egocentric displays are usually best [55]. This is why we use the perspective illustrated in Figures 3.1 and 3.3 instead of a top-down view like the one shown on the left of Figure 3.2. Many car racing or driving simulators for computers (e.g., computer games) utilize a fixed perspective a little above and behind the car. This allows the operator to view objects and obstacles to the side of their car as well as in the front.

This kind of perspective is even better for navigating through objects [55].

Our interface uses a display that uses a fixed perspective behind the robot, as shown in Figure 3.3. We built the display to use DirectX hardware acceleration to render a representation of the robot and obstacles in real-time. The obstacles we are displaying represent distances and angles detected by the robot's range-finding sensors. Both the robot and the obstacles are drawn by texture-mapping a metal texture onto a 16-sided barrel. These are presented in three dimensions with the robot at the front of the display and other objects projecting out in front of it as shown in Figure 3.3.

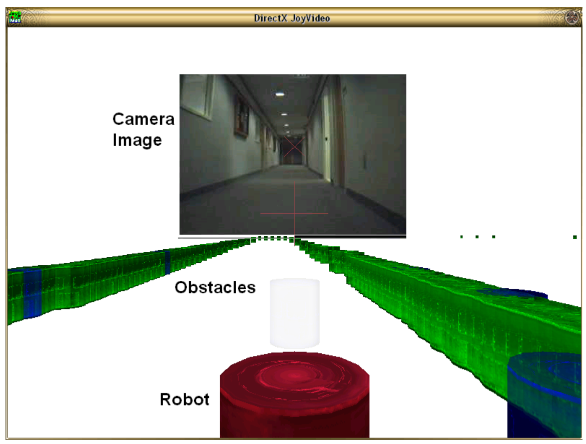


Figure 3.3: The interface draws 3-D barrels to represent obstacles in the world.

There are a number of ways that the display gives the operator information over the channels they are accustomed to using. The texture mapping and 3-D perspective give the operator range cues to determine how far away obstacles are. The size of the barrel representing the robot in comparison to spaces between range barrels help operators see if they can fit through gaps. Because the center of each range barrel is determined by the distance returned by a range device, there may be a little more room between the robot and obstacles than it would appear in the display. This allows for a little bit of a buffer or safety margin built into the display. These kind of visual cues are directly perceived by the operator because they are similar to cues that humans rely on to navigate [55].

In order to differentiate between the types of information presented in the display, we use colors to show which barrel is the robot, which barrels come from information provided by the laser range-finder and which barrels come from SONAR information. The barrel representing the robot is colored red because that is one of the primary colors on our Pioneer2-DXe mobile robot. All the barrels were designed to be approximately the size of the actual robot which gives the robot a little bit of cushion around each reading and makes the readings appear substantial. The barrels representing SONAR information are blue because that blends in with the black background which is supposed to communicate that they are less accurate. The barrels representing laser range-finder information are green because they are more accurate.

Unfortunately this color scale may not be intuitive to everybody and some people may have trouble differentiating between the red and green barrels. Because the robot stays in a fixed location in the display, however, it is not too hard for people to tell it apart from the laser barrels. Additionally, the difference between laser information and sonar information is usually not critical. Ultimately, the addition of color coding helps experienced users better understand the state of the world at a glance while still being accessible to novices.

Obstacles may be detected by multiple sensors, in which case barrels from both sensors show that information. The barrels are somewhat transparent which allows the operator to see through inaccurate sensor readings and readings close together reinforce the idea that there is a substantial obstacle at a particular point. Since range measurements are not exact, even if SONAR and laser detected the same object, there is a good chance that the two readings would differ by a small amount. This makes the readings show up as two separate barrels, one in front of the other, instead of combining colors. Since the barrels are approximately the size of the robot, barrels from the laser may overlap with each other. While we could account for this and space out the barrels, having the barrels overlap makes the display look less like a bunch of discrete barrels and allows users to naturally identify continuous surfaces and object sizes. There are some things, like glass, that SONAR picks up better than lasers. Other things are more readily identifiable by the laser range-finder, such as walls. Since the sensors complement each other to some degree, it is beneficial to display both sets of barrels.

In addition to the barrels, the most recently received image from the robot's camera is displayed in front of the robot. This is texture-mapped onto a rectangle, the bottom of which is approximately at the vanishing point of the barrels in the display. This allows the operator to navigate in the barrel world while looking at a projection of the camera image in front of the robot in the barrel world. We will present how quickening affects the image in Section 3.4, but first we describe how joystick activity is translated into robot movement.

3.2 Joystick Control

Ten times per second the interface sends a joystick movement command to the robot. This command includes a forward velocity, angular velocity and a timestamp. The mapping between joystick positions and movement commands is illustrated in Figure 3.4. In addition to being sent to the robot, the command is stored in a queue in the interface program. Because of bandwidth constraints, the robot may send back image and range data at a rate that differs from the rate of control inputs it receives due to robot processing power or bandwidth. Information packets from the robot include the timestamp from the last joystick command received by the robot. The command queue and time-stamped messages are used to manage how the display predicts the robot's current perspective. In order to predict where the robot will move since the information was last received from it, we no longer need nodes in the command queue that were processed by the robot before the latest update from the robot. Timestamps from the robot are used to manage the command queue; commands issued earlier than the last one the robot sent can be pruned. These unnecessary nodes are thus removed from the command queue when new information makes them obsolete.

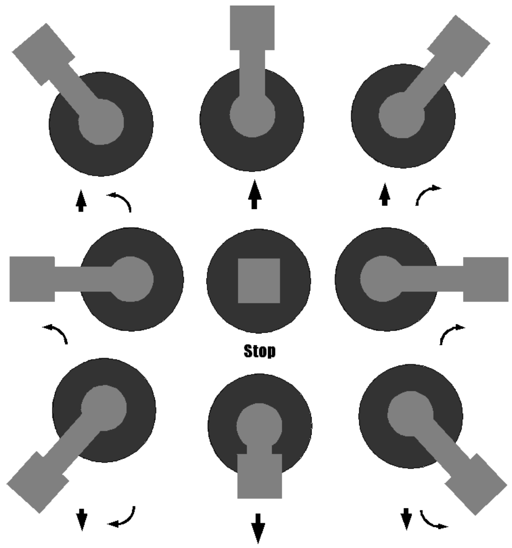


Figure 3.4: Joystick mappings used to translate joystick positions into forward and angular velocity. The black circles represent the base of the joystick and the gray line and squares represent the position of the joystick relative to the base. A representation of a corresponding forward velocity and angular velocity is shown beneath each joystick position.

3.3 Prediction used for Quickening

Because of delay inherent in communications, there is a short and possibly variable period of time between when a command is sent to a robot and when the effects of those commands are seen by the user. This delay is made up of three parts as shown in Figure 3.5. First, it takes some amount of time for commands to travel from the user to the robot due to communications delay. Second, the robot must act on this command and then send back new sensor information. Third, communications delay will once again slow the receipt of this new information. In order to effectively teleoperate a robot in the presence of even relatively short delays, the user must be able to either predict how past commands will affect the current and future state of the robot, or employ a move-and-wait strategy to compensate for it [12].

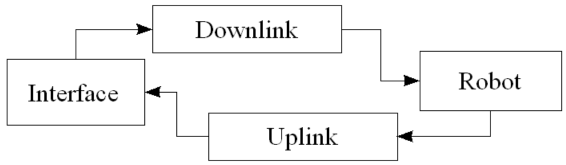


Figure 3.5: Simple representation of teleoperation delay.

3.3.1 Timing

A better way of handling delay than the simplistic model of Figure 3.5 is to send commands at a rate independent of the uplink channel rather than forcing users to wait for the results of their actions to send new commands. This could be a fixed rate or a variable rate, but we focus on a fixed rate model for simplicity. Figure 3.6 shows the modified downlink channel. Now several commands are sent and the robot simply processes the last one it has received. In the figure the interface is currently at time t_0, the current time. One time step ago it sent a command with a velocity, V_x and angular velocity ω to the robot. Along with this command it sent a timestamp for time $t-1$ timesteps which we will call t_1. There are a total of three commands in the downlink channel which have not been received yet by the robot. The times these commands were sent were $t-1$ timesteps, $t-2$ timesteps and $t-3$ timesteps so they are shown with timestamps t_1, t_2 and t_3 in the figure, respectively. The robot is operating on the command t_4, which was sent 4 timesteps ago.

There may also be sensor information which has been sent by the robot but not yet received by the interface. If the robot were to send sensor information at the current time, t_0, it would include the timestamp t_4. This allows the interface to know what command is currently active on the robot. In Figure 3.7 the last sensor information in the comms channel

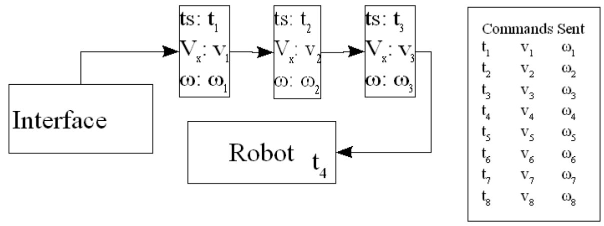


Figure 3.6: A better representation of the downlink portion of the command queue.

was sent at about 2 timesteps ago. The robot was acting on command t_6 at that time. The last sensor information received by the interface included the timestamp t_8, as indicated by the t_8 in the Interface box. In this example the robot sends fewer sensor updates on the uplink than it receives commands on the downlink; the boxes with timestamps t_5 and t_7 do not have any sensor data included with demonstrating that no new sensor information was sent by the robot when those commands were active.

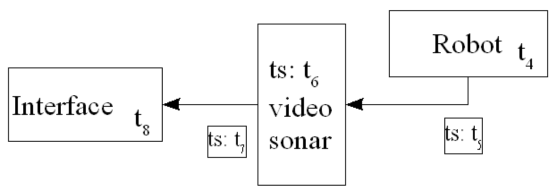


Figure 3.7: The uplink portion of the command queue.

In order to help the user make informed decisions, the robot's new position should be predicted from all the commands whose full effect have not been seen in the interface. As a reasonable approximation, we assume that commands are executed on the robot for the amount of time between when the command was sent from the joystick process and the time the next one was sent and that no command issued before the timestamp received in the latest sensor update will affect the robot. Prediction is accomplished by extrapolating where the robot will be after executing the commands currently in the command queue of the interface. The most recently issued command is handled a little differently than the others. Prediction based on the most recently issued command uses the amount of time since the command was sent to the robot instead of the amount of time we predict it will be processed on the robot. This allows the predicted position to move linearly instead of jumping to a new position every time we send a new command to the robot.

