Two-Handed Direct Manipulation on the Responsive Workbench

Lawrence D. Cutler* Bernd Fröhlich Pat Hanrahan

Computer Science Department
Stanford University
{larryc, bernd.froehlich, hanrahan}@cs.stanford.edu

Abstract

We have built a system that allows users to naturally manipulate virtual 3D models with both hands on the Responsive Workbench, a tabletop VR device. Our design is largely based upon Guiard's observations of how humans distribute work between the two hands in the real world. We show how to apply these principles for the workbench environment and describe many issues encountered during the design. We first develop a framework for two-handed interaction and then explore a variety of two-handed 3D tools and interactive techniques. Related issues include how constraints are implemented and controlled by the two hands and how transitions between one-handed and two-handed tasks occur seemlessly. Informal observations of the system in practice show that users can perform navigation and manipulation tasks easily and with little training using the two-handed environment. One of our interesting findings was that users often performed two-handed manipulations by combining two otherwise independent one-handed tools in a synergistic fashion. In these cases, we did not program two-handed behaviors explicitly into the system; instead they emerged naturally.

CR Categories: I.3.6 [Computer Graphics]: Methodology and Techniques—Interaction techniques; I.3.7 [Computer Graphics]: Three-Dimensional Graphics and Realism—Virtual reality; H.5.2 [Information Interfaces and Presentation]: User Interfaces—Interaction styles;

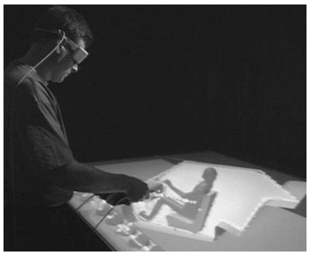
Keywords: two-handed interaction, virtual reality, three dimensional interaction, direct manipulation, 3D widgets, interactive 3D graphics

1 INTRODUCTION

Most current interactive 3D graphics applications are based on conventional desktop computing environments. Unfortunately these environments use two-dimensional input and output devices: a mouse for input and a CRT or flat-panel display for output. Such a 2D interface to a 3D world is often unnatural and unintuitive, and at worst frustrating and unproductive. 3D input and output technologies, such as a six degree of freedom (DOF) positional trackers and binocular stereo displays, provide much more flexibility than the mouse and CRT for modeling and visualizing 3D structures. But merely providing more channels of input and output is not enough to make the interface to a 3D virtual world natural and easy to use. In fact the flexibility afforded by more degrees of freedom makes the user interface design more complicated, not simpler. Design principles for such environments are largely unknown and certainly not in the same state of development as the modern desktop graphical user interface.

This paper describes two-handed interaction techniques that have been developed for the Responsive Workbench, a tabletop stereo display based on a workbench metaphor [9, 10]. It differs from most classical VR-systems, like head mounted displays (HMDs) and the CAVE [4], that try to immerse users completely in a virtual space. In contrast, the Responsive Workbench allows applications to set virtual objects on top of a real table which is integrated into the user's natural working environment. For example, an architectural design application places a virtual site model on the workbench to emulate the physical model. Two applications will be used in this paper to illustrate our interactive techniques: medical training and automotive design (Figure 1).

(a)



(b)

Figure 1: Two-handed interaction for two different applications on the Responsive Workbench. (a) Medical education and training: A skeleton rests on the workbench like a body on the operating table. Users can reach in and grab bones and organs as well as perform manipulations on the entire model. (b) Automotive Design: A finite volume model of a car interior is placed on the Responsive Workbench. Engineers use the model for air flow analysis. The images in this paper were taken by switching the viewpoint from the user's head position to the actual camera location.

The user stands in close proximity to virtual objects on the Responsive Workench, which enables two-handed direct manipulation techniques. The original interface to the system used simulated buttons at the front of the table to control different actions and a

*currently at Pixar Animation Studios

standard one-handed grasp-and-manipulate interface. Both button pushes and manipulations were controlled by a single 6 DOF stylus. However, it soon became obvious that this interface was very limiting and we sought to improve it.

In this paper we report on some of our initial results using two-handed input on the Responsive Workbench. In particular, we apply Guiard's framework on how humans manipulate objects with both hands in the real world. The most interesting techniques involve an asymmetric division of labor between the two hands. We discuss the basic building blocks implemented for two-handed interaction and explore various two-handed tools and techniques. We also describe many issues in the design, such as how to specify constraints and how to transition between different tasks in a natural way. Finally, we present results and observations of the system in practice.

2 RELATED WORK

Guiard [5] has studied everyday activities to understand how humans distribute work between their right and left hands. He classifies manual activities into three categories. Certain tasks are inherently *unimanual* such as throwing darts or brushing one's teeth. Other tasks are *bimanual symmetric*; both hands perform an identical action either in phase such as in weightlifting or out of phase such as milking a cow. A third class of activities is *bimanual asymmetric* where a complex coordination between the hands is required. Examples include dealing cards or playing a musical instrument. Guiard also defines an orthogonal division of labor, where the hands exhibit mutual independence, such as when working on two unrelated, unimanual tasks.

