

A Survey of Design Issues in Spatial Input

Ken Hinckley^{1,2}, Randy Pausch², John C. Goble¹, and Neal F. Kassell¹

University of Virginia
Departments of Neurosurgery¹ and Computer Science²

{kph2q, pausch, jcg7q}@virginia.edu, neal@msmail.neuro.virginia.edu

ABSTRACT

We present a survey of design issues for developing effective free-space three-dimensional (3D) user interfaces. Our survey is based upon previous work in 3D interaction, our experience in developing free-space interfaces, and our informal observations of test users. We illustrate our design issues using examples drawn from instances of 3D interfaces.

For example, our first issue suggests that users have difficulty understanding three-dimensional space. We offer a set of strategies which may help users to better perceive a 3D virtual environment, including the use of spatial references, relative gesture, two-handed interaction, multisensory feedback, physical constraints, and head tracking. We describe interfaces which employ these strategies.

Our major contribution is the synthesis of many scattered results, observations, and examples into a common framework. This framework should serve as a guide to researchers or systems builders who may not be familiar with design issues in spatial input. Where appropriate, we also try to identify areas in free-space 3D interaction which we see as likely candidates for additional research.

An extended and annotated version of the references list for this paper is available on-line through mosaic at address <http://uvacs.cs.virginia.edu/~kph2q/>.

KEYWORDS

Spatial input, virtual reality, 3D interaction, two-handed input, ergonomics of virtual manipulation, haptic input

INTRODUCTION

The term *spatial input* refers to interfaces based upon free-space 3D input technologies such as camera-based or magnetic trackers [52][53], as opposed to desktop devices such as the mouse or the Spaceball [58]. In the literature, a wide variety of interfaces for manipulating three-dimensional objects have been described as “3D interfaces,” but we find it useful to use the term *spatial input* to distinguish the class of 3D interfaces based upon free-space interaction.

Thus, rather than trying to identify issues which are applicable to all forms of 3D input, we restrict the present survey to interfaces that employ free-space input devices. Also, to maintain the focus of the survey, we do not discuss general techniques for graphical interaction, such as progressive refinement [4][11], nor do we describe algorithms to overcome artifacts of existing spatial input devices, such as techniques for filtering noise and lag from tracker data [1][42]. Instead we focus on issues which are specific to spatial interaction techniques.

Many results in spatial input are scattered across the literature, without an overall structure in which to view them. The interface designer is faced with numerous descriptions of applications and experiments, without order, organization, or a common nomenclature. There have been a few publications which extract common themes from the examples and studies available, or distill this information into practical suggestions. To make some headway on this problem, the present work seeks to synthesize many results into a common framework, in the form of a series of design issues.

The design issues we present are not well-formulated principles of design or ready-to-go solutions. Rather we present some issues to be aware of and some different approaches to try. Few of the design issues we present have been subjected to formal user studies, so they are supported only by possibly unrepresentative user observations. Nonetheless we believe the present survey of design issues will serve as a useful guide and starting point for the community of designers and researchers wishing to investigate spatial input.

PREVIOUS WORK

Previous work in spatial interaction consists largely of two flavors: single applications built for users with specialized tasks, and formal user studies which analyze individual phenomena in isolation. Example applications include the 3DM three-dimensional modeler [14], Ostby’s system for describing and modifying free-form surfaces [49], or Sachs’s 3-Draw computer-aided design tool [54]. Example formal user studies include Jacob’s multidimensional input experiment [37] or various studies of 3D point selection [62][64].

Unfortunately, there are few papers which attempt to bridge the gap between these two different types of research results. How can the techniques which work in one specific application be applied to another 3D interface? How can the results of a formal study be applied in a nuts-and-bolts fash-

ion to a given 3D interface design problem? While the present survey cannot adequately address these larger questions, we hope that the survey can help researchers and designers who may be unfamiliar with spatial input techniques to ground themselves in the field.

Brooks offers many insightful observations about 3D interfaces in his 1988 SIGCHI plenary address [11]. Our hope is to supplement Brooks's observations with some additional issues which are described in the literature and which we have experienced in our research. We reference some of Brooks's observations, but the reader should be aware that many important issues presented in Brooks's paper are not covered by the present survey.

Nielsen's discussion of noncommand user interfaces [47] covers similar ground, but the scope of Nielsen's work is much broader than this survey. Nielsen's goal is to describe trends in advanced interface design, while by contrast, our goal is to discuss design issues in one class of advanced interfaces, those that employ 3D free-space input.

DESIGN ISSUES

We have grouped our observations into two major categories, those dealing with human perception, and those dealing with ergonomic concerns. This is purely an organizational convenience, rather than an indication of some fundamental difference between the two types of issues.

Human Perception

1: *Understanding three-dimensional space vs. experiencing three-dimensional space*

Anyone who has tried to build a stone wall knows how difficult it is to look at a pile of available stones and decide which stone will best fit into a gap in the wall. There are some individuals, such as experienced stone masons, who have become proficient with this task, but most people simply have to *try* different stones until one is found that fits reasonably well.

In general, people are good at experiencing 3D and experimenting with spatial relationships between real-world objects, but we possess little innate comprehension of 3D space in the abstract. People do not innately *understand* three dimensional reality, but rather they *experience* it.¹

From a perceptual standpoint, we could argue that our difficulty in building stone walls, and in performing abstract 3D tasks in general, is a result of our sub-conscious, rather than conscious, perception of 3D reality. For example, the Shepard-Metzler mental rotation study [57] suggests that for some classes of objects, we must mentally envision a rigid body transformation on the object to understand how it will look from different viewpoints; that is, we must perceive the motion to understand the effect of the transformation.

