FinalReport:
Remote Manipulation for D&D Exhibiting Teleautonomy and Telecollaboration

Lead Principal Investigators: Thomas F. Ewing, J. Edward Colgate
Co-Investigators: Young S. Park, Michael A. Peshkin
Graduate Students: Brian P. DeJong, Eric L. Faulring
Research Scientist: Hyosig Kang

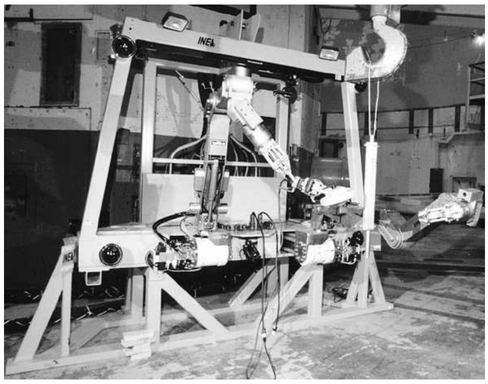


Plate 1. The Dual Arm Work Platform (DAWP) at Argonne National Laboratory. Experience using the DAWP to dismantle the inside of a retired nuclear reactor is the underlying motivation for this work.

Introduction

Teleoperation has found much application in today's society. While its benefits make it advantageous in many situations, teleoperation also introduces new challenges. For example, the operator is forced to work with limited sensory feedback such as that from video cameras. Second, the mapping from what the operator inputs at the master manipulandum to what the operator sees the slave robot do can be mentally challenging and tiresome. Finally, time-delay that may exist between master and slave makes force-feedback more confusing than helpful.

In this paper, we present techniques to improve teleoperation performance by making better use of the human's inherent skills. We do so by minimizing the mental transformations in a teleoperation interface and by locally implementing haptic virtual surfaces via the master. The following section presents a specific teleoperation application that is the main motivation for this work. We then introduce an internet teleoperation testbed that incorporates our techniques, and discuss seven lessons learned during the creation and use of that testbed. Finally, we present our conclusions with avenues for future research.

Motivating example

A specific, motivating example showing the difficulties of teleoperation is an application at Argonne National Laboratory.

Argonne operators have used teleoperation techniques to dismantle the interior of a retired nuclear reactor as part of an ongoing decontaminating and decommissioning project. The slave robot used was the Dual Arm Work Platform (DAWP) (See Plate 1), consisting of two six-degree-of-freedom Schilling robotic arms and several tilt/pan/zoom cameras. The user interface included two completely passive masters and several video monitors arrayed in a fixed arc around the operator (Argonne National Laboratory-East, 1998).

While teleoperation in the radioactive environment was cost-effective, Argonne personnel noted several problems. First, the operator selection and training process was time-consuming and expensive; only 60% of the tested operators were skilled enough even to complete tasks. Second, they found that operators spent over 90% of their time setting up rather than performing tasks. Finally, Argonne's operators found the teleoperation tiring, especially when performing tasks that required switching between multiple camera views. Other information regarding the DAWP and subsequent designs can be found in (Noakes et al., 2002) and (Draper and Blair, 1997).

A typical task for the DAWP that was both time-consuming and mentally challenging was the cutting of a pipe using a circular saw mounted on the end of the slave. The operator had to first properly align the teleoperated saw, and then cut the pipe in a straight line so as not to damage the saw blade. If the blade did break, as was often the case, then not only did workers have to put radiation suits on and replace the blade by hand, but the operator also had to precisely re-insert the blade into the existing cut, or start an entirely new cut. Because of the challenges involved in this pipe-cutting task, it has been selected as the baseline demonstration for the testbed presented here.

Teleoperation testbed

To implement haptic virtual surfaces and reduced mental transformations, a collaborative teleoperation testbed between Northwestern University and Argonne has been established. The user interface located at Northwestern University includes a master manipulandum and a graphical user interface (GUI) on two video displays. The remote site is located over thirty miles away at Argonne National Laboratory and consists of a 6-DOF hydraulic robot, two video cameras, and a structured light sensor system. See Plate 2 for a block diagram of the system's components and pictures of the two sites. The master, GUI, robot, and structured light system communicate via UDP protocol over Ethernet.