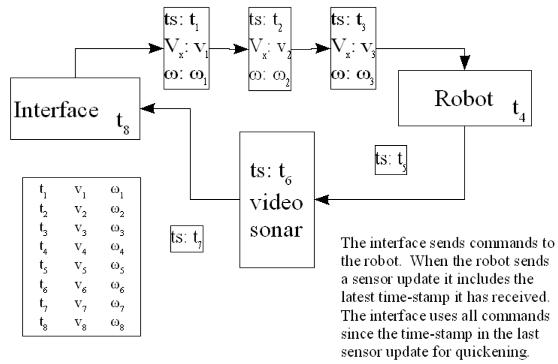


Figure 3.8: Commands sent to the robot that still need to be used for prediction.

Figure 3.8 shows a representation of how the full command queue works. The last command sent to the robot was (v_1, ω_1). This command and the two previous ones have not yet reached the robot. The robot is executing the last command it has received (v_4, ω_4). The last time the robot sent sensor information to the user the robot was executing (v_6, ω_6). The interface is displaying sensor information the robot sent when it was executing the command sent at timestamp t_8. This means that no sensor update was sent while the commands at timestep t_5 or t_7 were executed. Quickening uses the control inputs given at times t_8 through t_1, which correspond to the last eight timesteps, to predict where the robot will be. The prediction method used to translate the control inputs in the command queue to the future position of the robot is discussed below.

3.3.2 Movement

The first step in predicting where the robot will be is deciding where it currently is. Since we are only visualizing the objects detected by the robot's sensors in its last update, we basically generate a visual local occupancy grid with the range data. Local occupancy grids have often been used in mapping and localization [51]. We use only the latest sensor scans to build the occupancy grid; no attempt has been made to integrate past information, maps, or moving object detection [3] into the display. This means that we only need a local representation of robot movement in the prediction algorithm. For this reason, we only need to calculate change in x position, y position and heading for our predictions. This is especially easy if we define the pose of the robot when it sent new sensor information as the origin, and its heading as zero (Figure 3.9). Figure 3.10 shows the position of the robot at $(0, 0)$ heading directly out the x-axis (0°).

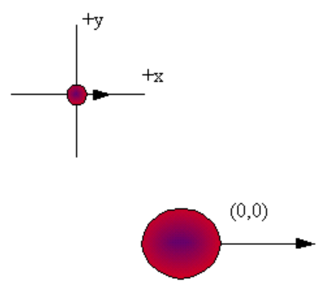


Figure 3.9: Initial position and heading in robot-centered coordinates.

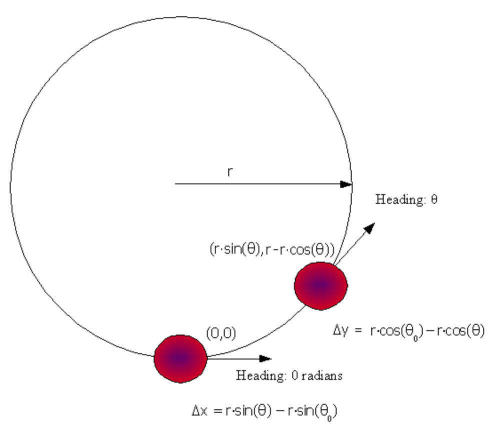


Figure 3.10: Ideal Prediction is based on traveling around a circle.

Since movement commands sent to the robot consist of a desired translational velocity and a desired angular velocity, dead-reckoning predictions are fairly easy. We start out with the robot at the origin of its local environment with respect to where the latest sensor information was obtained from the robot. The robots we are using use translational velocity (V_x) and angular velocity (ω). From this, the change in x position, Δx, change in y position, Δy and change in heading, $\Delta\theta$, can be calculated. When given a non-zero angular velocity, the robot will follow a circular course. If this course were followed for long enough, the robot would make a complete circle. The radius of this circle is proportional to the forward velocity, V_x, and inversely proportional to the angular velocity, ω. If the radius of the circle is fairly small, we can use the starting position on this circle and the ending position on the circle (see Figure 3.10) to calculate change in robot position:

$$\begin{aligned} r &= V_x/\omega \\ \Delta\theta &= \omega\,\Delta t \\ \Delta x &= r[sin(\theta_0 + \Delta\theta) - \sin(\theta_0)] \\ \Delta y &= r[cos(\theta_0 + \Delta\theta) - \cos(\theta_0)]. \end{aligned} \tag{3.1}$$

Since commands in the command queue could have different desired velocity and angular velocity, each node in the command queue could use a different size circle, (see Figure 3.11). This is acceptable because we can simply append an arc from one circle size onto the arc generated from the last command node. Using this method, we iteratively update $\Delta\theta$, Δx and Δy, using the previous values for θ_0, x_0 and y_0. Each prediction stage uses the velocity, angular velocity and the amount of time the command was active for V_x, ω and Δt. The new values of Δx, Δy and $\Delta\theta$ are generated from that command, and this is repeated for the next command. Since the ecological display can be updated much faster than sensor updates are received, we smooth the trajectory on the current command input by using the amount of time the command has been active for Δt. The display currently renders the ecological representation approximately 60 times per second.

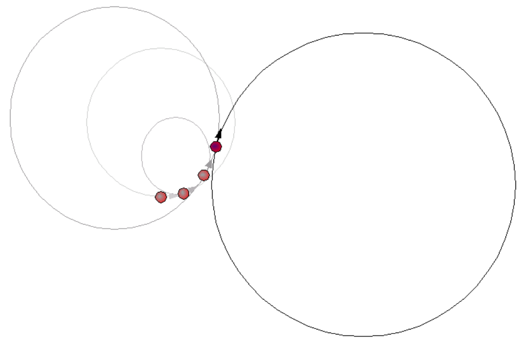


Figure 3.11: As V_x and ω change the robot follows arcs from circles of various sizes.

If the robot is not turning very quickly, ω will be very small, which can lead to significant floating point error. A simpler formula for dead reckoning is thus used when $|\omega|$ is small (less than $\pi/30$); these formulas were adapted from simple straight-line formulas [28]. Using these formulas, we first calculate the displacement of the robot, $\bar{s}$, and the amount the robot has turned, $\Delta\theta$. The other measures are related to the sine and cosine of the amount the robot has turned. The straight line approximation is as follows:

$$\begin{aligned}
\bar{s} &= V_x \, \Delta t \\
\Delta\theta &= \omega \, \Delta t \qquad (3.2) \\
\Delta x &= \bar{s} \, \cos[\theta_0 + (\Delta\theta/2)] \\
\Delta y &= \bar{s} \, \sin[\theta_0 + (\Delta\theta/2)].
\end{aligned}$$

Assuming the robot is traveling on a circular path, the angle between the initial heading and the line between where the robot was initially and its final position should be half the change in heading of the robot, assuming it has turned less than $360°$. For example, if the robot has turned $90°$ it has gone $1/4$ of the way around the circle. If the original position was at the origin, facing $0°$, the new position would be along the ray $45°$ from the origin (see Figures 3.10, 3.12). So, depending on the forward velocity, the new position would be at $(1,1)$ or (π,π), etc. Of course, there would be significant error in the fact that the robot has taken a curved path, instead of a straight path for the distance it has traveled. This error goes to zero as the change in heading goes to zero, however, which is why we use these formulas only when $|\omega|$ is small.

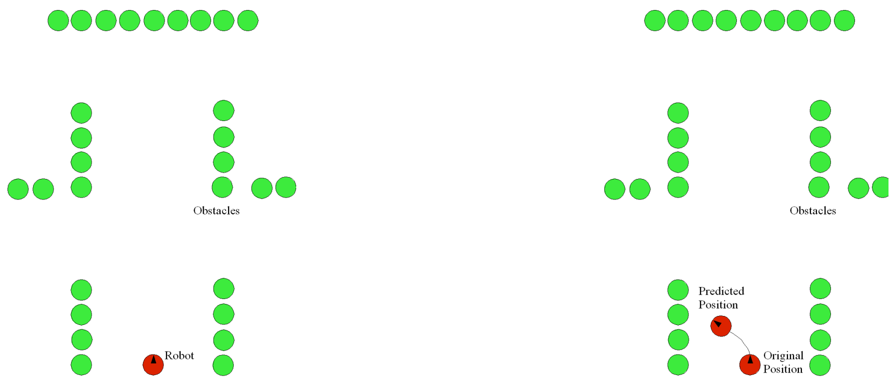


Figure 3.12: The robot moves forward and turns left about 70°.

3.4 Quickening Barrel World

Once we have extrapolated a new robot position, quickening the display is relatively easy. Since we are using somewhat of an egocentric display, as the robot moves we see the world move around the robot instead of the robot moving in the world. The display is designed to represent what the world will look like when all the commands sent to the robot have been followed. This includes moving the objects in the world as well as the last image from the robot in a way that depicts what the robot would see from the predicted location. In between the times when joystick commands are sent to the robot, there is a smooth transition that eliminates some of the "jerkiness" in the display caused by infrequent sensor updates.

Figure 3.13 shows what happens in the display when the interface predicts a new position for the robot. On the left is what the display would look like if the robot stayed

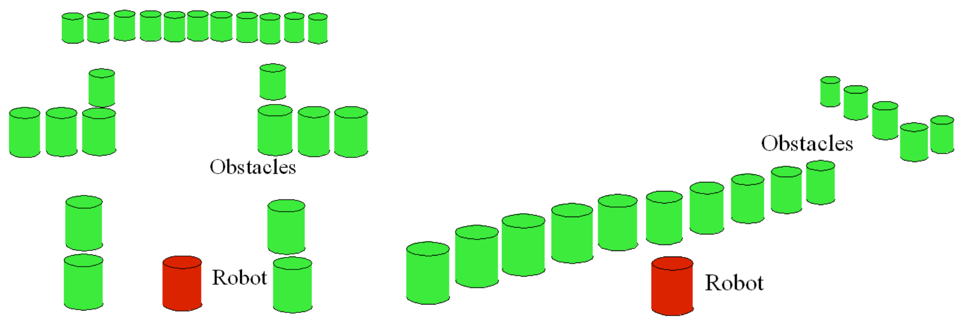


Figure 3.13: Perspective representations of the world from the two positions in Figure 3.12.

stationary like the robot on the left side of Figure 3.12. The darker barrel toward the bottom of the display represents the robot and the lighter barrels represent range information shown in perspective. In the right of Figure 3.12 we see that the robot is moving forward and turning about 45°. The right side of Figure 3.13 shows what this looks like in a perspective display. We now see a wall in front of the robot because it has moved closer to and turned toward the left wall. We can also see part of the first opening on the left to the right of the perspective display. In the perspective display of Figure 3.13 it is much easier to see where the robot is headed and which way to turn the robot than the exocentric display shown in Figure 3.12.

