2009-08-10

Supporting Remote Manipulation: An Ecological Approach

John A. Atherton
Brigham Young University - Provo

SUPPORTING REMOTE MANIPULATION: AN ECOLOGICAL APPROACH

by

J. Alan Atherton

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

December 2009

BRIGHAM YOUNG UNIVERSITY

GRADUATE COMMITTEE APPROVAL

of a thesis submitted by

J. Alan Atherton

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 A. Goodrich, Chair

Date ______________ Mark Colton

Date ______________ Michael Jones

BRIGHAM YOUNG UNIVERSITY

As chair of the candidate's graduate committee, I have read the thesis of J. Alan Atherton 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 A. Goodrich Chair, Graduate Committee

Accepted for the Department

Date	Kent E. Seamons Graduate Coordinator

Accepted for the College

Date	Thomas W. Sederberg Associate Dean, College of Physical and Mathematical Sciences

ABSTRACT

SUPPORTING REMOTE MANIPULATION: AN ECOLOGICAL APPROACH

J. Alan Atherton

Department of Computer Science

Master of Science

User interfaces for remote robotic manipulation widely lack sufficient support for situation awareness and, consequently, can induce high mental workload. With poor situation awareness, operators may fail to notice task-relevant features in the environment often leading the robot to collide with the environment. With high workload, operators may not perform well over long periods of time and may feel stressed. We present an ecological visualization that improves operator situation awareness. Our user study shows that operators using the ecological interface collided with the environment on average half as many times compared with a typical interface, even with a poorly calibrated 3D sensor; however, users performed more quickly with the typical interface. The primary benefit of the user study is identifying several changes to the design of the user interface; preliminary results indicate that these changes improve the usability of the manipulator.

ACKNOWLEDGMENTS

I thank my wife for her remarkable patience and support. Her encouragement and excitement about life made this journey much easier. I also thank my kids for their joy and the energy they brought along. I thank my parents and siblings for supporting me especially in earlier years and instilling the desire for education.

I express gratitude to my adviser, Mike Goodrich, for his exceptionally patient and useful counsel while I learned about how to research. I have great respect for him and his research ideals.

I am grateful to Idaho National Laboratory and my colleagues there who provided financial and other support to make this research possible. I am also grateful to Army Research Laboratory for providing financial support for this research.

Contents

List of Figures

Chapter 1

Introduction

1.1 Background On Remote Manipulation

A robot with manipulation capability, known as a *manipulator*, carries an instrument that can be used to operate on its environment. Robotic manipulators are often modeled after a human arm and hand. When operated from a distance, these robots can be used as human surrogates for certain tasks, and are called *remote manipulators*. Figure 1.1 typifies a remote manipulator robot.

Remote manipulators are useful tools in areas that are dangerous or inaccessible to humans. For example, they are used in planetary exploration (Mars rovers), urban search and rescue, and explosive ordnance disposal [7, 44, 17]. The manipulator is used to grab, push, poke, operate tools, and generally do what people do with

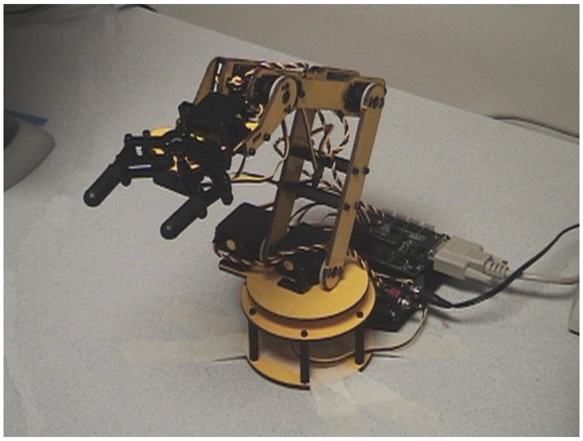


Figure 1.1: Example manipulator robot.

their hands. In order to safely benefit from the capabilities these robots provide, the human operator must control them remotely. Operators can use the manipulator to perform a variety of tasks from a safe distance.

One difficulty in remote manipulation is for the operator to understand the situation given limited data. In remote situations, the only connection the operator has with the robot is the user interface, and the operator is limited to whatever information the robot can provide. This is sometimes referred to as looking at the world through a “soda straw” [56]. Problems that come from this limited view of the world include missed events, difficult navigation in new situations, and an incomplete understanding of the explored world [56].

Factors that affect performance in manipulation tasks include the physical capabilities of the robot, the autonomous behaviors of the robot, the user interface, and the capabilities and skill of the human operator. Although all of these factors affect performance, we can focus on a few factors that a system designer can control. Assuming that we have a skilled operator, the design variables that we can freely manipulate become the robot capabilities, the robot behaviors, and the user interface. When the robot does not have the required capabilities for a task, then the usability of the user interface or autonomous behaviors make little difference for accomplishing the task. On the other hand, when the robot does have the necessary capabilities, then the user interface and autonomous behaviors can greatly affect performance.

In addition to manipulation capability, several applications also require the robot to be mobile. When this is the case, the manipulator arm is mounted to a mobile base that commonly moves by means of wheels or tank-like treads. When the manipulator is mobile in this manner, the robot is called a *mobile manipulator*. While our goal in this research is to support mobile manipulation as a whole, a full mobile manipulation system is complex and requires a significant amount of infrastructure. Because of the complexity, we focus solely on the manipulation aspect for this work.

The scope of this thesis is to create a new interface for *remote manipulation* and test the performance of operators using the interface.

1.1.1 Example Applications

Planetary exploration, explosive ordnance disposal, and urban search and rescue are applications where manipulators can be especially useful. These application domains are characterized by environments that are often dangerous or inaccessible to humans. We now explain how manipulators help operators to safely perform tasks in each of these applications.

Planetary exploration: The purpose of the Mars exploration rovers is to allow scientists on Earth to measure and understand the geology, geochemistry, etc., of the surface of Mars. Tasks include (a) navigating over rocky and sandy terrain, and (b) operating specialized instruments mounted on, for example, a five degrees-of-freedom arm [48]. Specific manipulations involve positioning the end of the manipulator (end effector) against a rock, abrading the rock surface, taking measurements with various instruments mounted on the end effector, etc.

With these tasks in mind, the manipulator must be very precise to position the instruments where the scientists want. Such manipulation is challenging not only because of the "soda straw" view, but also because of time delays [1] that can be as large as forty-five minutes round trip [54]. Since precision is important, these challenges make the tasks especially difficult.

Explosive Ordnance Disposal: The purpose of Explosive Ordnance Disposal (EOD) is either to disable or to destroy explosive devices while minimizing risk to humans. A closely related application is improvised explosive device (IED) defeat [30]. These operations are usually done wherever the explosives are found, as it is often dangerous to move them. Wherever possible, it is best to use a manipulator to

[1] This thesis addresses only the "soda straw" view and not time delays.

defeat the explosives while the crew is at a safe distance, rather than have EOD personnel don protective suits and defeat the explosives at close range. This implies that good user interfaces, which make it possible to perform remote work efficiently, are very important. Tasks include exploring and observing hard-to-reach places (under cars, inside cargo compartments), and discharging weapons to deactivate explosives. Manipulations include using tools, lifting, measuring, and pushing [46, 44].

Responsiveness, situation awareness support (especially visual), operating range, and reliability are important attributes for an EOD robot and user interface. Achieving these attributes is difficult due to time delay, the "soda straw" view, and radio interference [44]. EOD personnel must also work under time pressure and sometimes even while under enemy fire [46]. This suggests that reducing the mental workload on EOD personnel would be highly beneficial.

Urban Search and Rescue: The purpose in Urban Search and Rescue (USAR) is to use robots to locate and/or assist victims in places that cannot be quickly or easily reached by human operatives. Tasks include operating in extremely difficult terrain (rubble, inside collapsed buildings, etc.), mapping the environment, and locating victims. Manipulations include moving rubble, opening doors, lifting objects, providing medical support to victims, and stabilizing the environment surrounding victims [21, 8, 55, 35]. Challenges for USAR are similar to those for EOD, except USAR personnel are not often in danger of being attacked. In addition to these challenges, there are extreme physical and mental demands for USAR, which suggests that reducing mental workload is helpful.

While robot navigation is a significant challenge in USAR, manipulation brings additional utility and challenges. Robust manipulation capability would be especially useful for some medical support tasks, such as placing a device to inject the victim, administering oral medicine to a heavily incapacitated victim, checking vital signs, or applying first aid to critical wounds. Manipulation would also support positioning of shoring devices to stabilize a portion of a collapsed structure. In short, any task that requires more precision or dexterity than a mobile platform can provide can benefit from a manipulator.

1.2 Related Work

In this section we will discuss previous work related to manipulation. First we introduce work about the requirements of robot operators in the USAR/EOD domain. Then we will discuss current human-robot interface paradigms and explain the challenges associated with each paradigm.

1.2.1 Robot Operator Needs

Casper and Murphy reported the usage of robots for USAR during the World Trade Center disaster in 2001 [8]. They give a summary of the robotic systems that were considered for deployment as well as systems that were actually deployed in the rubble of the World Trade Center towers. In addition, they describe challenges that robot operators faced during the operation. Lack of sleep was a cause for the robot operators' cognitive fatigue. Because of this, reducing the mental workload caused by the robot interface could improve operator performance. Situation awareness was difficult to maintain due to limited sensors, lack of mapping, and lack of other world state information, etc.

Battelle prepared a report on the needs and uses of robots in law enforcement (including EOD, see [5]). The report includes a detailed survey of robot requirements as well as common tasks using a robot. The salient tasks are (in order of frequency): traverse through "clean" areas, disrupt bombs, negotiate curbs, operate more than 2 hours on one mission, traverse through cluttered areas, relocate objects, inspect and manipulate under vehicles and furniture, and enter doors. The tasks in this list involve a combination of navigating to a location and manipulating something in the world.

Scholtz et al. developed a preliminary test bed for evaluating manipulators in the USAR/EOD environment [44]. In the test bed, test subjects had to perform a variety of tasks using various manipulators. Pilot studies with experts indicated that the most difficult tasks were: navigating slopes, opening doors with knobs, climbing stairs, lifting objects, and looking under low furniture in a confined space. Operators that performed these tasks attributed a high NASA TLX workload rating to the tasks. Scholtz indicated that situation awareness is important for executing these tasks.

In an earlier paper, Scholtz et al. measured critical incidents, which are situations where the robot could damage itself, a victim, or the environment [45]. They presume that lack of situation awareness contributed to occurences of critical incidents.

1.2.2 Current Human-Robot Interfaces

In addition to robot hardware, there is also a need for designing user interfaces. Goodrich describes eight principles for efficient human-robot interfaces [15]. These principles are:

1. implicitly switch interfaces and autonomy modes,
2. let the robot use natural human cues,
3. manipulate the world instead of the robot,
4. manipulate the relationship between the robot and world,
5. let people manipulate presented information,
6. externalize memory,
7. help people manage attention, and
8. learn.

Requirements for an interface vary by the task to be accomplished [52]. Therefore, some of these requirements may not be relevant for a given task. They are useful points to consider when designing an interface with a task in mind.

Many interface paradigms exist for human-robot interaction (HRI). An interface paradigm consists of *information presentation* (display) and an *interaction scheme*. Information presentation shows data collected by the robot, and the interaction scheme describes the autonomy mode of the robot along with the user interface between the human and the robot [11].

One way to think about the relationships between the different components of an interface paradigm is the model-view-controller (MVC) software engineering pattern [40]. Applied to HRI, the model corresponds to all of the data recorded by the robot sensors, the view corresponds to the interface display, and the controller corresponds to the interaction scheme.

We now give some examples of common interface displays followed by examples of common interaction schemes.

1.2.3 Interface Displays

The *conventional* display shows information from the robot in several separate windows. A lot of information can be displayed very accurately using this approach. This presentation requires the operator to integrate information from all of the windows. As a result, the operator may have high workload and low situation awareness [37, 58]. The Idaho National Laboratory (INL) "2D Interface" is an example of a conventional display (see Figure 1.2). Another example of a conventional interface, shown in Figure 1.3, was designed by Yanco et al. [57]. This interface shows important information (such as distance to obstacles) surrounding the camera video where the operator's attention is often focused.

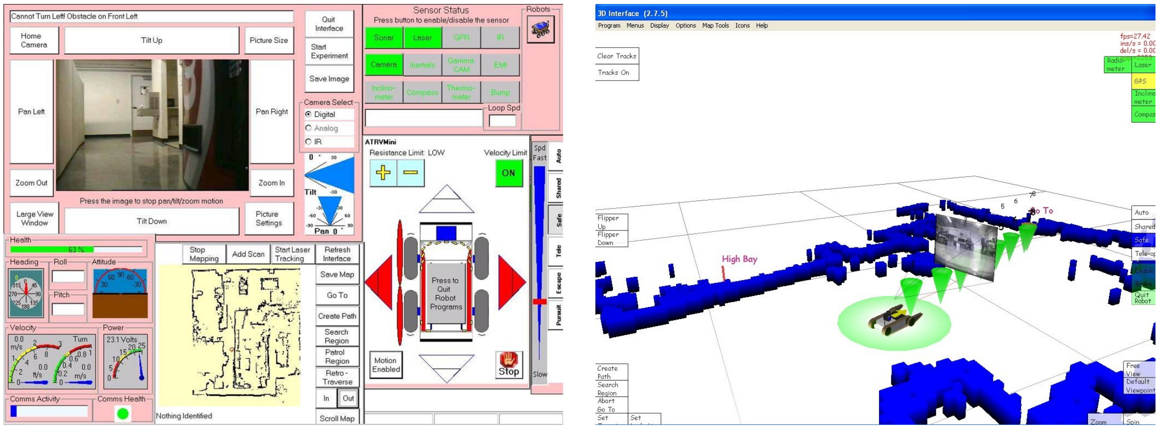


Figure 1.2: Left: INL conventional robot interface. Right: INL/BYU 3D augmented virtuality interface.

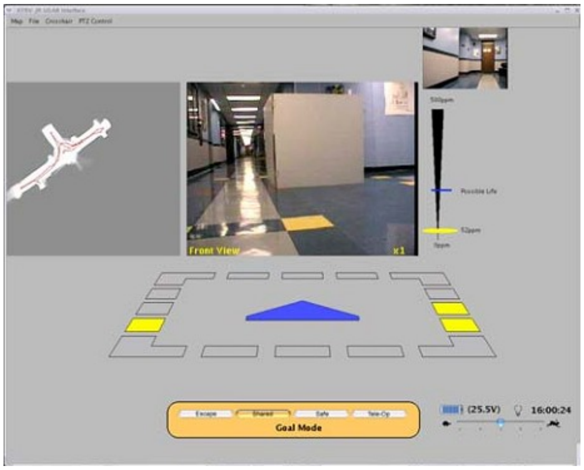


Figure 1.3: Conventional user interface adopted from [57].

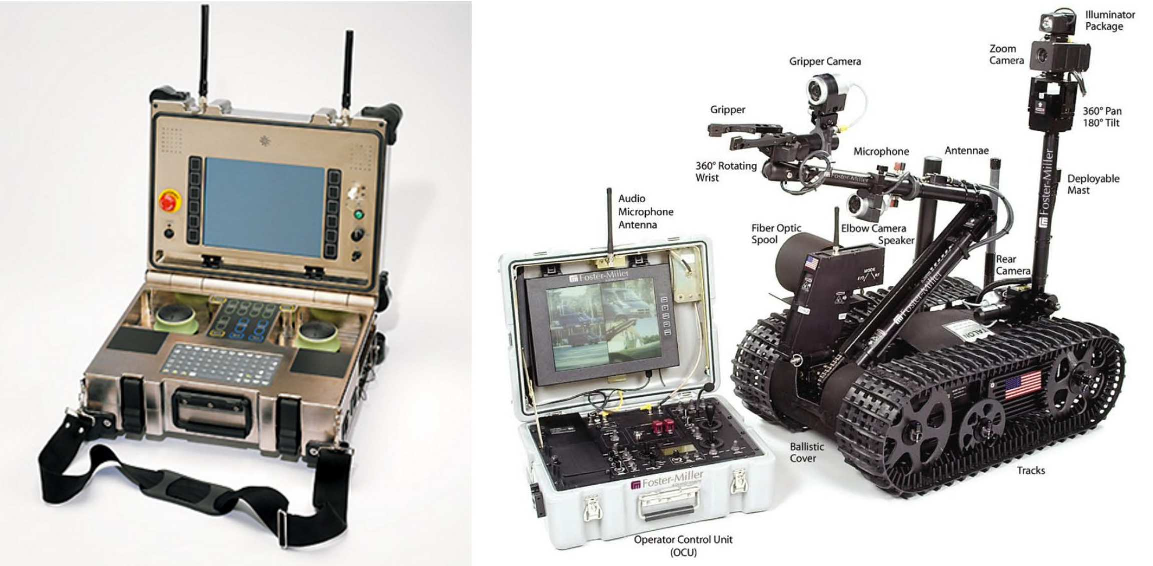


Figure 1.4: Left: iRobot Packbot OCU. Right: Foster-Miller Talon mobile manipulation robot and OCU.

