

# User Interfaces for Volumetric Displays



**Although still in the prototype stage, three-dimensional volumetric displays present unique interface challenges. The authors use physical mockups to explore solutions to interaction styles for 3D volumetric displays.**

Ravin  
Balakrishnan  
George W.  
Fitzmaurice  
Gordon  
Kurtenbach  
AliasWavefront

The constraints of current display technologies limit our ability to view and interact with 3D computer graphics. Although the data may be inherently 3D, we typically view it on a 2D system incapable of explicitly displaying the depth dimension. To perceive and interpret depth in the real world, humans rely on many visual cues, including perspective, occlusion or interposition, light and shadows, relative size, motion parallax, and stereopsis.<sup>1</sup> A truly rich perceptual experience requires all these cues, yet current 3D displays provide only some of them. For example, a flat 2D display can provide only perspective projection, occlusion, lighting, shading, and limited motion parallax through manual scene navigation.

Various types of head-mounted displays<sup>2</sup> and 3D projection systems<sup>3</sup> attempt to do better by providing stereoscopic views. However, such displays create a basic conflict between those two mechanisms that—combined—give humans stereoscopic vision:

- *convergence*, our eyes' ability to pivot inward toward the object in focus, and
- *accommodation*, the eye's ability to focus its lens on objects in the frontal field of view.

By providing two slightly different images for each eye, stereoscopic displays satisfy convergence, but the single image plane makes accommodation difficult. As a result, some users experience nausea and dizziness.

To overcome these effects, some systems use head tracking, which attempts to provide motion parallax. Plagued by technical difficulties such as lag and accu-

racy, head tracking also limits display use to a single user, which hampers collaboration.

Volumetric displays,<sup>4</sup> which enable true 3D image visualization, hold the promise of enhancing rendered 3D graphics' sense of realism by providing *all* the depth cues humans require.

## VOLUMETRIC-DISPLAY TECHNOLOGIES

Although several competing volumetric-display technologies exist,<sup>4</sup> all such displays generate true 3D volumetric images by actually illuminating points in 3D space rather than merely simulating the perception of viewing such points. This approach lets us use our natural physiological mechanisms for depth perception to gain a richer understanding of the virtual 3D scene. Further, we can view these displays from almost any direction and multiple individuals can view them simultaneously. All of this is achieved without requiring users to wear hardware such as shutter glasses or head-tracking gear.

From the output perspective, all volumetric displays generate true 3D images, but the technology underlying the displays can vary widely. We can classify these technologies into three broad categories:

- *Holographic displays*<sup>5</sup> produce a 3D image by reproducing the diffraction of light from a 3D scene. Microscopic patterns on a physical holographic imaging plane control the diffraction of light, while the spatial frequency of the patterns on the plane varies the light diffraction, creating a true 3D image.
- *Swept-volume techniques* sweep a periodically time-varying 2D image through a 3D spatial volume at a higher frequency than the human eye can

**Figure 1. Felix: a dome-shaped swept-volume display. When the volumetric display sweeps a periodically time-varying 2D image through a 3D spatial volume at a higher frequency than the human eye can resolve, it generates an image that the viewer perceives as a 3D volumetric image.**



resolve. Due to visual persistence, the human viewer perceives a 3D volumetric image. Figure 1 shows an example of a dome-shaped swept-volume display called Felix (<http://www.felix3d.com>).

- *Static-volume techniques*<sup>4</sup> create the volumetric image without any mechanical motion within the viewing volume. These techniques essentially create emissive voxels by direct excitation of points within a physical 3D medium.

Regardless of the underlying display technology, we can envisage using these displays for diverse applications with varying levels of interactivity. On one end of the interactivity spectrum, these displays could work much like a printer, providing an auxiliary device to display 3D images, with all interaction done via standard PC components. At the spectrum's other end, the volumetric display could function as the primary display and interaction platform. This highly interactive use of volumetric displays would likely drive new applications and transform existing ones, which would in turn require development of new user-interface techniques and input styles.

The presence or absence of an enclosure surrounding the volumetric display is a key user interface issue. Holographic volumetric displays suspend the image in midair, with no physical barrier between user and image. Because virtual reality systems are also perceived to have no barrier between image and user, previous research in this field provides guidance for interacting with enclosure-free displays. We thus focus on an area yet to be extensively studied: user interface issues for volumetric displays (VUIs) in which a physical 3D enclosure separates user and image.

Historically, user-interface research has often been reactive in that developers design the interface after the underlying display and computing technologies mature. We advocate a more prescriptive approach, proposing design possibilities for user interfaces that fully leverage these displays' unique capabilities. Achieving this goal requires exploring the design space of interaction techniques and input devices that VUIs will require.

## METHODOLOGY

Proactively proposing user-interface designs for highly interactive volumetric displays poses the challenge of designing for an infant technology in which the final accepted version could vary in scale, shape, display quality, usage context, and other aspects. Rather than constraining our design explorations to existing prototype volumetric displays, we use physical mockups to consider design possibilities for a variety of potentially viable displays and associated input mechanisms. Using high fidelity mockups and props lets us glimpse the future with minimal prototyping effort, before the display and input technologies themselves fully mature.

