

HoloSketch: A Virtual Reality Sketching / Animation Tool

MICHAEL F. DEERING

Sun Microsystems Computer Corporation

This article describes HoloSketch, a virtual reality-based 3D geometry creation and manipulation tool. HoloSketch is aimed at providing nonprogrammers with an easy-to-use 3D “What-You-See-Is-What-You-Get” environment. Using head-tracked stereo shutter glasses and a desktop CRT display configuration, virtual objects can be created with a 3D wand manipulator directly in front of the user, at very high accuracy and much more rapidly than with traditional 3D drawing systems. HoloSketch also supports simple animation and audio control for virtual objects. This article describes the functions of the HoloSketch system, as well as our experience so far with more-general issues of head-tracked stereo 3D user interface design.

Categories and Subject Descriptors: I.3.3 [**Computer Graphics**]: Picture/Image Generation—*display algorithms*; I.3.7 [**Computer Graphics**]: Three-Dimensional Graphics and Realism

General Terms: Human Factors

Additional Key Words and Phrases: 3D animation, 3D graphics, CAD, graphics drawing systems, graphics painting systems, man-machine interface, virtual reality

1. INTRODUCTION

Virtual reality (VR) is often promoted as the next generation of man-machine interface, but other than in simulation and entertainment, virtual reality has not yet found widespread use in everyday applications. Any application in which rapid, accurate human understanding of the shape, orientation, and distance to 3D objects is important is a candidate for productivity improvements through virtual reality. But there are many barriers to deploying virtual reality technology. Only recently has 3D graphics hardware of minimally acceptable rendering performance become affordable. Trackers and displays are fraught with technical limitations. Building *any* completely new software application has become progressively more expensive in recent years, as user’s expectations of functionality, quality, documentation, and support have steadily risen. Standardization of any form, whether hardware or software, is not yet on the horizon.

Author’s address: Sun Microsystems Computer Corporation, MS-UMTV18-211, 2550 Garcia Avenue, Mountain View, CA 94043; email Michael.Deering@eng.sun.com

Permission to make digital/hard copy of part or all of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage, the copyright notice, the title of the publication, and its date appear, and notice is given that copying is by permission of ACM, Inc. To copy otherwise, to republish, to post on servers, or to redistribute to lists, requires prior specific permission and/or a fee.

© 1995 ACM 1073-0516/95/0900-0220\$03.50

ACM Transactions on Computer-Human Interaction, Vol. 2, No. 3, September 1995, Pages 220–238

Given this, most virtual reality applications have been built by or for large organizations' specific internal applications, and even these are more often than not proof-of-concept prototypes, rather than something that is in everyday use. Today's commercial virtual reality software offerings are mostly toolkits or demos.

In contrast, our goal was to test the hypothesis that virtual reality technologies could be used in mass market applications, not just limited niches. A nearly universal need in 3D is the ability to create 3D objects and assemblies; this is the application area we chose to experiment with.

This article describes HoloSketch, an easy-to-use 3D sketching, drawing, painting, and animation system. It is roughly analogous to simple 2D drawing systems. Users can rapidly create 3D shapes, apparently in the air in front of them, and then edit and animate their constructs. The resulting animated 3D images can be saved for later redisplay, emailed to others, or used to give real-time animated presentations.

After a brief summary of the display technology and environment employed, the features of the HoloSketch system are described. Some important implementation issues are then discussed, followed by preliminary results of using the system and possible future applications.

2. PREVIOUS WORK

There have been a number of "direct interaction" 3D construction systems built in recent years, but in many of these systems "direct interaction" still means drawing with a 2D mouse over a monoscopic 3D image projection. The Brown system [Snibbe et al. 1992; Zeleznik et al. 1993] uses 3D widgets controlled by a 2D mouse. The system described in Hanrahan and Haerberli [1990] used a 2D mouse to paint 2D images directly on 3D surfaces.