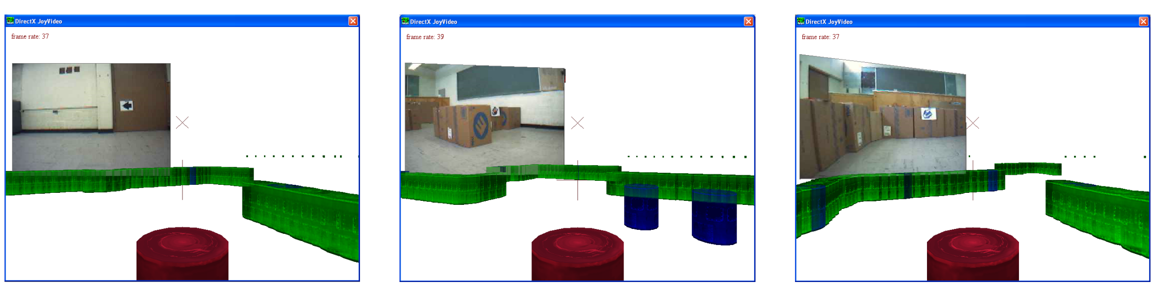


Figure 3.14: Comparison between the camera image sliding (left), rotating 1/4 as much as a flat image plane (middle) and a flat image plane (right). The left image looks like the robot slid to the right. The right image looks distorted because the image plane is viewed from too steep of an angle. The middle image represents a compromise between the two.

The camera image represents a section of the world depicted in the virtual environment. As the robot moves, the camera image also needs to move to show what area that camera image represents. Unlike obstacles generated from range sensors, we do not know the distance to the objects in the camera image nor do we know what objects are closer than other objects. As a rough heuristic, we estimate that the camera image is an image plane approximately 500 centimeters from the robot[1]. As the robot moves in the virtual environment, this image plane gets closer or farther and may move to a new position in the display due to translation and rotation. In order to keep the robot from running through the image plane, the plane moves asymptotically closer to the robot, but never so close or so far that the camera image cannot be seen. As we will demonstrate when we report the experimental results, this approximation produces a satisfactory user experience. When the robot turns a fair amount, such as greater than 30°or 45°, the image may no longer be visible because we may now be looking at the image plane from an oblique angle. This can warp the image so it is much taller than it is wide because we are looking at it from the side. In order to keep the image from being distorted and because we have no external knowledge that we are looking at a flat plane, we rotate the image plane 3/4 of the way back to normal (see Figure 3.14). This amounts to a compromise between viewing the image as a plane and viewing it as the inside of a cylinder. We validate the usefulness of this compromise as part of the entire display in the user studies but note that future work should consider user sensitivity to the rotation.

This quickened display enables the user to interact with the world as if he or she were controlling the robot in real-time despite the delay, as long as the predictions are reasonably accurate. This should mitigate much of the need for the move-and-wait strategy, a commonly used method for dealing with delays, making the robot more controllable. The ecological display and prediction together should improve users' situational awareness, decrease mental load and improve their performance.

[1]This is a heuristic which has been tuned to display a good sized image and move intuitively. Future work could improve this by correlating laser distances with the heuristic value.

Chapter 4

Validation in Simulation

In order to validate that the display does in fact increase performance, we did a series of studies in simulation. We also ran tests using a conventional interface as a baseline. (See Figure 4.1 for screenshots of the two interfaces.) The standard interface shows the latest laser, sonar, and camera information from the robot, along with some buttons and a map that were unused for these tests. This interface was chosen as a standard because it is similar to most current teleoperation interfaces and because the ecological display was designed to work with this interface already. This allowed the exact same control code to operate the robot regardless of which display the user was looking at. The ecological display shows the laser and sonar information as barrels and the camera data in front of the robot. The amount of delay between when subjects gave the robot a control and when the results of that action could be perceived was set to be approximately one second. This required the operator to figure out what the robot would be doing but did not require them to adopt a “wait and see” approach.

In our experiments, each test subject did eight tests, each of which took approximately four minutes. Tests consisted of using both the standard display and the ecological display on a simulated robot. Each test consisted of guiding the robot through one of the four mazes

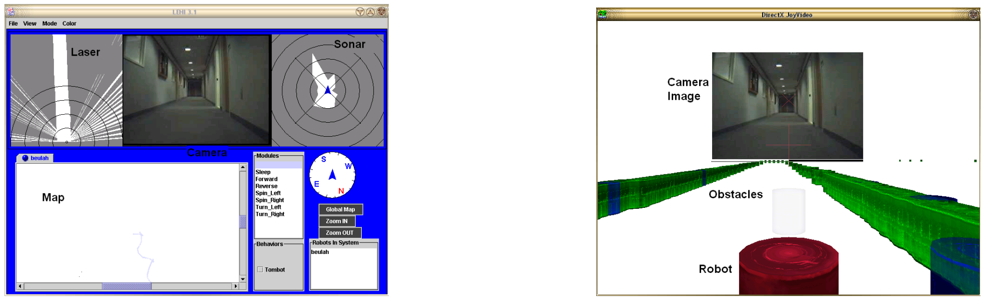


Figure 4.1: The two interfaces we compared in our study. On the left is the standard interface, the ecological interface is on the right.

in Figure 4.2. The display that the test subject used first and the order of the mazes was chosen randomly, but with the constraint that approximately the same number of people would be in each group.

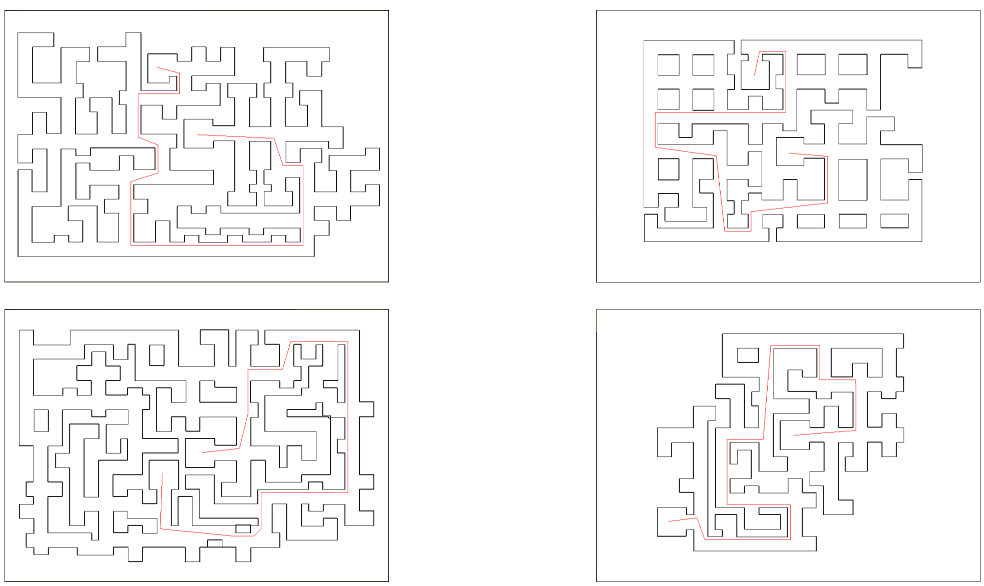


Figure 4.2: These are the maps we used to validate the interface in simulation. The map in the upper left was used for training. The map in the lower right was used both with and without the memory task.

Each subject was given as much time as they needed to "feel comfortable" with each interface. We measured this time to get an idea if the ecological display would be easier to learn than the standard interface. The robot was then placed in the start position of the first maze. The subject had to guide the robot through the maze following a pre-planned path to the end. In these simulated world experiments, the path was given to the operator as a series of red dots in the camera images and verbal instructions about what to do when they reached the next dot.

In addition to completing the mazes with the robot, a memory task was devised to try and determine the amount of working memory required to use each interface effectively. Other researchers have found that ecological displays often require less working memory [36]. The memory task worked as follows. Before each maze subjects were given a list of 5 images or words to memorize. The test subject had at most one minute to memorize these before the program would continue. After completing the maze, which was designed to take about 4 minutes, subjects were required to select the objects they had memorized from a list of 16 similar objects. They were then required to put the five objects in the same order they were originally given. Memory tasks alternated between image tasks and word tasks in order to keep from confusing people with what they had just seen.

Several measurements were taken to evaluate performance. For each test, we measured the amount of time needed to complete the maze, the number of times the robot collided with obstacles, average speed, performance on memory tasks, and joystick steering entropy [37, 9]. A Questionnaire, given in Appendix A, was given to each test subject after using each interface. This questionnaire was designed to measure task workload, how difficult the display was to learn, and the level of confidence they had in each interface. Scores on the questionnaire range from low to high, or easy to hard, similar to the scoring criteria used in NASA TLX [38]. After completing all the tests, the subjects completed another questionnaire, given in Appendix B, which asked them which display they preferred, which they did better on and which was more intuitive. In order to determine if results were biased by the amount of experience people had with robots, we also asked them how much experience they had with robotics.

4.1 Simulation Studies Summary

We had 32 people (8 women and 24 men) complete the test suite in the simulated worlds. Half of these (4 women and 12 men) used the standard interface first and the other half used the ecological interface first. Our results show that the average person did much better with the ecological display. We found that people averaged 14% less time with the ecological display and twice as many people crashed into walls with the standard display as the ecological display. Additionally, entropy was 31% lower with ecological display and subjects preferred it 4 to 1.

4.2 Objective Results

People did remarkably better using the ecological display than they did with the standard interface. Two of the most notable difference were the number of people that crashed using each interface and joystick entropy. There were over 7 times more collisions using the standard interface than the ecological interface which amounts to less than 12% of the total collisions. Additionally, more people were able to complete the worlds without crashing at all using the ecological interface. We also found that people were able to complete the worlds much faster (an average of 14% faster) and that their average velocity increased approximately 9%.