These physical props let us explore the *affordances* of volumetric displays that differ in size, form, and use. Affordance theory, first enunciated by psychologist J.J. Gibson, refers to the properties of objects with reference to the observer and environment. Understanding these affordances—which is critical in determining the optimal input and interaction styles for each display type—required constructing different physical mockups of volumetric displays.

Each of our mockups consists of a clear-plastic enclosure on a solid rotatable base. By placing a variety of physical objects within the enclosure, we can simulate the display of different data types in the mockup volumetric display. A removable platform on the enclosure's base allows for easy replacement of objects within. We also use physical objects to simulate input devices and user-interface elements.

Our explorations using these mockups generated several observations about fundamental design issues:

- *Physical rotation.* We believe that users will feel compelled to physically rotate the volumetric display to view the content from different perspectives.
- *Display size and shape.* The display enclosure's form factor encourages a variety of interaction styles and elicits different user expectations.
- *Touching the enclosure.* Given the true 3D realism of the graphical objects within the display's enclosure, users will try to interact with objects by using the enclosure as an input surface.
- *Devices on the enclosure.* If an application requires more precise or higher-degree-of-freedom input devices, it's possible to comfortably use these devices on the enclosure rather than away from it. This solution also avoids heavily dividing users' attention between data and input spaces.
- *Volume management.* Given the display's realism and volumetric nature, users will need to manage the space within the display, perhaps



**Figure 2. Volumetric-display mockups: (a) 2-foot diameter dome; (b) 1-foot cubic enclosure; (c) 4-foot large-scale dome. Developers place a variety of physical objects within the clear plastic enclosure to explore the affordances of volumetric displays that differ in size, form, and use.**

through gestures similar to those used to manipulate real physical volumes.

- *Viewpoint-independent user-interface widgets.* Letting users rotate or move around the display requires user-interface widgets accessible from any viewpoint.
- *Applications.* Different application domains may benefit from different display form factors. For example, application classes that require a tight coupling between an object's 3D and planar orthographic views will benefit from a cubic enclosure. Applications that only require perspective viewing, on the other hand, will benefit from edgeless spherical enclosures.
- *Reuse.* Certain technologies—including input devices, interaction techniques, and metaphors—from current 2D interfaces transfer well to volumetric displays and should be reused; others transfer poorly, and must be replaced with innovative solutions.

Throughout this exploration, we explicitly avoid championing a particular design option, preferring to describe a range of choices for further development.

## DISPLAY SCALE AND SHAPE

The design of our volumetric-display mockups, examples of which are shown in Figure 2, was influenced by existing working volumetric displays like the one shown in Figure 1. From a human perspective, we can conveniently classify these mockups into three groups:

- *Hand.* Our smallest display mockup, the 3-inch-diameter sphere, fits comfortably in a user's hand. We envisage that displays of this size would be used mainly for displaying a single object or a larger object's subcomponent.
- *Desk.* We built three mockups for desktop interaction. Figure 2a shows the smallest desktop mockup, a 1-foot-diameter dome. We also built a 2-foot-diameter dome that displays a fairly detailed small object or simple 3D scene. Figure

2b shows the third type of desktop mockup, a 1-foot-cubic enclosure, which has significantly different affordances than spherical enclosures.

Overall, these desk-size displays seem suited to either individual use or small groups. We also found that, while seated, users could only comfortably reach the front half of the display without rotating it. These affordances influence the types of interactions and applications used with the display.

- *Room.* Our largest mockup is a 4-foot-diameter dome. The display's size makes it possible for several people to use it simultaneously. This dome allows display of larger-scale objects, such as consumer products or one-fifth-scale automobiles. At this scale, a user cannot reach completely around the display, therefore, rotating it becomes a critical usability requirement.

Other shapes, such as cylinders, are possible. However, our mockups represent two important shape classes: shapes that are seamless and uniform across their entire surface, such as domes and spheres, and shapes that have clearly demarcated boundaries, such as cubes. Both shape classes have their pros and cons, making the final choice likely to depend on application or technology.