There are severe problems in using 2D mice to control 3D objects: in the physical world, our hands naturally control xyz positions and three axes of rotation. A 2D mouse can only directly affect two of these six variables at a time. Mechanisms to "guess" or constrain additional variables have had limited successes. Some systems have thus adopted a full 6-axis input device [Galyean and Hughes 1991; Liang and Green 1993; Sachs et al. 1991], but they are still coupled to a monoscopic display. These systems employ the 6D mouse for positioning, orienting, and stretching 3D objects, but in a relative mode (i.e., the control device is not at the visual site of object creation). There have been a few attempts to make systems run within a stereo head-mounted display (HMD) and 6D glove/mouse environment: Butterworth et al. [1992] and Halliday and Green [1994] and the DVise system by Division Ltd. U.K. Unfortunately, the visual resolution of current HMD technology is extremely limited; even displaying buttons with readable text labels is difficult in these systems.

Each of these systems has some similarities to HoloSketch, but none has the full combination of head-tracked high-resolution stereo with a 6D hand input device; this makes the HoloSketch system functionally quite distinct. At present very few of these "new-style" 3D systems discussed above exist,

and the author knows of no commercial user site in which 3D objects are routinely constructed *within* a head-tracked stereo virtual reality environment. Most virtual reality objects are constructed with 2D systems, or even just using text editors.

3. DISPLAY ENVIRONMENT

3.1 Display Technology

The HoloSketch system was designed to work within multiple and different virtual reality display environments, but the particular focus has been with the Virtual Holographic Workstation “fish-tank stereo” display described in Deering [1992]. This configuration, shown in Figure 1, consists of a high-performance desktop 3D workstation (the SPARCStation 10 ZX [Deering and Nelson 1993]) with a 960×680 20-inch stereo CRT, running at a 112.9Hz refresh rate (56.45Hz per eye). The user wears a pair of head-tracked field sequential stereo shutter glasses (from StereoGraphics/Logitech) and manipulates the virtual world through a hand-held six-axis mouse/wand.

For every frame of display, the computer calculates a new viewing matrix, separately for each eye, that corresponds to the current position of the user’s eye. This head-tracked stereo display produces high-quality virtual reality imagery, allowing the user to look freely around images hovering in space. For head-tracked stereo displays to be visually effective, the rendering frame rate must be kept in the range of human visual motion fusion, e.g., at least 13Hz. This constrains the complexity of 3D imagery displayed to that which can be redrawn at the required rates.

The 3D mouse is augmented with an offset digitizer rod protruding from it, effectively making the mouse a six-axis wand or a “one-fingered data glove.” The end of the rod (“wand tip”) functions as the direct-manipulation cursor for most of the HoloSketch system. Even after a short period of use, the wand tip feels like an extension of your index finger, similar to how a baseball glove extends the fingers. The 3D mouse has three top buttons and one side button. Through the remainder of this article, the input device and its buttons will be referred to as a wand.

3.2 Calibration Fanaticism

Most all other VR display systems are *not* capable of displaying virtual objects at precise and stable perceived positions in the physical world. With HoloSketch, if the system wants to present a 3cm on-a-side cube positioned 10cm from the center of the physical screen, then that is where the virtual object will appear (± 0.25 cm), no matter what angle or position the viewers move their heads relative to the CRT (within the tracking range). The relative accuracy of the system is much higher; objects can easily be positioned by hand to better than 0.05cm. In most all head-mounted display systems, the perceived position (and distance) to objects in virtual space will “swim” and distort by very large amounts as the viewer changes head position. But the perception of physical stability is essential to engage our



Fig. 1. Desktop virtual reality display system, with head-tracked stereo glasses and 3D mouse/wand.

haptic sense fully for interacting with virtual objects in an editing system like HoloSketch. HoloSketch is so accurate that one can hold up a *physical* ruler to a virtual object to make (accurate) measurements.