The most common activities involve an asymmetric division of labor between the left and right hand (we assume right-handed individuals throughout this paper). Guiard presents three high-level principles regarding the different roles of each hand during such a coordinated movement.

- The left hand adjusts the spatial reference while the right hand performs actions using this reference frame. As an example, the left hand is used to position and orient an object while the right hand operates a tool.

- Right-handed movements tend to have a higher temporal and spatial frequency in comparison to left-handed movements. The right hand is capable of producing fine-grained gestures, while the left hand performs gross manipulations.

- The left hand initiates the action.

Several researchers have applied Guiard's framework to design two-handed desktop interfaces. The best example is the Toolglass and Magic Lenses system developed by Bier et al.[1, 2]. In this system one hand controls the Toolglass, a transparent sheet containing overlaid tools, via a trackball, and the other hand controls a cursor that interacts with application objects through the Toolglass. Kabbash et al.[8] followed Guiard's framework in creating bimanual asymmetric techniques for a drawing/coloring task based on the Toolglass. They showed that two-handed techniques reduce the number of operations, minimize the cognitive load, and enhance performance. This evidence was supported by a follow-up study [11] with the Toolglass on two-handed techniques for 2D drawing tasks. All these studies suggest that Guiard's framework is useful in the design of two-handed computer interfaces.

In the past few years, several VR systems have been built that enable the use of both hands. THRED [18] is a 3D CAD system designed for sketching polygonal surfaces such as terrains. THRED uses two 6 DOF Polhemus trackers with added buttons for input. The division of labor is as follows: the non-dominant hand controls the interaction mode while the dominant hand handles spatial tasks such as picking and manipulating an object.

PolyShop [12] concentrates on symmetric two-handed techniques for scaling, rotating, and stretching objects and navigating through the scene. Users can also align objects with both hands via anchors and constraints. In the CHIMP system [14], the user performs a unimanual operation for translations and rotations, and a bimanual symmetric movement for scales. Recently work on CHIMP has focused on more asymmetric two-handed manipulation[13].

Hauptmann has studied how users specify graphical object manipulations with a mixture of conversation and hand gestures[6]. Many subjects expressed rotations by giving a steering wheel turn or a paddle wheel motion, and most specified scaling by moving the hands apart or together. These experiments suggest that meaning is often naturally conveyed via bimanual symmetric hand motions.

One of the most novel two-handed input systems is the environment for neurosurgical planning described by Hinckley et al.[7]. In this system the user manipulates "passive real-world props" with both hands. These props are physical, everyday objects with embedded 6 DOF trackers. For example, the left hand controls the head position with a doll's head prop while the right hand manipulates a cutting-plane with a rectangular plate prop. Both the 3-Draw system [17] and the Worlds in Minature (WIM) project [15] employ props in a similar manner. The advantage of props is that they give the user kinesthetic and tactile feedback which aids in manipulation, and their physical shape provides a crucial affordance as to their appropriate use in the system.

3 THE RESPONSIVE WORKBENCH

The Responsive Workbench [9, 10] is a virtual environment based on a high resolution tabletop display system. Users interact directly with three dimensional virtual objects, which are projected as stereoscopic images onto the surface of a table (Figure 2). A separate image is computed for each eye, and the computer quickly alternates the display of the two views. Users wear shutter glasses, which cover the left eye while the right eye's image is displayed, and vice versa, thus producing the stereoscopic effect. We attach a Polhemus 6 DOF sensor to the shutter glasses for head tracking. This allows the system to compute the correct perspective image for any user location.

Manipulation of virtual objects and navigation within the environment is controlled by a Polhemus stylus, a pen-like 6 DOF input device, and Fakespace's PINCH gloves equipped with Polhemus 6 DOF sensors on the back of each hand. The stylus tip provides a single distinguished point of action, whereas for the PINCH gloves, such a point is not well-defined. We decided to use the position where the thumb and index finger meet as the point of action, and we estimate this point by adding a constant offset to the position information provided by the Polhemus. The stylus is a one button device, while PINCH gloves detect different pinches between fingers. The original system as described in [9, 10] used a Virtual Technologies CyberGlove, which also provides joint angle information for gesture recognition, but requires more extensive calibration than the PINCH gloves.

Navigation in traditional immersive VR systems using HMDs entails flying, walking, or driving around. In contrast, navigation on the Responsive Workbench typically exploits the natural spatial reference frame provided by the tabletop. We identified four basic navigational tasks:

1. The user slides the model around on the table plane, lifts it up, or pushes it back down.

2. The model is rotated around one of the principal axes that are naturally defined by the tabletop or by the model.

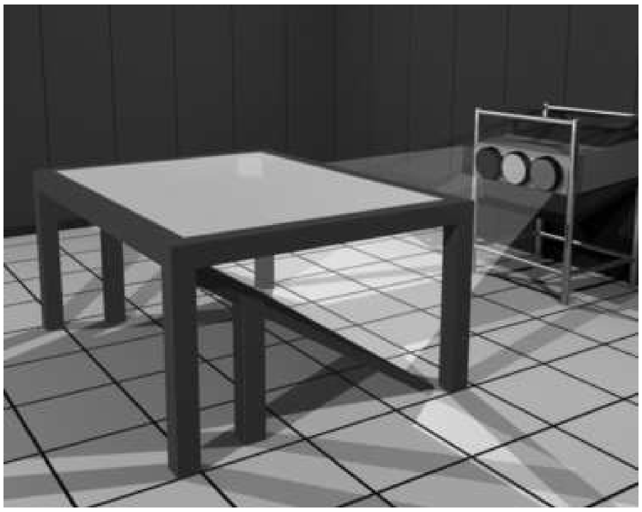


Figure 2: The Responsive Workbench. A video projector projects a high resolution stereoscopic image through a mirror onto the table top. The system is currently driven by a Silicon Graphics Onyx RealityEngine2 graphics system at a resolution of 1025x768 pixels at 96Hz, 48Hz per eye.