Figure 4.2 shows the number of test subjects that crashed in at least one world through those who crashed in every world for each interface. Although 34% of the participants crashed in at least one world, more than half of those crashed in only one world. In fact, the number of people that crashed using the ecological interface in at least one world seems to decrease exponentially when we look at the number that crashed in another world as well. On the other hand, about 69% of the test subjects crashed in at least one world using the standard interface. Additionally this number decreased much slower percentage-wise than it did for the ecological interface. Twice as many people crashed in every world using the standard display as the number of people that crashed in half the worlds using the ecological display.

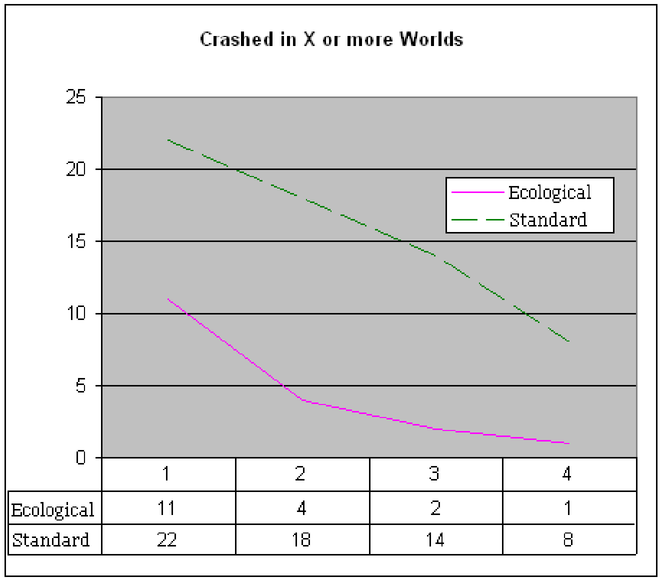
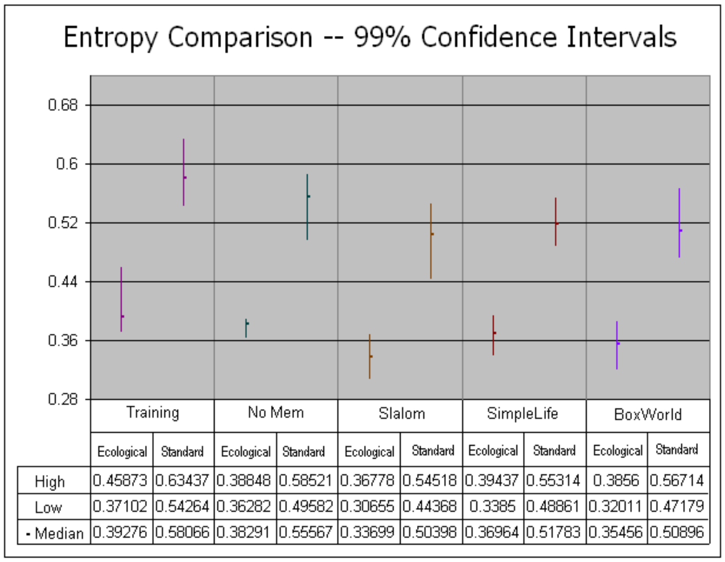


Figure 4.3: The number of people that crashed in at least 1, 2, 3, or all the worlds. This represents a cumulative distribution on the number of worlds people crashed in.

The average behavioral entropy of every single person was lower when using the ecological interface. This is a very good result because high entropy corresponds to higher workload or less efficient control. Since the entropy is much higher using the standard display, we conclude that the amount of human workload and effort required to plan actions was higher using this display. Overall, behavioral entropy decreased 31% when using the ecological display. Figure 4.4 shows 99% confidence intervals for each world on both interfaces. This strongly supports our claim that the ecological display is easier to use than standard displays.

Entropy was calculated by averaging the angle of the joystick every 150ms. A second-order Taylor series expansion was used to predict what the angle would be. The error was



	Training Ecological	Training Standard	No Mem Ecological	No Mem Standard	Slalom Ecological	Slalom Standard	SimpleLife Ecological	SimpleLife Standard	BoxWorld Ecological	BoxWorld Standard
High	0.45873	0.63437	0.38848	0.58521	0.36778	0.54518	0.39437	0.55314	0.3856	0.56714
Low	0.37102	0.54264	0.36282	0.49582	0.30655	0.44368	0.3385	0.48861	0.32011	0.47179
• Median	0.39276	0.58066	0.38291	0.55567	0.33699	0.50398	0.36964	0.51783	0.35456	0.50896

Figure 4.4: Entropy comparison between ecological and standard display for each world.

then put into bins according to how far the error deviated from zero. The size of the bins was adjusted so that the center three bins had 90% of the data. The entropy is related to the distribution of the data in the bins; the more evenly distributed across the bins the higher the entropy because that means it is harder to predict what people are doing with the joystick. This is a standard method of determining behavioral entropy and was proposed by Nakayama et al. [37].

Unfortunately, the results for the memory task were inconclusive. We had hoped that the memory task would also show that workload was lower using the ecological display, but the memory task was too easy and thus did not get us enough good data to show anything. The results for the memory task are shown in Table 4.1 with the results for the other metrics.

Display	Time	Collisions	Speed	Memory task	Entropy
Ecological	212.37 s	30	45.26 cm/s	98.04	0.358
Standard	248.56 s	237	41.41 cm/s	98.53	0.519
Improvement	36.19 s	6.47/subject	3.85 cm/s	-0.49	45%
p values	$8.60 \cdot 10^{-6}$	$2.24 \cdot 10^{-4}$	$2.31 \cdot 10^{-5}$	0.493	$3.75 \cdot 10^{-15}$

Table 4.1: Objective Test Results - Simulation

4.3 Subjective Results

In addition to the objective metrics we used to validate the interface, we also obtained several subjective measurements to help us validate the usefulness of the ecological interface. These measurements include the time it took for subjects to feel comfortable using the interface and a questionnaire that dealt with which interface they preferred, which was more learnable, etc. All of these metrics consistently rate the ecological interface higher than the standard interface we were comparing it against. Some of the statistical results that we obtained are shown in Table 4.2.

One subjective measure that we were able to measure somewhat objectively is the amount of time it took for someone to feel comfortable using the interface. This measurement is hard to really quantify because there are so many variables that can affect it besides the interfaces themselves. For example, the scripts for describing the interfaces were different because the interfaces themselves were different. Another random variable is how each individual interprets when they “feel comfortable” with the interface and how long they feel like training on each one. There were also a couple of situations where technical difficulties or other factors may have influenced the amount of time somebody took to complete training. For example, there were times when two subjects were located in the same room and came at the same time. In these circumstances one of them had to start first and the other subject heard about an interface before they actually began the tests. Additionally, such subjects may have been motivated by peer pressure or other factors to finish training when the other person had finished hearing about an interface. Lastly, some people began figuring out the interface while it was still being introduced while others waited until the entire introduction had been read.

Display	Training	Workload/Effort	Learnability	Preference	Confidence
Ecological	191.82 s	2.97	2.59	78.13%	8.28
Standard	240.61 s	5.47	4.75	18.75%	6.81
Improvement	48.79 s	54%	55%	4x	22%
p values	$5.22 \cdot 10^{-2}$	$1.93 \cdot 10^{-7}$	$3.09 \cdot 10^{-5}$	$2.02 \cdot 10^{-4}$	$5.46 \cdot 10^{-5}$

Table 4.2: Subjective Results - Simulation

Despite the fact that feeling comfortable using the interface is somewhat hard to measure, our results show an advantage in the time needed to learn the ecological interface over the standard interface. Due to the high variance, we are unable to show statistical significance with the raw data, but the fact that people averaged 49 seconds, or 20% less time, using the ecological display is hard to dismiss. Additionally, people averaged about 85 seconds longer on the first interface they saw independent of the interface being studied. To compensate for this factor we could take off 30 seconds or 10% of the time they took to learn the first interface and attribute it to learning the test instead of the interface. Either change would show there is a statistically significant ($p < 0.05$) difference between the time it took to learn each interface.. In other words, when the time taken to learn the interface is separated from other factors, it becomes readily apparent that the ecological display is easier to learn.

Another subjective measure of learnability was a question that asked how easy it was to learn each interface. Figure 4.5 shows a comparison of how difficult it was to learn each interface. The average learnability score on the questionnaire for the ecological interface was a 2.59 which is very close to the "easy" side of the scale. The average score for the standard interface was a 4.75 which is a little below the middle of the scale. This backs up the assertion that the ecological interface is easier to learn. These numbers indicate that, on average, people perceive the ecological display to be about twice as easy to learn as the standard interface.

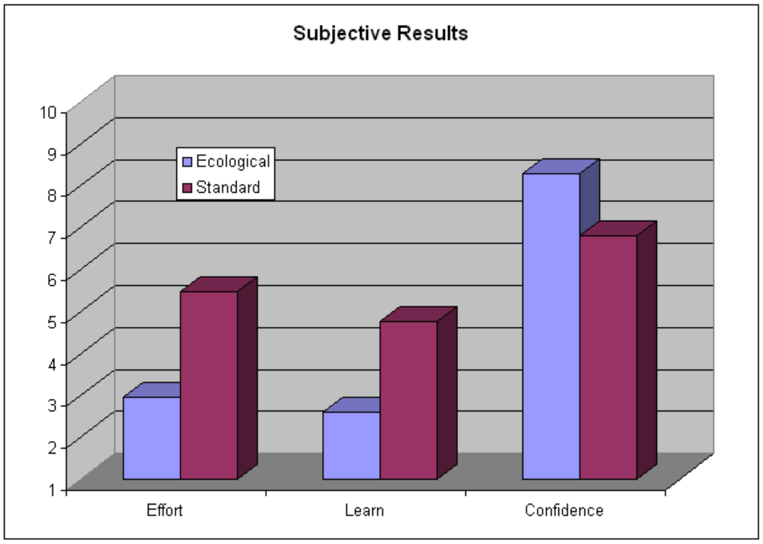


Figure 4.5: How much effort was required to use the robot, to learn how to use the interface, and how much confidence people had in the robot using each interface.

People also thought the ecological interface took less effort to use and had more confidence in the robot when using it as shown in Figure 4.5. The ecological display rated about half of the workload required for the standard interface. Additionally, people marked the ecological interface almost twice as close to "high" confidence than the standard interface. These results are consistent with the learnability measure (Figure 4.5).