In many commercial 3D display systems the user employs indirect controls to manipulate or modify a 3D object. For example, a set of three sliders on the screen or three knobs on a physical dial box might be used to control the 3D position of an object. More-recent, mostly experimental systems allow the user to place the 2D mouse cursor directly over the screen image of the 3D object when manipulating it. Such systems have been referred to in the literature as “direct manipulation.” But, in fact, in these systems the user interaction is still relative, even when a true 3D mouse is used. By definition, if the display system does not mirror the perceptual stability of the physical world, any manipulating system must be *relative*, rather than absolute. Perhaps the term “absolute manipulation” can be used for the case where the percepts of the virtual world and the physical world are equivalent to high accuracy.

HoloSketch’s high accuracy was not easily achieved; it required attention to numerous details. Here we summarize those described in Deering [1992]: first and foremost, a high absolute accuracy position and orientation tracker (Logitech’s ultrasonic system). Magnetic trackers, while excellent in other aspects including their *relative* accuracy, just do not have subcentimeter

absolute accuracy. Next, we touch on the use of a high-resolution, relatively flat direct-view stereo CRT. Furthermore, corrections are made for the distortions due to curvature of the CRT, as well as those due to the index of refraction of the thick glass between the CRT light-emitting phosphor and the viewer. The individual viewer's interocular distance is a calibration parameter, and the system corrects dynamically for interocular distance changes due to rotation of the viewer's eyes in their sockets. Other calibration includes precise registration of the tracker transmitter to the CRT bezel.

It must be stressed that many of HoloSketch's interaction techniques rely upon this high accuracy and would not work without it.

4. HOLOSKETCH MENU DESIGN PHILOSOPHY

In extending the concept of a computer-based sketching package to 3D, we had to strike a balance between the extremes of slavishly copying 2D systems versus gratuitously doing everything different in 3D. Our goal was more neutral: do what seems to work. Some universal concepts are retainable, such as "cut" and "select." Other concepts, like "move" and pop-up menus, need to be modified to work naturally in a 3D context.

In most 2D and 3D drawing packages, an array of screen buttons and menu selectors are present off to one edge of the drawing space (or as an independently positionable menu). One set of menu buttons selects the current drawing mode, while others independently control various drawing attributes such as: drawing color, line thickness, etc. Yet other buttons select momentary actions ("cut," "duplicate") or initiate more-extensive operations ("save," "print").

There are several problems with duplicating this setup in 3D. One is the cost in screen real estate. With a head-tracked display, shifts in head position lead naturally to shifts in the relative positions of 3D objects of different depths. This makes it quite a bit harder to position a menu area where it would not interfere visually (behind or in front of) with the objects being edited. Another problem is the 3D consequences of *Fitts' Law*: to select a menu item by physically rendezvousing the wand tip with it in three dimensions takes more time than in 2D, because further distances must be traveled. Finally, there are rendering-resource limits: a complex menu consumes valuable rendering cycles that otherwise could have been applied to displaying more-complex work pieces at reasonable frame rates.

For these reasons, HoloSketch had to depart somewhat from 2D screen systems, and instead, all the main controls are hung from a single huge pop-up menu engaged by depressing (and continuing to hold down) the right wand button. This menu might be described more properly as a "fade-up" menu: when the right wand button is first depressed, the existing work space is faded out to the background color over a quarter second time frame. Then the menu geometry is faded up from the background color in the next quarter second. The menu is organized as a 3D pie menu and always materializes centered around and slightly behind the present position of the wand tip. The menu layout is shown in Figure 2. The menu stays fixed in its initial position