The master manipulandum is a six-degree-of-freedom Cobotic Hand Controller, presented in detail in (Fauling et al., 2004). It employs six continuously variable transmissions connected in parallel to the end effector. By utilizing non-holonomic constraints within these transmissions, rather than powerful actuators, it can create haptic virtual surfaces that are very stiff (50 kN/m) in the constraint direction yet very smooth tangentially. The master position-controls the slave robot.

The GUI consists of a Windows PC that receives user input and presents the two video streams on LCD monitors with augmented reality. The PC receives position information from the robot and mode/parameter input from the user, analyzes it, and tells the master what mode to be in and what reference frame to control the slave in. The GUI allows the user to create, adjust, and implement virtual surfaces, adjust motion scaling, and toggle on/off components of the augmented overlays.

At the remote site, the structured light system is capable of measuring the 3-D position of points on an object or surface. The system (Park et al., 2004) projects a laser grid onto the object in question and records a video image. By image processing, it can determine the positions of the intersections of the gridlines. These points can be least-squares fit onto a desired primitive shape (such as a cylinder) or left as raw data. When sent to the GUI, they are displayed as the surface or shape, as selected by the user, and can be used to define a virtual surface. The structured light sensor system allows for on-the-fly modeling of unstructured environments, such as the interior of a partially dismantled nuclear reactor.

The video cameras are color CCD cameras. Their video stream is transmitted by commercially available video servers (Brightnoise Inc., 2004) over the internet to the GUI's video board. The graphical overlays are locally incorporated into the video images as defined by the user.

Lessons learned

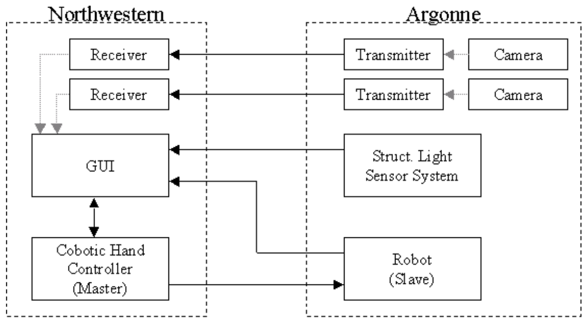
There were several lessons learned during the creation and use of this teleoperation testbed. The first lesson deals with properly arranging the testbed's components to minimize mental transformations, while the remaining six involve the implementation of haptic virtual surfaces applied at the master.

Lesson 1: Arranging the Interface to Reduce Mental Transformations

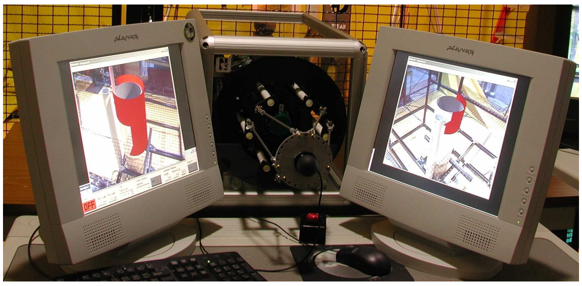
For a teleoperation interface, mental workload increases if the mapping between master and *perceived slave*, i.e. the slave as seen in the video images, is not identity. Conflicting orientation information leads to confusion and forces the operator to learn a new mapping from hand to eye, i.e. it forces the operator to mentally transform hand motions to perceived-slave motions. Sheridan (1992) briefly presents this idea as a misalignment of prioperception, and there is a large amount of cognitive science literature (Shepard and Metzler, 1971; Kosslyn, 1980; Wexler et al., 1998) discussing the cost of performing mental transformations.