Previous interfaces have demonstrated a number of issues which may facilitate 3D space perception, including the following:

- Spatial references
- Relative gesture vs. absolute gesture
- Two-handed interaction
- Multisensory feedback
- Physical constraints
- Head tracking techniques

We do not wish to suggest that all spatial interfaces must consider all these issues to be usable. Rather, the designer should consider these issues as a set of approaches which might be applied to a given design problem.

We now further explain these issues using examples drawn from instances of 3D interfaces.

1.1: *Spatial references*

Badler [2] describes an interface where a stylus is used to control the position of a virtual camera. One version of the interface allows the user to indicate the desired view of an imaginary object using the stylus. Badler reports that "the lack of spatial feedback [makes] positioning the view a very consciously calculated activity."

Badler repeated the experiment with a real (as opposed to imaginary) object. He digitized a plastic spaceship and allowed the user to specify the virtual camera view of the corresponding wireframe spaceship by positioning and orienting the wand relative to the real-world plastic spaceship. With this single change, Badler's "consciously calculated activity" suddenly became "natural and effortless" for the operator to control.

In general, to perform a task, the user's perceptual system needs something to refer to, something to *experience*. In 3D, using a spatial reference (such as Badler's plastic spaceship) is one way to provide this perceptual experience. More precisely, we define a *spatial reference* as a real-world object relative to which the user can gesture when interacting in 3D.

Ostby's system for manipulating surface patches [49] was a second early system to note the importance of spatial references. Ostby reported that "[locating] a desired point or area [is] much easier when a real object is sitting on the Polhemus's digitizing surface."

1.2: *Relative gesture vs. absolute gesture*

In Galyean's 3D sculpting interface [29], the user deforms a 3D model by positioning a single tracker in an absolute, fixed volume in front of a monitor. This leads to an interface which is not entirely intuitive. Galyean reports that "controlling the tool position is not easy. Even though the Polhemus pointer is held in a well-defined region, it is often difficult to correlate the position of the pointer in space with the position of the tool on the screen."

Compare this to Sachs's 3-Draw computer-aided design tool [54], which allows the user to hold a stylus in one hand and a palette in the other (both objects are tracked by the computer). These tools serve to draw and view a 3D virtual object which is seen on a desktop monitor. The palette is used to view the object, while motion of the stylus relative to the palette is used to draw and edit the curves making up the object.

1. Ivan Sutherland suggested this distinction between understanding 3D and experiencing 3D in the Fall of 1993. Also, Fred Brooks included this idea in his 1988 review paper where he observes that "3D understanding is difficult" [11].

3-Draw's use of the stylus for editing existing curves and Galyean's use of the "Polhemus pointer" for deforming a sculpture represent nearly identical tasks, yet the authors of 3-Draw do not report the difficulties which Galyean encountered. We attribute this difference to the palette-relative gesture employed by 3-Draw, as opposed to the abstract, absolute-space gesture required by Galyean's sculpting interface. As Sachs notes, "users require far less concentration to manipulate objects relative to each other than if one object were fixed absolutely in space while a single input sensor controlled the other" [54].

Thus, users may have trouble moving in a fixed, absolute coordinate frame. A spatial interface could instead base its interaction techniques upon relative motion, including motion relative to a spatial reference or the user's own body.

We have previously described an interface where users can manipulate virtual objects by moving real-world tools or "props" [35] which correspond to the virtual objects, and thus serve as spatial references. Based on our informal observations of test users at various stages of the design, using any spatial reference is better than none. Even an abstract object, such as 3-Draw's palette, a rubber ball, or the user's other hand, can serve as a source for relative gesture. If the spatial reference corresponds closely to the virtual object, the users's tactile and kinesthetic feedback reinforce the visual illusion, but such correspondence is desirable, rather than strictly necessary.

1.3: Two-handed interaction

Two-handed input has often been viewed as a technique to improve the efficiency of human-computer interaction, by enabling the user to perform two sub-tasks in parallel [15], rather than as sequentially selected modes. When interacting in three dimensions, we find that using two hands not only improves efficiency, but can also help to make spatial input comprehensible to the user. For example, during informal user observations of a virtual reality interface, we have noted that users of two-handed interaction are less likely to become disoriented versus users who interact with only one hand [50].

Enabling the use of both hands can allow users to ground themselves in the interaction space; in essence the user's own body becomes a spatial reference. Regarding two-handed interaction in free space, Sachs observes that "the simultaneous use of two [spatial input] sensors takes advantage of people's innate ability--knowing precisely where their hands are relative to each other" [54]. Our informal observation of several hundred test users of a two-handed spatial interface for neurosurgical visualization [35] strengthens and reaffirms Sachs's observation: we find that most test users can operate the two-handed interface effectively within their first minute of use. This also reinforces findings by Buxton [15] and Kabbash [39] that users can transfer everyday skills for manipulating tools with two hands to the operation of a computer, with little or no training.

Even when manipulating just a single object in 3D, using two hands can be useful and natural. In a classic wizard-of-oz experiment, Hauptmann [33] observed test subjects spontaneously using two hands for single-object translation, rota-

tion, and scaling tasks. Using two hands can also offer other practical advantages: it is often easier to grasp and rotate a spatial input device with two hands, and fatigue may be reduced since the hands can provide mutual physical support.