Many current EOD robot operator control units (OCUs) use some variation of the conventional display (see Figure 1.4). Typically, the OCU displays video from multiple cameras on a screen. Other information is shown with lights, gauges, and other instruments similar to the dashboard of a car. Operators may have high workload from needing to simultaneously integrate the different video views while paying attention to the "dashboard" instruments. Situation awareness may be difficult to acquire due to, for example, lack of a map and proximity sensors (such as lasers, sonar, etc.).

In contrast to the conventional 2D display, a *virtual environment* display attempts to reconstruct a 3D virtual replica of the real environment around the robot. Multiple perspectives allow the operator to see the environment from novel viewpoints generated by the interface. Some virtual environment displays show predicted future events based on a simulation of the robot (for example [18]). This allows the operator to see the expected result for a sequence of commands.

The Rover Sequencing and Visualization Program (RSVP) for controlling the Mars Exploration Rovers (MER) is an example of a virtual environment interface.

RSVP consists of a computer textual interface for entering commands (RoSE) coupled with a 3D virtual environment display (HyperDrive) for visualizing the command queue [18]. Interactive control of the rovers is infeasible due to round trip time delay between six and forty-five minutes. Instead, command sequences for an entire sol (Mars day) are uploaded to the rovers, and the rovers autonomously perform the scripted sequence.

Nguyen et al. present several NASA virtual reality interfaces for scientific and space robotics, the most similar to ours called Viz [36]. Viz is a highly flexible, modular interface system that can display virtual models of robots in context with data from various sources, such as automatically generated 3D terrain models. The interface and visualization we present is similar in many ways to Viz; however, we have designed with real-time operation in mind, while Viz seems to be designed more for planning due to the long delays associated with extraplanetary robotics.

As an alternative to virtual environment displays, *mixed reality* displays superimpose virtual views of data over camera imagery from the real world. Sometimes mixed reality is called augmented reality. Information overlaid on camera imagery shows the relationship between the world and the virtual information in a directly perceivable manner. This can reduce the workload on the operator and increase situation awareness.

Drascic and Milgram present a mixed reality interface in [34]. They show a stereoscopic camera view of a robotic arm superimposed with virtual information. The operator can use tools such as virtual landmarks, virtual tethers, virtual pointers, and virtual tape measures to assist in planning robotic arm movement.

In contrast to sophisticated mixed reality interfaces that require specialized equipment, Keskinpala et al. designed a simpler PDA-based mixed reality interface [25]. The display shows the video feed overlaid with semi-transparent virtual control buttons and data from range sensors. When a USAR operative is wearing safety gear, the large buttons on the display are more usable than intricate controls like a computer keyboard. In addition, the PDA platform makes transportation between search areas easier than other platforms.

Augmented virtuality (AV) views are similar to augmented reality with some key differences. In augmented reality, the interface view is typically limited to the camera video. In other words, any virtual elements in the interface are drawn onto the camera video as an overlay. In augmented virtuality, the interface view can expand beyond the camera video. In addition, the camera video is simply an element in a larger virtual scene. Users can view the virtual scene from any position and orientation. Multiple viewpoints are useful when information is available outside of the camera's field of view, such as range information from a laser.

Virtual environments, mixed reality, and AV displays are variations of the class of ecological interfaces. As discussed by Vicente in [50], the term *ecological interface* originates from psychologist James J. Gibson who defines *ecological psychology* as study of human-environment relationships in a natural setting, as opposed to a laboratory setting. An *ecological interface* makes the relationships, constraints, and affordances of a system perceivable by the operator. In terms of remote robot control interfaces, an ecological interface might display the robot, its surroundings, and camera video in a single integrated display so that the position of the robot and camera video relative to the environment are directly perceivable. Any of the previously discussed interface paradigms can potentially exhibit ecological characteristics.

Ricks et al. developed an AV interface for teleoperating a robot [41]. The display shows a simplified model of the robot, obstacles picked up by laser and sonar, and the most recent camera image in a 3D virtual scene. The robot model is displayed in a third-person "over the shoulder" view so obstacles behind and to the sides are visible. They show that this display increases performance for a teleoperated robot in a mapping task.

In contrast to the 3D virtual environment of Ricks' interface, Kaminka and Elmaliach designed a simple ecological interface to show the relationship between multiple robots on a team [23]. One part of the display shows video from each of the robots on the team, and another part shows the spatial relationship between the robots. The relational tool significantly improved performance over a conventional display in a formation maintenance task.

A more sophisticated AV display is shown in Figure 1.2. This interface shows an "over the shoulder" view of a 3D virtual icon of the robot surrounded by a map built from the robot's sensors. The video feed from the robot is projected onto a plane near the robot icon. The position and orientation of this plane reflects the pan and tilt state of the camera. Nielsen, the designer of the display, compares the utility of a map and video when performing a navigation task [38]. Results from Nielsen's work show that the map is more useful than video for navigating in some cases, which implies that the utility of given data depends on the task.

Nielsen's AV interface displays obstacles surrounding the robot in a planar fashion. Because of this, the appearance and height of obstacles is unknown. Kelly et al. present an AV interface that uses a range sensor combined with a color camera to construct a 3D virtual display of the environment around the robot [24]. They show that performance for a vehicle navigation task in an outdoor mixed-terrain environment is better with the AV interface compared to a video-only interface.

Nielsen also showed that ecological interfaces can be more effective for teleoperation than a conventional interface in a navigation task [37]. This was confirmed by Yanco et al., who compared INL's AV interface (developed by Nielsen, shown in Figure 1.2) against the UMass Lowell USAR interface (shown in Figure 1.3). All eight participants searched an arena for victims using both INL's system and UMass Lowell's system. Their results indicated that the operators covered more area in a search task with INL's system than with UMass Lowell's system. There was no significant difference between the number of victims found in the interfaces. Although operators bumped into objects behind the robot less often with UMass Lowell's interface, Yanco et al. attributed the addition of a rear-facing camera on their robot to the increased awareness of obstacles behind the robot.

Ferland and Michaud et al. present an AV interface that projects video onto virtual walls constructed from laser range data [14]. The resulting quality of the environment is very good for vertical, planar walls. They evaluated their user interface in a navigation and observation task where the operator teleoperated a robot around a small office area and indicated the location of several orange cones. Most users that tested the interface prefer the extruded video projection to a mesh created with a stereo disparity technique.

Perhaps the most sophisticated mobile manipulator is the "Segwanaut." Ambrose et al. created this teleoperated mobile manipulator using a Segway RMP mobility platform combined with the NASA Robonaut manipulator [1]. This combination exhibits exceptional capabilities for mobility and manipulation alike. Their work focuses on telepresence as the mode of control. Telepresence interfaces attempt to create an exact replica of the robot's environment for the human operator. On this platform, telepresence shows productivity higher than what can currently be achieved with higher level tasking; that is, by using sophisticated autonomy. Unfortunately, telepresence requires specialized equipment in a specific environment and demands full operator attention. Consequently, the operator has a much higher workload with telepresence than might be necessary with higher level tasking.

Haptic interfaces and devices are another promising way to give operators improved situation awareness. While visual and auditory feedback are common for user interfaces, haptic feedback uses the sense of touch to give information to the user. For example, a haptic device could push the operator's hand in order to steer the robot away from an obstacle. In addition, some haptic telemanipulation research strives to replicate the touch and feel of objects the manipulator interacts with. For some haptics and telemanipulation background, see for example [19, 27].

Operators must commonly control robots under a time delay that increases mental workload [44]. One way to reduce the effects of time delay is a technique called *quickening*. Quickening is a form of predictive display that shows the predicted future state of a system based on current commands [52]. For example, a quickened remote manipulation display would probably show the predicted future position of a robot arm. One of the primary benefits of quickening is that operators have immediate feedback for their actions, thus reducing workload associated with time delay.

1.2.4 Interaction Schemes

Besides the interface display, another part of an interface paradigm is the interaction scheme. While interaction schemes as defined previously include the interface display, robot autonomy, and user input method, we focus on the user input method in this section. There are more interaction schemes than we can discuss in the scope of this thesis. For our research, frames of reference play a greater role than what is usually discussed in terms of input methods and robot autonomy. As such, we will give a brief summary of interaction schemes and then discuss frames of reference.

Autonomy for interaction schemes varies from full human control (*teleoperation*) to nearly full robot control (*supervisory control*). Just beyond teleoperation is *safeguarding*, which nullifies a human command if the robot's sensors show that the command will cause an error or damage the robot. Next is *shared control*, where the robot autonomy uses human input as an expression of intent and the autonomy determines the best course of action based on sensor feedback. Just before supervisory control is *traded control*, which is when the human sometimes has control over the robot (using a lower level of autonomy), and the robot has control at other times. For a more descriptive treatment of interaction schemes, see for example [47].

For interaction schemes that have a lower level of autonomy (between teleoperation and shared control), the frame of reference for human-robot control is an important factor. Frames of reference indicate the coordinate system (or coordinate frame) for interpreting directions. For example, if someone tells you to move left, do you move to your left or their left?

Hiatt et al. explored various frames of reference for teleoperating a robot. They performed a user study to evaluate what they call viewpoint-centric, robot-centric, and task-centric frames of reference. Viewpoint-centric means that directions are based on video coming from a camera external to the robot. Robot-centric means that directions are based on the robot, as if the operator were sitting in the robot. Task-centric means that directions are based on a particular object being manipulated by the robot. In Hiatt's example task, the user must position a beam in a particular fashion, so the task-centric frame of reference is relative to the beam. Results from their study indicate that mental workload for all three frames of reference is similar, but the task was performed fastest with the task-centric frame of reference, followed by the robot-centric frame of reference. [20]

When the autonomy of the robot is on a higher level, the human has less direct control over the robot, so frames of reference play a smaller part. The following three examples show remote manipulation interaction schemes that use a form of supervisory control.

Kulkarni et al. have implemented a system that can autonomously navigate a mobile manipulator given a target [28]. In their system, the operator clicks a 2D video image, and the system uses an internal range image to determine the corresponding 3D world point. The system then navigates the mobile robot and arm autonomously and retrieves an object at the location given.

Kelly et al. implemented a nearly identical interaction scheme and performed a two-user study [24]. Their results indicate that to grasp an object within reach of the manipulator, the autonomous system was on average 13% faster than human teleoperation. When the object is out of reach of the manipulator and the mobile platform first had to be moved closer, the autonomous system performed slower in most cases. The performance of the autonomous system may have been affected by a lack of sensor accuracy.

Tsui and Yanco et al. designed an interface for tasking a wheelchair-mounted manipulator arm [49]. Unlike other related work, the operator is in close proximity with the robot, so there is less need for the user interface to support situation awareness. The interface facilitates interaction between the robot arm and the operator. The operator indicates an object of interest with a joystick or touchscreen, then the robot arm autonomously moves close to the object.

1.3 Thesis Statement

A 3D augmented virtuality interface for remote manipulation that displays range image data *ecologically* can potentially increase performance compared to conventional interfaces. This interface can improve performance by reducing mental load, increasing situation awareness, and reducing time to complete tasks.

1.4 Thesis Organization

The remainder of this thesis is organized as follows. In Chapter 2, we present a paper published in the AISB 2009 conference. We describe our user interface, experiment design, and results from a user study. In Chapter 3, we describe changes from the interface presented in Chapter 2. We also show preliminary evidence to support our claims that the changes we made improved performance. Chapter 4 summarizes the thesis and indicates some possible avenues for future work. Appendix A describes additional changes to the interface that have not been formally evaluated.

Chapter 2

Supporting Remote Manipulation with an Ecological Augmented Virtuality Interface

This chapter is a paper published in the AISB 2009 conference [3].

2.1 Abstract

User interfaces for remote robotic manipulation widely lack sufficient support for situation awareness and, consequently, can induce high mental workload. With poor situation awareness, operators may fail to notice task-relevant features in the environment often leading the robot to collide with the environment. With high workload, operators may not perform well over long periods of time and may feel stressed. We present an ecological visualization that improves operator situation awareness. Our user study shows that operators using the ecological interface collided with the environment on average half as many times compared with a typical interface even with a poorly calibrated 3D sensor; however, users performed more quickly with the typical interface possibly because of the poor calibration.

2.2 Introduction

A manipulator is a machine that can operate on its surrounding environment. Manipulators are commonly modeled after a human arm and hand, and aim to perform actions similar to what humans can do with their arms and hands. A *remote* manipulator is located away from the operator, so that the operator depends on sensors near the robot to provide information about the remote environment. Figure 2.1 shows an example of a remote manipulator and supporting sensors. A *mobile* manipulator is manipulator mounted to a mobile base, and is often also a remote manipulator.

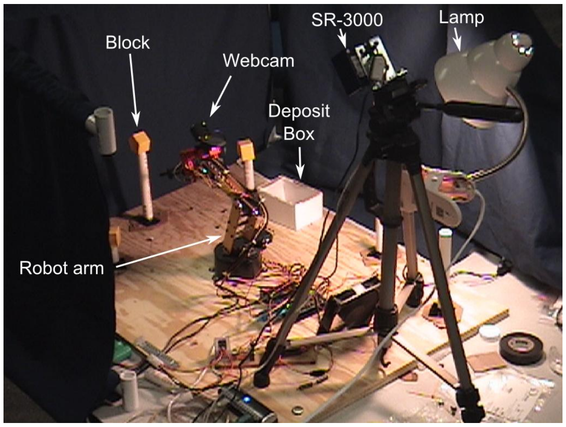


Figure 2.1: Remote manipulator robot, surrounding environment, and supporting sensors.

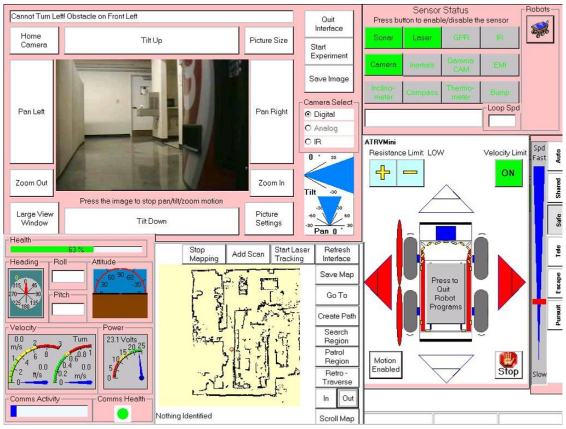
Mobile manipulators are useful tools in areas that are dangerous or inaccessible to humans. For example, they are used in planetary exploration (Mars rovers), urban search and rescue (USAR), and explosive ordnance disposal (EOD) [43, 8, 44]. Operators use the manipulator to grab, push, poke, operate tools, and generally try to do what people do with their hands. In order to safely benefit from the capabilities these robots provide, a human operator must control them remotely. Operators can use the mobile manipulator to perform a variety of tasks from a safe location. Although supporting mobile manipulation is our end goal, the complexity involved is more than we can test in one study, so we limit the scope of this work to remote manipulation as a step toward supporting mobile manipulation.

One difficulty in remote manipulation is for the operator to understand the situation given limited data. In remote situations, the only connection the operator has with the robot is the user interface, and the operator is limited to whatever information the robot or the interface can provide. This is sometimes referred to as looking at the world through a “soda straw” [56]. Problems that come from this limited view of the world include missed events, difficult navigation in new situations, and an incomplete understanding of the explored world [56]. For example, depth perception is a very important aspect of understanding the environment for manipulation, but single-camera video displays provide limited depth perception in unstructured environments.

Factors that affect performance in remote manipulation tasks include the physical capabilities of the robot, the autonomous behaviors of the robot, the user interface, and the capabilities and skill of the human operator. Although all of these factors affect performance, we can focus on a few factors that a system designer can control. Assuming that we have a skilled operator, the design variables that we can freely manipulate include the following key elements: robot capabilities, the robot behaviors, and the user interface. When the robot does not have the required capabilities for a task, then the usability of the user interface or autonomous behaviors make little difference for accomplishing the task. On the other hand, when the robot does have the necessary capabilities, then the user interface and autonomous behaviors can significantly affect performance.

Typical robotic user interfaces present several sets of information in multiple windows with each window showing a distinct set of information (see, for example, Figure 2.2). When information is displayed disjointly in this manner, operators must mentally integrate the displays to understand relationships between different sets. While such interfaces can be very useful, including for testing and debugging a system, such a design can increase the mental workload on operators. By splitting the information into multiple windows, operators have poorer manipulation or task-specific situation awareness as they may miss an important event from one display while focused on a different display.