A full analysis of the types of mental transformations found in a teleoperation setup, and how to minimize them, can be found in (DeJong, 2004); we briefly summarize it here. There are three types of mental transformations that can be found in a teleoperation setup and each adds to the setup's mental difficulty. The first two types, *Control Rotation* and *Control Translation*, involve misalignments (rotational and translational, respectively) between master and perceived slave. An example of Control Rotation is when horizontal motion of the master results in vertical motion of the slave on the display. When such misalignments exist in the control mapping from master to slave, the human must mentally rotate and/or translate the desired motion of the perceived slave to get the required motion or force at the master. The third type of transformation, called *View Rotation*, exists if the operator is not on a display's centerline. If so, the operator must mentally rotate the view to determine the desired motion of the slave.

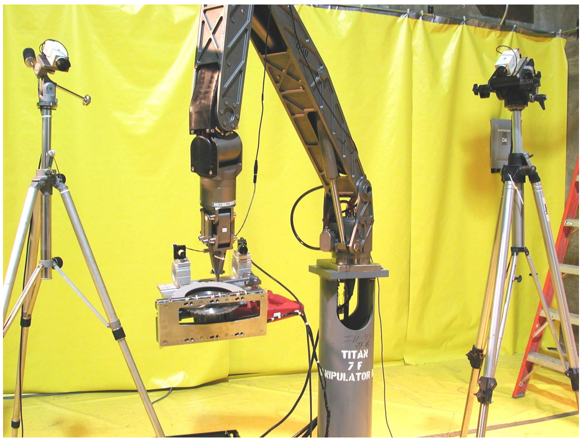
Therefore, in designing a setup with only one camera



(a)



(b)



(c)

Plate 2. The teleoperation testbed between Northwestern and Argonne . (a) Block diagram of the system showing communication directions. Solid lines represent Ethernet connections while dotted lines show composite connections. (b) The operator interface at Northwestern , with master manipulandum and two displays. (c) The robot with saw and structured light system onboard and the two video cameras, at Argonne .

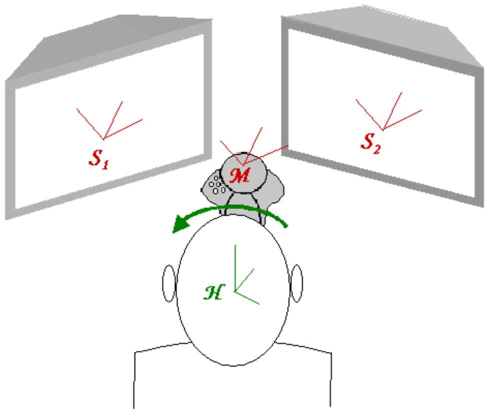


Plate 3. An example of a teleoperation interface with minimized mental transformations. Note that the human operator is at the intersection of the displays' centerlines. Reference frames *M* and *H* are the master's and human's frames, respectively; frame S_i is the slave's frame as seen through camera and display *i*.

and one display, the designer should position the camera, master, and video display properly so as to eliminate or minimize the transformations. Given the locations of any two of the three components, one can calculate the location for the third that minimizes transformations. For example, given the camera and display locations, there is a specific position and orientation for the master that minimizes the mental transformations. Note that if the camera's location changes, such as with pan/tilt or a mobile robot, then ideally the display and/or the master should move to compensate.

When a setup involves more than one camera/display, mental transformations may be even more costly. If the operator is using one of the displays and it involves mental transformations, he/she may eventually learn the mapping such that control is relatively easy. However, once the operator switches attention to another display, the mapping has changed and must be relearned. Cognitive science literature shows that switching between different perspectives of an environment is already mentally challenging (Rieser, 1989; Farrell and Robertson, 1998) without the additional workload of relearning mappings. If the teleoperation interface instead has reduced mental transformations (see one such example in Plate 3), then every control mapping is the same and very close to identity, hence trivial to learn. When inputting a force or motion on the master, the operator sees the robot move in the same direction in every video image.