Guiard's analysis of human skilled bimanual action [32] provides an insightful theoretical framework for hypothesizing which classes of two-handed interfaces might improve performance without inducing additional cognitive load. Guiard has proposed the following principles based on his observations of right-handed subjects:

- "Motion of the right hand typically finds its spatial references in the results of motion of the left hand." For example, when writing, the left hand controls the position and orientation of the page, while the right hand performs the actual writing by moving the pen relative to the left hand.
- The right and left hands are involved in asymmetric temporal-spatial scales of motion. In the writing task, for example, the movements of the left hand adjusting the page are low in temporal and spatial frequency compared to the high-frequency, detailed work done by the right hand.
- "The contribution of the left hand to global manual performance starts earlier than that of the right." The left hand first positions the paper, then the right hand begins to write.

We note, however, that Guiard's principles have not been formally demonstrated, and may also represent an incomplete set of conditions for usable two-handed interfaces. For example, Kabbash [39] describes a two-handed interface (the "palette menu") where the user moves an opaque menu using a trackball in the left hand and a selection cursor using a mouse in the right hand. Although this interface apparently conforms to Guiard's principles, Kabbash's results suggest that the palette menu interface may induce a cognitive load.

1.4: Multisensory feedback

A key challenging facing spatial interaction is identifying aspects of the proprioceptive senses that we can take advantage of when interacting in real space. Interacting with imaginary, computer-generated worlds can easily bewilder users; presumably, providing a wide range of sensory feedback might help the user to more readily perceive their virtual environment. Psychologist J. J. Gibson has long argued that information from a variety of feedback channels is crucial to our understanding of space [30].

Brooks [11] discusses interfaces which employ multisensory feedback techniques, including force feedback [12][36][46], space exclusion (collision detection), and supporting auditory feedback. To these techniques we add *physical manipulation of tools with mass*.

For example, we have experimented with a virtual reality interface for positioning a virtual flashlight using a glove, which users can use to grab and position the virtual flashlight. However, during public demo sessions, we found that users have inordinate difficulty grasping and manipulating the virtual flashlight using the glove. By replacing the glove with a tracked *physical* flashlight, we found that users could position the virtual flashlight with ease. For this application,

physical manipulation of a flashlight worked well, while glove-based manipulation of a virtual flashlight was a disaster.

We see several factors which can contribute to ease-of-use for the physical manipulation paradigm:

- When utilizing glove-based input, the user must rotate the entire hand to indicate the rotation of a virtual object. But as Liang notes, “the hand has certain kinematic constraints. For example, it is far more easy to rotate something held by the fingers than to rotate the whole hand itself” [43].
- The mass of the tool can damp instabilities in the user’s hand motion. For example, surgeons are very particular about the *weight* of their surgical instruments, as the proper heaviness can help decrease the amplitude of small, involuntary hand tremors.
- A physical tool provides kinesthetic feedback, due to the tool’s inertia and the force of gravity.
- The physical properties of a tool suggest its use and constrain how the user can manipulate it. For example, a screwdriver affords rotation about its vertical axis while a wrench affords rotation about a horizontal axis. This type of haptic feedback would not be possible if the rotational constraints were purely visual, as is the case with graphical 3D widgets [20].

1.5: Physical constraints and affordances

Physical constraints and affordances are widely used in industrial design (Norman [48] provides many examples) and we believe spatial interfaces can take advantage of these physical properties of objects. Software constraints are often useful, but they do have limitations: the user must understand the constraints and their feedback, which may impose a small cognitive load. Using physical constraints can remove this cognitive load and also lends support: users can *try* configurations of objects by moving their hands until they hit something.

For example, Schmandt describes an interface for entering multiple layers of VLSI circuit design data in a 3D stereoscopic work space [55]. The user enters the data by pressing a stylus on a stationary 2D tablet; the user can adjust the depth of the image so that the desired plane-of-depth lines up with the 2D tablet. Versions of the interface which constrained the 3D stylus position to lie on grid points via software mapping were less successful; the physical support of the tablet proved essential.

Other useful 2D constraining surfaces include the physical surface of the user’s desk, the glass surface of the user’s monitor, or even a hand-held palette or clipboard.

For example, we use a clipboard (held in the non-dominant hand) and a stylus (held in the dominant hand) in a virtual reality application which allows the user to edit the architectural layout of the room they are standing in [59]. The stylus is used to edit a miniature model of the room, which is seen on the virtual counterpart of the real-world clipboard. The clipboard provides a convenient work surface which can be moved out of the way when it is necessary to view the larger context, and also provides an effective metaphor for action-at-a-distance: the user can, for example, move an object on the opposite side of the room by moving its representation

on the virtual clipboard. Based on our informal observations of users of this interface, we find that using a combination of physical and software constraints works well.

1.6: Head tracking techniques

In a non-immersive spatial interface, desktop-based head tracking can allow the interface to “give back” some of the information lost by displaying 3D objects on a flat display, via head motion parallax depth cues. We merely note head tracking as a technique for spatial feedback; previous research [45][22][66][43] discusses the advantages of head tracking and the implementation details. An additional user study [51] shows performance improvement for a generic search task using an immersive head-tracked, head-mounted display vs. a non-head-tracked display.

2: *User perception of multidimensional tasks: related vs. independent input dimensions*

The Jacob / Sibert study [37] compares user performance for two tasks: the first asks the user to match the (x, y, size) of two squares, while the second task requires matching the (x, y, greyscale) of two squares. Both tasks require the control of three input dimensions, but Jacob reports that user task performance time for the (x, y, size) task is best with a 3D position tracker, while performance for the (x, y, greyscale) task is best with a mouse (using an explicit mode to change just the greyscale).

Jacob argues that the 3D tracker works best for the (x, y, size) task since the user thinks of these as related quantities (“integral attributes”), whereas the mouse is best for the (x, y, greyscale) task because the user perceives (x, y) and (greyscale) as independent quantities (“separable attributes”). The underlying design principle, in Jacob’s terminology, is that “the structure of the perceptual space of an interaction task should mirror that of the control space of its input device” [37].