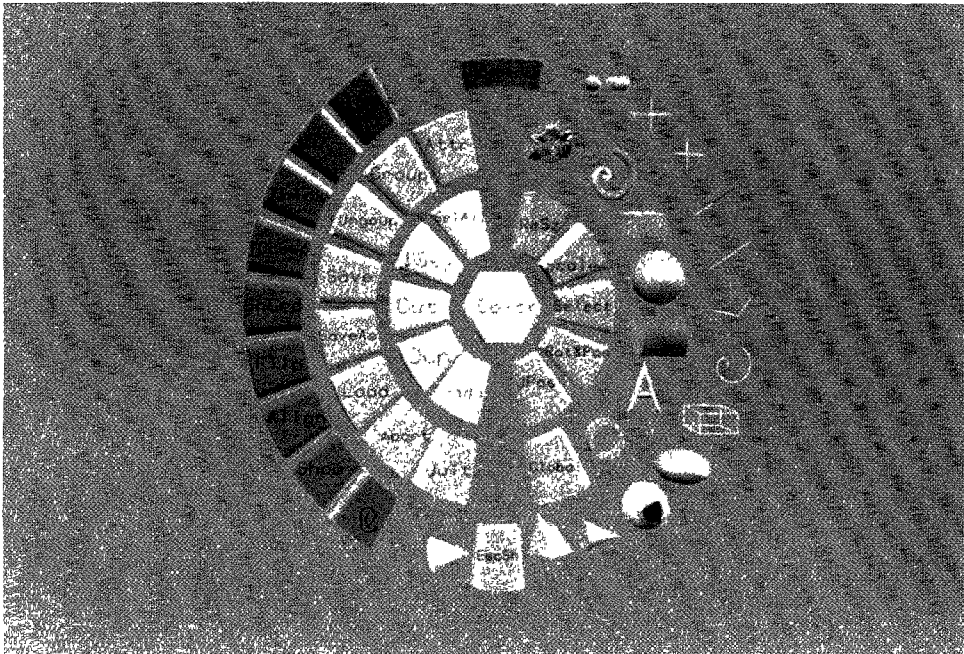


Fig. 2. The 3D fade-up menu system interface, invoked by holding down the right wand button. The 3D icons on the two outer right rings are individual geometry drawing modes. The buttons on the inner right ring are editing modes. The buttons to the left are one-shot actions.

for the duration of the invocation, remaining visible until the user releases the right wand button, resulting in a reversal of the fade-up process.

To select a menu item, the user pokes the wand tip at the desired button. The poked button “lights up” as soon as the wand tip is close enough in three-space. These buttons either change current modes (such as the current drawing primitive) or cause one-shot actions, such as “cut.” A one-shot button lights up when approached, but the action does not take effect unless the wand tip remains within the capture range of the button when the right wand button is released. In this case, the successfully engaged menu button changes to a bright white color during fade-out, confirming the selection.

The 3D fade-up menu supports submenus. When a button with a submenu attached is approached by the wand tip, the whole pie menu moves back into the screen a few centimeters, and the submenu fades up around (but slightly behind) the wand tip (Figure 3). If the wand is moved too far from the plane of the submenu, the submenu fades back out, and the main menu comes back forward. The interface has a fairly natural feel to it, but the capture limits of the menu buttons take some practice to get used to.

Like many drawing systems, HoloSketch is a modal editor. At any one time, there is a single current drawing or editing mode in force. To “draw” in the current drawing mode, the wand tip is moved while holding down the left