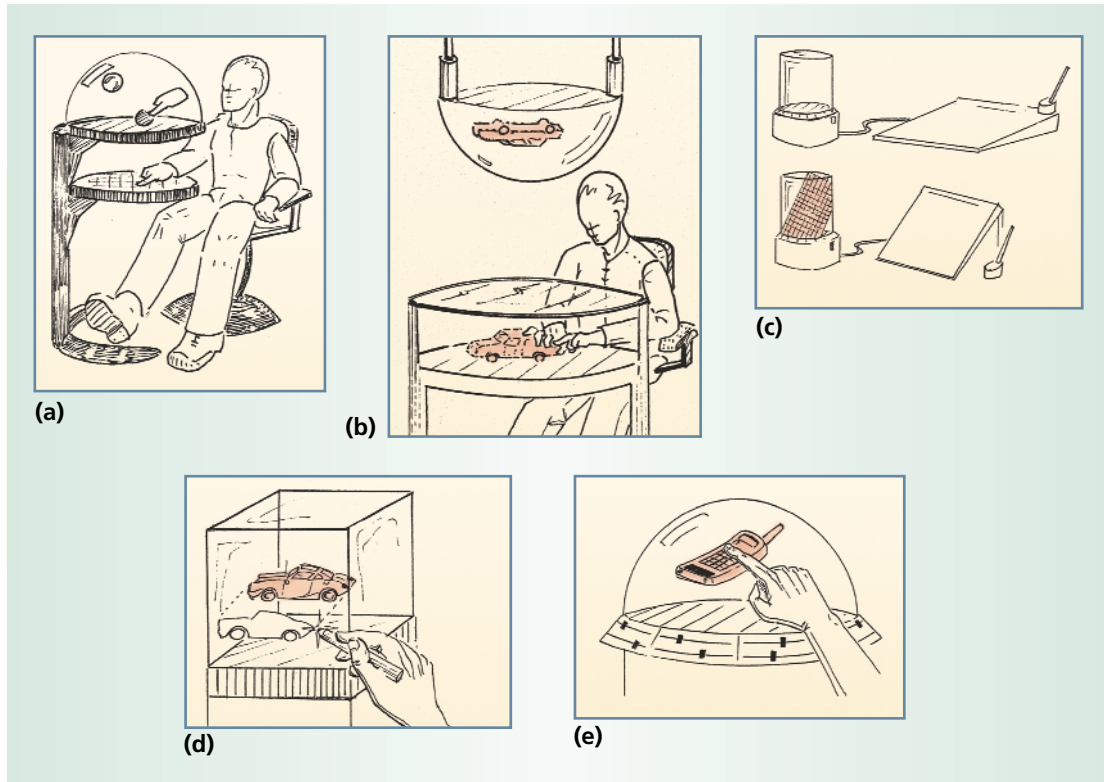
## INPUT ISSUES

Our focus on using volumetric displays both as an output device and a highly interactive display platform requires consideration of fundamental issues for input configurations.

### Shared versus separated spaces

The level of separation between input and output spaces is immediately significant. A shared space makes the input and output spaces the same. For example, pen-based computers, like the Palm Pilot, combine both input and output spaces into one unified space.

Most computer systems, however, have separate input and output display spaces. For example, the keyboard and mouse operate in different spaces from the CRT display.



**Figure 3.** Various input mappings: (a) a separated 3D to 3D mapping lets users work in a volume below the display volume; (b) a perceptually integrated 3D to 3D mapping uses a half-silvered mirror to give the illusion of working directly with 3D data; (c) top—a planar 2D to 3D mapping uses a digitizing tablet to control 3D data via 3D abstract controllers; bottom—a planar 2D to 2D within 3D mapping maps a digitizing tablet to a plane in the 3D scene; (d) a planar 2D to 2D within 3D mapping uses a cubic enclosure's planar surfaces as image and input planes; and (e) a nonplanar 2D to 3D mapping uses a nonplanar enclosure surface as the input device.

Volumetric displays with enclosures, by their very design, do not facilitate complete sharing of input and output spaces. Thus, we focus on designs for physically separated input/output spaces.

### Mapping separated I/O spaces

Using enclosed volumetric displays requires working with different possible mappings of input space to output space. Our mockups revealed four basic mappings.

**3D input space to 3D output space.** Figure 3a shows a mapping of a 3D volumetric input space to a 3D volumetric display space. This method tracks the user's hands—perhaps using cameras and computer vision techniques—in a volume directly below the display volume. A virtual representation of the hands would be superimposed into the 3D output volumetric display.

Figure 3b shows a configuration in which half-silvered mirrors combine the volumetric image with the users' view of their hands so that users perceive their hands to be operating within the display.

Both of these approaches offer the advantage of providing easily understood one-to-one mappings of input space to output space. However, previous work in virtual reality systems indicates that the fatigue and inaccuracy inherent in unsupported freehand manipulations can pose problems.

**Planar 2D input space to 3D output space.** This method maps a planar input space to control some subset of the 3D volumetric display space. Examples include using a 2D digitizing tablet, as Figure 3c shows, or a regular mouse to control aspects of the 3D scene. Although configurations of this type can avoid the problems of freehand manipulation, mismatch of input and output can lead to spatial ambiguity.

**Planar 2D input space to planar 2D space within the 3D output space.** To alleviate problems with 2D input to 3D output mapping, we can imagine mapping a planar input space to another, varying, planar output space, as current computer-aided design (CAD) software does. Figure 3c shows a configuration that maps a digitizing tablet to a plane in the 3D output space.

Figure 3d shows another configuration where the cubic enclosure's planar surface is used as an input plane that maps to the corresponding planar view of the 3D volumetric image.

This approach affords the ability to work directly on the 3D scene's orthographic views. Typically, CAD software users manipulate these views in separate windows. The enclosure, however, organizes the orthographic views spatially to reflect their relationship with the underlying 3D scene being manipulated.