3. The user zooms in or out by enlarging or shrinking the scene.
4. The user changes his or her position relative to the table and consequently relative to the model, e. g. by walking around the table or by moving the head closer to or away from the model.

The table serves in some sense as a large physical prop, since the model is anchored on the table and the user's head position is tracked with respect to the table. The visible parts of the model are mostly within arms length reach of the user, which enables easy direct manipulation of the scene and of objects in the scene.

4 BASIC BUILDING BLOCKS

We developed a system framework to support multiple input devices and two-handed interaction. During the initial phases of design, we decided on a virtual tools-based approach similar to [16, 19], mainly because the Responsive Workbench resembles a physical workbench. We created three basic building blocks: manipulators which encapsulate input devices, tools which define the interactions, and toolboxes which allow for transitions between different tools.

Manipulators

Manipulators provide a logical abstraction for 3D input devices. Each device supplies the manipulator with position and orientation data as well as button click information. Manipulators also provide a mechanism for attaching to the various unimanual and bimanual tools. Our system supports both one-handed and two-handed manipulators. One-handed manipulators encapsulate a single-handed device and allow it to pick up a one-handed tool. The unimanual manipulators in our system are the stylus, the left glove, and the right glove. Two-handed manipulators bind two one-handed manipulators together. The two-handed manipulator can pick up a two-handed tool, or it can allow either one-handed manipulator to pick up a one-handed tool. In our system a two-handed manipulator can bind two gloves or a glove and a stylus (Figure 3).

We designed the manipulators so that two-handed behaviors would be developed independent of specific devices. This leads to

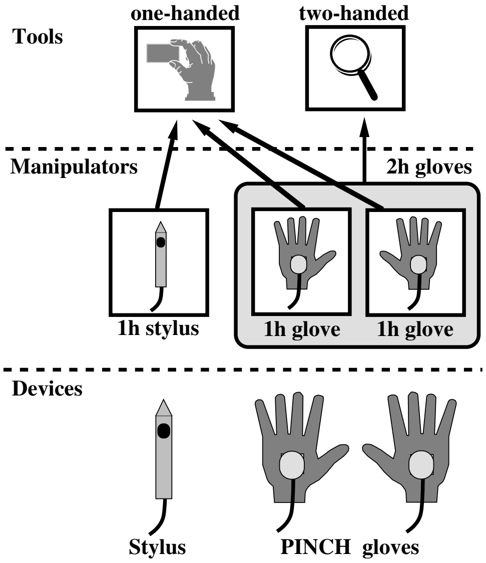


Figure 3: Basic Building Blocks. Devices refer to the physical input hardware, and manipulators provide an abstraction for these devices. Each of the devices is encapsulated by a one-handed manipulator. Layered on top of that, two-handed manipulators bind together two one-handed manipulators for bimanual interactions. The arrows in this diagram represent the possible attachments between manipulators and tools. By combining the one and two-handed manipulators in the above manner, we allow the user to pick up both one-handed and two-handed tools with the pinch gloves.

an extensible system that can easily incorporate new input technologies. Furthermore, we wanted to experiment with different combinations of input devices, such as the glove with the stylus. An interesting combination uses all three devices simultaneously. The user wears both gloves for two-handed manipulation but picks up the stylus in one hand to perform precision tasks. Thus, the user can choose the more appropriate manipulator for a given situation.

Tools

Tools are used to perform specific tasks. Table 1 lists all of the one and two-handed tools referenced in this paper. We first implemented a one-handed 6 DOF grab, which allows the user to pick up a single object and move it around freely. We also created one-handed visualization tools, such as a cutting plane and an opacity tool. The automobile application has tools specifically designed for scientific visualizations: a temperature plane provides visualizations of 2D temperature slices, while two other tools emit particles or streamlines into the airflow. One-handed tools are usually initiated with a stylus click or pinch (thumb-to-index).

In contrast, two-handed tools engage both hands in a synergistic fashion. These tools display two 3D virtual cursors, one for each hand, to indicate the appropriate division of labor. Typically, the left hand has a coordinate system to signify positional subtasks, while the right hand shows a cursor implying specific functionality, such as a magnifying glass for zooms. Most of our two-handed tools deal with global scene positioning (zooms, rotations, and translations) or single object manipulations. Two-handed tools are usually initiated when both hands are pinched.