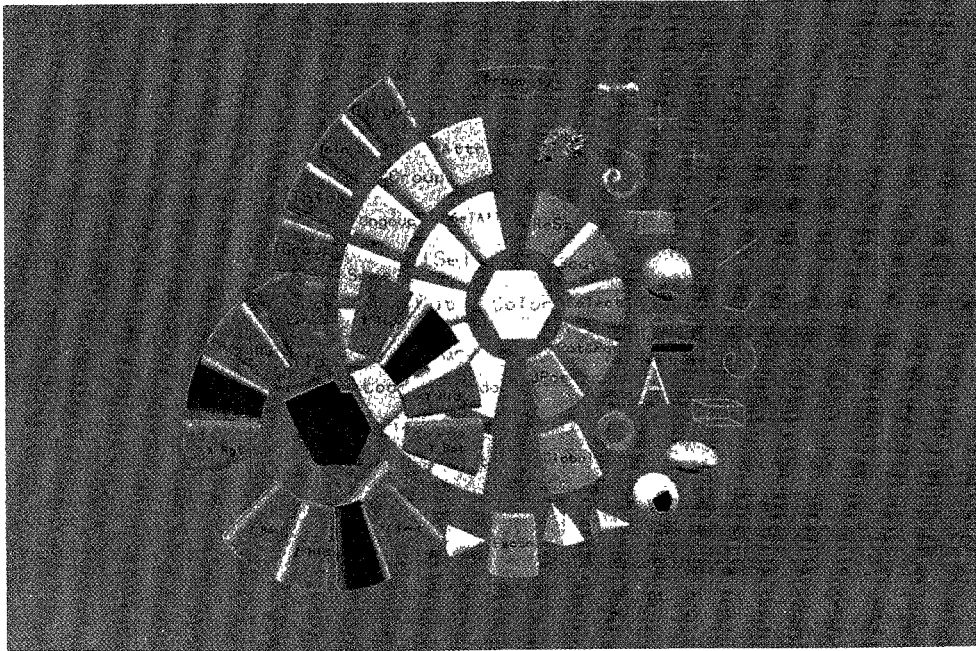


Fig. 3. Align submenu of main menu. All three coordinate axes can have separate alignment constraints.

wand button. Many drawing and editing features are controlled by the position, orientation, and movement of the wand in space and will be described individually with their functions below.

5. HOLOSKECH FEATURES

5.1 Drawing Modes

Currently, HoloSketch supports several types of 3D drawing objects: rectangular solids, spheres, ellipsoids, cylinders, cones, rings, free-form tubes, 3D text, isolated line segments, free-form and polyline wires, and several others. All objects are drawn in a similar manner: first the appropriate drawing primitive is selected as the current drawing mode (through the fade-up menu). Then, when the left wand button is depressed, a new instance of this form of object is created, initially centered at the wand tip's current position. As long as the user continues to hold down the left wand button, the size and/or shape/form of the newly created object can be changed by moving the wand in 3-space.

Simple objects just snap their diagonal to the current position of the wand tip when drawing. Another class of objects is created by making controlled sweeps through space with the wand, while the left button is down. As the wand is moved, a trail of material is put down behind it. Materials include

wire-frame lines, fixed or variable diameter “toothpaste,” or clouds of random small triangle particles.

Other primitives include solid 3D text (input via keyboard) and random fractal mountains, with more primitives being added all the time.

5.2 Importing Geometry

Besides creating new geometry from the primitives listed above, HoloSketch also supports importing polygonal geometry from other modeling environments. Once imported, geometry is treated as any other HoloSketch object. Two-dimensional RGB rasters can be imported as texture maps.

5.3 Drawing Attributes

HoloSketch has a number of drawing attributes, the most important being current color. The current color is controlled via a 3D pop-up RGB color cube engaged by touching the center button of the fade-up menu (Figure 4). The wand tip cursor is replaced with a small colored sphere, whose color directly reflects the current new color selected by the wand tip’s *present location within the RGB color cube*. Upon wand right button release, the last color selected becomes the new current color. If the wand tip moves outside the color cube before release, the colored sphere disappears, indicating that no change will be made to the current color if the wand right button is released now.

5.4 Selection

As with many systems, the last object created is automatically selected. Otherwise, a particular object is selected by depressing the middle wand button with the wand tip inside the object to be selected. The selected object then does a slow continuous color change flash between its intrinsic color and white, to indicate that it is selected. In contrast to a more-static 2D application, for a real-time 3D animation system there is a metaproblem with how to show selection; any animated graphical effect that indicates that an object is currently selected might also be a useful standalone effect. A secondary selection mechanism of blinking the object (appear/disappear) is automatically used when just color flashing would be ambiguous.