We present an ecological augmented virtuality (AV) user interface.



Cannot Turn Left! Obstacle on Front Left

Press the image to stop pan/tilt/zoom motion

Sensor Status

Press button to enable/disable the sensor

Camera Select: ● Digital ○ Analog ○ IR

Health: 63 %

Power: 23.1 Volts

ATRVMini

Resistance Limit: LOW

Velocity Limit: ON

Nothing Identified

Figure 2.2: Idaho National Laboratory (INL) mobile robot control interface with multiple windows. Adopted from [6].

As discussed by Vicente in [50], the term *ecological interface* originates from psychologist James J. Gibson who defines *ecological psychology* as study of human-environment relationships in a natural setting, as opposed to a laboratory setting. Ecological interface design aims to make relationships in the environment perceptually evident to the user in order to minimize workload for understanding those relationships. The term *augmented virtuality*, as defined by Paul Milgram, means the interface presents real-world information (e.g. camera video) inside a virtual environment [33]. Augmented *reality*, on the other hand, displays virtual information on top of camera video.

The ecological AV interface is inspired by work done by Milgram [34], Ricks [41], Nielsen [37], and Michaud [32]. The primary component of the interface is an ecological visualization for obstacles and objects around the robot. The visualization displays a 3D range scan in context with a virtual robot arm to show the spatial relationship between the robot and its surroundings; see Figure 2.3. We claim that the visualization gives the operator more viewpoints than are typically afforded by video camera systems and provides a grounded frame of reference, potentially improving performance.

The primary contribution of this research is the analysis of the benefits of using an ecological AV interface to support robotic manipulation. Another contribution is the interface design itself; in particular, the ecological 3D range scan visualization. We examine the effects of such an approach on overall performance and situation awareness.

For the remainder of the paper we will show how our approach improves situation awareness in a remote manipulation task. We discuss the motivating factors in the design of our user interface. We describe our experiment design, explain how the results show improved situation awareness, and present conclusions.

2.3 Related Literature

Milgram and Drascic et al. show that stereo vision displays improve performance and accuracy in manipulation tasks, and that virtual tools augmenting the display further improve performance and accuracy [34]. People were able to more accurately position a robot arm with virtual tools including tethers, tape measures, pointers, landmarks, and object overlays. In this paper we look at a method that gives some of the benefits of stereo vision without the high bandwidth requirements and viewing hardware.

Nguyen et al. present several NASA virtual reality interfaces for scientific and space robotics, the most similar to ours called Viz [36]. Viz is a highly flexible, modular interface system that can display virtual models of robots in context with data from various sources, such as automatically generated 3D terrain models. The interface and visualization we present is similar in many ways to Viz; however, we have designed with real-time operation in mind, while Viz seems to be designed more for planning due to the long delays associated with extraplanetary robotics.

Nielsen shows that an augmented virtuality interface improves performance for mobile robot navigation and exploration tasks [37]. The interface displays a live video feed from a pan-tilt-zoom camera in context with a 2D map built with a laser rangefinder. This paper explores whether there are similar improvements when applying ecological augmented virtuality design to a manipulation task.

Tsui and Yanco et al. designed an interface for tasking a wheelchair-mounted manipulator arm [49]. Unlike other related work, the operator is in close proximity with the robot, so there is less need for situation awareness. The interface facilitates interaction between the robot arm and the operator. The operator indicates an object of interest with a joystick or touchscreen, then the robot arm autonomously moves close to the object.

2.4 Methods

Our method of improving user interfaces for remote manipulation has four steps: choose a relevant problem, identify interface goals and requirements for the problem, design interfaces using different approaches, then test to see which approach is better. Thus, we break up the discussion of methods into four respective sections.

2.4.1 Application

The real-world application most similar to what we will use for experimentation is remote sample acquisition. The objective in sample acquisition is to navigate the robot through an environment and use a manipulator to collect samples, such as rocks, and analyze them on-site or return them to a lab for further analysis [43]. We will mimic this task and analyze the performance of operators using a variety of interface designs.

2.4.2 Interface Goals and Requirements

To guide the design of our interface, we choose some goals and requirements inspired by Goodrich and Olsen [16]. The primary goals are to:

1. maintain a manageable workload and
2. support situation awareness.

Requirements for the interface are to:

1. integrate information from multiple data sources,
2. provide views of the environment from multiple perspectives,
3. act as a grounding frame of reference for interaction with the robot, and
4. externalize memory.

Goals

Maintain a manageable workload. A manageable workload means that the operator can perform the tasks without feeling overly stressed, operate in a situation where there are multiple competing tasks, and sustain operation for some period of time [47, p. 301]. Reducing the operator workload is especially important in EOD and USAR, where operators may have to work while physically and mentally fatigued [8]. To reduce the workload, we must (a) think about information presentation that leverages human perception and (b) devise control methods so that users can build correct, simple mental models [52, p. 132].

Support situation awareness. Endsley defines situation awareness as "The perception of the elements in the environment within a volume of time and space, the comprehension of their meaning, and the projection of their status in the near future" [12]. In the context of remote manipulation, situation awareness involves understanding the environment surrounding the robot to avoid obstacles and to position the manipulator with sufficient precision to accomplish a task.

Requirements

Integrate information. When information coming from different sources is presented in separate contexts, it can be difficult to see and understand relationships that may exist between different types of data. For example, if robot position is displayed as text on one computer screen, and a map of the environment around the robot is displayed as an image on a separate computer screen, it may be difficult to quickly determine where the robot is located on the map. Integrating information can reduce mental workload by reducing the number of mental transformations that must take place to establish relationships between different sets of data [53, p. 189].

View the environment from multiple perspectives. Providing the ability to view multiple perspectives can help operators to acquire situation awareness, especially since they must understand a 3D environment on a 2D display. Several views can help the operator understand the relationship between the robot and its surroundings. This is particularly useful in manipulation tasks where depth perception plays a large role. Although a single camera only provides a single perspective, creating a virtual environment can be a way to provide multiple perspectives [37, 10]. With a virtual environment, the operator can move a virtual camera around to change perspectives.

Grounded frame of reference. Macedo et al. [31] describe how people perform better when information display is grounded with control. This means that the robot moves in a way the operator expects. When the display is rotated it may not immediately be clear what direction the robot arm should move when given a command. Grounding simplifies the operator's mental model of how the controls affect the robot arm by connecting the motion of the arm to the display [52, p. 135]. A simpler mental model means that the mental workload is lower and performance is higher. More recent work by Hiatt et al. suggests that a task-centric frame of reference yields better performance than robot-centric or view-centric frames of reference [20].

Externalize memory. When the interface includes real-time data, such as robot telemetry or a live video stream, it can be difficult to remember what happened in the past. For example, if an obstacle comes into view on the camera, then leaves the view, it is difficult to remember where the obstacle is after a while [16]. Relieving burden on short-term memory can help reduce workload and improve performance. Externalizing memory means we place information in the interface so the operator no longer needs to keep what happened in working memory [53, p. 191].

2.4.3 Interface Design

We designed a new interface with these requirements in mind. The interface is a type of 3D ecological AV display similar to the work done in [34, 37, 41, 32]. Our interface differs from [34] in that they use stereo vision with a fixed camera viewpoint and overlay virtual elements as in traditional augmented reality, whereas we create a virtual environment to display a 3D model that can be seen from several viewpoints. While [37, 41, 32] apply ecological AV interfaces to real-time robot navigation, we look at this type of interface for real-time robotic manipulation.

In order to reduce workload on the operator and improve situation awareness, we implemented a display that: (1) integrates information from different sources, (2) provides views from multiple perspectives, (3) gives a grounded frame of reference, and (4) externalizes memory. We explain the design of each of these parts and how each meets the requirements in the previous section.

Integrating information into one display may reduce the number of mental transformations the operator must perform to establish the relationships between different data sources, such as robot state, map, manipulator arm state, and camera video. We integrate information by rendering virtual representations of the data sources such that their spatial relationships are directly perceivable. We render a virtual graphic of the manipulator arm that reflects the position and orientation of all the robot's parts. We also display a 3D scan generated from the range image data that we explain in greater detail later in this section. See Figure 2.3 for a screen-shot that depicts these representations, and also shows the view the user has while using the interface. By integrating these sources into a single display, the mental workload on the operator is potentially reduced [41].

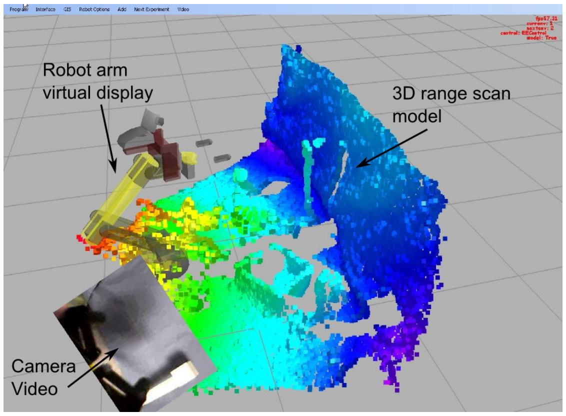


Figure 2.3: Augmented virtuality user interface for remote manipulation. This is the user's view in the interface.

Since we display information in a virtual environment, we can provide **virtual viewpoints** from anywhere in the environment. The user is enabled to change the "look-from" point of the virtual camera, and the focal point of the camera is set on the center of the workspace of the robot. These adjustments give multiple perspectives to the user and enhance understanding of the world surrounding the robot arm.

While multiple perspectives are nice, they introduce a problem with controlling the robot arm. When the user indicates a direction for the robot arm to move, should the robot move according to its own frame of reference or from the virtual perspective's frame of reference? In essence, this is a problem of **grounding**. We implemented two control methods: the *robot frame-of-reference control* method is direct, independent control of each joint, and the *grounded control* method is a "flying the end effector" control method where the user controls an invisible target point to which the arm reaches. In other words, the user moves the target position for the gripper, and the robot decides how to move each joint to reach that position. Controlling the end effector has been shown to reduce workload dramatically compared to controlling the joints, although situation awareness is negatively impacted when controlling the end effector [22]. We hypothesize that the negative situation awareness effects will be reduced as a result of the integrated nature of our display. With end-effector control, the orientation of the virtual view affects the direction the target point moves (see Figure 2.4). In other words, commanding the target point to move right will cause the target to move right in the computer screen frame of reference, and not necessarily in the robot's frame of reference. This can cause confusion while focused on the camera video (because the video is in the robot's frame of reference), so we rotate the camera video to give a sense for what direction the robot will move relative to the camera video. The grounded control connects the control of the robot arm to the virtual display of the world.

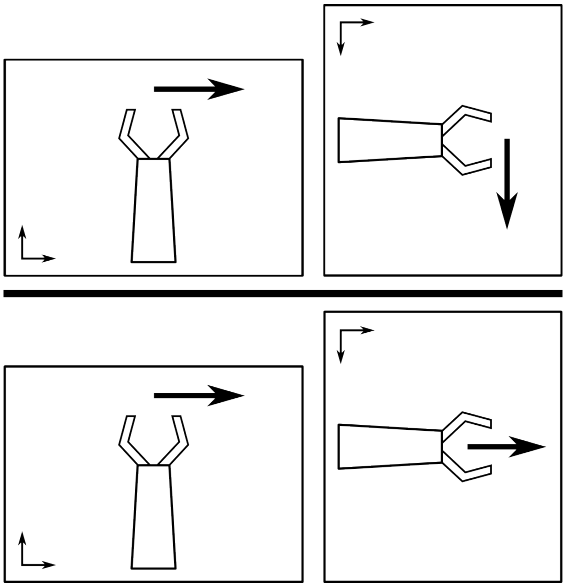


Figure 2.4: Two frames of reference for controlling a robot arm. All boxes show the robot arm from a top-down perspective. The robot arm is in the same position in each box, but the view is rotated. The top row shows the direction the end effector would move when commanded to move right when using a robot-centric reference frame. The bottom row shows the direction when using a view-centric reference frame.

Part of the world around the robot is represented by a 3D scan. The scan is captured from a ranging camera (but only when the user requests the scan) and then displayed as a set of colored splats. The color assigned to each splat is a function of depth in a "heat map" gradient. We align the 3D scan with the graphic of the robot arm simply by manual measurements that result in a degree of misalignment. The misalignment is not constant and in some cases appears to be perfect, but in some cases is off by as much as 2 cm. We leave improved alignment to future work.

The 3D scan presents a perceptual display of information about the world and thus potentially improves situation awareness while reducing workload. Such a display also provides **externalized memory**, as the 3D scan provides a greater field-of-view than the video camera.

2.5 User Study

In the user study we evaluate the usefulness of the 3D scan compared with the camera video as well as two different control methods: (1) joint control and (2) end-effector control. We hypothesized that users would perform tasks most quickly and with the fewest collisions while using the 3D scan, camera video, and end-effector control together. We measure the performance of users as they collect blocks with a robot arm in two control-type conditions (joint and end-effector) and three visualization conditions (scan-only, video-only, and scan+video).

2.5.1 Experiment Setup

For our experiment, we designed a simple object collection task. Participants in the experiment were required to pick up 3 small yellow blocks (2.8 cm wide) and drop them into a cardboard box (8W x 10L x 6H cm). These target blocks are placed on top of posts with heights 6, 10, and 18 cm. Seven different layouts specify the position for each post and the deposit box to prevent users from memorizing the course. In most of the layouts, the posts are placed within relatively easy reach of the robot arm, but a few configurations require the operator to almost fully extend the arm to reach the block. Curtains obscure the robot from the operator's view, although audio cues are present (primarily when a block falls from its post or is deposited).

The robot system is comprised of a robotic arm, sensors, computers, and a joystick controller. See Figure 2.1 for a photograph of the experiment setup.

The robot arm used in the experiment is a modified Lynxmotion 6 degrees-of-freedom (DOF) arm (5 DOF + gripper). The modified arm can reach approximately 30 cm and lift approximately 0.5 kg. The gripper measures 5 cm when fully open, so there is a 2 cm clearance between the gripper "fingers" and the target blocks. The robot servos provide no position or force feedback. We modified the arm with higher-power servos and added position feedback for the gripper. Feedback on the gripper allows us to display the gripper correctly when an object is grasped; otherwise, the gripper would appear completely closed when grasping an object even though it should appear only partially closed.

The experiment used a ranging image sensor and a webcam. The ranging image sensor is a Swiss Ranger SR-3000 mounted on a stationary tripod above and behind the robot arm. This positioning simulates a configuration where the sensor is mounted above and behind the arm on a mobile robot base. In a real mobile manipulator situation, the base is often stationary during manipulation. Future work will consider the arm mounted to a mobile base. Upon user request, the SR-3000 provides a 3D point for each pixel in the image (dimension 176 by 144 pixels). A generic USB color webcam, mounted to the arm, has a resolution of 320 by 240 pixels and a frame rate of approximately 8 frames per second in our environment. There is approximately a 0.25-second delay on the camera video and robot commands, and about a 1-second delay on the 3D range data.

The controller is a Logitech WingMan RumblePad (see Figure 2.5), and of its features we use two analog thumb joysticks, a digital directional pad, six thumb buttons, and two finger "shoulder" buttons. This controller is very similar to present-day common video game controllers. The thumb joysticks control the motion of the robot arm, the directional pad controls the virtual view orientation, and the buttons control the virtual view zoom, operate the gripper, send the arm to stow position, and request a 3D snapshot.

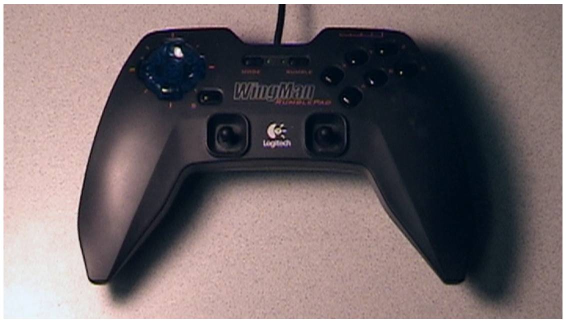


Figure 2.5: Logitech WingMan RumblePad controller used to control the robot arm and user interface in the experiment.

2.5.2 Procedure

Before running the actual tests, each participant watched a tutorial video and performed a small training exercise to become familiar with the system. The tutorial included instructions to work quickly while trying to avoid collisions with the environment. The training exercises allowed the participant to explore how to change the view, how to control the robot arm in two different modes, and how to interpret the 3D range scan. After the tutorial, each participant collected at least one block with the whole system running the same as a normal experiment run, except the data was not recorded. Most participants spent about 15 minutes for the entire training portion. Participants were allowed to ask questions freely during the training. During the actual tests, however, only questions related to how to use the system were answered

Table 2.1: Variations in the user study. Parentheses show acronyms for each variant.