Type	Description
Unimanual	
one-handed grab	Pick up a single object and move it freely.
panning	Slide the model on the table top.
cutting plane	Cut away a portion of the model or a portion of a single object.
opacity	Adjust the transparency of the skin for the medical application.
temperature	Visualize 2D temperature slices within an automobile.
particle	Emit particles into an air stream.
streamline	Seed stream lines into an air stream.
Bimanual symmetric	
symmetric scale	Shrink or enlarge objects by moving both hands apart or together.
slide-and-turn	Slide and turn the model on the table top.
turntable	Turn the model on the table top about a fixed axis of rotation.
grab-and-twirl	Carry and turn an object around with both hands. Each hand can also be used independendly as a one-handed grab tool.
grab-and-carry	Similar to the grab-and-twirl tool except it does not allow roll around the line connecting the two hands.
Bimanual asymmetric	
grab-and-scale	Left hand positions object while right hand moves towards or away from it.
trackball	Left hand positions object while right hand rotates it about its center.
zoom	Left hand positions the model and specifies the zoom region, right hand moves towards or away from the left hand to specify the zoom factor.
free rotation	Left hand positions the model. The axis of rotation is specified by the left hand's orientation. Right hand rotates around left hand.
axis rotation	Similar to the free rotation except the axis of rotation snaps to one of the principal axes.
heuristic rotation	Similar to the free rotation except the axis of rotation is constrained to one of the principal axes and infered from the motion of the right hand.
pinch rotation	Similar to the free rotation except the axis of rotation is specified by a pinch gesture with the left hand.
constrained translation	Left hand specifies a line or plane constraint, right hand translates.

Table 1: List of tools referenced in this paper

Toolboxes

Toolboxes allow the user to transition between different tools. They also give structure and organization to the system (Figure 4). In our original design, the user could place tools anywhere on the table, similar to a real-world workbench. Although such a system gives the user great flexibility, the table soon becomes cluttered and messy (like the real workspace). The toolbox groups the tools in a clear manner, much as toolbars do on many desktop applications. We did not implement hierarchical and movable toolboxes, but these extensions could be easily integrated.

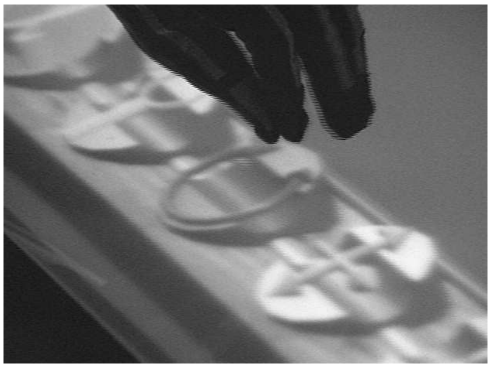


Figure 4: Closeup of a toolbox containing various tools. Our system supports multiple toolboxes which group similar tasks together. The tools are displayed as 3D icons that visually represent their functionality. Users can pick up a tool by clicking on it with the stylus or by pinching it with the gloves. After finishing with a tool, the user can either pick up a different tool or simply drop the current one in the toolbox, in which case the system returns to a default behavior.

5 TWO-HANDED INTERACTION

Once we developed an underlying system foundation, we explored two-handed interactions on the Responsive Workbench. We implemented both bimanual symmetric and bimanual asymmetric tools for the Workbench.

Coordinated Symmetric Interaction

We present five different types of two-handed symmetric tools: symmetric scale, slide-and-turn, turntable, grab-and-carry, and grab-and-twirl. The scaling tool shrinks or enlarges objects by moving both hands together or apart, similar to [3, 12, 14].

The slide-and-turn (Figure 5) allows the user to perform a steering-wheel motion on the table top. This tool exploits the fact

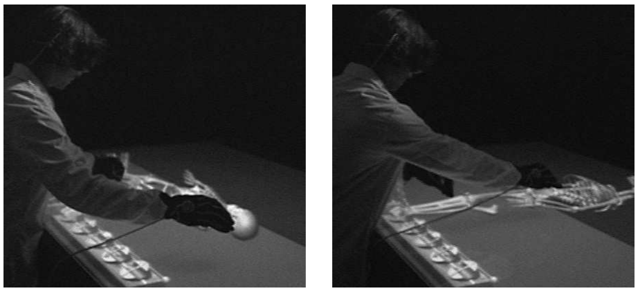


Figure 5: The slide-and-turn tool: We use both hands in a symmetric manner to simultaneously rotate and slide the model on the table. The axis of rotation is always constrained to be perpendicular to the table plane.

Figure 6: The zoom tool: The left hand positions the skeleton and provides the focal point, while the right hand zooms in by moving away from the focal point. Similarly, one can zoom out by moving the left hand towards the focal point.

that many of our models rest on the table. The user pinches with both hands, which locks the scene to the center of the line connecting the two hands. The scene's movement is defined by a translation of the center of the line segment and a rotation around the center. The axis of rotation is fixed to be perpendicular to the table top, and the translation is constrained to the table plane. This tool gives users the flexibility of sliding the model on the table plane in addition to rotating it. We also implemented a second variation, the turntable, which fixes the rotation axis position at the start of the rotation and does not translate. This alternative behaves more like a real turntable, but provides less flexibility than the slide-and-turn.