**Nonplanar 2D input space to 3D output space.** In our mockups, we found it effective to use the display's enclosure as the input surface, as Figure 3e shows. However, this style of input—when used with enclosures such as domes, spheres, and cylinders that have nonplanar surfaces—presents the problem of mapping the nonplanar 2D input to the underlying 3D data. Our mockups suggest that casting a ray from a point on the enclosure into the scene provides a possible solution.

### Physical intermediaries

Using physical input devices as intermediaries between user and 3D output facilitates the various input/output mappings. These devices can add numerous degrees of freedom such as tilting, rotation, and pressure and they also affect input style, precision, comfort, and functionality.

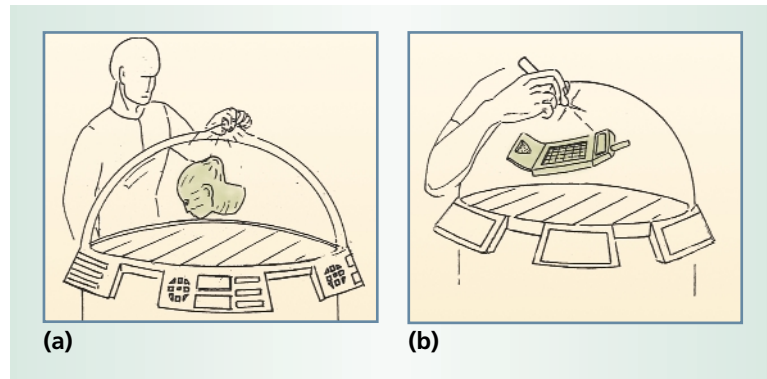
The first input style we explored involved simply using a finger as the input actuator. In addition to pointing with the finger, we could also use pinching, stretching, and pushing gestures, especially for deformation operations. To facilitate these activities, the transparent digitizer that covers the volumetric display could be a deformable membrane that detects multiple pressure points on its surface, as shown in Figure 4a. Research systems such as the Haptic Lens<sup>6</sup> and deformable cube<sup>7</sup> use this input membrane and deformable surface technique, although neither uses a transparent medium.

A digitizing input surface that tracks a variety of input devices is also feasible. Figure 4b shows how a digital stylus could be comfortably used to specify a 2D coordinate. Sensing the stylus tilt information would enhance a ray-casting pointer input model. Another option that worked well was the Rockin'Mouse device.<sup>8</sup> This device combines the familiar buttons and comfort of the traditional mouse with a curved-base design that allows the mouse to sense tilt information.

Other possibilities include applying specialized devices or familiar everyday artifacts with embedded sensors on the volumetric display's digitizing surface. These objects would act as graspable or tangible user interfaces<sup>9</sup> that serve as physical handles to virtual objects, proxy objects, or dedicated controllers.

Input devices can be *parkable* or *nonparkable*. Parkable input devices stay in place when the user releases the device, making it easy for the user to reacquire the device and maintain the cursor's position. Nonparkable devices retain neither their position nor their orientation upon release. Normally parkable devices such as a mouse could lose this property when used on a dome or spherical surface. A mouse that remains parkable when used on a volumetric-display enclosure may thus be desirable.

We also found that large enclosures afford the simultaneous use of multiple devices on the 3D volu-



**Figure 4. Volumetric display physical intermediaries: (a) a deformable input surface that detects pressure and pinching gestures; (b) the stylus.**

metric display. A single user can operate these devices using both hands, or multiple users can use the devices simultaneously. This affordance requires that we develop appropriate two-handed and multiuser interaction techniques.

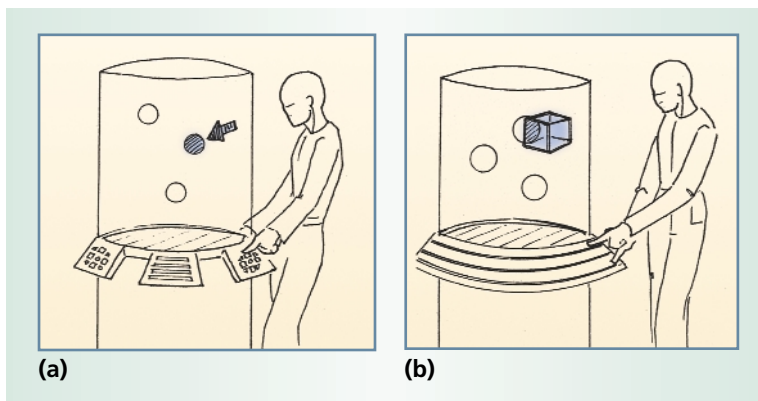
We also experimented with using physical controllers along the perimeter of the display enclosure, as Figure 5 shows. These controllers could consist of buttons, keyboards, sliders, touchpads, mice, and current multi-degree-of-freedom devices such as the SpaceBall.

Using the standard mouse and keyboard as the controllers is also viable. These two familiar and robust devices already offer a rich set of interaction techniques for operating within the 2D and 3D worlds, and we acknowledge that some developers may want to start with this as an initial interface, especially given the high backward compatibility.