This result points away from the standard notion of logical input devices. It may not be enough for the designer to know that a logical task requires the control of three input parameters (u, v, w). The designer should also know if the intended users perceive u, v , and w as related or independent quantities. In general it may not be obvious or easy to determine exactly how the user perceives a given set of input dimensions.

2.1: Extraneous degrees of freedom

Most spatial input devices return six dimensions of input data, but this does not mean that all six dimensions should be used at all times. If, for example, the user’s task consists only of orienting an object, it makes little sense to allow simultaneous translation, since this only makes the user’s task more difficult: the user must simultaneously orient the object and keep it from moving beyond their field of view. Extraneous input dimensions should be constrained to some meaningful value.

In general, it makes good common sense to exploit task-specific needs to reduce dimensionality. For example, the mouse-based interactive shadows technique [34] allows constrained movement in 2D planes within a 3D scene. If the user’s task consists only of such constrained 2D move-

ments, this may result in a better interface than free-space 3D positioning. Presumably this general strategy can scale to the use of spatial input devices.

3: *Control metaphors*

Ware [65] identifies three basic control metaphors for 3D interaction:

- *Eyeball-in-hand metaphor (camera metaphor)*: The view the user sees is controlled by direct (hand-guided) manipulation of a virtual camera. Brooks has found this metaphor to be useful when used in conjunction with an overview map of the scene [10][11].
- *Scene-in-hand metaphor*: The user has an external view of an object, and manipulates the object directly via hand motion. Ware suggests this metaphor is good for manipulating closed objects, but not for moving through the interior of an object [65].
- *Flying vehicle control (flying metaphor)*: The user flies a vehicle to navigate through the scene. Ware found flying to be good for navigating through an interior, but poor for moving around a closed object [65]. Special cases of flying include the “car driving metaphor,” as well as the “locomotion metaphor,” which requires the user to physically walk through the scene [10].

We add a fourth metaphor:

- *Ray casting metaphor*: The user indicates a target by casting a ray or cone into the 3D scene. The metaphor can be used for object selection [43] as well as navigation [44]. It is not yet clear under which specific circumstances ray casting may prove useful.

The selection of an appropriate control metaphor is very important: the user’s ability to perform 3D tasks intuitively, or to perform certain 3D tasks at all, can depend heavily on the types of manipulation which the control metaphor affords. Brooks addresses this issue under the heading “metaphor matters” [11].

4: *Issues in dynamic target acquisition*

The term *dynamic target acquisition* refers to target selection tasks such as 3D point selection, object translation, object selection, and docking. As previously suggested, specifying a target based on the absolute (x, y, z) position of the tracker can be a fatiguing, consciously calculated interaction. Instead targeting can be based upon relative motion; options include movement of the user’s hand relative to the user’s body, relative to the user’s other hand, relative to a real object, or relative to the starting point of the gesture.

We present several issues related to dynamic target acquisition tasks:

- Use of transparency to facilitate target acquisition
- Ray casting vs. direct positioning in 3D
- Cone casting vs. ray casting

The first two issues suggest general strategies, while the second two issues address 3D point selection and 3D object selection, respectively.

4.1: *Use of transparency to facilitate target acquisition*

Transparency is a good general technique to aid in dynamic target acquisition tasks for two reasons:

- *Occlusion cues*: Placing a semi-transparent surface in a 3D scene provides occlusion cues. The user can easily perceive which objects are in front of, behind, or intersected by a transparent surface.
- *Context*: Since the surface is semi-transparent, objects behind it are not completely obscured from view. This allows the user to maintain context as the transparent surface is manipulated.

Zhai [68] describes the use of a semi-transparent volume, known as the “silk cursor,” for dynamic target acquisition. Zhai’s experimental results suggest that for the 3D dynamic target acquisition task which he studies, transparency alone leads to greater performance improvements than stereopsis alone. Zhai’s work is the first we know of to generalize the benefits of transparent volumes for target acquisition tasks.

Other example uses of transparency to aid target acquisition include use of a 3D cone for object selection [43], use of a semi-transparent plane for selecting cross-sections of a polygonal brain [35], and use of a semi-transparent tool sheet in the Toolglass interface [7].

4.2: *Ray casting vs. direct positioning in 3D*

Perhaps the most obvious way to implement point selection is to base it on the (x, y, z) position of the tracker, but in many circumstances 3D ray casting may be a superior strategy for selecting 3D points. Instead of directly specifying the 3D point, the spatial input device is used to shoot a ray into the scene, allowing the user to hold the input device in a comfortable position and rotate it to change the ray direction [43].

The 3D points selectable by casting a ray are constrained to lie on the surface of virtual objects in the scene. In many circumstances this is exactly what is desired. If it is necessary to select points on objects which are inside of or behind other objects in the scene, the ray casting can be augmented with a mechanism for cycling through the set of all ray-object intersection points.

For disconnected 3D points, 3D snap-dragging techniques [6] can be used if the disconnected points are related to existing objects in the scene. If the disconnected points are on the interior of objects, ray casting can be combined with a “cutting plane” operator, which is used to expose the interior of the objects [35][43].

Digitizing points on the surface of a real object is an instance where ray casting may not be helpful. In this case, the real object provides a spatial reference for the user as well as physical support of the hand; as a result, direct 3D point selection works well [49].

4.3: *Cone casting vs. ray casting*

For gross object selection, ray casting may become less appropriate, especially if the object may be distant. One could alternatively use a translucent 3D cone to indicate a region of interest; distance metrics can be used to choose the closest object within the cone. Note that “spotlighting” visual effects afforded by many graphics workstations can provide real-time feedback for this task.