Results from an experiment reported in (DeJong, 2003) show that these mental transformations affect task performance. Task performance was significantly poorer for interfaces with more mental transformations. When regular and minimized-transformation setups were both used by Argonne personnel, they were surprised by how much the reduced transformations made teleoperation easier. Proper alignment in an interface can also improve task precision and make the teleoperation less tiresome for the user.

To reduce the transformations in our teleoperation testbed involving master, slave, and two cameras and displays, we set up our components as follows. First, we chose camera locations that gave us useful views of the slave workspace.

Second, we placed the two displays in front of the operator such that the operator was at the intersection of their centerlines. Doing so minimized human-display misalignment, i.e. minimized View Transition, with minimal transition needed when shifting from one view to the other. Furthermore, because we wanted all of the perceived slave frames to the same (as in Plate 3), we needed

$$ {}^{D_2}_{D_1}\boldsymbol{R} = {}^{C_2}_{C_1}\boldsymbol{R}, \tag{1} $$

where D_i and C_i are display *i*'s and camera *i*'s reference frames (DeJong, 2004) , and $\boldsymbol{R}$ is a four-by-four homogeneous transformation matrix. This constraint told us the angle required between the two displays.

Third, we needed to chose, based on now-known positions of a camera and display, the proper master position so as to minimize mental transformations. Because of the size and shape of our Cobotic Hand Controller, we positioned it on a table between our two displays. Doing so also minimized translational misalignment between master and perceived slave, i.e. minimized Control Translation.

Therefore, all that remained was to properly orient the master reference frame to eliminate rotational misalignment between master and perceived slave, i.e. eliminate Control Rotation. There are two ways to achieve this and at two levels of accuracy. We could have either physically rotated the master or computationally rotated its commanded motions, and we could have aligned the master frame to the perceived slave frame by roughly eye-balling the master/slave mapping or by calculating it more accurately from the constraint

$$ \boldsymbol{I} = {}^{S}_{M}\boldsymbol{R} = {}^{S}_{C}\boldsymbol{R} \cdot {}^{C}_{D}\boldsymbol{R} \cdot {}^{D}_{M}\boldsymbol{R} = {}^{S}_{C}\boldsymbol{R} \cdot \boldsymbol{I} \cdot {}^{D}_{M}\boldsymbol{R}, \tag{2} $$

where *S*, *M*, *D*, and *C* are the coordinate frames for the slave, the master, a display, and a camera, respectively(DeJong, 2004). Because of the shape of our master, we chose to computationally rotate the frame, and for accuracy, we decided to calculate the rotation needed. That is, we needed to know

$$ {}^{M}_{Int}\boldsymbol{R} = {}^{M}_{D}\boldsymbol{R} \cdot {}^{D}_{Int}\boldsymbol{R} = {}^{S}_{C}\boldsymbol{R} \cdot {}^{D}_{Int}\boldsymbol{R}, \tag{3} $$

where *Int* is the internal reference frame by which the Cobotic Hand Controller records inputs. (The ${}^{M}_{D}\boldsymbol{R}$ to ${}^{S}_{C}\boldsymbol{R}$ substitution is known from (2).) We knew *C*, *S*, *Int*, and *D*, thus we calculated ${}^{M}_{Int}\boldsymbol{R}$. The GUI sends this matrix to the master, which subsequently multiplies all motion commands by it.

Lesson 2: Applying Virtual Surfaces at the Master, Rather than at the Slave

One of the benefits of the Cobotic Hand Controller is its ability to physically impose multidimensional virtual surfaces on the operator. Virtual surfaces, or virtual constraints

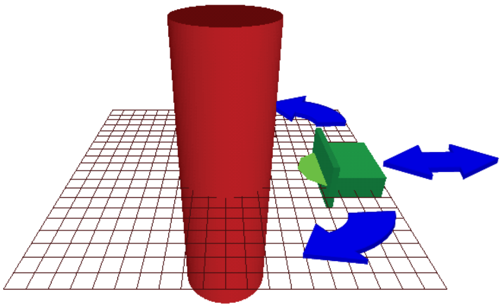


Plate 4. A environmental representation of our example virtual surface. The tool (i.e. a saw) is constrained to be on the plane (i.e. the cutting plane of the saw blade), pointing towards and outside the cylinder (i.e. the pipe to be cut). Arrows in the figure represent allowed motions of the tool.

on motion, were first suggested by Rosenberg (1994). They simplify a teleoperation task by reducing the number of degrees of freedom available to the operator. Thus a six-dimensional task, such as moving our robotic slave, can be constrained to a simpler, lower-dimensional task.