Most people preferred the ecological interface and felt they did better using it as shown in Figure 4.6. 78% of the test subjects preferred the ecological interface with only 19% preferring the standard interface. There was one person who preferred both interfaces equally. Related to this metric, 84% of the test subjects though that the ecological interface was more intuitive while 9% thought the standard interface was more intuitive. 84% thought they did better using the ecological interface while 9% felt they did equally well on both. The fact that people overwhelmingly preferred the ecological interface, thought

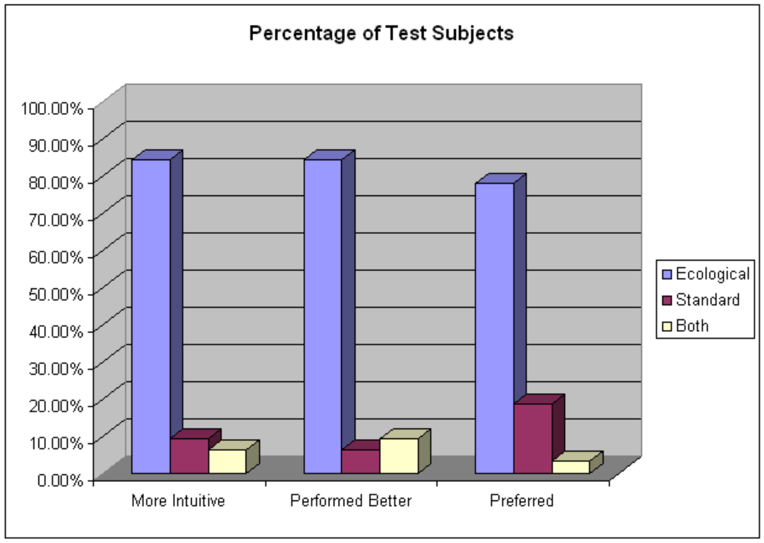


Figure 4.6: Percentage of test subjects who felt each interface was more intuitive, they performed better using a particular interface and which they preferred.

it was more intuitive and felt they did better using it strongly suggests that our ecological interface improves the current state of teleoperation.

4.4 Comments

In addition to the standard questions, space was left on the final questionnaire for comments. Some of these comments illustrate what made the ecological interface easier to use.

Some of the most informative comments come from those who preferred the standard interface. One operator who preferred the standard interface commented that he thought he could do better if the rate images were received was increased. This shows that this individual was probably trying to drive the robot mostly with image information instead of taking advantage of other sensors and features that could have helped. Another individual liked the standard interface better because it had more “neat stuff” to look at like the map display and the compass. This shows that some people preferred one display or the other for merely cosmetic reasons.

Another person commented that he liked the standard display because “it presented a challenge” even though he preferred the ecological interface. This shows that the task required may have been too easy for some people, although it was a challenge for others. This may be due to the fact that some people had a really hard time learning the standard interface. The same person further commented that he thought he could learn to use the standard interface, but would prefer the ecological interface in “life-or-death” situations until he had mastered the standard interface. One operator said the standard interface was “more pleasant to use” but that the ecological interface would be preferable if “asked to perform a task with more precision than driving around a maze.”

Some people did not like the ecological interface because it’s prediction sometimes makes the display jump around a little. The main reason this happens is that errors in the prediction are corrected in future updates which may cause the position to jump. Another effect is that the jump makes it appear that the robot is traveling in a certain direction when it is not. With a little training people could probably be taught to recognize this and correct for it, but we were looking at the effects of the interfaces when very little training was allowed. Another source of jumpiness is the camera image, which jumps to a new location when a new image is received. While this can bother those new to the interface, we believe it serves a useful purpose and would probably be harder to understand if it did not move to match up with the obstacles.

Some of the test subjects seemed to feel the control interface was different because they did not realize why there was some delay between actions they took and the effects of those actions. For example, someone commented that the standard interface was harder because “the steering is touchy.” Along those same lines, another person felt the standard interface allowed more “decisive” moves and someone felt the biggest difference was “the joystick sensitivity.” Another operator commented that the standard interface “had a delay in turning and moving that really annoyed me." Even though the video comes across at the same rate in both interfaces, one person felt the camera on the standard interface "was choppy."

Some people only had comments about the structure of the experiments. Some people did not like the way path planning was implemented because it did not give instructions quite like most people would give instructions. Others would have like to see waypoints in the obstacle portion of the ecological display, but this would be impossible in many real-world situations where people would have to find things the live video as well. Another insightful comment was that "instead of having a synthesized computer voice, (we should) get some sexy girl voice."

Some people seemed to really appreciate the ecological interface. One person said "it was nice to have laser and sonar translated and combined in to one graphical display - it did all the translation work for me therefore diminishing my workload." Another commented that they were more sure "where the robot was more of the time." Finally, one subject commented that the perspective of the ecological interface made it easier to negotiate the turns and "easier to plan a turn in advance."

People with experience in video games often commented about similarities between the ecological display and 3-D video games. One person said this made it "feel more accurate, easier to control."

Chapter 5

The Real World

To validate that our simulation results will carry over into real-world systems, we ran similar experiments in the real world. We had 8 people (2 women and 6 men) complete the test suite in the real world. None of these people had done the simulation studies. Half of these (1 woman and 3 men) used the standard interface first and the other half used the ecological interface first. We used space in the old UVSC building (B-77) to run our experiments in the real world. A map of this space can be seen in Appendix C, although we only had rooms S209, S210, S211, S212, S213, and S215 available for our tests. Because we did not want to require back-tracking and to leave one room for training, we really could only use rooms S213, S212 and S209 and the hallway S210A. In order to require several turns and make a decent-sized course we added boxes to create a small obstacle course through these three rooms for the tests.

There are a couple of factors that required the experiments be different in the real world than they were in simulation. Due to limited space in the old UVSC building it was necessary for the hallways to be closer together, which required a small change to the joystick control mappings to allow tighter turning. The lasers and sonars on the real robot are also a lot noisier than the sensor on the simulated robot. Additionally, a safeguarding

Figure 5.1: Some pictures of the space we used for real-world experiments. The picture on the left shows the robot halfway through the first room of the test environment. The middle picture shows the second room, and the right picture shows the last room and the hallway between the second and third rooms.

behavior was added to keep the robot from banging into walls and boxes (in order to keep the robot working). We also only made two mazes, one for practice and one for the tests, although we designed the test maze to be drivable forwards and back again. Instead of having the computer guide people through waypoints, we had a person down in the other building give simple instructions that subjects were able to hear in a synthesized voice and large arrows to show the way. Finally, memory task items were located throughout the mazes instead of being shown before each test, as discussed in more detail in Section 5.4.

We found that we were unable to run as many experiments in the real world as we were able to in simulation. Most of this was due to the fact that the changes mentioned above made the maze more difficult than we anticipated. Another reason is that it takes a little longer to set up between mazes in the real world than it does in simulation because memory tasks have to be changed and the robot may need to be moved or turned. We also sometimes ran into communications delay induced by wireless network conditions and hardware that were much worse than those used in the simulated studies.

5.1 Objective Results

Our results from our real-world experiments are similar, but more dramatic, than our simulation results. People averaged 105% longer to complete a maze with the standard interface than the ecological interface. Additionally, people crashed[1] thirteen times less and their entropy was 29% lower with the ecological interface. We also found that memory task performance was 12% better with the ecological interface. Complete results of the objective measurements can be found in Table 5.1.

Display	Time	Collisions	Speed	Memory task	Entropy
Ecological	269.86 s	6	26.70 cm/s	95.86	0.393
Standard	553.11 s	83	13.53 cm/s	85.88	0.509
Improvement	283.25 s	9.63/subject	13.17 cm/s	9.98	29%
p values	$4.53 \cdot 10^{-3}$	$5.51 \cdot 10^{-3}$	$7.84 \cdot 10^{-4}$	$4.15 \cdot 10^{-2}$	$3.58 \cdot 10^{-2}$

Table 5.1: Objective Test Results - Real World

5.2 Subjective Results

The benefits of the ecological interface were also very apparent in the subjective measurements. These are summarized in Table 5.2. Note that all these results are statistically significant. In fact, the difference between the two interfaces is much more striking in the real world. People were felt it was about three times as easy to learn, took one third the effort and were much more confident when using the interface. All eight test subjects preferred the ecological interface in every way.

[1] Crashes are defined, in this context, as safeguarding slowing their progress 70% or more to avoid a collision.

Display	Training	Workload/Effort	Learnability	Preference	Confidence
Ecological	170.85 s	2.75	2.25	100%	6.38
Standard	309.89 s	7.625	7.50	0%	3.25
Improvement	139.04 s	64%	70%	100%	46%
p values	$4.83 \cdot 10^{-2}$	$6.30 \cdot 10^{-4}$	$8.39 \cdot 10^{-5}$	-	$1.12 \cdot 10^{-2}$

Table 5.2: Subjective Results - Real World

5.3 Delay

Since socket communication in the real world experiments involved at least one wireless ethernet link, packets sometimes took a lot longer to arrive than they did in simulation. This can be especially troublesome when delays are sporadic and inconsistent because it is hard to tell which command the robot is executing, how long each command has executed and how old the latest information is. Fortunately some of this estimation is handled by the ecological interface which simply predicts that each command is executed for the amount of time before another command was sent.

$$delay_t = (0.95) * delay_{t-1} + (0.05) * (t_{now} - t_{robot}) \tag{5.1}$$

Actual delay was estimated by a simple formula which gives a rough estimation of the average delay, with recent delays counting more toward the average than older delays. Each time a message is sent the average is updated by making the new average equal to 95% of what it was before and 5% of the difference between the timestamp of the message being sent and the timestamp of the latest information received from the robot, this formula is shown in Equation 5.1. The average is also updated when new information is received from the robot. Note that this calculation makes approximately 40% of the average delay come from information gathered within the last second[2].

$$Delay = DesiredDelay - AverageDelay + SendDelay \tag{5.2}$$

In order to keep the average delay around one second we delay messages from the interface to the robot. The amount we want to delay outgoing messages is found using the formula in Equation 5.2. *Send delay* is the amount we have been delaying messages, calculated with Equation 5.1. If we have been delaying messages too long, then average delay will increase, which causes messages to be delayed for less time. This creates a cycle where messages are delayed for shorter periods of time for a while, then as the average decreases messages get delayed a little longer until the average catches up. Since 60% of the average comes from older information, delay figures tend to be fairly close to one second as long as network conditions are good. In fact, when people were using the robot in the practice room, the standard deviation for the delay was always less than $1/30^{th}$ of a second. It is important to keep delay measurements consistent in order to make conditions as similar as possible between subjects and between the real world and simulation.