We base this strategy on the implementation reported by Liang [43]. It is not presently clear if other strategies, such

as using ray casting to sweep out a cone, might provide better results in some cases.

5: *Recalibration mechanisms*

At a low level, all spatial input devices provide the software with an absolute position in a global coordinate frame. The user interface should provide a *recalibration mechanism* for mapping this *absolute* position to a new *logical* position, which allows the user to specify a comfortable resting position in the real world as a center point for the interaction space. We are aware of three basic recalibration strategies:

Command-based: The user explicitly triggers a recalibration command, sometimes referred to as a “centering command” or a “homing command.” JDCAD, for example, uses this strategy [43] to bring the 3D cursor to the center of the visible volume.

Ratcheting: Many spatial interfaces (e.g. [18], [64]) utilize the notion of *ratcheting*, which allows the user to perform movements in a series of grab-release cycles. (The user presses a clutch button, moves the input device, releases the clutch button, returns his or her hand to a comfortable position, and repeats the process).

Continuous: In some cases recalibration can be made invisible to the user. For example, in a virtual reality system, when the user moves his body or head, the local coordinate system is automatically updated to keep their motions body-centric. Another example is provided by our desk-top system [35], where a tool held in the non-dominant hand is used to define a dynamic frame-of-reference relative to which other tools may be moved with the dominant hand. Based on informal observations of several hundred test users, we find that the technique is natural and intuitive.

These strategies can be composed. In a virtual reality application, for instance, the position of the hands will be continuously recalibrated to the current position of the head, but an object in the virtual environment might be moved about via ratcheting, or brought to the center of the user’s field of view by a homing command.

Ergonomics and Facility of Interaction

6: *Multiple degree-of-freedom input in coarse positioning tasks vs. precise positioning tasks*

In two dimensions, the direct manipulation paradigm allows rapid, imprecise object placement. But to perform useful work in the context of a complex application such as a document editor, direct manipulation often needs to be constrained by techniques such as gridding or snap-dragging [5]. Corresponding three-dimensional constraint techniques and feedback mechanisms need to be developed for use in spatial interfaces.

Users may have difficulty controlling an interface which requires simultaneous, precise control of an object’s position and orientation. The biomechanical constraints of the hands and arms prevent translations from being independent of rotations, so rotation will be accompanied by inadvertent translation, and vice versa. Even in the real world, we typically break down 6DoF tasks, such as docking, into two sub-

tasks: translating to the location and then matching orientations [12].

The design hurdle is this: provide an interface which effectively integrates rapid, imprecise, multiple degree-of-freedom object placement with slower, but more precise object placement, while providing feedback that makes it all comprehensible. As Stu Card has commented, a major challenge of the post-WIMP interface is to find and characterize appropriate mappings from high degree-of-freedom input devices to high degree-of-freedom input tasks.

Applications such as 3-Draw [54] and abstractions such as Gleicher’s snap-together math [31] make good initial progress toward providing constrained input in 3D, but we believe the general “spatial input constraint problem,” and the issue of providing appropriate feedback in particular, is still a challenging area for future research.

7: *Dynamics and size of the working volume of the user’s hands*

Guiard’s observations of subjects performing writing tasks [32] as well as observations of users of our two-handed interface [35] suggest that people tend to move their hands in a surprisingly small working volume. This volume is not only small, but also tends to move over time as the user changes body posture.

Guiard’s analysis of handwriting tasks suggests that the writer tends to define an active volume relative to his or her non-dominant hand. Guiard also reports that “the writing speed of adults is reduced by some 20% when instructions prevent the nonpreferred hand from manipulating the page” [32].

This suggests that users of a spatial interface which requires movements relative to a fixed frame-of-reference in their environment may experience reduced task performance due to cognitive load, fatigue, or both. This also reinforces the possible importance of using relative gesture (*section 1.2*) and providing recalibration mechanisms (*section 5*).

8: *Use of mice and keyboards in combination with free-space input devices*

It can be awkward and fatiguing to repeatedly switch between spatial input devices and traditional input devices such as mice and keyboards. Keyboards are especially problematic because they can get in the user’s way. We have noted that users frequently rest their hands on the desk-top while manipulating spatial interface tools [35]; if the keyboard is present, it frequently entangles the cabling for the trackers or otherwise gets in the way.

Alternatives include:

- *Voice input:* Mouse-activated commands and keyboard hotkeys can be replaced by voice commands.
- *Touchscreen:* A touchscreen could be also used for command selection, but might furthermore allow the user to perform 2D direct manipulation tasks [56]. Note the facility with which a touchscreen can be utilized: users can touch the screen directly with their spatial input devices, instead of putting them down to use a mouse. This remains an untested idea, but we have observed neurosurgeons spontaneously reaching out to touch the

screen during discussions of our neurosurgical visualization interface [35], suggesting that surgeons will find touching the screen with spatial interface tools to be natural.

The general issue of constructing *hybrid interfaces* which combine 2D and 3D interaction in a unified framework (both in terms of user interaction, and from the standpoint of support provided by user interface toolkits [50]) remains largely unexplored. Feiner's integration of a 3D augmented reality head-mounted display with a standard 2D desktop display [25] offers one of the few examples of which we are aware.

9: *Clutching mechanisms*

Most spatial interfaces incorporate some type of *clutching mechanism*, that is, a software mode which allows the spatial input device to be moved without affecting the 3D cursor. In our experience, some of the most confounding (for the user) and hard-to-fix (for the implementor) usability problems and ergonomic difficulties can arise due to poor clutch design.