There are several commercial devices that can create virtual surfaces, including the DELTA (Dimension, 2004), the VIRTUOSE (Haption, 2004), and the PHANTOM (SensAble Technologies, 2003). In addition, previous generations of cobots (Peshkin et al., 2001) have succeeded in imposing one- and two-dimensional surfaces using the same technique as the six-degree-of-freedom Cobotic Hand Controller discussed here.

While the cobotic master used in our testbed can create many user-defined virtual surfaces, the surface chosen for our demonstration constrains the operator to a two-dimensional, partially bounded workspace to aid in the cutting of a large pipe using a circular saw. Geometrically, the virtual surface can be described as a plane with a forbidden hole in it; with respect to the tool and environment, it involves a cylinder about the pipe and the cutting plane (see Plate 4). When the surface is in effect, the manipulandum is physically constrained to the cutting plane, with rotation in the plane forced such that the saw is always pointing to the pipe's center (i.e. saw's guard is tangent to the pipe). Thus the operator has two degrees of freedom: rotation around, and translation towards, the pipe's axis. Furthermore, the operator is restricted from entering the cylinder (i.e. the saw's guard rides along the pipe).

This virtual surface is automatically generated via processed structured light data. In addition, during teleoperation the operator is able to turn on and off the constraints as desired. Thus he/she can move the saw close to the pipe in unrestricted mode, use the virtual surface for the cutting of the pipe, and return to unrestricted mode to move away when finished. Using the structured light information as the basis for a virtual surface makes the system versatile in unstructured environments.

Once a virtual surface is defined, it needs to be enforced. There are two ways to enforce virtual surfaces: by computationally restricting the commanded motion to the slave, or by physically constraining the operator's motion. While systems exist that use the first technique, doing so has a significant drawback in that the user does not feel the constraints. What the operator does with the master is not what he/she sees the slave do. This discrepancy between hand and eye is often confusing (Sheridan, 1992), causing more harm than good. Furthermore, with a wall constraint such as the cylinder in our example, the operator must overcompensate by continuously moving "into" the surface to keep the slave against it.

Instead, by physically imposing the virtual surfaces at the master, what the operator does is what he/she sees. Thus the mapping from master to slave is preserved, and the operator can use the haptic information as an additional source of feedback. The operator can feel the master's stiff constraints and physically stay against them as he/she moves the master.

Lesson 3: Overcoming Time-Delay Difficulties with the Virtual Surfaces Implemented at the Master

The third lesson learned from this testbed is that the locally generated virtual constraints counteract the detriments of time-delay. The round-trip of information, from a motion on the master to the motion of the slave in the video images is approximately a quarter of a second. While this amount is small compared to other applications such as space teleoperation (Landis, 2003) it can be detrimental to precise teleoperation (Sheridan and Ferrel, 1963) without haptic virtual surfaces. If the operator is forced to perform tasks, such as pipe cutting, without virtual constraints, then this slight time-delay may degrade performance. Meanwhile, if the operator *has* the benefit of virtual constraints but they are applied computationally such that he/she still has to completely rely on visual feedback, then time-delay could still be an issue. However, because our virtual surfaces are applied locally at the master, the operator can cut the pipe in real-time by using real-time haptic feedback. Even if the time-delay were larger, the operator's cutting of the pipe should not be affected.