The average delay was recorded every 150ms, which allowed us to compare network conditions between subjects. As recorded in Table 5.3, the standard deviation on the delay measurement was always more than $1/30^{th}$ of a second because network conditions were much worse in some parts of the course than they were in the training room. Table 5.3 shows the average of these measurements along with the longest such measurement for each test subject for the first test run of four test subjects. It is interesting to note that performance was not closely correlated with delay, although it may have been a contributing

[2]This figure comes from the fact that each timestep 95% of the weight goes to the previous information. Since this is updated about 10 times per second, information a second old or older should account for 0.95 to the 10^{th} power. This means about 60% of the average comes from information over a second old and about 40% comes from information received in the last second.

factor in some cases.

Sched	Display	Delay	St Dev	Max	Time	Entropy	Mem Task	Safeguarding
0	Ecological	1.01	0.071	1.57	233.31	0.399	91	0
	Standard	1.02	0.136	2.47	893.11	0.620	91	19
1	Ecological	1.02	0.120	1.81	160.91	0.386	100	0
	Standard	1.00	0.057	1.14	266.17	0.616	91	4
2	Ecological	1.04	0.240	3.14	190.89	0.422	100	0
	Standard	1.01	0.093	2.23	480.53	0.262	91	8
3	Ecological	1.00	0.065	1.61	289.64	0.382	100	2
	Standard	1.00	0.058	1.98	475.11	0.455	75	23

Table 5.3: Delay statistics for the first forward run on each interface for test subjects that used the standard interface before the ecological interface. More complete delay statistics are found in Appendix D.

One of the rooms in the real-world test environment had a much worse network connection than the rest of the rooms. This room, S209 (see the map in Appendix C) was the at one of the ends of the test course, the last room when going forwards and the first room when going backwards. This caused problems as people tried to maneuver in this room while fighting long and inconsistent delays. There were times when the connection in the room got so bad it was virtually unusable. One time packets got behind about a minute (see Table D.1 in Appendix D) and then improved when the robot left room S209.

People were usually able to deal with the delay better with the ecological interface. Table 5.3 shows that people tended to see higher average delays with the standard interface than with the ecological interface. While this may have contributed to the fact that they tended to take longer using the standard interface, the two factors are actually dependent on each other. Since people had a harder time dealing with delays using the standard interface, they tended to spend longer in the areas where they were experiencing higher delays. Staying longer in areas with worse network conditions caused their average delay to increase and made it harder to control the robot with the standard interface which could cause them to take even longer to get through the course. This vicious cycle is not nearly as bad with the ecological interface because it allowed people to better anticipate and deal with varying network conditions.

5.4 Memory Tasks

In the real world mazes, test subjects were required to find five words or images that had been placed toward the beginning of each maze. They were required to remember each item and put it in the order it appeared in the world at the end of each test. This makes the memory task more difficult because people have to find each memory item, and usually they would need to stop in front of each to identify it later on. Instead of just measuring workload as in the simulation experiments, the memory tasks also measure how well people are able to maneuver the robot to see the memory item with the robot's forward-facing camera. These maneuvers also affected the other measurements such as time taken, entropy, and velocity because people needed to take some time and effort to see the memory items. This makes the memory task much more realistic in terms of what people might be doing when operating a robot; people are more likely to be looking for things in the world than simply remembering lists.

Memory task performance is calculated based on a few factors. First, up to 38 points are awarded for correctly remembering each memory task item. Second, up to 42 points are awarded for correctly placing these items with respect to each other. Finally, 4 points are given to each of the five memory task items placed in the correct position. These 100 points are divided in such a way that they award people primarily for remembering what items came before other items and secondarily for remembering the exact ordering of those items. Both of these rely on being able to remember the objects in the first place.

Because recall contributes directly or indirectly to all three of the factors that make up the score, test subjects receive more points for correctly remembering the first item, less for the second and so on. The amount they receive for remembering partial orderings also decreases with the number they already have correct. This is simply one way to get utility values for memory task performance so we can rank how well people did. Scores range from 20 to 24 if the test subject remembered only one item, and someone that remembered four of the five items would score between 37 and 91 points. This is important because much more of the score is based on remembering the sequence of items than recall, especially because most people remembered at least four of the five items regularly, as evidenced by scores above 76 (the maximum when a test subject only remembers only three items) in Table 5.3. This utility-based measurement of memory task performance performs pretty well for our purposes because it gives us a good idea of how well they remember the ordering of the memory task items.

5.5 Comments

Some of the comments people made about the interfaces were very interesting. One test subject, who experienced much longer delays when using the ecological interface due to network conditions, said that the standard interface "had significantly more lag than the [ecological] interface." This is most likely due to the fact that the quickening employed by the ecological interface allowed this operator to interact with the robot more readily than the standard interface. The same person also commented that it was more difficult to work with the range sensors separated from the image data in the standard interface than it was to have them all together like they are in the ecological interface.

We also got comments that the standard interface was much harder and some people incorrectly thought the joystick worked differently in the standard interface than it did with the ecological interface. For example, one person said that "the robot did what I wanted it to" when using the ecological interface whereas the standard interface was "very frustrating." Several people commented that immediate feedback in the ecological interface was helpful. One person mentioned it was easier to integrate information from the sonar, lasers and camera in the ecological interface.

5.6 Correlation with Simulation Experiments

Since the experiments turned out to be quite different, it is difficult to correlate a lot of the data we gathered in the real world with the data we gathered in simulation. These experimental differences occur both in terms of memory task structure and the size, number, and difficulty of the courses we used in the two situations. Table 5.4 shows a few of the results which are the most comparable between the experiments. The subjective results tend to be more comparable since the same scale was used in both experiments. Entropy is included because the numbers are fairly similar, although we would expect entropy to be higher in courses that are more difficult to navigate.

Display		Entropy	Effort	Learnability	Preference	Confidence
Ecological	Real World	0.393	2.75	2.25	100%	6.38
Ecological	Simulation	0.358	2.97	2.59	78.13%	8.28
Standard	Real World	0.509	7.625	7.50	0%	3.25
Standard	Simulation	0.519	5.47	4.75	18.75%	6.81
Improvement	Real World	29%	64%	70%	-	46%
Improvement	Simulation	45%	54%	55%	4x	22%

Table 5.4: Correlation - Real World and Simulation

There is not a lot of correlation between the results in Table 5.4, although there are a few interesting trends. Improvement tends to be more dramatic in the real world, although this does not apply to entropy because the courses were more difficult in the real world. Confidence in the robot's actions was higher using both interfaces in simulation than either in the real world; once again probably due to the added difficulty. Finally, people found the ecological interface about as easy to use in the real world as in simulation, but thought the standard interface was much harder to use in the real world. This is interesting in light of the fact that they had much less confidence in the real world. The data shows that while the experiments are fundamentally different, they yield similar conclusions showing that the ecological interface is dramatically better than the standard interface.

Chapter 6

Limitations and Future Work

In the future we expect many of the principles of our ecological interface to be applied to many other systems, and we expect ecological principles to improve the state of the art in these systems as well. The ecological interface could be improved by using better prediction, especially prediction that takes into account likely actions by intelligence on the robot or collisions with known obstacles. Image processing could also be used to give users a better idea of where objects detected by the camera are located in relation to obstacles detected by the range sensors. Ecological interfaces like the one we created can be used for other methods of user control of robotics besides teleoperation, especially if combined with better prediction methods mentioned above. Additional features could be added to the ecological display, which would enhance telepresence such as the ability to use a pan and tilt camera and readily determine the current direction of the camera in relation to the pose of the robot. Another improvement would be to integrate map-building and localization into the interface so that the user would be able to see obstacles detected previously in addition to current sensor readings.

There are a number of ways that the prediction algorithms used in the ecological interface described in this thesis could be improved. First and foremost, prediction of future robot positions should take into account either collisions with obstacles and/or actions taken by intelligence on the robot itself. For example, if the robot hits a corner as it is trying to turn, then the prediction may tell the user that the robot is around the corner in the next hallway when in fact it is stuck on the corner. There should be some mechanism for deciding that the robot is or will get stuck in the prediction algorithm in order to inform the user of this problem. Another improvement to the prediction algorithm would be to integrate the robot's current velocity into estimates of the robot's future position and velocity – i.e., include inertia effects and robot kinematics. Finally, more accurate information about the amount of time a particular command will execute on the robot could be obtained by passing an additional timestamp that includes the amount of time a command has been running on the robot when it sends back state information. Another possibility would be to send commands more frequently. These improvements to the prediction algorithm, while not trivial, could improve operators' sense of presence by giving them a better idea of where the robot is and what it is doing.

Another way in which better prediction could be used to enhance the interface is to use image processing. Currently the display simply moves and rotates the camera image to represent the objects in the display. This is based on an image plane a fixed distance from the robot. This could be improved by doing image processing to determine the distance to parts of the image and warping it in a way that better represents what would we would see from the position the robot has moved to. A more advanced approach would identify actual objects in the image and move the objects separately so objects in the foreground of the camera image would move more than objects in the background. Additional image processing could also be used to combine information from past camera images into the display in such a way that the user could get more complete visual information. Predicting what objects in the image are doing could enable the part of the display from the camera to be more meaningful and useful.

In addition to teleoperation, ecological displays could be used to help improve human-robot interaction with a variety of different control modes and autonomy levels. For example, users might benefit from an interface for point-to-point control that had both a view from behind the robot and a top-down view for path planning. Control modes that take advantage of robot autonomy could also benefit from using an ecological display by implementing some prediction of what the autonomy will do into the prediction algorithm. Shared control schemes allow humans to give directions to a robot with a joystick, but the robot decides if that is safe and modifies it if necessary to keep out of danger. This control scheme would require prediction that takes into account what the robot will probably do in the future to fully implement an ecological interface like the one discussed in the paper, but would probably greatly benefit from it. One reason shared control would benefit is that people would be able to see the world in the context of an ecological display so they could see objects around them easily, yet the robot would help them avoid the obstacles.