Interface/Control	*Joint control*	*End-effector control*
Video only	Variant 1 (VJ)	Variant 2 (VE)
3D scan only	Variant 3 (SJ)	Variant 4 (SE)
Video and 3D scan	Variant 5 (SVJ)	Variant 6 (SVE)

and not questions regarding the state of the world. Once finished with training, the users transitioned into the actual testing.

For the actual tests, each participant performed the task with 3 out of the 6 possible interfaces. The six variations are listed in Table 2.1. The participants progressed through the tests in pseudo-random, counterbalanced order to reduce order-dependent effects.

At the end of each test, a small survey had the users rate the interface on a scale of 1 to 10 with the following questions: 1. How much effort was required to use this interface effectively? 2. How difficult was it to learn this interface effectively? 3. How much confidence did you have in the robots actions? At the end of all tests, the users completed an additional survey with the following questions: 1. How much time in a week do you spend playing video games? 2. Rank the interfaces in order of your preference (selection from a list). 3. Any comments? At this point the experiment procedure was finished.

The independent variables in the experiment structure are (a) user interface display and (b) manipulator control mode. The user interface displays and manipulator control modes are explained in greater detail in Section 2.4.3.

As shown in Table 2.1, variant 1 corresponds to the typical interface and control model. While not every traditional interface looks like this, we tried to capture the most common elements between these interfaces, namely the live video stream and current robot arm position. We left out some elements common to interfaces that are not relevant to the task, such as battery level indicators or robot "health status"

Table 2.2: Metrics and their respective methods for measurement in the user study.

Metric	Measurement
Manipulation time	Time from first arm movement to block deposit
Operator workload	Time performance, subjective survey
Situation awareness	Collisions
Quality of interactions	# interactions, time spent interacting
Operator preferences	Subjective survey

information. The remaining variants of the interface use different arm control modes and the AV interface previously shown in Figure 2.3.

We measured operator performance primarily by time to complete tasks and number of collisions. To make these measurements, we recorded video footage from a separate camera behind the robot arm, then analyzed the video post-hoc to measure timings and collisions. See Table 2.2 for a listing of metrics and their measurement methods.

2.6 Results

We present results for 30 people that participated in the user study. Twenty-four male and six female students at Brigham Young University participated. Participants were primarily young adults near age 18. Although we present results for 30 people, 34 participants actually attempted the experiment. One participant simply was not able to perform the task under any conditions, two participants were highly distracted, and one performed the task when a bug in the system had significantly slowed things down. Because we want as much as possible to avoid measuring such effects, we do not include data from these participants in the analysis. On three runs the system crashed after an hour of usage, though in each of these cases the participant was nearly finished, so we kept this data in the analysis. The primary results of interest that we found relate to (1) manipulation time, (2) collisions, and (3) subjective measures.

2.6.1 Manipulation Time

Participants performed the tasks fastest in the video-only conditions followed closely by the scan+video conditions, with scan-only slowest by a significant margin. People worked faster with joint control compared to end-effector control. See Figure 2.6 for the distributions of manipulation times, and Table 2.3 for significance between conditions. For the statistical significance analysis, we first use log correction to warp the data to a normal distribution. The D'Agostino-Pearson normality test shows that the warped data is significantly different from normal unless we remove statistical outliers (measured as 1.5 times the distance between quartiles 1 and 3), so we remove these outliers for significance analysis. The results in Table 2.3 show p-values for the corrected data.

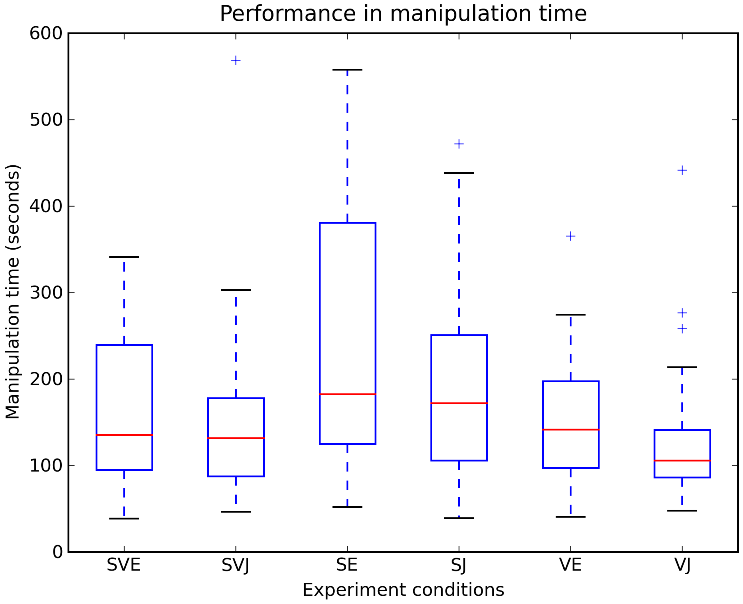


Figure 2.6: Performance measured as time required to perform a manipulation task. Note: To make this plot more readable, we do not show outliers above 600 seconds.

Several things might have caused the difference in time performance but we will discuss only a few. The likely largest factor in the slowness of the scan-only interfaces is misalignment between the 3D range scan and the virtual representation

Table 2.3: Statistical significance analysis with p-values for two sample two-tailed t-tests. Bold numbers are significant at $\alpha = 0.05$. Note: S denotes 3D scan display, V denotes camera video display, E denotes end-effector control, and J denotes joint control.

Conditions	SVE	SVJ	SE	SJ	VE
VJ	**0.014**	0.080	$<$ **0.001**	$<$ **0.001**	**0.021**
VE	0.550	0.575	$<$ **0.001**	0.185	
SJ	0.574	0.065	**0.023**		
SE	**0.010**	$<$ **0.001**			
SVJ	0.287				

of the robot arm. In some cases, the misalignment was as much as 2 cm, which is substantial considering the clearance between the gripper opening and the block is about 1 cm on either side. Participants generally trusted the 3D range scan at first, but as they realized that the alignment was imperfect, they often grew frustrated in scan-only conditions. Because of the misalignment, users had to simply guess where and how the alignment deviates. On the other hand, with video conditions, users had undistorted visual feedback on the position of the gripper in relation to a target block, although depth perception is decreased.

Another factor in time performance involves view adjustments. When the 3D range scan was present, users spent some of the time adjusting the virtual perspective, while with the video-only interfaces almost no adjustments were made. Adjusting the view does not affect the viewpoint of the video camera, so people did not need to adjust the view much. One exception to this is in the end effector control mode, where the video is oriented corresponding to the rotation of the robot arm and the virtual viewpoint to give a sense for what direction the robot arm will move in the camera view. As such, users adjust the view somewhat in end effector control mode, although very little compared to when the 3D scan was present.

Another factor in time performance is the speed of the manipulator arm between control modes. In end-effector control the gripper moves at constant velocity in cartesian space regardless of position in the environment. On the other hand, joint control moves at constant angular velocity, which means that the farther the arm is extended, the faster the gripper moves in cartesian space. Participants tended to operate the arm in a more extended configuration, so end-effector control ended up moving somewhat slower overall.

2.6.2 Collisions

We separate collision types into two classes: (1) environment collisions and (2) target block collisions. Target block collisions are contact between the robot arm and the target blocks, except when a grasp action is underway. Environment collisions are contact between the robot arm and any part of the environment except target blocks. We report the total number of collisions for each condition across all participants for environment collisions in Figure 2.7 and for target block collisions in Figure 2.8. Because of high variability and a small sample size, this data has low statistical significance. We will look at statistical significance for collisions in future work.

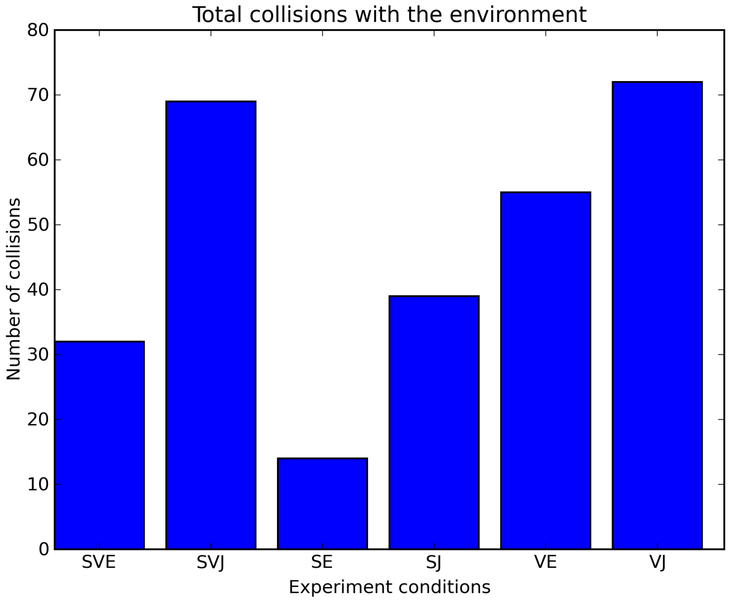


Figure 2.7: Total collisions with the environment.

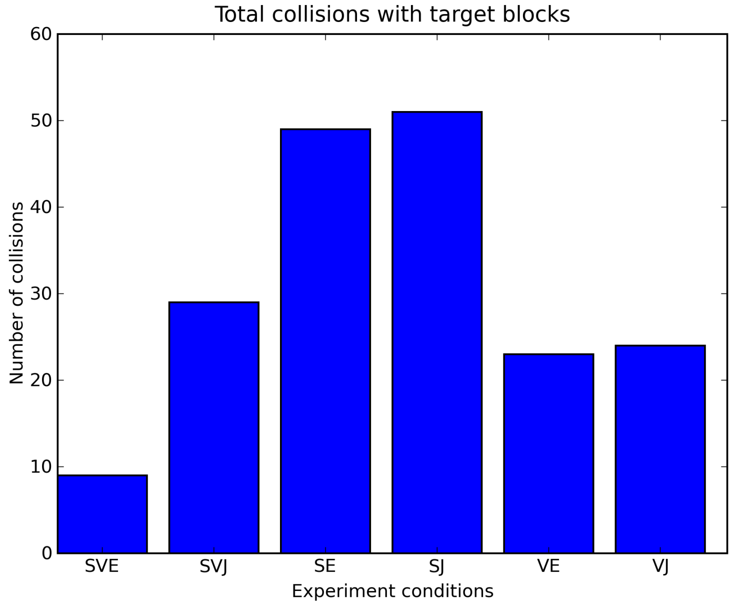


Figure 2.8: Total collisions with target blocks.

In terms of displays, more environment collisions occurred with the video-only and scan+video interfaces than with the scan-only interfaces, while block collisions happened more with the scan-only interface than with any of the video interfaces. The alignment of the 3D scan is poor and coarse, so fine-grained, precise manipulation is difficult with scan-only interfaces. However, the granularity is sufficient for overall situation awareness and avoiding obstacles on a larger scale.

Even though the 3D scan is present in the scan+video interfaces, the video draws users' attention away from the 3D scan and so users appear to fail to notice imminent collisions [26].

Another factor in the number of collisions has to do with sensors. The webcam is closer to the environment than the SR-3000, so less of the environment is visible through the webcam. Such a view gives no information about what is to the sides of the camera, so users bump into things before they realize an obstacle is in the way. The webcam gives no depth information directly, so the user must use visual cues (for example shadows, relative sizes, and motion parallax) to determine depth.

When depositing a block with a 3D scan present, users often would drop the block well above the deposit box, while with video-only interfaces they usually had to make sure the block was inside the box before releasing the block. This reach-in approach generally caused several collisions with the deposit box, whereas the drop-from-above approach kept the arm clear of obstacles.

When we consider control type, more collisions occurred with joint control than with end-effector control. Joint control has a fixed angular resolution; that is, moving the control stick to the left rotates the robot arm a fixed number of degrees. On the other hand, end-effector control moves in cartesian space, so the resolution is finer as the arm extends as compared to joint control. Control resolution impacts the results because as the gripper nears an object, the user must finely adjust the position of the gripper for the final alignment.

We will present ANOVA for pairwise effects in future work.

2.6.3 Subjective Measures

In addition to objective measures for manipulation time and collisions, we also recorded a few subjective measures. Users answered the questions mentioned in Section 2.5.2. The results have sufficiently high variance that no conclusions can be made based on statistical significance, but we discuss trends anyway. For each of the subjective measures, we look at how many times a particular condition was rated better than the other conditions. Then we sum each of the condition votes for a global ranking. We look at subjective results for preference in overall interface, workload, learning required, and confidence in the robot's actions.

More users preferred joint control over end-effector control, except with the SVE condition. Users preferred variants with the 3D scan and video both present (SVE, SVJ). The preference ordering for all interfaces is as follows ($\sim$ denotes indifferent to and $\succ$ denotes preferred to): SVE $\sim$ SVJ $\sim$ VJ $\succ$ SJ $\succ$ SE $\succ$ VE. One possible reason for the low ranking of the scan-only interfaces (SE, SJ) is the misalignment between the virtual robot graphic and the 3D scan display. This disconnect led to frustration in the users, and that is reflected in this ranking. Video rotation (see Section 2.4.3) is probably the cause of dislike in the case of the video-only end-effector (VE) condition.

No obvious classification appears in the results for workload between the conditions. The preference ordering for workload is: SVE $\succ$ SJ $\succ$ VJ $\succ$ SVJ $\sim$ VE $\succ$ SE. Note that although performance was not the highest for the SVE condition, users apparently feel that the workload is lowest for the SVE condition. Strangely, SJ also ranks well for workload keeping in mind the misalignment between the 3D scan and robot arm graphic.

In terms of learning required to effectively operate the interface, joint control was generally easier to learn than end-effector control. The preference ordering for learning is: VJ $\succ$ SJ $\succ$ SVJ $\sim$ VE $\succ$ SVE $\succ$ SE. Joint control is most similar to many common types of control that people may have already been exposed to, such as driving a remote control car. End effector control is a new concept for most people, so more training is required to understand and use it effectively.

When it comes to understanding how the robot will move when given commands, users tended to feel more confident with joint control, except for the SVE condition. The preference ordering for confidence is: SVE $\succ$ VJ $\succ$ SJ $\succ$ SVJ $\succ$ VE $\succ$ SE. Part of the lack of confidence with end-effector control is likely due to the constraint for the target point. Normally it is best to constrain the target point to the robot arm's workspace, but due to implementation error we constrain it to a box that covers the robot's workspace. The side effect of such a constraint is that when the target point moves outside of the workspace, the robot's actions are confusing until the target point moves back inside the workspace. In the future it would be better to constrain the target point to the robot's workspace.

2.6.4 Discussion

The results seem to indicate that the 3D scan may actually increase workload, although it also improves situation awareness. Multiple perspectives with the 3D scan seem to provide better situation awareness, as fewer environment collisions happened when the 3D scan was present. In addition, it appeared from subjective observation that users who changed their view to check alignment had fewer collisions with the environment; future work should test this claim. Only a coarse understanding of the environment was possible with the resolution and calibration of the 3D scan, as indicated by a higher number of object collisions when video was absent. Without a good calibration for the robot arm and ranging camera (both intrinsic and exterior calibration), operators cannot trust the 3D scan to be accurate, and it may encourage them to depend more on camera video.

End-effector (grounded) control was poorly rated except when combined with the 3D scan and camera video even though fewer environment collisions happened with this control. This may be due in part to the view-dependent nature of the control. As an example case, we can consider when a user moved the gripper close to an object, and then changed the virtual viewpoint to gather more understanding. After the view change, the controls also change, and so when the user indicates the same direction on the controller as before, the robot moves differently. Apparently users think more from the robot's frame of reference than the computer screen's frame of reference. Perhaps more training with this control method would improve understanding, appeal, and performance.

Since the scan-only display with end-effector control exhibits the fewest world collisions and the scan+video display with end-effector control exhibits the fewest block collisions and has reasonable time performance, we predict that a better camera video integration with the 3D scan would reduce collisions while maintaining reasonable performance. This better integration could be done by virtually projecting the video in the 3D space near the virtual display of the robot arm, as if the virtual depiction of the camera was projecting the video.

2.7 Conclusions and Future Work

Remote manipulation requires operators to control robots in complex environments with limited information. Typical user interfaces present information to the operator in the form of several camera video feeds and data displayed in several other channels. Although information required to perform tasks is available, the mental workload seems to be high enough that some of the information is forgotten or missed.