Lesson 4: Giving Visual Feedback of Virtual Surfaces via Augmented Reality

In additional to being presented haptically, the virtual surfaces are presented visually in the form of augmented reality overlays. Augmented reality is a common tool in teleoperation literature (Fuchs et al., 2002; Cao, 2000; Azuma, 1997), and is usually used to create predictive displays and to aid in depth perception. In our testbed, it is created using OpenGL and implemented at the video board; to achieve occlusion effects, environmental objects such as the robot and tool are modeled with a transparent color. Plate 5 shows a screen capture of the video image and overlay. The overlayed virtual surfaces give visual feedback congruent with the haptic information felt at the master and allow the user to visually check the definition of the surface.

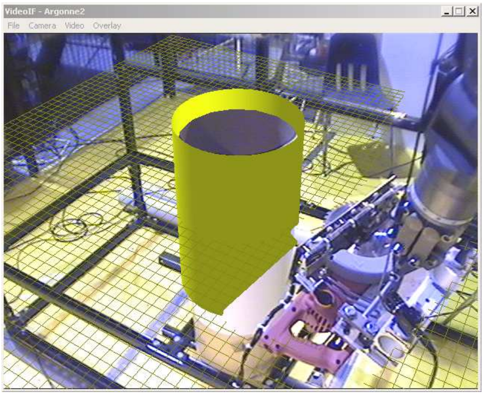


Plate 5. A screen capture of one of the video images with augmented reality of virtual surfaces. The horizontal grid and solid-colored cylinder are the overlayed virtual surfaces. Note that they are properly occluded into the physical scene.

Lesson 5: Using the Master to Define Virtual Surfaces

Suppose the operator wants to define a virtual surface that cannot be obtained from sensor-based structured light data, such as a spherical slave workspace or a vertical wall-constraint that will protect multiple pipes from being hit. In our testbed, the operator can easily define such a surface, by using the master to align it into the scene. The user can define a surface via the GUI, display it in the overlay, and move it around in the physical environment and augmented scene using the master. Furthermore, when a surface *is* defined by the structured light system, the operator has the option of adjusting its components' (e.g. the cylinder's and cutting plane's) positions. Thus, the Cobotic Hand Controller is used during one part of operation to drive the slave and in another part to move the virtual surface components. This allows for on-the-fly generation of user-defined virtual surfaces and compensation for sensor errors or failures, making the system more versatile and robust for unstructured environments.

Lesson 6: Re-registering the Surface After a "Mouse Jump"

Because the Cobotic Hand Controller, like any input device, has a limited translational workspace, we implemented the capability to "mouse jump" the master during use. Similar to picking up and repositioning a PC mouse, this capability allows the operator to reposition the master without moving the slave or virtual surface being adjusted. Mouse-jumping occurs in our testbed when the user holds down a key on the keyboard. However, if mouse-jumping when a virtual surface is defined, the virtual surface needs to be repositioned in the master's workspace. For example, suppose the saw in our demonstration is against the physical pipe and thus the operator's input is constrained by the virtual cylinder. If the operator mouse-jumps the master but the virtual cylinder is not repositioned, then his/her input may now be unconstrained although the saw and pipe have not moved! Therefore, when the mouse-jump motion is completed, the master repositions the virtual surface accordingly.

Table I. Saw blade rotation out of the cutting plane, for the act of cutting a pipe with and without virtual surfaces. For simplicity, angles are given generically as degrees rotated out of the cutting plane; ideally they would be zero.

	Angle 1	Angle 2
With V.S.	-0.1 to 0.1	-0.1 to 0.0
Without V.S.	-1.2 to 8.3	-9.9 to 3.3

$$ {}^{Int}\boldsymbol{V}_{New} = \boldsymbol{T}_{Rereg} \cdot {}^{Int}\boldsymbol{V}. \qquad (4) $$

Lesson 7: Improving Task Performance via the Virtual Surfaces

Once the testbed and its functionality were completed, demonstrations were run using it. One such demonstration was the one mentioned previously: practice cutting a large pipe with and without the aid of virtual surfaces. The task involved approaching the pipe, moving in both directions around the pipe as if cutting it, and moving away from the pipe. To reduce the risk of damaging the robot, the pipe used was made of cardboard and loosely held at its base. Thus, it was able to tilt or even tip over during collisions.