For example, we have seen users struggle with many different clutch designs in our two-handed spatial interface [35]. In versions of the interface which used more than one clutch (one clutch was provided for each tool), users could operate the interface easily once the operation of the clutches was explained to them, but most users could not infer the operation of the clutches without any instruction. In versions of the interface which used an ill-placed or hard-to-press clutch button, users became fatigued in as little as five minutes of use. A clutch based on voice input also did not seem to work very well. Based on this experience, we suggest that a poor clutching interface can jeopardize the usefulness of spatial input.

As an example clutching mechanism, the University of North Carolina has constructed an input device which consists of a 3D tracker encased in a pool ball, which has a clutch button mounted on its surface [18]. When the user holds the clutch button down, the virtual object follows movements of the pool ball, and when the button is released, movement of the pool ball has no effect.

When a clutch button is mounted at a fixed location on a spatial input device, the user must have a fixed grip on the input device, to keep their fingers in a position to press the clutch button. Due to the kinematic constraints of the wrist, a fixed grip limits the possible rotations which can be performed. If arbitrary, large-angle rotations are required, the resulting interface can be very awkward. In such cases the clutch button should be separated from the input device. For example, one interface which requires arbitrary rotations uses a foot pedal as a clutch [35], allowing the associated spatial input device to be rotated with ease.

If the user's task seldom requires arbitrary rotation, it is preferable to mount the clutch button directly on the input device. Such a button, unlike the foot pedal, is visibly connected to the input device it controls, and its operation is therefore self-revealing.

Another alternative is to have no clutch button at all. If the interface provides a mechanism to take a snapshot of the

screen, in some cases the need for clutching might be eliminated altogether.

10: *Importance of ergonomic details in spatial interfaces*

Manipulating input devices in free space can easily fatigue the user. The designer of a spatial interface must take special pains to avoid or reduce fatigue wherever possible. A poor design risks degraded user performance, user dissatisfaction, and possibly even injury to the user. An exhaustive list of human factors requirements is beyond the scope of this paper, but we can make a few suggestions:

- Users should be able to move around and shift their body posture. The interface should not require the spatial input devices to be held within a fixed volume that cannot easily be adjusted by the user. Use of recalibration mechanisms is one way to address this problem.
- For desk top configurations, provide an adjustable height chair with arm rests. Also, using a **C** or **L** shaped desk can provide additional surface area to rest the arms.
- If the interface is designed well, fatigue should only be associated with prolonged, uninterrupted use. It may be useful to build time-outs into the system which remind the user to take an occasional break.

Also note that, based on our user observations, the posture of users' hands while manipulating spatial interface tools [35] is not the same as the hand posture required during typing. The palms face each other (instead of facing downward) and users usually either rest the sides of their palms on the desk top, or they alternatively support their forearms at the elbows using chair arm rests, and hold their hands in the air above the desk top. This suggests that the ergonomics requirements for spatial manipulation may be different than those for typing.

SUMMARY AND FUTURE WORK

This paper represents a first attempt to extract design issues from a large body of work. We have identified common themes in what has worked well for spatial input, and what has not. The issues we have presented are not formally proven principles of design. They are meant to serve as a guide to designers who are getting started in spatial input, and should not be expected to serve as a substitute for user testing of any spatial interface based upon strategies we have suggested.

Situations where the issues and strategies we have discussed work well, or where they do not work well, need to be better defined and characterized, and ultimately subjected to formal study. In contemplating formal studies of some of the observations herein, we have been struck by the apparent interdependency of it all: it is extremely difficult to devise experiments which will give insight into one specific phenomenon, without the results being confounded by other effects. Nonetheless, we welcome suggestions for formal experiments.

Multidimensional input is still a hard, unsolved problem, so we cannot hope that the present attempt to distill design issues will address every important issue; we are still learning something new every day. But we believe this paper is at least a good start, and we hope that in the future other researchers will be able to formulate more precise principles

of design which will augment or supercede the preliminary results presented here.

Writing this paper has led us to ask many questions which we are currently unable to address, but they should form an agenda for possible future research:

- Which design issues apply to both spatial interfaces on the desktop and virtual reality or augmented reality interfaces using head-mounted, head-tracked displays? Which do not? Why?
- What ideas, metaphors, or working patterns might 3D computer interfaces adopt from people who must understand spatial relationships to perform their real-world tasks, such as sculptors, surgeons, radiologists, masons, pilots, furniture movers, architects, or molecular chemists? What spatial reasoning skills, or classes of skills, can be found across this wide range of expertise? Is spatial interaction inherently prone to be very task- and user-specific?

ACKNOWLEDGEMENTS

We wish to thank the Department of Neurosurgery for their support. We thank David Kurlander and Shumin Zhai for providing detailed comments about the paper. We also thank the reviewers and the UIST'94 program committee for pointing out several problems with the original draft of this paper. The final product is much improved as a result of their feedback.