One feature often found on robots that is often hard to figure out on robot interfaces is a pan and tilt camera. This allows users to look at objects to the side of the robot without moving the actual robot. Studies have shown that it is often hard for people to keep track of which direction the camera is pointed when the robot is stopped [25]. This could be alleviated by moving the camera in the scene to show what part of the world is currently being shown in the camera image like our ecological interface does. Additionally, the ecological interface would make it easier for people to drive the robot with the camera pointed in another direction because the integrated lasers and sonars allow navigation even when the camera is being used for other tasks. Pan and tilt cameras could also be used to improve the ecological interface by panning to the side when the robot is about to make a turn. This would allow the use to see down the hallway he is about to turn into before the front of the robot comes around to see down the hallway. While pan and tilt cameras have been used on robots for years, the union of this technology and ecological interfaces would complement each other and could make both much more effective.

Other features could be added to the ecological interface that would allow the user to better understand the robot's current state and anything they can do about it. It is often important to get good information about the status of the robot's batteries and communications. This could be displayed as text in a heads-up display, but people often don't keep track of those kind of things when they are under stress or new to the system. Another way to display this information would be to color code the robot to give the user a quick idea of the status of the robot's state. An improvement would be to have the robot flash or put a flashing light on the robot when a problem occurs or something falls below some critical level.

There are often times when a robot revisits an area that has been visited or mapped out before. By integrating mapping and/or localization this information could be integrated into the interface which would allow better interaction with the environment. For example, when the robot turns a corner it would be nice to know how big the new area is in advance, especially when there are long delays. This could also be used in a top-down view for path planning and other activities. As long as the robot was able to localize itself with reference to previous information this could be very valuable.

Chapter 7

Conclusion

Teleoperation is difficult because of delay and lack of perceptual cues. Most research to address this problem has focused on either adding sensors or intelligence to the robot. This thesis has proposed an alternative approach that combines an ecological display with prediction to help operators better understand what the robot is doing. Our experiments demonstrated that people have a much easier time using the new interface to control robots than they do with a standard interface. Additionally, most people preferred the new interface and felt it is easier to use.

In order to validate that the ecological interface improves performance we performed user studies to determine how well untrained users would be able to use the ecological interface. These studies were performed both in simulation and in the real world. In simulation, we found that people were able to complete the mazes 17% faster with 80% fewer collisions and 70% of the workload (as measured by behavioral entropy). In the real world, people completed the mazes in half the time with 90% fewer collisions and 71% of the workload. Standard t-tests show that these results are all statistically significant at p=0.05. Additionally, people preferred the ecological interface 8 to 0 in the real world and

4 to 1 in simulation. These user studies show that the ecological interface improves the effectiveness of teleoperation over a conventional display.

Bibliography

[1] Kevin Arthur. *Effects of Field of View on Performance with Head-Mounted Displays*. PhD thesis, University of North Carolina at Chapel Hill, Department of Computer Science, 2000.

[2] Ronald T. Azuma. A survey of augmented reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, Aug 1997.

[3] Rahul Biswas, Benson Limketkai, Scott Sanner, and Sebastian Thrun. Towards object mapping in non-stationary environments with mobile robots. In *Proceedings of the Conference on Intelligent Robots and Systems (IROS)*, 2002.

[4] Brian Blau, Charles E. Hughes, Michael J. Moshell, and Curtis Lisle. Networked virtual environments. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, pages 157–160, 1992.

[5] Shawn Bonham, Daniel Grossman, William Portnoy, and Kenneth Tam. Quake: An example multi-user network application - problems and solutions in distributed interactive simulations. CSE 561 term project report, University of Washington, 2000. http://www.cs.washington.edu/homes/grossman/projects/561projects/quake/Quake.ps.

[6] David J. Bruemmer, Julie L. Marble, David D. Dudenhoeffer, Matthew O. Anderson and Mark D. McKay. Mixed-initiative control for remote characterization of hazardous environments. In *HCISS*, 2003.

[7] Catherine M. Burns, Johnson Kuo, Renee Chow, Angela Garabet, Olivier St-Cyr, and Kim J. Vicente. Future networks: How will people manage? Technical report, University of Waterloo, 2001. http://www.ngi-supernet.org/lsn2000/U_Waterloo.pdf" displays in marine navigation

[8] Peter I. Corke. Experiments in high-performance robotic visual servoing. In *International Symposium on Experimental Robotics, Kyoto, Japan*, 1993.

[9] Jacob W. Crandall and Michael A. Goodrich. Principles of adjustable interactions. In *Proceedings of the 2002 AAAI Fall Symposium Human-Robot Interaction Workshop*, 2002.

[10] R. P. Darken, K. Kempster, and B. Peterson. Effects of streaming video quality of service on spatial comprehension in a reconnaissance task. In *Proceedings of the Interservice/Industry Training and Simulation Education Conference*, 2001.

[11] David Drascic and Paul Milgram. Perceptual issues in augmented reality. In *SPIE Vol. 2653: Stereoscopic Displays and Virtual Reality Ssytems III*, 1996.

[12] Jeff Ellis. An investigation of predictive and adaptive model-based methods for direct ground-to-space teleoperation with time delay. Master's thesis, Wright State University, 1998.

[13] M. R. Endsley. Automation and situation awareness. In R. Parasuraman and M. Mouloua, editors, *Automation and human performance: Theory and applications*. Mahwah, NJ: Lawrence Erlbaum, 1996.

[14] Mica Endsley, Randy Sollenberger, and Earl Stein. The use of predictive displays for aiding controller situation awareness. In *Proceedings of the Human Factors and Ergonomics Society 43rd Annual Meeting*. Santa Monica, CA: Human Factors and Ergonomics Society, 1999.

[15] Terrance W. Fong and Charles Thorpe. Vehicle teleoperation interfaces. *Autonomous Robots*, 11(1):09–18, Jul 2001.

[16] Terrence Fong, Charles Thorpe, and Charles Baur. Advanced interfaces for vehicle teleoperation: collaborative control, sensor fusion displays, and remote driving tools. *Autonomous Robots*, 11(1):77–85, Jul 2001.

[17] Terry Fong, Henning Pangels, and David Wettergreen. Operator interfaces and network-based participation for Dante II. In *SAE 25th International Conference on Environmental Systems (ICES)*, Jul 1995.

[18] Terry Fong, Charles Thorpe, and Charles Baur. A safeguarded teleoperation controller. In *IEEE International Conference on Advanced Robotics 2001*, Aug 2001.

[19] James Jerome Gibson. *The ecological approach to visual perception*. Houghton Mifflin, Boston, 1979.

[20] Michael Goodrich, Dan R. Olsen Jr., Jacob W. Crandall, and Thomas Palmer. Experiments in adjustable autonomy. In *Proceedings of the IJCAI01 Workshop on Autonomy, Delegation, and Control: Interacting with Autonomous Agents*, 2001.

[21] Rob Gray, Hong Z. Tan, and J. Jay Young. Do multimodal signals need to come from the same place? Crossmodal attentional links between proximal and distal surfaces. In *Proceedings of the 4th IEEE International Conference on Multimodal Interfaces*, Oct 2002.

[22] I. Harvey, P. Husbands, and D. Cliff. Issues in evolutionary robotics. In *Proceedings of SAB92, the Second International Conference on Simulation of Adaptive Behaviour*, 1992.

[23] Butler Hine, Phil Hontalas, Terrence Fong, Laurent Piguet, Erik Nygren, and Aaron Kline. Vevi: A virtual environment teleoperations interface for planetary exploration. In *SAE 25th International Conference on Environmental Systems (ICES)*, Jul 1995.

[24] B. L. Hooey, D. C. Foyle, and A. D. Andre. The design of aircraft cockpit displays for low-visibility taxi operations. In A. G. Gale, editor, *Vision in Vehicles IX*. Holland: Elsaevier Science Publishers, 2001.

[25] Stephen Hughes, Joseph Manojlovich, Michael Lewis, and Jeffrey Gennari. Camera control and decoupled motion for teleoperation. In *Proceedings of the 2003 IEEE International Conference on Systems, Man, & Cybernetics*, 2003.

[26] Eric Krotkov, Reid Simmons, Fabio Cozman, and Sven Koenig. Safeguarded teleoperation for lunar rovers: From human factors to field trials. In *Proceedings of the IEEE Workshop on Planetary Rover Technology and Systems*, Apr 1996.

[27] Peter Lincroft. The Internet sucks: Or, what I learned coding X-Wing vs. Tie Fighter, 1999. http://www.gamasutra.com/features/19990903/lincroft_01.htm.

[28] G. W. Lucas. A tuturial and elementary trajectory model for the differential steering system of robot wheel actuators. The Rossum Project, 2001. http://rossum.sourceforge.net/papers/DiffSteer/DiffSteer.html.

[29] Michael R. Macedonia, Donald P. Brutzman, Michael J. Zyda, David R. Pratt, and Paul T. Barham. NPSNET: A multi-player 3D virtual environment over the Internet. In *Symposium on Interactive 3D Graphics*, 1995.

[30] Giuseppe Mantovani and Giuseppe Riva. Real presence: How different ontologies generate different criteria for presence, telepresence and virtual presence. *Presence: Teleoperators and Virtual Environments*, 8(5):538–548, Oct 1999.

[31] Joshua A. Marshall. An essay on effective telepresence. Technical report, University of Toronto, Apr 2002. http://www.control.toronto.edu/ marshall/docs/telepresence.pdf.

[32] Santosh Mathan, Arn Hyndman, Karl Fischer, Jeremiah Blatz, and Douglas Brams. Efficacy of a predictive display, steering device, and vehicle body representations in the operation of a lunar vehicle. In *CHI Conference Companion 1996*, 1996.

[33] Roger Meier, Terrence Fong, Charles Thorpe, and Charles Baur. A sensor fusion based user interface for vehicle teleoperation. In *International Conference on Field and Service Robotics (FSR)*, 1999.

[34] Paul Milgram, David Drascic, Julius J. Grodski, Anu Restogi, Shumin Zhai, and Chin Zhou. Merging real and virtual worlds. In *Proceedings of IMAGINA '95*, Feb 1995.

[35] Max Mulder. *Cybernetics of Tunnel-In-The-Sky Displays*. Delft University Press, 1999.

[36] Anne K. G. Murphy. Effective information display and interface design for decomposition-based quantitative electromyography. Master's thesis, University of Waterloo, 2002.