Position data for both trials is shown in Plate 6. The results clearly show that the virtual surfaces improved task performance with respect to positioning. With virtual surfaces, the saw was easily held flush against the cylinder, and restricted vertically to the cutting plane. On the other hand, without virtual surfaces the saw did not stay flush with the cylinder and in fact collided with it several times. Such collisions in actual application can be harmful to both the slave and the environment.

Furthermore, the virtual surfaces greatly decreased the chance of the saw blade binding or breaking (Plate 6 and Table I). With the virtual surfaces, the saw stayed in the cutting plane and did not rotate in either of perpendicular directions. Without the virtual surfaces, the saw blade moved vertically up and down from the cutting plane, and rotated up to almost ten degrees out of the plane. In actual application, such misalignment would most likely cause binding or even breaking of the saw blade.

Finally, the addition of the virtual surfaces made the task feel less mentally challenging. Instead of having to rely solely on the limited visual feedback, the operator was able to physically feel the cylinder via the virtual surfaces. Thus, the limited video views were used as a double check of saw's position, rather than the main source of feedback.

Therefore, it is clear from the results that utilizing haptic virtual surfaces improves task performance. With the virtual surfaces, the slave motion was smoother, there was less chance for binding/breaking of the saw blade or harmful collision between slave/saw and pipe, and the task appeared easier mentally.

Conclusions and future work

We have demonstrated the capability to create an over-the-Internet teleoperation testbed with virtual surfaces generated physically at the master manipulandum and reduced mental transformations in the interface. Our system allows

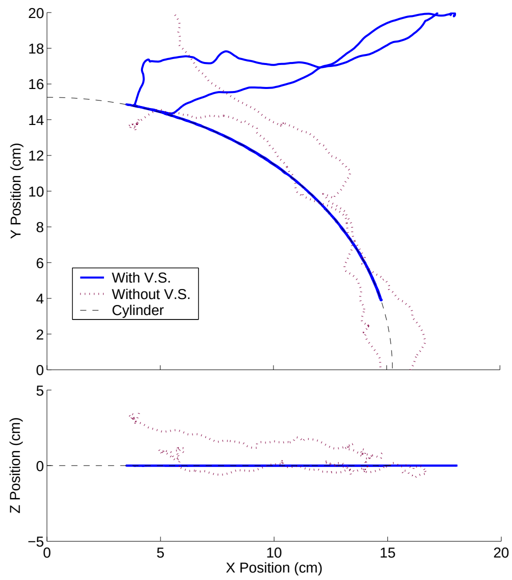


Plate 6. Saw blade position data for the act of cutting a pipe, with and without virtual surfaces. Note that with the virtual surfaces, the saw was held smoothly against the cylinder. Without the virtual surfaces, the saw moved in and out from the cylinder and up and down from the cutting plane (placed horizontal at Z=0). During actual operation, such collisions between saw and cylinder can damage the saw or slave robot, while such misalignment to the cutting plane can cause the saw blade to bind or even break.

for the creation and implementation of arbitrary virtual surfaces, defined by the structured light system and/or the operator with the aid of the master. Furthermore, the locally-defined nature of the virtual surfaces means that the operator has real-time haptic feedback information, thus helping to overcome time-delay difficulties. Most importantly, our teleoperation setup leads to dramatically improved task performance.

There are several avenues for future work. First, it would be beneficial to perform a more thorough examination of mental transformations, such as defining metrics so as to obtain a cost-function, or running additional experiments. Second, there are many teleoperation interfaces that cannot be easily configured to reduce mental transformations, including those in search and rescue and the military (which are often PDA- or suitcase-based). There may be other ways to decrease the mental transformations involved in such interfaces. Third, the language currently used to discuss and define virtual surfaces (especially those in higher dimensions) is very limited, as can be seen from the descriptions in this paper. A well-thought-out classification and language for virtual surfaces would be extremely useful. Finally, we need to continue our research into the Cobotic Hand Controller and its capabilities.