Regardless of the choice of physical controller, the need for user convenience and interaction flexibility offers some possible design variations for placement of the input devices with respect to the volumetric display. For example, a single physical control panel could be placed along a rail on the display's perimeter, letting the user drag the panel along while moving around the display. Based on the control panel's orbital placement along the rail, the system can infer the user's approximate location and adjust the input controls and UI accordingly. Alternatively, multiple identical physical control panels could be positioned around the perimeter, as Figure 5a shows. Another option, shown in Figure 5b, is to use touch-based controllers placed in continuous rings around the display.

### Nonhaptic interactions

In addition to haptic input designs, speech or eye-gaze are also potential interactive modalities. Speech input may be valuable for executing commands and mode switching. A head-mounted eye-tracker could help detect areas of interest and adjust rendering qual-



**Figure 5.** Volumetric-display cursors and their controllers, positioned on the display's perimeter: (a) a 3D-point cursor manipulated by button-and-slider control panels repeated around the enclosure; (b) a volume cursor manipulated by touch-sensitive rings. A single user can operate these devices using both hands, or multiple users can use the devices simultaneously.

ity and priority. Volumetric displays, however, do not significantly alter the challenges that developers of speech and eye-tracker technologies currently face.

## CURSORS

Regardless of the choice of input device, the input stream is often used to control a cursor. We have identified three classes of cursors for 3D data.

### 2D point cursor

We could use the status quo 2D cursor in volumetric displays in two ways:

- *Restrict the cursor to moving along the volumetric display's outer surface.* This approach resembles the use of 2D cursors on conventional CRT displays—although its use in volumetric displays would require adapting the ray-casting technique typically used for 3D applications.
- *Constrain the cursor to move along a plane within the 3D scene.* The plane could be specified either via a secondary device or the orientation of the input plane itself, as shown in Figure 3c.

In the first approach, the orientation of the cast ray can take several standard forms. The most obvious choice is to cast the ray perpendicular to the enclosure into the 3D scene. Alternatively, the ray could snap to the nearest object in the 3D scene. For nonplanar enclosures such as domes, casting the ray straight down from the 2D cursor into the 3D scene and ending on the ground plane of the dome may be effective.

The ray itself should probably be displayed as a visible line. This is unnecessary in a traditional CRT display because the user's typically fixed viewpoint means that the ray selects only the first object hit. However, with volumetric displays, instead of looking down along the ray, users may be viewing it from any angle. Thus, the problem is determining which object along the ray's path to select. Solutions include simply selecting the first object hit, the last object hit, or all objects

hit. Another option is using a controller on the input device to move a point along the ray so that it acts as a constrained 3D point cursor.

Instead of manifesting as a line, the ray could be a cone or cylinder. Although existing applications have explored this approach, they may benefit from the enhanced depth perception that volumetric displays afford.

### 3D point cursor

A cursor can also be a small 3D object within a scene, as Figure 5a shows. Although this option offers the advantage of an easily understood metaphor, when used with conventional displays, other objects obscure these cursors. Also, users often struggle to perceive where the cursor resides in the depth dimension. Volumetric displays alleviate this problem somewhat because of their enhanced depth perception and wider field of view. Further, because we expect that volumetric displays will facilitate easy scene rotation, this feature will further increase the efficiency of pointing in 3D.

### 3D volume cursor

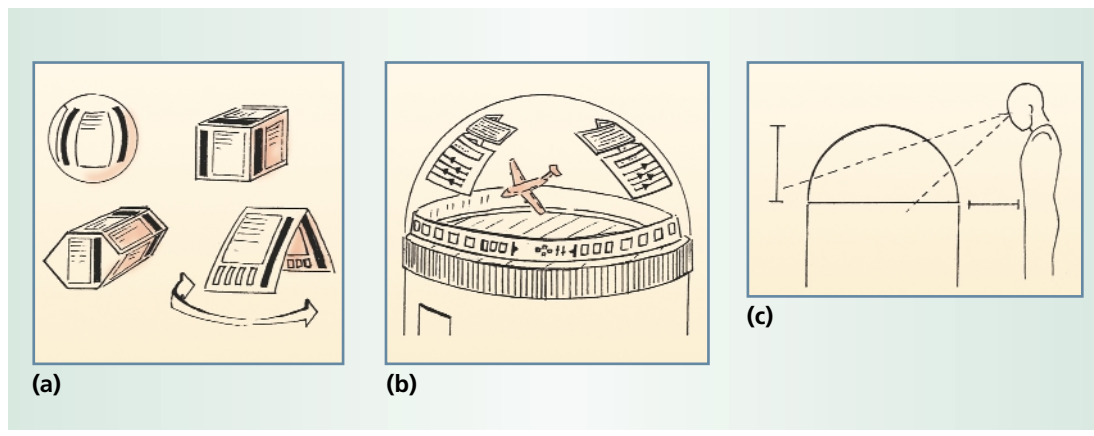
Shumin Zhai and colleagues<sup>10</sup> have demonstrated cursors represented as volumes rather than points, as Figure 5b shows. We can use these cursors' properties advantageously in volumetric displays. First, if the volume cursor is semitransparent, objects behind the cursor remain visible. Second, the volumetric nature of the cursor can enable operations such as selecting multiple objects at once—although selecting specific objects within an object crowd can be challenging. The development of smart cursor shrinking and wrapping algorithms may alleviate this problem.