The grab-and-carry lets the user hold onto an object with both hands, and "carry" it as well as turn it around. This tool is functionally identical to the slide-and-turn, except that its axis of rotation is not constrained nor is the translation. This widget has five degrees of freedom as we do not roll the object around the line connecting the two hands. The grab-and-twirl adds this sixth degree of freedom. The object's roll can either be controlled by the left hand's roll, the right hand's roll, or a combination of the two. We decided to use the right hand's roll, which introduces a slight asymmetry into the tool but provides the user with more direct control than a combination of the two rolls.

Both tools work well for large objects where it is easy and natural to pinch at the ends of the object. In such cases, it gives the user more control over an object than the one-handed grab. This interface is very similar to how we grab and maneuver objects with both hands in the real world, and it shows how two 6 DOF manipulators can effectively interact to specify a 6 DOF motion for a virtual object.

Coordinated Asymmetric Interaction

For scene navigation, we implemented the following two-handed asymmetric tools: constrained translation, zoom, and several variations of a rotation tool. With the translation tool, the left hand specifies a line or plane constraint while the right hand translates the scene.

Two-handed zooming (Figure 6) allows the user to focus on a specific region of the scene. This tool introduces a tight coupling between the two hands that conforms to Guiard's principles on asymmetric bimanual activity. The left hand initiates the action and sets up the reference frame for the right hand in two ways: it positions the model before the zoom, and it provides the focal point for the manipulation. This type of zoom operation is perceived like a three-dimensional version of a zoom with a camera, but it does not have an equivalent in the real world. In an immersive environment, the exact same operation would make the user feel as if he or she were shrinking or growing with respect to the surrounding scene.

The free rotation tool divides the labor between the left and the right hand in the following manner: The left hand translates the model, but also specifies the position and orientation of the rotation axis by holding onto a virtual axis. The right hand performs the actual rotation by circling around this virtual axis. The non-dominant hand provides the reference frame for the dominant hand in three different ways. First, the user can position the scene in preparation for the rotation, much as with the zoom tool. Next, the user specifies the axis position, similar again to the zoom tool. Finally, the user specifies the orientation of rotation axis with the left hand. We realized that for certain applications users only wanted to rotate the scene around one of the principal axes, which are naturally defined by the table top. The next section deals with the specification of constraints for such situations.

We also implemented a set of two-handed tools for object manipulations, where the left hand grabs the object and positions it, while the right hand performs the intended action. The grab-and-scale works similarly to the two-handed zoom above. Since it is difficult to specify a focal point for small objects, we always scale about the center of the object's bounding sphere. Another tool of this type is the trackball, which rotates an object about the center of its bounding sphere.

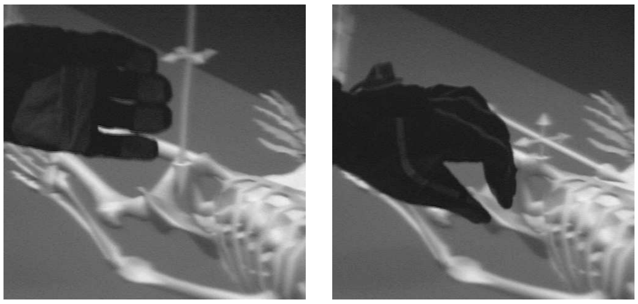
6 CONSTRAINTS

Constraints can greatly simplify tasks in a virtual environment. Many of our tools constrain the axis of rotation. We also constrain translations to occur along an axis or plane. We have not yet implemented alignment tools that restrict object position with respect to other objects.

There must also be a natural means for the user to specify the constraint. In our system, the left hand usually articulates the constraint while the right hand performs the intended action. We explored four different techniques:

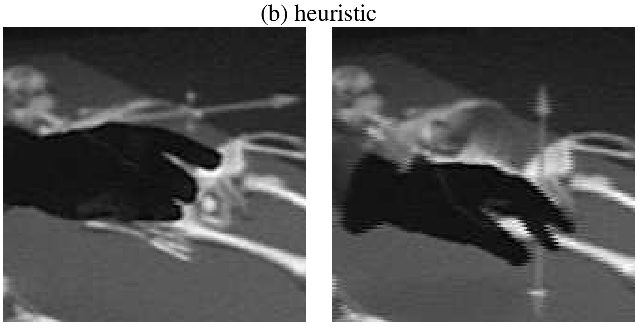
- **Built Into the Tool:** Many tools have a constraint built into their behavior. On the Responsive Workbench, built-in constraints often exploit the horizontal table top. The symmetric slide-and-turn described earlier (see Figure 5) restricts rotations around the axis perpendicular to the table plane. Similarly, the panning tool forces translations along the table top.

- **Hand Orientation:** The axis rotation tool computes the rotation axis based upon the user's hand orientation (see Figure 7a). Similarly, the constrained translation determines the line or plane from the orientation of the non-dominant hand. An open palm signifies planar translations while a closed fist denotes movements along a line.



(a) hand orientation

(b) heuristic



(c) pinching

Figure 7: Constraining rotations around one of the three principal axes. We implemented three different methods for specifying the axis of rotation. (a) Hand orientation: The user orients the left hand along the desired axis. The axis of rotation snaps to the closest principal axis. (b) Heuristic: We infer the axis once the right hand begins rotating the scene. (c) Pinches: The user selects the axis by specifying one of three different pinches with the left PINCH glove.