Evidence from the experiment suggests that although the ecological AV user interface for remote manipulation slows performance, it can also increase situation awareness and lessen mental workload. We believe that users were able to better understand the spatial relationship between the robot arm and its environment with the ecological interface compared to traditional interfaces. This represents a small step toward providing support for *mobile* manipulation. As this is an exploratory study, we have identified several problem areas with room for improvement. Future work needs to further test and refine these ideas.

In the future, we plan to make the display more integrated and ecological. We can add more autonomy to the robot arm, such as a “move close to that object” behavior. Because users spend much time adjusting the virtual perspective, we can improve the interaction method for adjusting the view with, for example, head tracking. Although we looked at operator workload with subjective measures and based on time performance, we can also use formalized methods such as NASA TLX or more objective measures such as secondary task performance.

2.8 Acknowledgments

We give thanks to Idaho National Laboratory and the employees there who supported this work. We also give thanks to ARL for support.

Chapter 3

Interface Improvements Motivated by the User Study

Trends and anecdotal evidence from the user study, as well as subjective experience, suggest several ways that the interface can be improved:

1. alignment between the 3D scan model and the virtual robot arm model could be better, which could lead operators to trust and use the model more,
2. the internal parameters of the robot arm need more precise calibration, which could also help to improve the alignment of the virtual model,
3. time delay in the controls could be reduced, and
4. the 3D scan model could look cleaner.

In this chapter, we discuss how we improved the interface based on these observations. We also give preliminary evidence to support our claims that the changes that we made do indeed improve performance.

3.1 Stereo Camera Exterior Orientation Calibration

For the user study discussed in Chapter 2, we used a SwissRanger SR-3000 range camera to produce the 3D scan model. This sensor has some nice features and capabilities, but noise and error make it difficult to rely on the depth readings. For example, if we scan our task environment (with blocks on posts) once with a curtain behind the posts, then scan again with the curtain removed, the position of the blocks

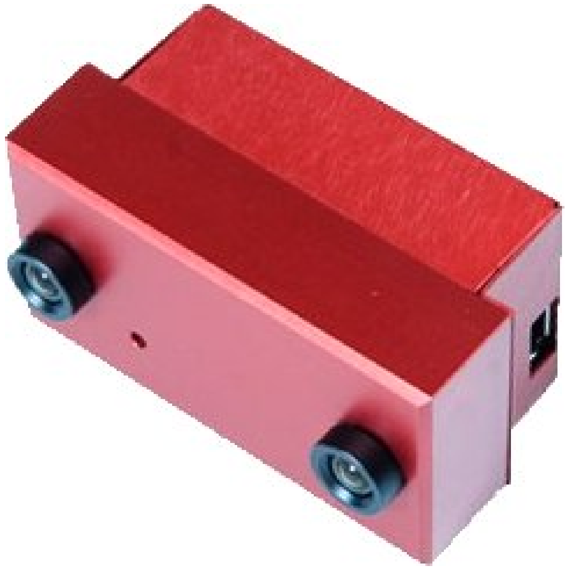


Figure 3.1: Videre STOC stereo camera

in the scan changes by as much as 2 cm. This happens because a pixel's depth error depends on its intensity, the measured distance, and the intensity of surrounding pixels [13]. The error introduced in the 3D scan by the sensor is part of the cause for misalignment between reality and the virtual model. From our observations in the user study, a better alignment may improve performance with the 3D scan model. As a result of the environment dependency and error, we decided that for the precision we desire, we needed to obtain a different sensor.

We replaced the SwissRanger with a Videre STOC-DCSG stereo camera (STOC). The camera has on-board stereo processing to reduce the computational cost of extracting 3D information from a stereo pair of images. We discovered, however, that the most costly part of providing our 3D visualization was creating the model to be displayed in OpenGL, due to the high number of points generated (approximately 50,000 points per scan). Because it is a visible-light stereo camera, occlusions or regions lacking texture are difficult to deal with, so such regions are simply ignored by the camera software.

In order to avoid any problems with alignment between the real world and our virtual display, we decided to properly calibrate the exterior orientation of the camera. We set up several targets, like the one shown in Figure 3.2, with known world positions that the STOC could easily see. Next, the interactive program shown in Figure 3.3 allows the user to indicate where the targets are located in the camera image. From this interaction we can create a multiple-point correspondence between the STOC coordinate frame and the world coordinate frame. With this correspondence, we do a least-squares minimization to find the best transformation from the STOC coordinate frame to the world coordinate frame as described in [2]. Given two 3D points sets defined as $\{p_i\}, i = 1, 2, ..., N$, where p_i and $p_i{}'$ are 3×1 column matrices, we have to solve the following problem:

$$p_i{}' = Rp_i + T + N_i \tag{3.1}$$

where R is a 3×3 rotation and scale matrix, T is a 3×1 translation matrix, and N_i is a 3×1 noise matrix. To solve equation 3.1, we must find R and T to minimize

$$\Sigma^2 = \sum_{i=1}^{N} \left\| p_i{}' - (Rp_i + T) \right\|^2 . \tag{3.2}$$

To see further detail on how the coordinate transformation matrices are found, see [2].

We display two points in the ecological interface for each calibration point: one that represents the point transformed from the STOC coordinate frame (using the current calibration) and one that represents the point in the world coordinate frame. Figure 3.4 shows this display for a poor calibration. This display allows the user to quickly see if any of the points in the STOC coordinate frame are abnormally distant from the known coordinates, and to see how good the calibration is visually.

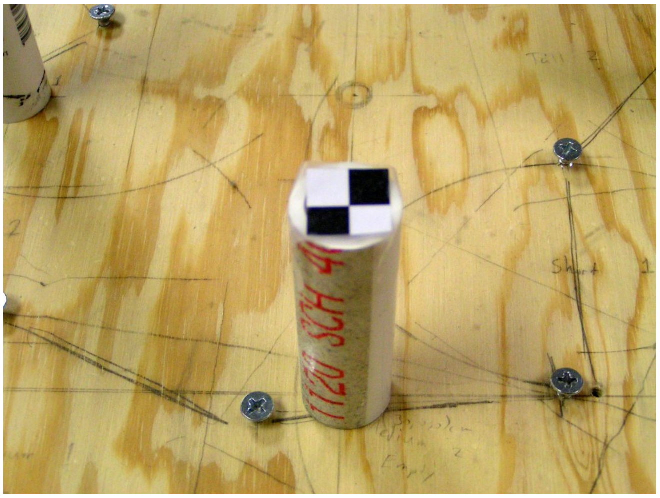
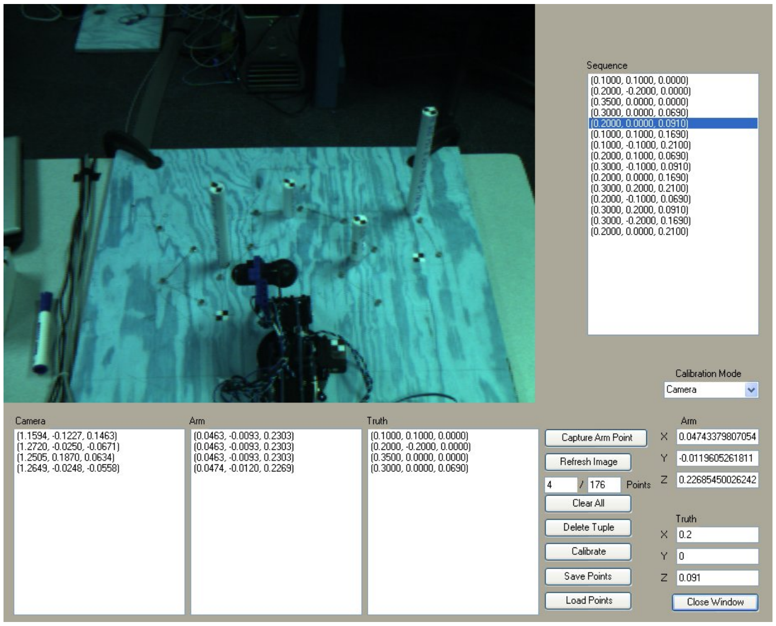


Figure 3.2: Closeup of stereo camera calibration target.



Sequence
(0.1000, 0.1000, 0.0000)
(0.2000, -0.2000, 0.0000)
(0.3500, 0.0000, 0.0000)
(0.3000, 0.0000, 0.0690)
(0.2000, 0.0000, 0.0910)
(0.1000, 0.1000, 0.1690)
(0.1000, -0.1000, 0.2100)
(0.2000, 0.1000, 0.0690)
(0.3000, -0.1000, 0.0910)
(0.2000, 0.0000, 0.1690)
(0.3000, 0.2000, 0.2100)
(0.2000, -0.1000, 0.0690)
(0.3000, 0.2000, 0.0910)
(0.3000, -0.2000, 0.1690)
(0.2000, 0.0000, 0.2100)

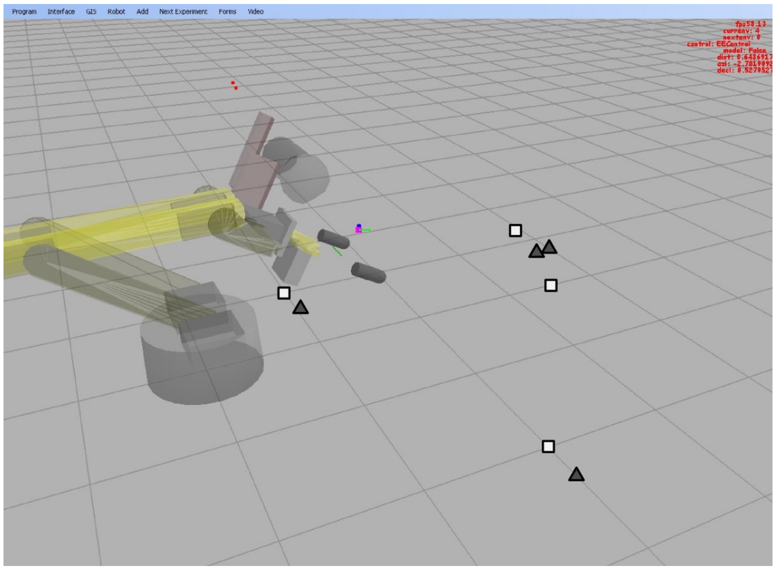
Calibration Mode: Camera

Camera	Arm	Truth
(1.1594, -0.1227, 0.1463)	(0.0463, -0.0093, 0.2303)	(0.1000, 0.1000, 0.0000)
(1.2720, -0.0250, -0.0671)	(0.0463, -0.0093, 0.2303)	(0.2000, -0.2000, 0.0000)
(1.2505, 0.1870, 0.0634)	(0.0463, -0.0093, 0.2303)	(0.3500, 0.0000, 0.0000)
(1.2649, -0.0248, -0.0558)	(0.0474, -0.0120, 0.2269)	(0.3000, 0.0000, 0.0690)

4 / 176 Points

	Arm	Truth
X	0.04743379807054	0.2
Y	-0.0119605261811	0
Z	0.22685450026242	0.091

Figure 3.3: Stereo camera calibration user interface.



```
fps58.13
currenv: 4
nextenv: 0
control: EEControl
model: False
dist: 0.6436917
azi: -2.7019092
decl: 0.5279527
```

Figure 3.4: Interface to display calibration points. Squares represent known world points and triangles represent points sampled from the stereo camera.

The resulting calibration is accurate enough to perform the block collection task using only the 3D scan without making any mistakes (collisions or missed grasps), as long as sufficient care is taken. One novice user in our second user study managed to collect all three blocks without a mistake with only the 3D scan model. This is a vast improvement over the alignment achieved for the first study, where many mistakes were made even when operators were extremely careful.

3.2 Interactive Robot Arm Calibration

Camera calibration, as discussed in the previous section, is only part of obtaining a good overall system calibration; the kinematic model of the robot arm also needs calibration. Our method for camera calibration yields a static model that remains calibrated as long as the camera-robot configuration does not change. Since cameras are usually mounted in such a way that the robot-camera configuration does not easily change, a static calibration is appropriate. For a robot arm, however, collisions or excessive strain could cause the arm's physical configuration to change and the calibration would no longer be valid.

Live, interactive arm calibration may be useful to robot operators doing field work. Live, interactive calibration means that the calibration can be done while the robot is in its operating environment (as opposed to a controlled setting) and the operator is actively involved in the calibration process (as opposed to a fully automatic calibration). With live, interactive arm calibration it may be possible to calibrate the arm remotely using video cameras and/or a 3D scan. The calibration may not be as accurate as what can be created in a controlled setting, but it may be good enough to perform some tasks.

In addition to poor camera calibration, the robot arm had an inaccurate kinematic model for the user study reported in Chapter 2. The poor kinematic model contributed to the 3D scan model alignment issues discussed in Chapter 2. The main effect of misalignment is that the utility of the 3D scan model is essentially unmeasured.

The main reason for the inaccuracy of the kinematic model is an assumption that the mapping from servo position to joint angle was linear. In reality, the weight of the arm causes it to sag as the arm extends, so the mapping is non-linear. In addition, the angle of one joint can depend on the angle of another. For example, the angle of the elbow joint is affected by the angle of the shoulder joint, because rotating the shoulder joint changes the direction of gravity applied to the arm beyond the shoulder joint. Figure 3.5 illustrates how the deflection angle depends on the joint angles. The dashed line in the figure represents the desired elbow angle, and the solid line represents the actual angle. The difference between these two angles is represented by θ_a before rotating the shoulder joint, and θ_b represents the difference after rotating the shoulder joint. In Figure 3.5(a), we see one position of the arm,

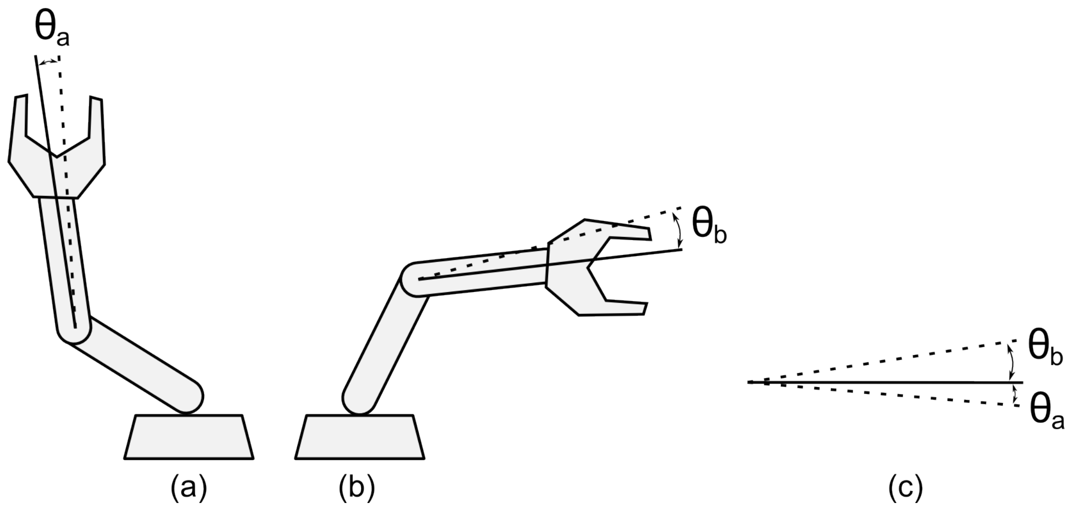


Figure 3.5: Deflection of joints due to gravity.

and in Figure 3.5(b), we only rotate the shoulder joint, but θ_b is different from θ_a. Figure 3.5(c) shows θ_a and θ_b graphically superimposed.

To constrain the calibration problem somewhat, we use only the base rotation joint, the shoulder joint, and the elbow joint. The remaining joints are set to a fixed position. With this constraint only the shoulder and elbow joints are interdependent, so the problem can be solved in two dimensions. Without this constraint, interdependence between the wrist joint and the elbow and shoulder joints would increase the dimensionality of the problem to three dimensions. Although solving the problem computationally in two dimensions extends simply into three dimensions, samples must be acquired by a person. Fewer samples are required with a two-dimensional problem, thus requiring less time from the person collecting samples.