## WIDGETS

Regardless of type, cursors operate *widgets*, virtual interactive mechanisms that handle a broad range of tasks and interface preferences. Developers have extended the 2D widgets traditionally found in conventional graphical user interfaces to 3D,<sup>11</sup> and these widgets currently represent the status quo for manipulating 3D data projected onto 2D display systems.

Given that a 3D volumetric display can reproduce the common 2D display, we expect that either 2D or 3D standard GUI widgets could function in VUIs without redesign. For example, we could display the common GUI desktop on a 2D plane within a 3D volumetric display. We would then map the input technique to this 2D plane.

Current 3D manipulators for 2D displays will not work flawlessly on volumetric displays without some adjustments, however. In conventional 2D displays, a 3D manipulator assumes a fixed viewpoint while the user operates it. Typically, a user grabs the 3D manipulator handle with the mouse and drags the handle relative to the fixed projection plane. The user's view-



**Figure 6. Volumetric user interface widgets: (a) four omniviewable widgets; (b) widgets located on the enclosure's surface or interior, with some widgets repeated in the display for viewing from nearly any position; (c) diagram of user height and distance from the display, which determines an optimal viewing zone.**

point—and therefore the projection plane—cannot be changed without terminating the drag. This operation style could pose problems with volumetric displays, which have variable viewpoints and lack an explicit projection plane.

One solution may be to define an input plane as the plane orthogonal to the user's line of sight. However, if we fix input mapping at the beginning of the drag, the mapping may become confusing as the user's viewpoint changes significantly. Solutions to this problem include changing the input plane dynamically in response to the changing viewpoint or making the fixed input plane highly visible so that users can understand the mapping.

Volumetric displays, which users can view from any angle, pose challenges for reusing current 2D GUI techniques. Because a user can change the view either by moving around the display or rotating it, user interface widgets at a fixed location may not be readable. Employing VUI components that automatically orient themselves in place relative to the user may be one solution.<sup>12</sup> Alternatively, the VUI widgets could correspondingly rotate in the volumetric display as the user moves to another viewpoint.

Rotating the volumetric display requires a mechanism for coarsely tracking user location. Implicit solutions include electric field sensing, referential GPS, or simplistic devices such as pressure pads around the volumetric display. An explicit solution is to have users indicate their locations by touching the display's enclosure after they move, whereupon the widgets position and orient themselves appropriately.

These follow-the-user techniques do not, however, work when several users need to view the same user interface widgets at dramatically different viewpoints. Duplicating the user-interface widgets for each viewer or duplicating the widgets at varying angles could solve this problem—although such duplication rapidly consumes display space.

Developing a single set of *omniviewable* user-interface widgets could sidestep this constraint. Figure 6a shows multifaceted widgets that users can read and manipulate from any viewing angle. While some omniviewable

widgets are static, dynamic widgets that rock to-and-fro slowly are also viable. In current GUIs, user-interface widgets typically appear along the display's edges, while the data occupies the display's center. VUIs could adopt a similar approach, with the user-interface widgets occupying certain regions within the volume. Because the data must be to be operated upon, the widgets cannot encase it, but we can imagine a VUI ring like that shown in Figure 6b surrounding the data.

As Figure 6c shows, the user's position relative to the display creates an optimal viewing zone. The central part of this viewing zone could be used to display items of primary interest, whether 3D data or user-interface widgets. Items of secondary importance could be relegated to areas outside this optimal viewing zone.

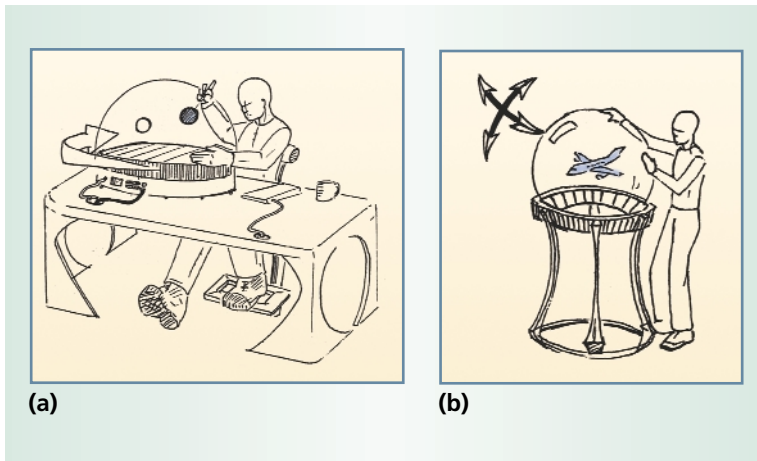
## FUNDAMENTAL TASKS

An interactive volumetric-display system must support several fundamental tasks. These tasks often combine to offer a more complex and rich set of user interaction activity. Table 1 lists tasks to consider when developing an overall user interaction style. While using our mockups to support these tasks, we discovered some significant design variations and issues.