- **Heuristic from Motion:** The heuristic rotation infers the axis of rotation based on the direction of the right hand's motion (see Figure 7b). The heuristic chooses a principal axis once the rotation angle of the right hand around this axis exceeds a certain threshold. We can also determine the line or plane of a constrained translation based on the translation path of the right hand, but this is currently not implemented.

- **Pinch Gestures:** A fourth approach maps the axis of rotation to different finger pinches (see Figure 7c), e. g. thumb to index maps to the x, thumb to ring to y, and thumb to pinky to z.

In comparing these approaches, heuristics involve an implicit specification of the constraint which tends to place minimal cognitive load on the user. However, any heuristic still has the potential of choosing incorrectly. Built-in constraints are also implicit and work very well with our system. Of course, users lose flexibility with built-in constraints as they can only rotate about one axis or translate within a single plane. The hand orientation approach generally requires brief training since we do not manipulate objects in the real world with gestures. However, this solution is explicit and visual with a direct mapping between the hand orientation and the axis or plane. Once learned, it gives users extensive control over the rotations or translations. Finally, specifying constraints with pinches involves an arbitrary mapping between the pinch and the constraint. With the pinch rotation tool, users can quickly learn the mapping since there are only three options. On the other hand, if we consider a pinch translation tool with six choices (three plane constraints and three line constraints), the mapping quickly becomes complicated and difficult to perform.

7 TRANSITIONS

Many tasks decompose into a number of sequential subtasks. Transitions refer to the change from one subtask to another. Two-handed input introduces an additional layer of complexity in handling transitions between one-handed and two-handed tools. We explore two explicit transition methods, toolbox and power widget transitions, as well as one implicit approach.

Toolbox Transitions

The toolbox provides an explicit means for the manipulators to transition between different tools. A manipulator attaches to a tool when the user picks it up and detaches when it gets dropped off (Figure 4). Toolbox transitions raise the interesting issue of what happens when a manipulator has not selected a tool (which occurs at startup time or after a tool has been dropped off). In these situations, the manipulator reverts back to a default tool. One-handed manipulators associate with a one-handed default. Two-handed manipulators can have a two-handed default or each hand can associate with a different one-handed default tool. For example, in the medical application, the default for each PINCH glove is a one-handed grab. When a tool has not been selected from the toolbox, the user can still pick up and move individual bones with the left and right hands. Other applications might define a different set of default behaviors. Figure 8 shows how toolbox transitions are handled in our system for a two-handed manipulator with two independent one-handed defaults.

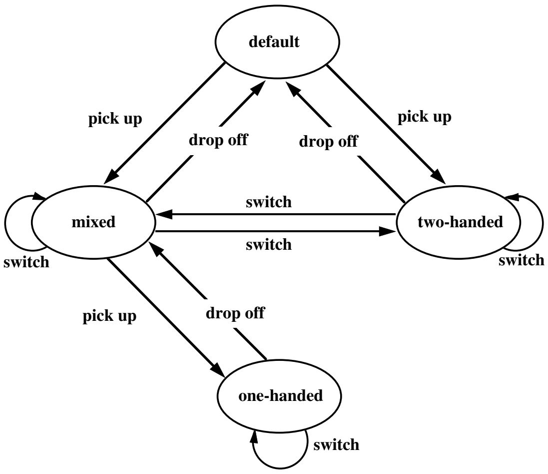


Figure 8: The state diagram for toolbox transitions for a two-handed manipulator with two one-handed defaults. The transitions define if a one-handed or a two-handed tool gets picked up, switched, or dropped off. The two-handed manipulator switches between one of the following states: (**Default**) Both hands have the one-handed default tool. (**Mixed**) One hand has a default tool, the other has a one-handed tool. (**Two-Handed**) Both hands are attached to a two-handed tool. (**One-Handed**) Each hand has a one-handed tool.

Power Tool Transitions

A disadvantage in using the toolbox is that users have to alternate back-and-forth between the toolbox and the area of interest. Power tools provide an explicit mechanism which can reduce the time spent switching back-and-forth. A power tool allows the non-dominant hand to control transitions between tools as well as engage in two-handed interactions. It combines the functionality of two or three tools by mapping each to a different pinch gesture. Thus, power tools can group related tasks and minimize the cognitive load from context switching.

Implicit Transitions

We also experimented with a more subtle method of transitioning, where the user is less aware that the transition is occuring. We define the grab-and-twirl (Figure 9) as the two-handed default behavior for an interaction. Initially, the system is identical to having the two one-handed grab defaults. But, often times the user will reach in with the second hand to help manipulate a grabbed object. At this point our default tool switches to the two-handed operation. The user can now twirl the object around with both hands. This transition occurs naturally much as one would fluidly switch from a one-handed to two-handed grab in the real world. The implicit transition was implemented by coupling a one-handed behavior with a two-handed behavior in the same tool. This technique could be extended to other situations as well.