To sample calibration points, we developed an interactive tool. The tool runs inside the user interface, and is activated as a separate window. Before performing this calibration, it is necessary to have a good exterior orientation calibration for the 3D scan sensor. We described how to obtain this calibration in the previous section. The tool first requires that a 3D scan be present. Then the user positions the arm while watching the arm directly or through a camera (instead of through

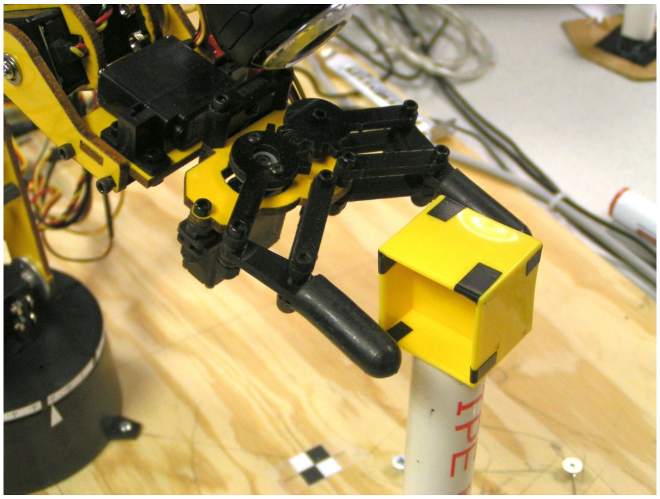
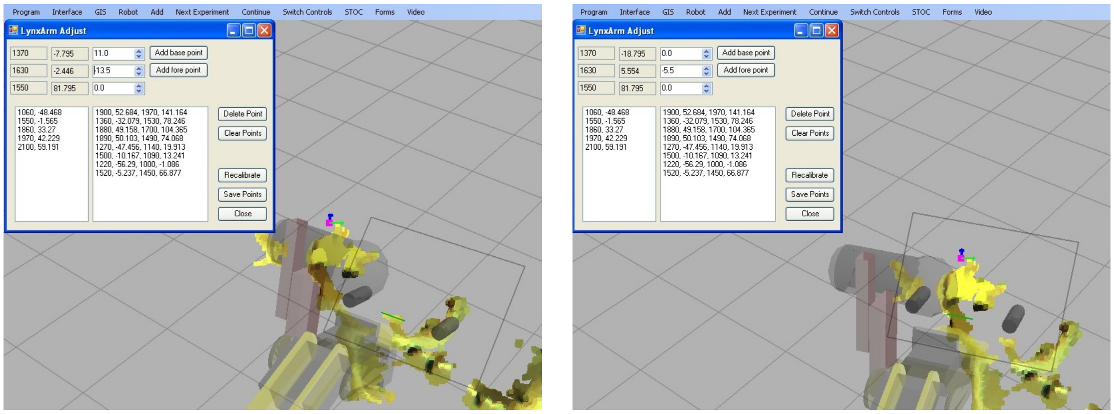


Figure 3.6: Good arm positioning to grasp block.

the interface with the virtual arm graphic) at a point that is visible in the 3D scan. An example of such a position is having the arm ready to grasp a block, as shown in Figure 3.6. Then the user looks at the depiction of the arm in the interface relative to the 3D scan (see Figure 3.7(a)) and uses interface spinner widgets to adjust the rotation of each joint angle until the graphical representation of the arm matches reality to their best judgment (see Figure 3.7(b)). Another button allows the user to capture the calibration point. A calibration point for the base rotation is a pair, (P_b, θ_b), consisting of the servo pulse width command, P_b, and rotation angle, θ_b. A calibration point for the shoulder-elbow system is a 4-tuple, $(P_s, P_e, \theta_s, \theta_e)$. The components of this tuple are the servo pulse width command for the shoulder, P_s, and elbow, P_e, and the actual rotation angle for the shoulder, θ_s, and elbow, θ_e.

We calibrate the arm kinematic model using piecewise linear interpolation. The base rotation joint appears to be independent of the other joints, so we use 1D linear interpolation for the base. Because the shoulder and elbow joints are interdependent, we use 2D linear interpolation for those joints.

To calculate the interpolation, we first create a Delaunay triangulation of the servo command points, as shown in Figure 3.8. The servo command points are made



1370	-7.795	11.0
1630	-2.446	13.5
1550	81.795	0.0

```
1060, -48.468
1550, -1.565
1860, 33.27
1970, 42.229
2100, 59.191
```

```
1900, 52.684, 1970, 141.164
1360, -32.079, 1530, 78.246
1880, 49.158, 1700, 104.365
1890, 50.103, 1490, 74.068
1270, -47.456, 1140, 19.913
1500, -10.167, 1090, 13.241
1220, -56.29, 1000, -1.086
1520, -5.237, 1450, 66.877
```

(a) Poor alignment.

1370	-18.795	0.0
1630	5.554	-5.5
1550	81.795	0.0

```
1060, -48.468
1550, -1.565
1860, 33.27
1970, 42.229
2100, 59.191
```

```
1900, 52.684, 1970, 141.164
1360, -32.079, 1530, 78.246
1880, 49.158, 1700, 104.365
1890, 50.103, 1490, 74.068
1270, -47.456, 1140, 19.913
1500, -10.167, 1090, 13.241
1220, -56.29, 1000, -1.086
1520, -5.237, 1450, 66.877
```

(b) Correct alignment.

Figure 3.7: Calibration of the virtual arm graphic display.

up of a pair of the shoulder command and elbow command. Next, when a query is given, we determine which triangle contains the query point, or which triangle is closest if no triangle contains the query point. Then we use barycentric coordinates to weight each of the triangle's points for the linear interpolation. Barycentric coordinates work nicely for this situation because they extrapolate values that are outside the triangle according to the plane of the triangle. The values that are actually interpolated are the joint angles in the 4-tuple.

Before including arm calibration, the interface display of the arm model showed only the open-loop prediction of the true arm position (obtained through the kinematic model) as illustrated in Figure 3.9. With calibration, the display shows the prediction corrected by the calibration as illustrated in Figure 3.10. The display correction implies that when the arm is commanded to move to a certain point, the arm will move to a different position than the operator believes it will move to (subjectively, up to 2 cm away), but the display of the arm is more accurate (casual observation shows between 1 and 10 mm with bias toward 1 mm). We chose to implement the calibration this way to allow for live, interactive adjustment of the calibration. If the calibration affects the internal kinematic model of the arm, then with a poor calibration (as in the early stages of adding calibration points while calibrating

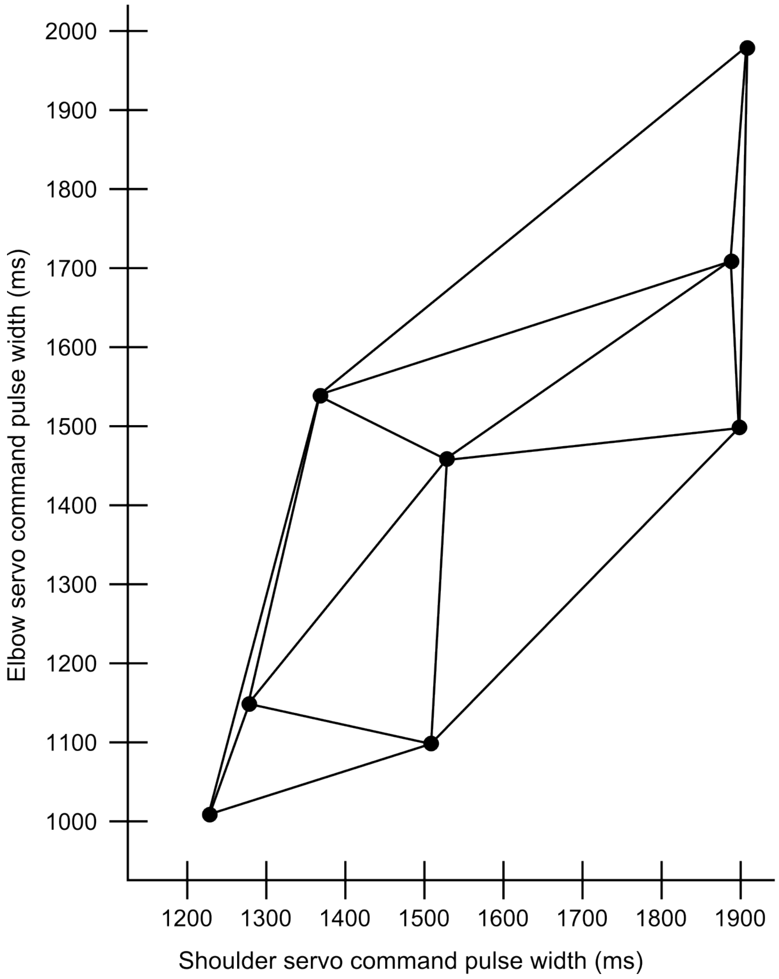


Figure 3.8: Delaunay triangulation of shoulder and elbow arm calibration points.

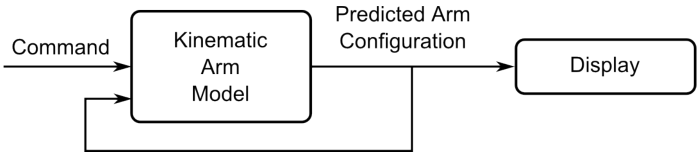


Figure 3.9: Arm model display procedure before introducing calibration.

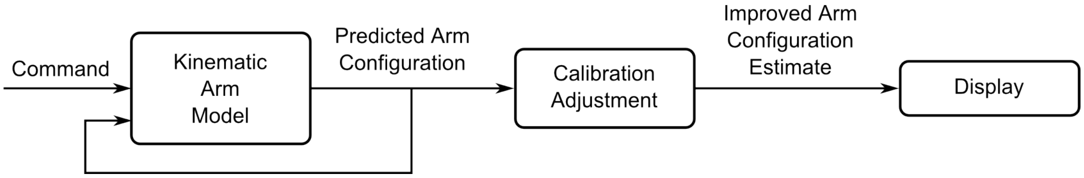

Command
Kinematic Arm Model
Predicted Arm Configuration
Calibration Adjustment
Improved Arm Configuration Estimate
Display

Figure 3.10: Arm model display procedure including calibration.

live), the arm will appear to behave erratically, and the operator may not even be able to position the arm throughout the entire workspace. When the calibration only affects the display, the arm can still be positioned anywhere at any time, although the display may be inaccurate in the early stages. Once the calibration is good, then we could modify the kinematic model of the arm. Due to time constraints, we did not change the kinematic model, and we leave it to future work.

This procedure could also be done autonomously with fiducials mounted on the arm. We implemented a naïve version of autonomous calibration, but the accuracy of the stereo camera was not good enough to provide a good alignment given our implementation. Part of the reason for failure in this could have to do with the placement of our fiducial. We mounted a single fiducial approximately at the wrist joint of the arm because the wrist joint is visible to the stereo camera at all arm positions. Unfortunately, when the arm is in its retracted position, the wrist stays mostly in the same position while the base rotates. This means that two very different arm positions can have similar fiducial positions. The problem with several similarly positioned fiducials is that small amounts of noise are seemingly amplified. When the mapping function looks for the closest measured points between which to interpolate, it may choose incorrect neighbors and yield a position with high error. If we were to place the fiducial in a better place, such as on an extension from the gripper, or place several fiducials (with a cost of higher complexity), autonomous calibration could possibly work well.

3.3 Simple Quickening

Delays that are commonly present in remote systems have adverse effects on operator performance and workload [9]. One way of mitigating the adverse effects is a type of predictive display called *quickening*. Quickening uses a known model for the motion of the remote robot to display nearly instantaneous feedback for user control. Studies have shown that quickening improves time performance and relieves some of the mental workload. One pitfall for quickening is that the model may not be perfectly accurate or some factor in the remote environment causes the model to be inaccurate for some period of time.

We implemented a simple form of quickening for our display by simply showing the target point the manipulator is trying to reach. Although we could predict the position for the whole robot arm, it was trivial to show the target point, and it seems like many of the benefits of quickening come with this simple addition. Operators can focus on the target point for positioning the arm close to the desired location, then wait for the feedback to catch up for tuning the final position. See Figure 3.11 for a sequence depicting our implementation of quickening. We leave testing and validation of this simple form of quickening to future work, though subjective feedback from lab members and visitors is generally positive.

3.4 3D Scan Pruning

Our block collection task environment is relatively clean and simple, but even so, the 3D scan appears cluttered due to noise. Several objects appear in the 3D scan that are irrelevant for the task: floor, curtains, robot arm, and noisy outlier points. We

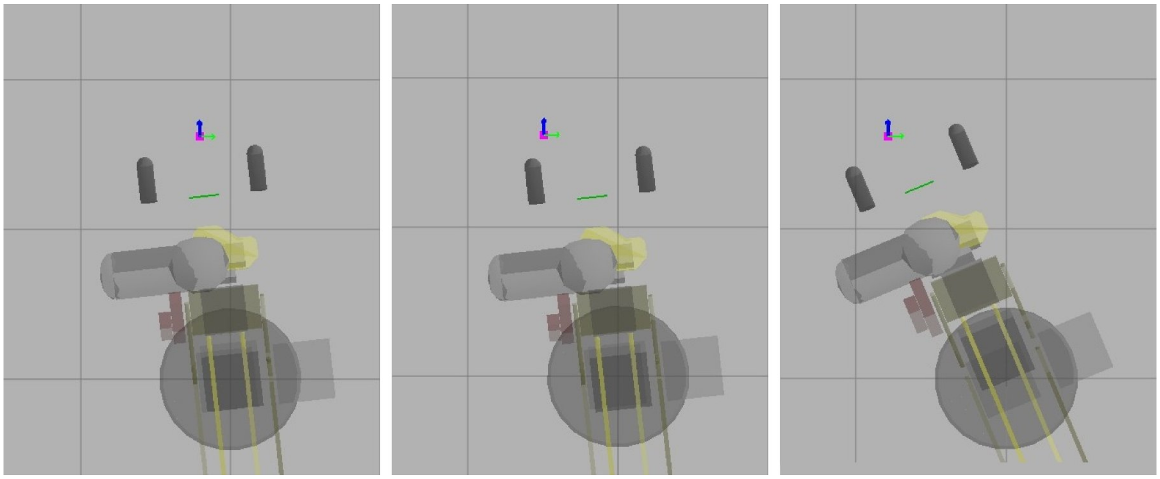
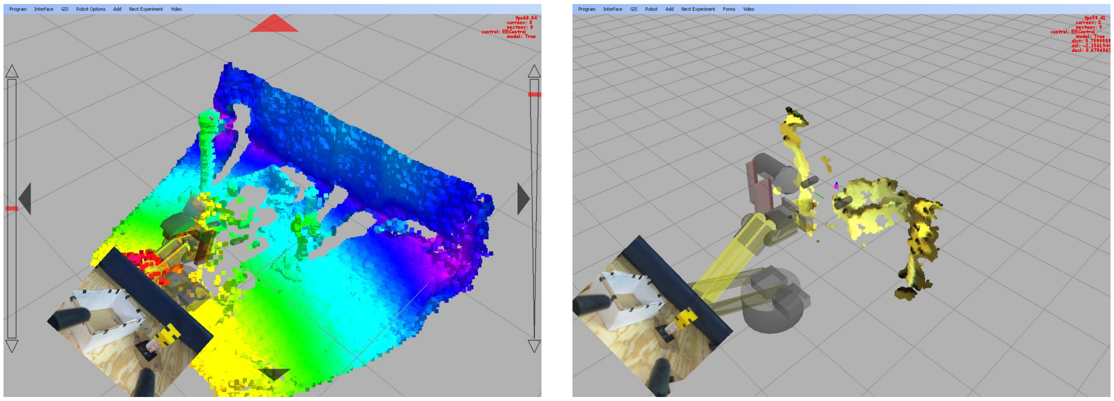


Figure 3.11: Simple quickening sequence. Notice the target point within the gripper in the left frame. In the center frame, the operator has given a command to move left, and the point reflects this action, although the arm has not yet started to move. In the right frame, the operator has stopped issuing commands and has waited for the feedback to catch up to the commanded position indicated by the target point.

need only display objects relevant to the task, such as the target blocks, posts, and deposit box. Since our environment is simple, we can prune away irrelevant points.

We can divide the task environment into axis-aligned bounding boxes to classify points in the 3D scan. Two bounding boxes can effectively classify relevant points for our task environment: one box includes the workspace of the robot arm above the floor, and another box excludes the space of the robot arm in its home position. Determining whether a point is inside a bounding box is a simple and efficient test that can run in real-time. However, when the robot arm is extended outside of its home-position bounding box, the points that represent the robot arm are not pruned and the display looks cluttered. The limitation with this method is that it requires the robot arm to be in the home position when creating a new 3D scan. Figure 3.12 compares the previous 3D scan model (unpruned and without texture) display with the pruned 3D scan model.

We can improve pruning of the robot arm by projecting the virtual arm into the stereo camera's image plane and determining more precisely which points belong to the robot arm regardless of the arm position. We leave this to future work.



(a) Previous 3D scan

(b) Newer, pruned 3D scan

Figure 3.12: Comparison of previous 3D scan model and new, pruned 3D scan model.

3.5 Second User Study

The results from the user study described in Chapter 2 suggested the several changes discussed in this chapter. A second partial user study evaluated the resulting design. The purpose of the second user study is to show that the changes discussed previously improved the user's ability to perform the block collection task under all of the conditions. We will not evaluate whether the 3D scan improves situation awareness in this thesis. However, we will publish such an evaluation in future work. This partial study did not evaluate the individual impact of each change, but rather it served to confirm that the lessons from the first user study led to improved performance. We will talk about the design of the second study and look at some preliminary results.