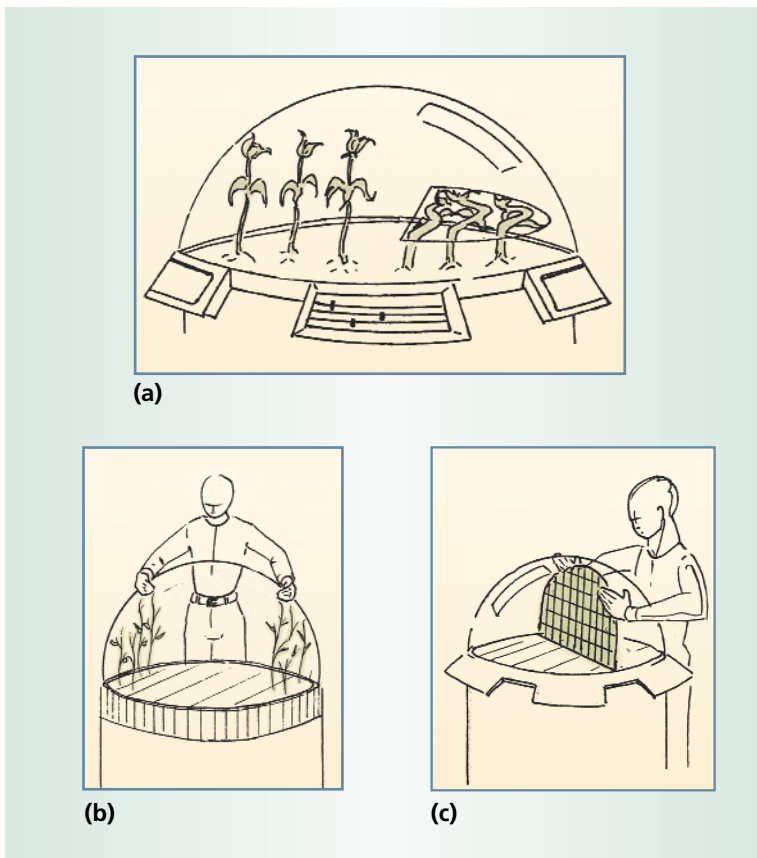
Given a sample device's enclosure shape, size, and input style, determining how to support fundamental tasks raises several questions:

**Table 1. Fundamental tasks for volumetric user interfaces.**

Task	Description
Select	Choose one or more objects within a scene and provide feedback such as highlights or vibration
Move	Place an object in the 3D scene or drag it through one to three dimensions
Rotate	Orient an object in the 3D scene
Scale	Shrink or enlarge an object
Navigate	Pan, tilt, or zoom the 3D scene
Command	Execute commands via menu-item selection or text entry
Filter	Using techniques such as magic lens or projection shadows, adjust scene contents to better view or interpret all 3D information



**Figure 7. Mechanisms for physically navigating a volumetric display: (a) a dome-shaped display rests on a turntable that the user can easily rotate; (b) a spherical display enclosure that can rotate in two degrees of freedom to tumble the data within.**



**Figure 8. Volume management: (a) After the user applies a crushing operation on the right side of the dome, the plant geometry is squashed but still visible under the crush plane; (b) the user's separation gesture parts the 3D space, moving the objects to the sides and either scaling them to fit or clipping them off; (c) using gestures to create subvolumes by specifying a plane within the display.**

- When do these designs fail?
- What aspects of a particular design cause fatigue?
- Can altering the input device or input style ease design tension?
- Which techniques work across many of the input styles?

Using Table 1's task list as a framework for exploring these issues, we consider two system configurations and usage scenarios.

### Two usage scenarios

Our first example assumes a dome-shaped volumetric display with a tilt-sensitive stylus as the input device. To select an object, a user positions and orients the stylus to cast a ray that hits the target object. We found this style of selection extremely compelling because the user can comfortably and easily move and orient the cast ray.

However, this technique can pose difficulties when moving the object. Translating the object to any position within the dome's volume requires not only moving and orienting the cast ray, but also varying its length. One solution to varying the ray's length is to use a roller wheel on the stylus, as with the Wacom Intuous tablet's airbrush (<http://www.wacom.com/>).

To rotate an object, we propose adopting a revolving skewer metaphor in which the cast ray serves as the axis for object rotation. Rotating the stylus device rotates the skewered object about the cast ray.

Our second example, a volumetric display with a cubic enclosure, similarly uses a stylus on one cube face and ray casting into the 3D scene for object selection. Moving an object may be easier than when using the dome because one cube face can specify 2D placement while an adjacent cube face specifies depth. The cubic display's rectilinear faces suggest a design in which 2D orthographic projections appear on the cube faces. Alternatively, the faces can display virtual shadows of elements within the 3D scene.

These alternative views could serve as 3D scene filters that assist the user in viewing, inspecting, aligning, and manipulating objects.