REFERENCES

An expanded and annotated version of the following list of references is available on-line through mosaic at address <http://uvacs.cs.virginia.edu/~kph2q/>. The document *Spatial.bib* is also available via anonymous ftp from uvacs.cs.virginia.edu (128.143.8.100) in the *pub/kph2q/* directory. Submissions of new material or corrections to the bibliography are encouraged and may be mailed to kph2q@virginia.edu. Please make the subject line read "*Spatial.bib contribution.*"

1. Adelstein, B., Johnston, E., Ellis, S., "A Testbed for Characterizing Dynamic Response of Virtual Environment Spatial Sensors," UIST'92, 15-22.
2. Badler, N., Manoochehri, K., Baraff, D. "Multi-Dimensional Input Techniques and Articulated Figure Positioning by Multiple Constraints," ACM Workshop on Interactive 3D Graphics, 1986, pp. 151-170.
3. T. Baudel, M. Beaudouin-Lafon, "Charade: Remote Control of Objects Using Hand Gestures," Communications of the ACM, 36 (7), 1993, 28-35.
4. Bergman, L., Fuchs, H., Grant, E., "Image Rendering by Adaptive Refinement," Computer Graphics, 20 (4), 1986, pp. 29-37.
5. Bier, E. A., Stone, M. C., "Snap-Dragging," Computer Graphics, 20 (4), 1986, pp. 233-240.
6. Bier, E. A., "Snap-Dragging In Three Dimensions," Proc. 1990 Symposium on Interactive 3D Graphics, Computer Graphics, 24 (2), pp. 193-204.

7. Bier, E., Stone, M., Pier, K., Buxton, W., DeRose, T., "Toolglass and Magic Lenses: The See-Through Interface," SIGGRAPH '93, pp. 73-80.
8. Bolt, R., "Put-That-There: Voice and Gesture at the Graphics Interface," SIGGRAPH '80, 262-70.
9. Bolt, R. A., Herranz, E., "Two-Handed Gesture in Multi-Modal Natural Dialog," UIST '92, pp. 7-13.
10. Brooks, F. P. Jr., "Walkthrough--a Dynamic Graphics System for Simulating Virtual Buildings," Proc. ACM Workshop on Interactive 3D Graphics, 1986, pp. 9-21.
11. Brooks, F., "Grasping Reality Through Illusion: Interactive Graphics Serving Science," CHI'88.
12. Brooks, F., Ouh-Young, M., Batter, J., Kilpatrick, P., "Project GROPE--Haptic Displays for Scientific Visualization," Comp. Graph. 24 (4), 1990.
13. Bryson, S., Levit, C., "The Virtual Wind Tunnel," IEEE CG&A, July 1992, pp. 25-34.
14. Butterworth, J., Davidson, A., Hench, S., Olano, T. M., "3DM: A Three Dimensional Modeler Using a Head-mounted Display," Proc. 1992 Symp. on Interactive 3D Graphics, pp. 135-138.
15. Buxton, W., Myers, B., "A Study in Two-Handed Input," CHI'86, pp. 321-326.
16. Card, S., Mackinlay, J., Robertson, G., "The Design Space of Input Devices," CHI'89, 117-124.
17. Card, S., Robertson, G., Mackinlay, J., "The Information Visualizer, an Information Workspace," CHI'91, pp. 181-187.
18. Chung, J. C., "A comparison of Head-tracked and Non-head-tracked Steering Modes in the Targeting of Radiotherapy Treatment Beams," Proc. 1992 Symp. on Interactive 3D Graphics, 193-196.
19. Cohen, P., Sullivan, J., "Synergistic Use of Direct Manipulation and Natural Language," CHI'89, pp. 227-233.
20. Conner, D., Snibbe, S., Herndon, K., Robbins, D., Zeleznik, R., van Dam, A., "Three-Dimensional Widgets," 1992 Symp. on Int. 3D Graph, 183-188.
21. Cruz-Neira, C., Sandin, D., DeFanti, T., "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE," SIGGRAPH '93, pp. 135-142.
22. M. Deering, "High Resolution Virtual Reality," Computer Graphics, 26 (2), pp. 195-202.
23. Feiner, S., MacIntyre, B., Haupt, M., Solomon, E., "Windows on the World: 2D Windows for 3D Augmented Reality," UIST'93, pp. 145-155.
24. Feiner, S., Macintyre, B., Seligmann, D., "Knowledge-Based Augmented Reality," Comm. of the ACM, 36 (7), 1993, pp. 53-61.