[37] O. Nakayama, T. Futami, T. Nakamura, and E. R. Boer. Development of a steering entropy method for evaluating driver workload. In *SAE Technical Paper Series: #1999-01-0892: Presented at the International Congress and Exposition*, Detroit, Michigan, Mar 1999.

[38] NASA Task Load Index (TLX) users manual. http://iac.dtic.mil/hsiac/docs/TLX-UserManual.pdf.

[39] Laurent A. Nguyen, Maria Bualat, Laurence J. Edwards, Lorenzo Flueckiger, Charles Neveu Kurt Schwehr, Michael D. Wagner, and Eric Zbinden. Virtual reality interfaces for visualization and control of remote vehicles. *Autonomous Robots*, 11(1):59–68, Jul 2001.

[40] Jens Rasmussen. The role of hierarchical knowledge representation in decision-making and system management. In *IEEE Transactions on Systems, Man and Cybernetics*, volume SMC-15, 1985.

[41] David W. Schloerb. A quantitative measure of telepresence. *Presence*, 4(1), Winter 1995.

[42] Dirk Schulz, Wolfram Burgard, and Armin B. Cremers. Predictive simulation of autonomous robots for reliable visualization over the internet. In *Proceedings of the IEEE IROS'98 Workshop on Robots on the Web*, 1998.

[43] Thomas D. Sharp and Arthur J. Helmicki. The application of the ecological interface design approach to neonatal intensive care medicine. In *Proceedings of the 42nd Annual Meeting of the Human Factors and Ergonomics Society*, 1998.

[44] Thomas B. Sheridan. *Telrobotics, Automation, and Human Supervisory Control*. Massachusetts Institute of Technology, 1992.

[45] Jonathon Steuer. Defining virtual reality: Dimensions determining telepresence. *Journal of Communication*, 42(2):73–93, Autumn 1992.

[46] David L. Still and Leonard A. Temme. Oz: Human-centered computing cockpit display. In *Proceedings of the Interservice/Industry Training, Simulation, and Education Conference*, 2001.

[47] Jussi Suomela and Aarne Halme. Tele-existence techniques of heavy work vehicles. In *Autonomous Robots 11*, pages 29–38. Kluwer Academic Publishers, 2001.

[48] Tim Sweeney. Unreal networking architecture. Technical report, Epic MegaGames, Inc., 1999. http://unreal.epicgames.com/Network.htm.

[49] Gregoire Terrien, Terrence Fong, Charles Thorpe, and Charles Baur. Remote driving with a multisensor user interface. In *SAE 30th International Conference on Environmental Systems (ICES)*, 2000.

[50] Geb Thomas, William D. Robinson, and Steven Dow. Improving the visual experience for mobile robotics. In *Seventh Annual Iowa Space Grant Proceedings*, pages 10–20. Drake University, Nov 1997.

[51] Sebastian Thrun. Learning metric-topological maps for indoor mobile robot navigation. *Artificial Intelligence*, 99(1):21–71, Feb 1998.

[52] Martin Usoh, Ernest Catena, Sima Arman, and Mel Slater. Using presence questionnaires in reality. *Presence*, 9(5):497–503, Oct 2000.

[53] Leo van Breda. Prediction information to improve navigational performance. Technical report, TNO Human Factors Research Institute, Soesterberg, The Netherlands, 2000.

[54] Kim J. Vicente, Klaus Christoffersen, and Alex Pereklita. Supporting operator problem solving through ecological interface design. *IEEE Transactions on Systems, Man, and Cybernetics*, 25(4):529–545, Apr 1995.

[55] Christopher D. Wickens and Justin G. Hollands. *Engineering Psychology and Human Performance*. Prentice-Hall, New Jersey, third edition, 2000.

[56] David Woods. Creating foresight: How resilience engineering can transform NASA's approach to risky decision making. Technical report, Testimony on The Future of NASA for Committe on Commerce, Science and Transportation, John McCain, Chair, October 2003. http://csel.eng.ohio-state.edu/woods/news/woods_testimony.pdf.

[57] David D. Woods, James Tittle, Magnus Feil, and Axel Roesler. Envisioning human-robot coordination in future operations. *IEEE Transactions on Systems, Man and Cybernetics*, May 2004.

[58] Yukichi Yamaguchi and Fumiya Tanabe. Creation and evaluation of an ecological interface system for operation of nuclear reactor system. In *Enlarged Halden Programme Group Meeting*, Sep 2002.

[59] Pavel Zahorik and Rick L. Jenison. Presence as being-in-the-world. *Presence*, 7(1):78–89, Feb 1998.

Appendix A

Questions about each interface

We asked these questions to get an idea how easy it was to use each interface.

Question: How much effort was required to use the interface effectively? (circle one)

Low									High
1	2	3	4	5	6	7	8	9	10

Question: How difficult was it to learn to use the interface effectively? (circle one)

Easy									Hard
1	2	3	4	5	6	7	8	9	10

Question: How much confidence did you have in the robot's actions?(circle one)

Low									High
1	2	3	4	5	6	7	8	9	10

Appendix B

Questions comparing the interfaces

These are the questions we asked to gather general information from test subjects and get them to compare the two interfaces.

Question: How much experience do you have with robotics?(circle one)

None									Expert
1	2	3	4	5	6	7	8	9	10

Question: Which interface was more intuitive for you? (circle one)

First   Neither was intuitive   They worked equally well   Second

Question: Which interface do you think you did better on? (circle one)

First   Neither worked for me   They worked equally well   Second

Question: Which display did you prefer? (circle one)

First   Neither was acceptable   They worked equally well   Second

Please use the remaining space for any additional comments you may have.

Appendix C

Space in B-77 for the real robot

This is a map of the space we were able to use in B-77, the old UVSC building. Rooms S216, S214 store equipment and thus could not be used for our experiments. Rooms S210, S211 and the closets were too small to use and undesirable because we did not want to require backtracking.

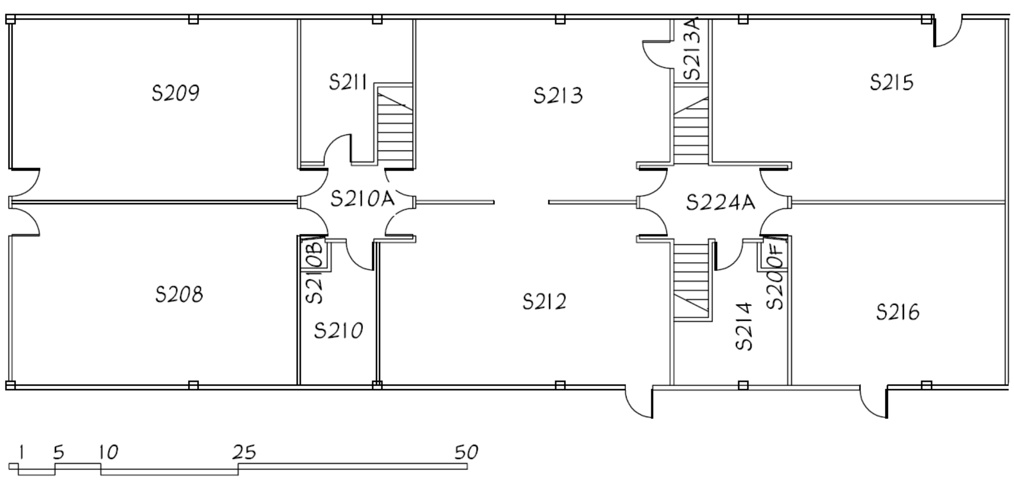


Figure C.1: Map of the area we used in B-77 for the robot experiments.

The practice course consisted of a loop in room S215 and the test course ran from room S213 through S212 to S209. In each room boxes followed the walls around each room, around one or two obstacle and through the doors. The hallways in the simulated maps (Figure 4.2) were about six feet wide, which is about twice as wide as ordinary doorways (Figure C). This made it more difficult to use the standard interface in the real world, especially compared to the ecological interface.

Appendix D

Additional information about delay

Following are more extensive tables on delay statistics than those found in Table 5.3. In addition to including all test subjects in Table D.2, a few of the more interesting statistics are found in Table D.1. Of particular note are the statistics that show that test subjects 1 and 2 were much faster with the ecological interface even though they encountered higher delays with the ecological display (see Table D.2). The individual that ran schedule 22 encountered delays up to nearly a minute using the ecological interface (see Table D.1), yet was still able to complete the course relatively quickly when the network situation improved.

Sched	Course	Display	Delay	St Dev	Max	Time	Entropy	Safeguards
1	No Mem	Ecological	1.04	0.133	2.00	120.03	0.182	0
		Standard	1.00	0.084	1.10	204.828	0.547	4
2	Backward	Ecological	1.32	0.879	7.23	192.89	0.416	2
		Standard	1.11	0.420	3.86	413.062	0.179	4
21	No Mem	Ecological	1.24	1.028	8.20	179.95	0.392	1
21	Backward	Standard	1.00	0.087	1.53	443.78	0.528	13
22	Backward	Ecological	5.86	10.610	51.85	408.91	0.291	1

Table D.1: Delay statistics for runs besides the forward course where the delay was longer using the ecological interface and other interesting cases.

Sched	Display	Delay	St Dev	Max	Time	Entropy	Mem Task	Safeguards
0	Ecological	1.01	0.071	1.57	233.31	0.399	91	0
	Standard	1.02	0.136	2.47	893.11	0.620	91	19
1	Ecological	1.02	0.120	1.81	160.91	0.386	100	0
	Standard	1.00	0.057	1.14	266.17	0.616	91	4
2	Ecological	1.04	0.240	3.14	190.89	0.422	100	0
	Standard	1.01	0.093	2.23	480.53	0.262	91	8
3	Ecological	1.00	0.065	1.61	289.64	0.382	100	2
	Standard	1.00	0.058	1.98	475.11	0.455	75	23
20	Ecological	1.00	0.63	1.23	225.17	0.427	76	0
	Standard	1.10	0.491	6.33	385.72	0.555	81	5
21	Ecological	1.00	0.039	1.24	272.72	0.407	100	2
	Standard	1.01	0.079	1.90	403.30	0.577	81	4
22	Ecological	1.08	0.352	4.09	486.20	0.336	100	0
	Standard	1.28	1.247	15.33	977.72	0.524	77	9
23	Ecological	1.00	0.056	1.19	300	0.386	100	2
	Standard	1.01	0.074	1.76	543.234	0.281	100	11

Table D.2: Delay statistics for the first forward course in the real world.