### Navigation issues

Navigating a 3D scene within volumetric displays requires choosing between physical or virtual techniques. As Figure 7a shows, we can design a dome-shaped display to physically rotate as if resting on a turntable. Our mockups indicate that the dome shape has affordances that suggest this style of interaction.

We can also adapt virtual techniques such as the arcball to specify 3D scene navigation. Stroke gestures along the dome equator revolve the scene, while gestures closer to the center tilt the scene in the corresponding direction. Virtual techniques may be better than physical techniques when a simple physical action



cannot easily perform the navigation. For example, Figure 7b shows that, depending on the mechanical and electronics design of a spherical-shaped enclosure, rotation about two axes requires special designs.

Navigation techniques also depend on the display's size. For example, in a desk-size display, it may be possible for the seated user's head movements to perform simple navigations. However, in a larger display, the user may need to walk around the display to navigate different regions. Virtual techniques that require input gestures must also be sensitive to display size as users may find that a very large display exceeds their reach.

## VOLUME MANAGEMENT

Volumetric displays, like conventional 2D displays, require some mechanism for managing the placement of data within the display. In our physical mockups, we found ourselves using everyday gestures to delineate sub-portions of the working volume. As Figure 8 shows, such delineating gestures could be used to effectively create and manage space within a volumetric display.

**O**ur physical mockups effectively elucidate a wide range of user-interface design challenges and volumetric-display opportunities. We could argue that, thanks to technology momentum, designers will most likely adapt standard user-interface techniques to VUIs. However, our mockups have revealed a rich set of alternative user-interface design options. Because they are based on fundamental affordances and technology expectations, these options may produce a more effective user interface. However, much evaluation work remains to be done before we can harness the full potential of interacting with volumetric displays. \*

---

## Acknowledgments

We thank Azam Khan, Shane Clodd, Greg Favalora, and Knut Langhans for their assistance.

---

## References

1. R. Haber and M. Hershenson, *The Psychology of Visual Perception*, Holt, Rinehard, and Winston, Philadelphia, 1973.
2. W. Buxton and G.W. Fitzmaurice, "HMD's, Caves, and Chameleon: A Human-Centric Analysis of Interaction in Virtual Space," *Computer Graphics*, vol. 32, no. 4, 1998, pp. 64-68.
3. M. Czernuszenko et al., "The ImmersaDesk and Infinity Wall Projection-Based Virtual Reality Displays," *Computer Graphics*, vol. 31, no. 2, 1997, pp. 46-49.
4. D. Ebert et al., "Realizing 3D Visualization using Crossed-Beam Volumetric Displays," *Comm. ACM*, vol. 42, no. 8, 1999, pp. 101-107.
5. M. Lucente, "Interactive Three-Dimensional Holographic Displays: Seeing the Future in Depth," *Computer Graphics*, May 1997, pp. 63-67.
6. M. Sinclair, "The Haptic Lens," *Proc. Siggraph 97*, ACM Press, New York, 1997, p. 179.
7. T. Murakami and N. Nakajima, "Direct and Intuitive Input Device for 3D Shape Deformation," *Proc. Computer-Human Interaction (CHI 94)*, ACM Press, New York, 1994, pp. 465-470.
8. R. Balakrishnan et al., "The Rockin'Mouse: Integral 3D Manipulation on a Plane," *Proc. Computer-Human Interaction (CHI 97)*, ACM Press, New York, 1997, pp. 311-318.
9. G. Fitzmaurice, H. Ishii, and W. Buxton, "Bricks: Laying the Foundations for Graspable User Interfaces," *Proc. Computer-Human Interaction (CHI 95)*, ACM Press, New York, 1995, pp. 442-449.
10. S. Zhai, W. Buxton, and P. Milgram, "The 'Silk Cursor': Investigating Transparency for 3D Target Acquisition," *Proc. Computer-Human Interaction (CHI 94)*, ACM Press, New York, 1994, pp. 459-464.
11. B. Conner et al., "Three Dimensional Widgets," *Computer Graphics*, vol. 22, no. 4, 1992, pp. 121-129.
12. G. Fitzmaurice et al., "An Exploration into Supporting Artwork Orientation in the User Interface," *Proc. Computer-Human Interaction (CHI 99)*, ACM Press, New York, 1999, pp. 167-174.

*Ravin Balakrishnan is a PhD candidate in the Department of Computer Science, University of Toronto, and a consulting researcher at AliasWavefront. His research interests include user interfaces for 3D graphics, novel input devices, haptic feedback interfaces, interfaces for mobile devices, and the fundamentals of human perception and motor control. Contact him at ravin@acm.org.*

*George W. Fitzmaurice is a research scientist at AliasWavefront. His research interests include high-dimensional computer input, computer augmented environments, and novel interaction techniques. He received a PhD in computer science from the University of Toronto. Contact him at gf@aw.sgi.com.*

*Gordon Kurtenbach is director of the Interactive Graphics Research Group at AliasWavefront. The group's focus is on 3D interactive graphics with research in the areas of input technologies, manipulation, modeling, animation, and rendering. He received a PhD in computer science from the University of Toronto. Contact him at gordo@aw.sgi.com.*