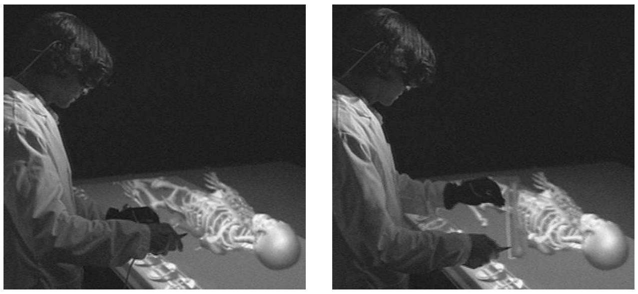


Figure 9: The grab-and-twirl tool. The user first grabs an object with either hand and manipulates the object as a one-handed 6 DOF grab. At some point, the user pins the object with the other hand and performs a symmetric object twirl. Thus the user experiences a non-explicit transition between a one-handed grab and a two-handed grab.

8 RESULTS

We observed people using the Responsive Workbench while giving demos and during a planned informal user observation session. We showed users a skeleton and asked them to complete several positioning and manipulation tasks such as "zoom in on the kneecap" or "orient the skeleton vertically towards you and zoom in on the heart." In observing users, we hoped to ascertain whether our two-handed tools and the system as a whole were natural and intuitive for people. Specifically, we attempted to answer the following questions during these observations:

- Does Guiard's framework provide a good basis for designing two-handed interactions?
- Do users find constrained operations useful for positioning and orienting the entire scene (or is it more effective to simply do a 6 DOF grab)?
- Are transitions fluid and unobtrusive to the user?

On the whole, users found the two-handed tools natural and easy to manipulate (see video proceedings). Users became proficient after no more than a minute or two of instruction. During our observations, we also found that users often picked up two seemingly independent one-handed tools and used them together in a coordinated fashion. We noticed the following examples of this emergent behavior: First, users positioned the skeleton with the left hand while grabbing a bone or applying a cutting plane with the right hand. Second, users positioned a temperature cutting plane into the car with the left hand while injecting particles with the right hand. These examples might be seen as the juxtaposition of two independent activities, but in each case the left hand sets up a reference frame for the right hand. This shows that Guiard's observations are a sensible framework for the implementation of direct two-handed manipulations in a virtual environment such as the Responsive Workbench.

Users noticed that constraints often provide direct means to achieve the task they had in mind. Even basic tasks like turning around the skeleton on the table plane turned out to be quite difficult with the one-handed grab tool. Users usually needed multiple grab and move operations, since our hand is in fact not an unconstrained 6 DOF device. Using the slide-and-turn or one of the axis rotation tools allowed them to perform this task with ease.

One of the more suprising results was that the asymmetric combination of a PINCH glove for the left hand and stylus for the right hand worked much better in many situations than the two PINCH gloves, especially for asymmetric tasks. The stylus is a thinner input device with a distinguished point of action, and it serves at the same time as a pen-like physical prop. The asymmetric combination of input devices mirrors the asymmetric distribution of labor and is very much in tune with Guiard's observation that the right hand is capable of performing finer grained gestures than the left hand.

9 SUMMARY AND FUTURE WORK

We have described a system that allows a user to naturally manipulate virtual models with both hands as they are displayed on the Responsive Workbench. The most interesting two-handed interactions are coordinated and asymmetric: both hands perform different subtasks in a synergistic way to get a complex task done. When designing the system we took advantage of several design principles developed by Guiard from studies of how people use their hands. These principles are effective guidelines for VR environments. We have also investigated a variety of methods for interactively specifying 3D constraints, and for transitioning smoothly between subtasks.

When beginning this work we thought that all the two-handed input techniques would need to be explicitly designed and programmed. However, when using the system we found that perhaps the most interesting tasks emerged when the user combined two otherwise independent unimanual tools. For example, in the scientific visualization system for automobile cabin modeling, the user controlled the slicing plane showing temperature with one hand and the source of particles used to generate streamlines with the other hand. Presumably the user was testing a hypothesis that temperature distribution depends on the air flow. In retrospect, such emergent interactions are not that surprising since this is how many two-handed operations arise in the real world. From a systems point of view, adding uncoordinated two-handed input to an existing one-handed system is relatively easy, but already very powerful.

One of the areas that needs further studies are the methods that map the additional degrees of freedom provided by more input channels into simple actions. New technologies to sense the user increase the numbers of channels of input data: e. g. we receive position and orientation data for both hands, and potentially multiple joint angles. This additional data is needed to capture "natural" motions, but using all of the input channels directly can make precise manipulations difficult. Instead, intelligently mapping the various input degrees of freedom into a lower degree tool often provides the user with more control. For example, the grab-and-twirl combines two 6 DOF inputs into an easy to use 6 DOF manipulator. Another means of reducing the degrees of freedom is by specifying constraints. In our system, the additional degrees of freedom from one input device are actually used to restrict the degrees of freedom in the other input.

A very interesting area of future research is to have the system infer that the user is using both hands in a cooperative manner and to help coordinate the task further. Consider the example when each hand holds a one-handed grab tool which can be used to pick up two different objects. If the same object were picked by both hands, the two one-handed grabs could be coordinated as if they were a tightly coupled, two-handed tool. This potential scenario raises a host of interesting issues. For example, how does the system decide when both hands are being used together? How does it transition between one-handed and two-handed modes? And finally, are there interesting three-handed interactions, where the user provides two hands and the computer a third to assist?