References

Argonne National Laboratory-East. Dual arm work platform teleoperated systems. *U.S. Department of Energy, Rep. DOE/EM-0389*, December 1998.

R. T. Azuma. A survey of augmented reality. *Presence*, 6 (4):355–385, August 1997.

Brightnoise Inc. www.brightnoise.com, 2004.

Caroline G. L. Cao. Augmented reality displays for endoscopic orientation and navigation. In *CHI '00 extended abstracts on Human factors in computing systems*, pages 91–92, 2000.

Brian P. DeJong. Improving teleoperation interfaces: Creation of a testbed and thinking inside the box. Master's thesis, Northwestern University, 2003.

Brian P. DeJong. Improving teleoperation: Reducing mental rotations and translations. In *IEEE International Conference on Robotics and Automation*, pages 3708–3714, April 2004.

Force Dimension. DELTA haptic device: 6-DOF force feedback interface. www.forcedimension.com/products/files/delta6dof.pdf, July 2004.

J. V. Draper and L. M. Blair. Dual arm work package performance estimates and telerobot task network simulation. In *American Nuclear Society Seventh Topical Meeting on Robotics and Remote Systems*, 1997.

Martin J. Farrell and Ian H. Robertson. Mental rotation and the automatic updating of body-centered spatial relationships. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 24(1):227–233, 1998.

Eric L. Fauling, J. Edward Colgate, and Michael A. Peshkin. A high performance 6-dof haptic cobot. In *IEEE International Conference on Robotics and Automation*, pages 1980–1985, April 2004.

P. Fuchs, F. Nashashibi, and D. Maman. Assistance for telepresence by stereovision-based augmented reality and interactivity in 3d space. *Presence*, 11(5):525–535, October 2002.

Haption. Virtual touch. www.haption.com/v3/eng/indexENG.htm, July 2004.

Stephen M. Kosslyn. *Image and Mind*. Harvard University Press, Cambridge, MA, 1980.

Geoffrey A. Landis. Robots and humans: Synergy in planetary exploration. In *Conference on Human Space Exploration*, February 2003.

M. W. Noakes, L. J. Love, and P. D. Lloyd. Telerobotic planning and control for DOE D&D operations. In *IEEE International Conference on Robotics and Automation*, pages 3485–3492, May 2002.

Young S. Park, Hyosig Kang, Thomas F. Ewing, Eric Faulring, J. Edward Colgate, and Michael Peshkin. Enhanced teleoperation for D&D. In *IEEE International Conference on Robotics and Automation*, pages 3702–3707, April 2004.

Michael A. Peshkin, J. E. Colgate, W. Wannasuphoprasit, C. A. Moore, R. B. Gillespie, and P. Akella. Cobot architecture. *IEEE Transactions on Robotics and Automation*, 17:377–390, 2001.

John J. Rieser. Access to knowledge of spatial structure at novel points of observation. *Journal of Experimental Psychology: Learning, Memory, and Cognition*, 15(6): 1157–1165, 1989.

L. B. Rosenberg. *Virtual Fixtures: Perceptual Overlays* *enhance operator performance in telepresence tasks*. PhD thesis, Stanford University, 1994.

SensAble Technologies. Phantom product brochure. www.sensable.com/products/phantom_ghost/phantom.asp, 2003.

R. N. Shepard and J. Metzler. Mental rotation of three-dimensional objects. *Science*, 171:701–703, February 1971.

Thomas B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, Cambridge, MA, 1992.

Thomas B. Sheridan and William R. Ferrel. Remote manipulative control with transmission delays. *IEEE Transactions on Human Factors in Electronics*, 4:25–29, 1963.

Mark Wexler, Stephen M. Kosslyn, and Alain Berthoz. Motor processes in mental rotation. *Cognition*, 68:77–94, 1998.