The task and general design of the second study are patterned after the first study. We attempted to replicate the design of the first study while removing the largest confounding factors. As such, the experiment has the same 2 by 3 structure as shown in Table 3.1. In the first study, we used pseudo-random counterbalanced

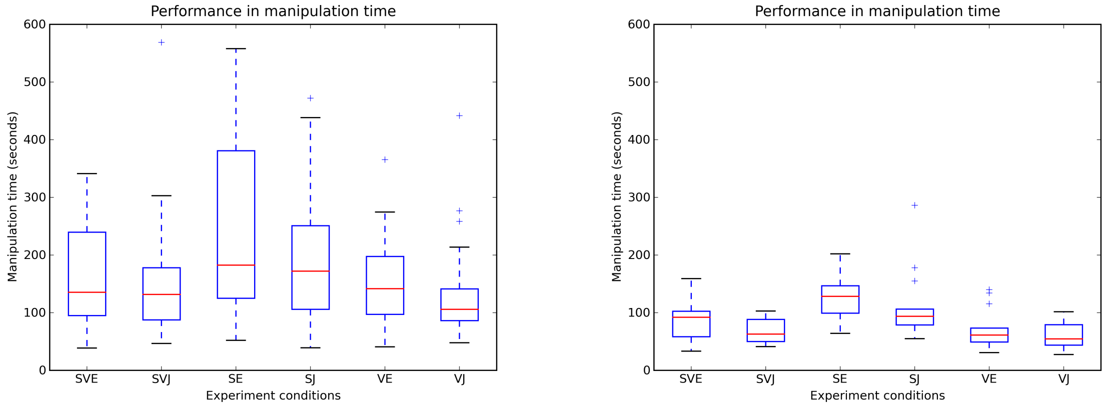
Table 3.1: Variations in the user study. Parentheses show acronyms for each variant.

Interface/Control	*Joint control*	*End-effector control*
Video only	Variant 1 (VJ)	Variant 2 (VE)
3D scan only	Variant 3 (SJ)	Variant 4 (SE)
Video and 3D scan	Variant 5 (SVJ)	Variant 6 (SVE)

ordering to minimize learning effects, but we had some users test both control types. The consequence of this decision was that we had to do strictly between-subject analysis of the data. In the second study, we decided to limit each subject to one type of robot control and test all three display variables. We retain the pseudo-random counterbalanced ordering, and we also ensure that there is an equal representation of the two robot control types, but now we are able to look at within-subject measures for the display variables.

The improvements discussed previously that we included in this user study are: stereo camera calibration, robot arm calibration, 3D scan pruning, and simple quickening. We also smoothed out robot arm motion, changed the gripper camera (increasing quality and framerate), added an overview video display for the video-only (VJ and VE) conditions, and provided more and better training.

Preliminary results show that users perform the tasks in about half the time, with the most improvement in the scan-only (SJ and SE) conditions. Box-and-whisker plots in Figure 3.13 show a comparison between the first study and second study results. Notice that all the times are reduced, but the general relationship between conditions appears to be mostly the same. This indicates that the improvements helped everything overall, but video-only remains the fastest mode, and scan-only remains the slowest. We have not looked at how many collisions happen in each mode, but it appears from our subjective judgment that far fewer collisions happened in the second study. We will report whether the changes made show improved situation awareness for the 3D scan in future work.



(a) First user study

(b) Second user study

Figure 3.13: Comparison of time performance between the first and second user studies measured as time required to perform a manipulation task. Note: To make this plot more readable, we do not show outliers above 600 seconds.

Chapter 4

Conclusion and Future Work

4.1 Conclusion

We have presented an augmented virtuality interface that presents spatial data in an ecological manner. This interface is designed to support remote manipulation, and represents a step toward supporting mobile manipulation as a whole. We use a ranging camera to build a 3D model of the environment in front of the robot. This model provides operators with a rich visualization that promotes situation awareness, especially in terms of the geometry of the robot's surroundings.

The thesis statement describes hypotheses about the benefits of our 3D visualization. The hypotheses are that mental workload is reduced, situation awareness is increased, and time to complete tasks is reduced. Although the experimental results suggest that mental workload is reduced, future work is needed to reach this conclusion. However, the results strongly suggest that situation awareness improves. Experimental results do not support the thesis statment in terms of time to complete tasks, but the results were probably influenced by the following issues: alignment error with sensors, jerky robot motion, and slow view adjustment. Changes were made that improved each of these areas, and evaluation of the resulting system shows substantial improvement. With this improved system we can reconsider the thesis statement, but we leave that evaluation to future work.

When operators used the 3D model, fewer collisions happened with the environment compared to a video-only interface, suggesting that the 3D model supports improved situation awareness. Because situation awareness improves with the 3D model, such an interface is better suited for tasks that involve precise manipulation or a low tolerance for collision errors. On the other hand, video-centric interfaces are better suited for tasks with a critical time component, where collisions may not matter as much.

4.2 Future Work

While we have shown promising results that situation awareness for remote manipulation improves with a 3D augmented virtuality (AV) interface, there are still several interesting experiments to try. We looked at a subjective measure of mental workload for our interface, but there are objective methods that we could use to better determine workload differences between interface types. Adding more autonomy is a trend that seems to be accepted by many as the way forward, so we can evaluate whether autonomous behaviors, such as the behavior described in Appendix A.3, are truly helpful in the long run.

In addition to evaluating factors with our current system, we can improve our system or evaluate other systems. Haptics provide another feedback modality that may be useful for remote manipulation tasks and improving situation awareness. Alternative controllers, such as 6 degree-of-freedom controllers, may be more useful for teleoperating a manipulator with our 3D visualization. Multiple manipulators open up a variety of tasks or make some tasks simpler, but they are inherently more complex to operate, and it would be interesting to evaluate the performance of an AV interface to support multiple manipulators. Refinement of the 3D scan over time to improve resolution or to deal with dynamic environments may further improve situation awareness.

Appendix A

Other Technology Developments

In addition to the changes discussed in Chapter 3, we made some additional changes that have not yet been evaluated. We added head tracking for view adjustment, a more ecological camera video display, and autonomy to support direct interaction with the 3D display.

A.1 Head Tracking for View Adjustment

Adjusting the virtual camera viewpoint is cumbersome and slow with a joystick. The typical workflow for skilled participants in the user study reported in Chatper 2 was something like this: move the view for initial awareness, move arm closer, adjust view some more, move arm closer, adjust view some more, etc. Each of these adjustments requires the operator to stop moving the arm and concentrate on adjusting the view. With this interaction scheme, it seems (subjectively) that operators were discouraged from changing the view often, and instead rarely changed the view. By improving the interaction scheme, we can possibly increase efficiency and encourage more use of the 3D scan, possibly increasing situation awareness.

To give better freedom with controlling the virtual viewpoint, we can use head tracking. Previous work by Barfield et al. shows that head tracking in a virtual environment improves spatial understanding [4]. We use a Nintendo Wii Remote paired with infrared LEDs to provide head position tracking. This idea came from

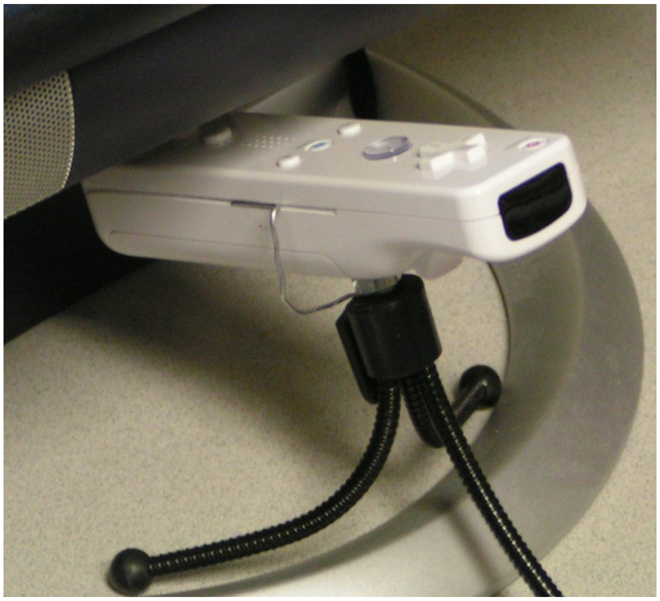


Figure A.1: Nintendo Wii remote positioned under computer monitor.

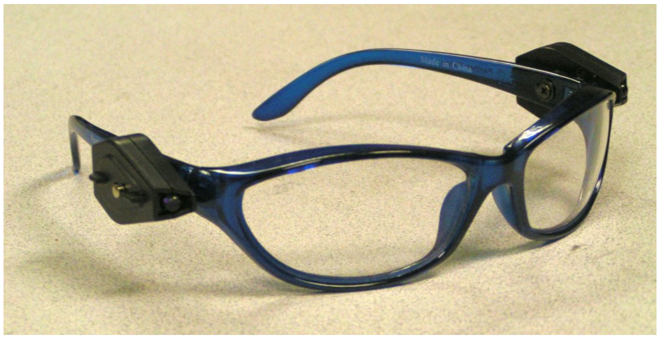


Figure A.2: Safety glasses with infrared LEDs mounted to the sides.

work done by Johnny Lee at CMU [29]. Lee provided a nice framework to do head tracking with the Wii Remote. The Wii Remote, shown in Figure A.1, is essentially an infrared camera with hardware blob tracking of up to 4 points.

When two infrared LEDs are attached to safety glasses (as shown in Figure A.2) and the operator wears the glasses, the Wii Remote can track the position of the two LEDs and infer the position of the operator's head. Lee's framework (WiiDesktopVR) gives the 3D position of the head relative to the screen after some brief calibration. The system may not be perfectly accurate, but its accuracy and precision are excellent for the low cost (about 50 US dollars) involved.

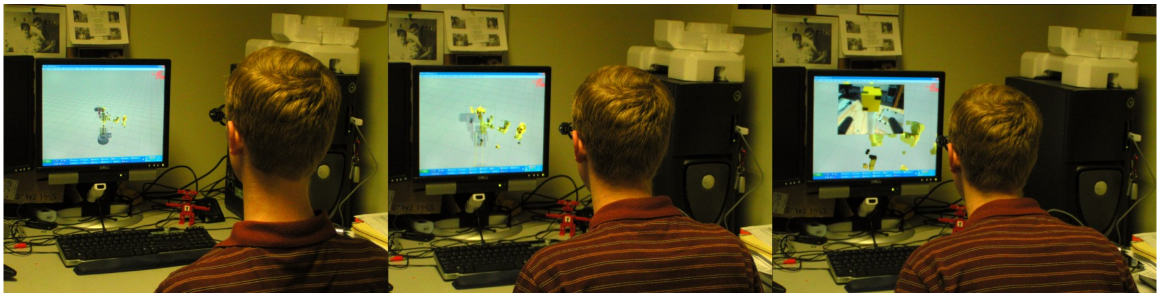


Figure A.3: Sequence of head tracking for user moving head toward monitor.

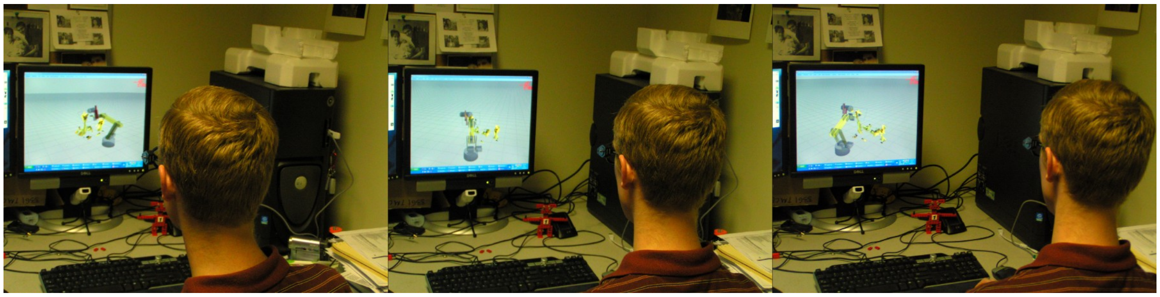


Figure A.4: Sequence of head tracking for user moving head side to side.

Once we have the position of the operator's head, we can use that to control the virtual viewpoint. When the operator leans closer to the screen, the view zooms in, as shown in Figure A.3. Moving the head left and right rotates the view, as shown in Figure A.4. The view tilts when the operator moves their head up or down, as shown in Figure A.5.

The view does not try to mimic a virtual window and make it appear that the operator is really viewing the 3D scene through their monitor (as in [51]), but instead links natural head motions to intuitive view controls.

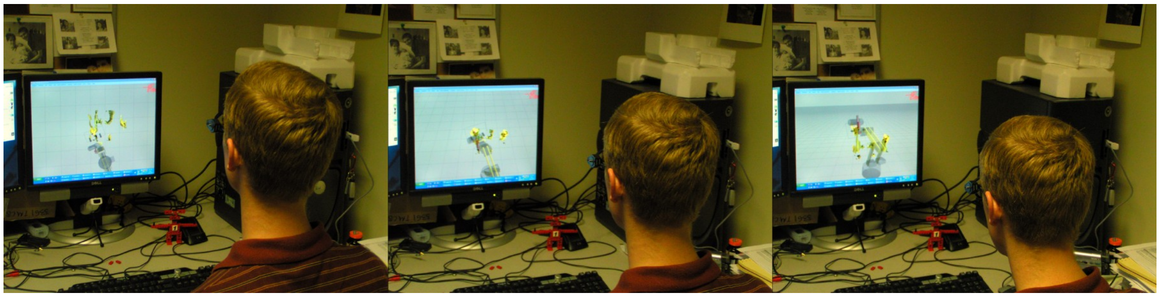


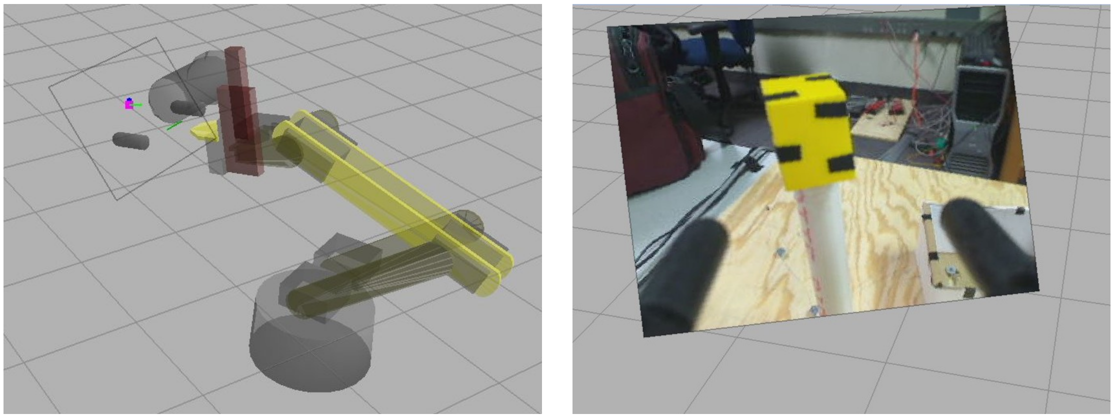
Figure A.5: Sequence of head tracking for user moving head from high to low.

The result is very quick and intuitive view controls, but a problem arises with this interaction scheme. Now that the view is intuitive to change, the frame-of-reference for controlling the arm is again important. Results from the user study in Chapter 2 show that the view-dependent control mode leads to fewer collisions compared to the robot-centric control mode. With head tracking, the virtual view changes frequently and so the view-dependent controls also change frequently. The frequently changing controls might end up causing confusion, so the robot-centric control mode might be better in this case. We leave evaluation of the appropriate arm control frame-of-reference to future work.

A.2 Ecological Camera Video Display

Kubey and Csikszentmihalyi found that video displays depicting motion draw peoples’ attention [26]. Operators focused on the video display can miss important information from the 3D scan, sometimes leading to a collision. Video feedback is important, especially for observation tasks such as USAR, so we cannot simply remove it from the display. Instead, we can try to direct the operator’s attention to the most useful display for different subtasks.

We can divide our manipulation task into three subtasks: (1) identify object of interest, (2) navigate arm to close proximity, and (3) perform final alignment. Identifying the object can be done with either video or the 3D scan, although a detailed exploration task might require video feedback. Navigating the arm is best done with the 3D scan as it gives a better understanding of where obstacles are in relation to the arm, as we determined based on the results from the user study reported in Chapter 2. Final alignment is best done primarily with the 3D scan, since the 3D scan provides depth information, but video gives real-time information about the state of the world in higher resolution than the 3D scan.