25. Feiner, S., Shamash, A., "Hybrid User Interfaces: Breeding Virtually Bigger Interfaces for Physically Smaller Computers," UIST '91, pp. 9-17.
26. S.S. Fisher, M. McGreevy, J. Humphries, W. Robinett, "Virtual Interface Environment for Telepresence Applications," Oct. 1988, Proc. Human Factors Society 32nd Annual Meeting.
27. Fitzmaurice, G. W., "Situated Information Spaces and Spatially Aware Palmtop Computers," Comm. of the ACM, 36 (7), 1993, pp. 39-49.
28. Foley, J. D., Wallace, V., Chan, P., "The Human Factors of Computer Graphics Interaction Techniques," IEEE CG&A, Nov. 1984, pp. 13-48.
29. Galyean, T. A., Hughes, J. F., "Sculpting: An Interactive Volumetric Modeling Technique," Computer Graphics, 25 (4), pp. 267-274.
30. Gibson, J., The Ecological Approach to Visual Perception. Lawrence Erlbaum, Hillsdale, NJ.
31. Gleicher, M., "Supporting Numerical Computations in Interactive Contexts," Graphics Interface '93.
32. Guiard, Y., "Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model," The Journal of Motor Behavior, 19 (4), 1987, pp. 486-517.
33. Hauptmann, A. G., "Speech and Gestures for Graphic Image Manipulation," CHI'89, 241-245.
34. Herndon, K., Zeleznik, R., Robbins, D., Conner, B., Snibbe, S., van Dam, A., "Interactive Shadows," UIST '92, pp. 1-6.
35. Hinckley, K., Pausch, R., Goble, J., Kassell, N., "Passive Real-World Interface Props for Neurosurgical Visualization," CHI'94, 452-458.
36. Iwata, H., "Artificial Reality with Force-feedback: Development of Desktop Virtual Space with Compact Master Manipulator," Computer Graphics, 24 (4), pp. 165-170.
37. Jacob, R., Sibert, L., "The Perceptual Structure of Multidimensional Input Device Selection," CHI'92, pp. 211-218.
38. Kabbash, P., MacKenzie, I. S., Buxton, W., "Human Performance Using Computer Input Devices in the Preferred and Non-Preferred Hands," INTERCHI'93, pp. 474-481.
39. Kabbash, P., Buxton, W., Sellen, A., "Two-Handed Input in a Compound Task," CHI'94, pp. 417-423.
40. Kaufman, A., Yagel, R., "Tools for Interaction in Three Dimensions," Proc. 3rd International Conf. on HCI (Boston, MA), Vol. 1, Sept. 1989, pp. 468-475.
41. Krueger, M., "Environmental Technology: Making the Real World Virtual," Communications of the ACM, 36 (7), 1993, pp. 36-37.
42. Liang, J., Shaw, C., Green, M., "On Temporal-Spatial Realism in the Virtual Reality Environment," UIST'91, pp. 19-25.
43. Liang, J., Green, M., "JDCAD: A Highly Interactive 3D Modeling System," 3rd International Conference on CAD and Computer Graphics, Beijing, China, Aug. 1993, 217-222.
44. Mackinlay, J., Card, S., Robertson, G., "Rapid Controlled Movement Through a Virtual 3D Workspace," Comp. Grap., 24 (4), 1990, 171-176.
45. McKenna, M., "Interactive Viewpoint Control and Three-dimensional Operations," Proc. 1992 Symposium on Interactive 3D Graphics, pp. 53-56.
46. Minsky, M., Ouh-young, M., Brooks, F. P., Behensky, M., "Feeling and Seeing: Issues in Force Display," Comp. Graph., 24 (2), 234-244.
47. Nielsen, J., "Noncommand User Interfaces," Communications of the ACM, 36 (4), pp. 83-99.
48. Norman D., The Design of Everyday Things. Doubleday: New York, New York, 1990.
49. Ostby, E., "Describing Free-Form 3D Surfaces for Animation," Proc. ACM Workshop on Interactive 3D Graphics, Oct. 1986, pp. 251-258.
50. Pausch, R., "Support for Rapid Prototyping of Two- and Three-Dimensional User Interfaces," Proposal for ARPA BAA 93-42. Comp. Science Department, University of Virginia, March, 1994.
51. Pausch, R., Shackelford, M. A., Proffitt, D., "A User Study Comparing Head-Mounted and Stationary Displays," Proc. IEEE Symposium on Research Frontiers in Virtual Reality, Oct. 1993.
52. Pixsys Inc., 3522 22nd St., Boulder, CO 80304. (303) 443-0771.
53. Polhemus Navigation Sciences, Inc., P. O. Box 560, Colchester, VT 05446. (802) 655-3159.
54. Sachs, E., Roberts, A., Stoops, D., "3-Draw: A Tool for Designing 3D Shapes," IEEE Computer Graphics & Applications, Nov. 1991, pp. 18-26.
55. Schmandt, C. M., "Spatial Input/Display Correspondence in a Stereoscopic Computer Graphic Work Station," Computer Graphics, 17 (3), 1983, pp. 253-262.
56. Sears, A., Plaisant, C., Shneiderman, B., "A New Era for High Precision Touchscreens," in Advances in Human-Computer Interaction, Hartson, Hix, eds., Vol. 3, 1992, pp. 1-33.
57. Shepard, R. N., Metzler, J., "Mental Rotation of Three-Dimensional Objects," Science, Vol. 171, 1971, pp. 701-703.
58. Spaceball Technologies, Inc. (508) 970-0330.

59. Stoakley, R., Pausch, R., "Virtual Kit of Parts," unpublished manuscript, available through mosaic at <http://uvacs.cs.virginia.edu/~rws2v/plinth.html>.
60. Sturman, D., Zeltzer, D., Pieper, S., "Hands-On Interaction with Virtual Environments," UIST'89, pp. 19-24.
61. I. E. Sutherland, "A Head-mounted Three Dimensional Display," Proc. the Fall Joint Computer Conference, 1968, pp. 757-764.
62. Takemura, H., Tomono, A., Kayashi, Y., "An Evaluation of 3-D Object Pointing Using a Field Sequential Stereoscopic Display," Proc. Graphics Interface '88, June 1988, pp. 112-118.
63. Taylor, R., Robinett, W., Chi, V., Brooks, F., Wright, W., Williams, R., Snyder, E., "The Nanomanipulator: A Virtual-Reality Interface for a Scanning Tunneling Microscope," SIGGRAPH'93, pp. 127-134.
64. Ware, C., "Using Hand Position for Virtual Object Placement," Visual Comp., 6 (5), 1990, 245-253.
65. Ware, C., Osborne, S., "Exploration and Virtual Camera Control in Virtual Three Dimensional Environments," Comp. Graph., 24 (2), 175-183.
66. Ware, C., Arthur, K., Booth, K. S., "Fish Tank Virtual Reality," INTERCHI'93, pp. 37-41.
67. Wellner, P., "Interacting with Paper on the DigitalDesk," Communications of the ACM, 36 (7), 1993, pp. 87-97.
68. Zhai, S., Buxton, W., Milgram, P., "The "Silk Cursor": Investigating Transparency for 3D Target Acquisition," CHI'94, pp. 459-464.
69. Zimmerman, T., Lanier, J., Blanchard, C., Bryson, S., Harvill, Y., "A Hand Gesture Interface Device," CHI+GI'87, pp. 189-192.