Acknowledgements

We would like to thank Maneesh Agrawala and Andrew Beers for contributing many ideas, Terry Winograd for valuable discussions, Tamara Munzner for helping with the video editing process, Sina Mostafawy for modeling the toolbox, and Gerold Wesche for help with encapsulating the visualization tools. GMD, Germany's National Research Center for Information Technology, developed part of the basic workbench software that was used for this project. GMD and Daimler-Benz AG implemented the visualization tools for the automotive application. The finite volume dataset is courtesy of Daimler-Benz AG. This work was supported by NASA under contract NAS2-14308, Interval Research Corporation, and Silicon Graphics, Inc.

References

[1] Eric Bier, Maureen Stone, Ken Fishkin, William Buxton, and Thomas Baudel. A taxonomy of see-through tools. In *ACM Annual conference on Human Factors in Computing Systems*, pages 358–364. ACM, Addison-Wesley, April 1994.

[2] Eric A. Bier, Maureen C. Stone, Ken Pier, William Buxton, and Tony DeRose. Toolglass and Magic Lenses: The see-through interface. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 73–80, August 1993.

[3] Richard A. Bolt and Edward Herranz. Two-handed gesture in multi-modal natural dialogue. In Gerrit C. van der Veer, Sebastiano Bagnara, and Gerard A. M. Kempen, editors, *Proceedings of the ACM SIGGRAPH Symposium on User Interface Software and Technology*, pages 7–14, 1992.

[4] Carolina Cruz-Neira, Daniel J. Sandin, and Thomas A. DeFanti. Surround-screen projection-based virtual reality: The design and implementation of the CAVE. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 135–142, August 1993.

[5] Yves Guiard. Symmetric division of labor in human skilled bimanual action: the kinematic chain as a model. *The Journal of Motor Behaviour*, 19(4):486–517, 1987.

[6] Alexander G. Hauptmann. Speech and gestures for graphic image manipulation. In Martin Helander, editor, *Proceedings of ACM CHI'89 Conference on Human Factors in Computing Systems*, pages 241–245, 1989.

[7] Ken Hinckley, Randy Pausch, John C. Goble, and Neal F. Kassell. Passive real-world interface props for neurosurgical visualization. In Gavriel Salvendy, editor, *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, pages 452–458, 1994.

[8] Paul Kabbash, William Buxton, and Abigail Sellen. Two-handed input in a compound task. In Gavriel Salvendy, editor, *Proceedings of ACM CHI'94 Conference on Human Factors in Computing Systems*, pages 417–423, 1994.

[9] Wolfgang Krüger, Christina-A. Bohn, Bernd Fröhlich, Heinrich Schüth, Wolfgang Strauss, and Gerold Wesche. The responsive workbench: A virtual work environment. *IEEE Computer*, pages 42–48, July 1995.

[10] Wolfgang Krüger and Bernd Fröhlich. The responsive workbench. *IEEE Computer Graphics and Applications*, pages 12–15, May 1994.

[11] Andrea Leganshuk, Shumin Zhai, and William Buxton. Bimanual direct manipulation in area sweeping tasks. *http://www.dgp.utoronto.ca/people/andrea/bimanual.html*, 1996.

[12] Daniel P. Mapes and J. Michael Moshell. A two-handed interface for object manipulation in virtual environments. *Presence*, 4(4):403–416, 1995.

[13] Mark R. Mine. *Personal communications 9/20/96*.

[14] Mark R. Mine. Working in a virtual world: Interaction techniques used in the chapel hill immersive modeling program. *Technical Report 1996-029*, 1996.

[15] Randy Pausch, Tommy Burnette, Dan Brockway, and Michael E. Weiblen. Navigation and locomotion in virtual worlds via flight into Hand-Held miniatures. In Robert Cook, editor, *SIGGRAPH 95 Conference Proceedings*, Annual Conference Series, pages 399–400. ACM SIGGRAPH, Addison Wesley, August 1995. held in Los Angeles, California, 06-11 August 1995.

[16] Timothy Poston and Luis Serra. The virtual workbench: Dextrous VR. In *Virtual Reality Software and Technology (Proceedings of VRST'94, August 23-26, 1994, Singapore)*, pages 111–122, Singapore, August 1994. World Scientific Publishing.

[17] Emanuel Sachs, Andrew Roberts, and David Stoops. 3-draw: A tool for designing 3D shapes. *IEEE Computer Graphics and Applications*, 11(6):18–26, November 1991.

[18] Chris Shaw and Mark Green. Two-handed polygonal surface design. In Gerrit C. van der Veer, Sebastiano Bagnara, and Gerard A. M. Kempen, editors, *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 205–212, 1994.

[19] Matthias M. Wloka and Eliot Greenfield. The virtual tricorder: A uniform interface for virtual reality. In Gerrit C. van der Veer, Sebastiano Bagnara, and Gerard A. M. Kempen, editors, *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 39–40, 1995.