(a) Quadrilateral video frame

(b) Video visible

Figure A.6: Ecological video display, showing the quadrilateral frame where video appears when the virtual viewpoint is close.

To support this dependence on both video and the 3D scan, we integrated the video into the 3D scan in a way that we believe is more ecological than before. The video is projected onto a plane just in front of the graphical depiction of the camera mounted on the arm in a manner similar to [39]. The video does not always show, but is hidden from view when the view is zoomed out (see Figure A.6). In other words, when the virtual camera distance from the 3D scan is greater than a certain amount, the video is hidden. As the virtual camera moves closer to the 3D scan, the video fades into view until it becomes fully opaque and is rendered on top of everything in the scene. When the video is fully visible, it occupies most of the screen to show all detail available in the video.

When combined with head tracking, this ecological display has some possibly useful benefits. When operators desire to see something in more detail, it is natural to lean in to get a closer view. With this interface, leaning in shows the camera video, which provides more detail in terms of pixel resolution. Leaning in for long periods of time is uncomfortable, so operators will likely occasionally lean back, which in turn hides the video and shows more of the environment in the 3D model. The head tracker is sensitive enough to pick up slight head movements, so the virtual perspective frequently changes even when someone is trying to sit still. The result of this frequent motion is that the user gets a better sense of depth than a completely static view. This idea of motion parallax, especially when the motion is self-induced, is explained in [42].

A.3 Autonomy to Support Interaction with the 3D Scan

Our focus up to this point has been to improve the accuracy of the 3D scan model. Another aspect of the scan model that we can benefit from is interactivity. During the user study in Chatper 2, we observed that people with little or no experience with robot teleoperation had difficulty understanding how to control the robot even after training. However, everybody that participated in the user study was able to use a mouse effectively for general computing, and it seems that more people have experience using computers than people with experience controlling robots. With some autonomy support, we can enable mouse interactions directly with the scan model display. The interactions so far are preliminary and are not yet fully designed.

We previously designed the motion controller of the arm to use inverse kinematics to position the end effector directly. This sort of controller lends itself well to a "put the end effector there" interaction scheme with the robot arm. The controller only needs a destination for the end effector, and the intermediate positions for the arm are automatically generated over time.

To provide the motion controller with a target point for the end effector, we can use mouse interactions with the 3D scan model. OpenGL allows us to query the depth of a given pixel on the screen. Using this information along with some coordinate frame transformations, we can determine the point represented by a pixel in the robot arm's coordinate frame.

An operator can use the mouse to command the robot arm to move its gripper to a position 3 cm above a point clicked on the 3D scan. In this way, the operator can task the robot to quickly move near the object of interest, and then the operator need only make fine adjustments to the positioning. This can possibly speed up overall operation.

One pitfall that can arise with this balance between autonomy and teleoperation is autonomy failure. There may be situations where the arm will not move in the expected manner after the operator clicks the 3D scan. Such behavior may be confusing or possibly even dangerous, depending on the situation and task. An example of such autonomy failure is indicated in [24] where an autonomous object grasping behavior would knock over the intended target. These errors can make the task more difficult for the human operator.

With supervisory control, an operator could simply click on a 2D video to indicate an object of interest, and the system could determine the corresponding 3D position and autonomously retrieve the object as in [24, 28]. Such an interaction scheme would make the 3D scan irrelevant for an object retrieval task. While supervisory control seems like the most efficient way to perform an object retrieval task, limited sensor accuracy and other automation errors are still prevalent. Because of this, we decided to let autonomy coarsely position the manipulator, and then allow the operator to adjust the position of the manipulator and grasp the object. With this mixed-autonomy interaction scheme, the 3D scan remains useful for spatial understanding. Mixed autonomy also allows more flexibilty for when operators need to perform other tasks besides object retrieval.

Bibliography

[1] R.O. Ambrose, R.T. Savely, S.M. Goza, P. Strawser, M.A. Diftler, I. Spain, and N. Radford. Mobile manipulation using NASA's robonaut. In *Proceedings of 2004 IEEE International Conference on Robotics and Automation*, 2004.

[2] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 9(5):698–700, 1987.

[3] J. A. Atherton and M. A. Goodrich. Supporting remote manipulation with an ecological augmented virtuality interface. In *Proceedings of the AISB Symposium on New Frontiers in Human-Robot Interaction*, Edinburgh, UK, April 2009.

[4] W. Barfield, C. Hendrix, and K. Bystrom. Effects of stereopsis and head tracking on performance using desktop virtual environment displays. *Presence: Teleoperators and Virtual Environments*, 8(2):237–240, 1999.

[5] Battelle. National institute of justice final report on law enforcement robot technology assessment. http://www.justnet.org/Pages/RecordView.aspx?itemid=1275, April 2000.

[6] D. J. Bruemmer, D. A. Few, R. L. Boring, J. L. Marble, M. C. Walton, and C. W. Nielsen. Shared understanding for collaborative control. *IEEE Transactions on Systems, Man and Cybernetics, Part A*, 35(4):494–504, July 2005.

[7] D.J. Bruemmer, D.A. Few, C. Kapoor, and M. Goza. Dynamic autonomy for mobile manipulation. In *ANS / IEEE 11th Annual Conference on Robotics and Remote Systems for Hazardous Environments*, Salt Lake City, UT, February 2006.

[8] J. Casper and R.R. Murphy. Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center. *IEEE Transactions on Systems, Man and Cybernetics, Part B*, 33(3):367–385, 2003.

[9] J.Y.C Chen, E.C Haas, and M.J Barnes. Human performance issues and user interface design for teleoperated robots. *Systems, Man and Cybernetics, Part C: Applications and Reviews, IEEE Transactions on*, 37(6):1231–1245, 2007.

[10] J. Cooper and M. A. Goodrich. Towards combining UAV and sensor operator roles in UAV-enabled visual search. In *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction*, pages 351–358, Amsterdam, The Netherlands, 2008. ACM.

[11] J.W. Crandall and M.A. Goodrich. Characterizing efficiency of human robot interaction: A case study of shared-control teleoperation. In *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, volume 2, pages 1290–1295 vol.2, 2002.

[12] M. R. Endsley. Design and evaluation for situation awareness. In *In Proceedings of the Human Factors Society 32nd Annual Meeting*, pages 97–101, Santa Monica, CA, 1988.

[13] D. Falie and V. Buzuloiu. Noise characteristics of 3d time-of-flight cameras. In *In Proceedings of International Symposium on Signals, Circuits and Systems*, volume 1, pages 1–4, July 2007.

[14] F. Ferland, F. Pomerleau, C. Dinh, and F. Michaud. Egocentric and exocentric teleoperation interface using real-time, 3D video projection. In *Proceedings of the 4th ACM/IEEE International Conference on Human-Robot Interaction*, pages 37–44, La Jolla, California, USA, 2009. ACM.

[15] M. A. Goodrich. Using models of cognition in HRI evaluation and design. In *Proceedings of the AAAI 2004 Fall Symposium Series: The Intersection of Cognitive Science and Robotics: From Interfaces to Intelligence*, Arlington, Virginia, 2004.

[16] M. A. Goodrich and Jr. Olsen, D. R. Seven principles of efficient human robot interaction. In *Proc. IEEE International Conference on Systems, Man and Cybernetics*, volume 4, pages 3942–3948 vol.4, 2003.

[17] F. R. Hartman, B. Cooper, C. Leger, S. Maxwell, J. Wright, and J. Yen. Data visualization for effective rover sequencing. In *2005 IEEE International Conference on Systems, Man and Cybernetics*, Pasadena, CA, 2005.

[18] F. R. Hartman, B. Cooper, S. Maxwell, J. Wright, and J. Yen. Immersive visualization for navigation and control of the Mars Exploration Rovers. In *SpaceOps*, Montreal, Canada, 2004.

[19] V. Hayward, O.R. Astley, M. Cruz-Hernandez, D. Grant, and G. Robles-De-La-Torre. Haptic interfaces and devices. *Sensor Review*, 24:16–29, February 2004.

[20] L. M. Hiatt and R. Simmons. Coordinate frames in robotic teleoperation. In *Proc. IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 1712–1719, 9–15 Oct. 2006.

[21] A. Jacoff, B. Weiss, and E. Messina. Evolution of a performance metric for urban search and rescue robots. In *Proceedings of the 2003 Performance Metrics for Intelligent Systems (PerMIS) Workshop*, Gaithersburg, MD, September 2003.

[22] D.B. Kaber, J. Riley, R. Zhou, and J.V. Draper. Effects of visual interface design, control interface type, and control latency on performance, telepresence, and workload in a teleoperation task. In *In Proceedings of the XIVth Triennial Congress of the International Ergonomics Association and 44th Annual Meeting of the Human Factors and Ergonomics Society*, 2000.

[23] G. A. Kaminka and Y. Elmaliach. Experiments with an ecological interface for monitoring tightly-coordinated robot teams. In *ICRA-06*, 2006.

[24] A. Kelly, D. Anderson, E. Capstick, H. Herman, and P. Rander. Photogeometric sensing for mobile robot control and visualisation tasks. In *Proceedings of the AISB Symposium on New Frontiers in Human-Robot Interaction*, Edinburgh, UK, April 2009.

[25] H.K. Keskinpala, J.A. Adams, and K. Kawamura. PDA-based human-robotic interface. In *Systems, Man and Cybernetics, 2003. IEEE International Conference on*, volume 4, pages 3931–3936 vol.4, 2003.

[26] R. Kubey and M. Csikszentmihalyi. Television addiction is no mere metaphor. *Scientific American*, 286(2):79–86, 2002.

[27] K. J. Kuchenbecker and G. Niemeyer. Induced master motion in force-reflecting teleoperation. *Journal of Dynamic Systems, Measurement, and Control*, 128(4):800–810, 2006.

[28] A. Kulkarni, D. Bruemmer, C. Kapoor, R. Kinoshita, J. Atherton, J. Whetten, C. Nielsen, and M. Pryor. Software framework for mobile manipulation. In *Proceedings of ANS 2nd International Joint Topical Meeting on Emergency Preparedness and Response and Robotic and Remote Systems*, Albuquerque, New Mexico, March 2008.

[29] J. Lee. Head tracking for desktop VR displays using the Wii remote. http://johnnylee.net/projects/wii/, December 2007.

[30] H. Levine. Improvised response. *Bulletin of the Atomic Scientists*, 62(4):22–23,67, July/August 2006.

[31] J. A. Macedo, D. B. Kaber, M. R. Endsley, P. Powanusorn, and S. Myung. The effect of automated compensation for incongruent axes on teleoperator performance. *Human Factors*, 40(4):541–553, December 1998.

[32] F. Michaud, P. Boissy, H. Corriveau, A. Grant, M. Lauria, D. Labonte, R. Cloutier, M. A. Roux, and D. Iannuzzi. Telepresence robot for home care assistance. In *AAAI Spring Symposium: Multidisciplinary Collaboration for Socially Assistive Robotics*, March 2007.

[33] P. Milgram and F. Kishino. A taxonomy of mixed reality visual displays. *IEICE Transactions on Information Systems*, E77-D(12):1321–1329, 1994.

[34] P. Milgram, S. Zhai, D. Drascic, and J. Grodski. Applications of augmented reality for human-robot communication. In *Intelligent Robots and Systems '93, IROS '93. Proceedings of the 1993 IEEE/RSJ International Conference on*, volume 3, pages 1467–1472 vol.3, 1993.

[35] R. Murphy. Human-robot interaction in rescue robotics. *IEEE Transactions on Systems, Man and Cybernetics, Part C: Applications and Reviews*, 34:138–153, 2004.

[36] L. A. Nguyen, M. Bualat, L. J. Edwards, L. Flueckiger, C. Neveu, K . Schwehr, M. D. Wagner, and E. Zbinden. Virtual reality interfaces for visualization and control of remote vehicles. *Autonomous Robots*, 11(1):59–68, 2001.

[37] C. W. Nielsen. *Using Augmented Virtuality to Improve Human-Robot Interactions*. PhD thesis, Brigham Young University, 2006.

[38] C. W. Nielsen and M. A. Goodrich. Comparing the usefulness of video and map information in navigation tasks. In *HRI '06: Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pages 95–101, New York, NY, USA, March 2006. ACM.

[39] C. W. Nielsen, M. A. Goodrich, and R. W. Ricks. Ecological interfaces for improving mobile robot teleoperation. *IEEE Transactions on Robotics and Automation*, 23(5):927–941, Oct. 2007.

[40] T. Reenskaug. Models-views-controllers. Technical report, Xerox PARC, December 1979.

[41] B. Ricks, C. W. Nielsen, and M. A. Goodrich. Ecological displays for robot interaction: A new perspective. In *Proceedings of IROS 2004*, Sendai, Japan, 2004.

[42] B. Rogers and M. Graham. Motion parallax as an independent cue for depth perception. *Perception*, 8(2):125–134, 1979.

[43] P. S. Schenker, E. T. Baumgartner, S. Lee, H. Aghazarian, M. S. Garrett, R. A. Lindemann, D. K. Brown, Y. Bar-Cohen, S. Lih, B. Joffe, S. S. Kim, B. D. Hoffman, and T. L. Huntsberger. Dexterous robotic sampling for Mars in-situ science. In David P. Casasent, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 3208 of *Presented at the Society of Photo-Optical Instrumentation Engineers (SPIE) Conference*, pages 170–185. SPIE, September 1997.

[44] J. Scholtz, M. Theofanos, and B. Antonishek. Development of a test bed for evaluating human-robot performance for explosive ordnance disposal robots. In *In Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction*, pages 10–17, New York, NY, USA, 2006. ACM Press.

[45] J. Scholtz, J. Young, J.L. Drury, and H.A. Yanco. Evaluation of human-robot interaction awareness in search and rescue. In *In Proceedings of the IEEE International Conference on Robotics and Automation*, volume 3, pages 2327–2332 Vol.3, 2004.

[46] N. Shachtman. The Baghdad bomb squad. *Wired*, 13(11), November 2005.

[47] T. B. Sheridan. *Telerobotics, automation, and human supervisory control.* MIT Press, Cambridge, MA, USA, 1992.

[48] A. Trebi-Ollennu, P. C. Leger, E. T. Baumgartner, and R. G. Bonitz. Robotic arm in-situ operations for the Mars Exploration Rovers surface mission. In *IEEE International Conference on Systems, Man, and Cybernetics*, Waikoloa, HI, 2005.

[49] K. Tsui, H. Yanco, D. Kontak, and L. Beliveau. Development and evaluation of a flexible interface for a wheelchair mounted robotic arm. In *Proceedings of the 3rd ACM/IEEE International Conference on Human-Robot Interaction*, pages 105–112, Amsterdam, The Netherlands, March 2008. ACM.

[50] K.J. Vicente and J. Rasmussen. Ecological interface design: Theoretical foundations. *Systems, Man and Cybernetics, IEEE Transactions on*, 22(4):589–606, 1992.

[51] C. Ware, K. Arthur, and K. S. Booth. Fish tank virtual reality. In *Proceedings of the INTERACT '93 and CHI '93 Conference on Human Factors in Computing Systems*, pages 37–42, Amsterdam, The Netherlands, 1993. ACM.

[52] C. D. Wickens and J. G. Hollands. *Engineering Psychology and Human Performance*. Prentice Hall, Upper Saddle River, NJ, 3rd edition, 2000.

[53] C. D. Wickens, J. Lee, Y. D. Liu, and S. Gordon-Becker. *Introduction to Human Factors Engineering (2nd Edition)*. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2003.

[54] D. R. Williams. Mars fact sheet. http://nssdc.gsfc.nasa.gov/planetary/factsheet/marsfact.html, September 2004.

[55] J. Wong and C. Robinson. Urban search and rescue technology needs: Identification of needs. Technical Report NCJRS 207771, Savannah River National Laboratory, 2004.

[56] D.D. Woods, J. Tittle, M. Feil, and A. Roesler. Envisioning human-robot coordination in future operations. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Applications and Reviews*, 34(6):749–756, November 2004.

[57] H. A. Yanco, M. Baker, R. Casey, B. Keyes, P. Thoren, J. L. Drury, D. Few, C. Nielsen, and D. Bruemmer. Analysis of human-robot interaction for urban search and rescue. In *Proceedings of the IEEE International Workshop on Safety, Security and Rescue Robotics*, Gaithersburg, MD, August 2006.

[58] H.A. Yanco and J. Drury. "Where am I?" Acquiring situation awareness using a remote robot platform. In *Systems, Man and Cybernetics, 2004 IEEE International Conference on*, volume 3, pages 2835–2840 vol.3, 2004.