With many systems, the selection action first deselects all existing selected objects and then selects a new object, if any is hit. The middle mouse button plus the keyboard shift key is used for cumulative selections of multiple objects. Whether singular or cumulative, reselecting an already selected object deselects it. In both selection cases, reselecting an already selected object deselects it. If no object is found intersecting the wand tip’s location when the wand button is initially depressed, then a rubberband selection inclusion cube is dragged out from the initial point of button selection, analogous to the 2D selection box of many 2D drawing systems. To be selected this way, objects must be entirely within the final released selection box. There are two one-shot selection main menu buttons: “deselect all” and “select all.”

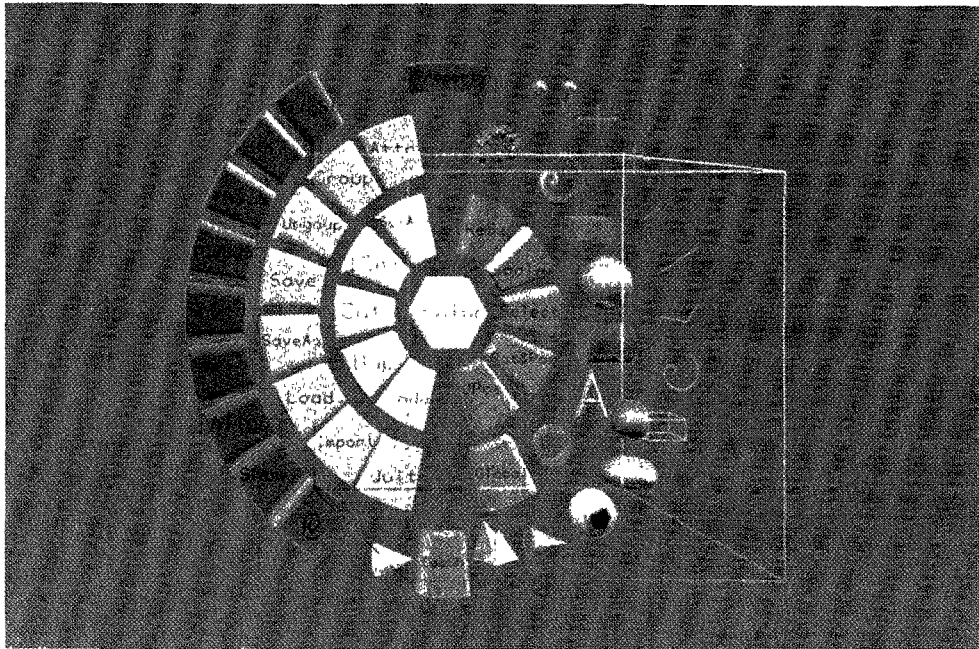


Fig. 4. The color selector: an RGB color cube. The pink sphere to the lower right is the wand cursor reflecting the color that would become current if the right wand button were released now.

5.5 Editing Operations

Movement. HoloSketch supports a variety of editing operations. To be edited, an object must first be selected. The most-important editing operation is simple movement. It is even more essential that movement be rapid and intuitive in 3D than in 2D. Because of this, we dedicated the wand side buttons strictly to movement control. These buttons are engaged by “squeezing” the wand: this action is very similar to physically grasping an object. Now, to support multiple/different complex 3D movement modes, and to guard against accidental wand side button engagement, additional keyboard keys must be held down as an object is grasped. HoloSketch supports *three* movement modes: pure positional movement (orientation preserving), pure orientational, and a combined position and orientation movement operation. These actions are bound to the main keyboard control key, control-shift, and shift keys, respectively.

Which of these movement modes is more appropriate to a particular operation is fairly intuitive; placing a hat at a jaunty angle on the head of a snowman clearly calls for the more-general movement; sliding figures around on a floor is more appropriate in the simpler movement mode.

To orient an object, a center of rotation must be defined. Choosing the point where the object was selected by the wand tip is similar to how origins are established in the real world when grasped by hand; but in practice, when the

change in orientation is relative (and offset), we seem to expect objects to rotate about their visual center. Thus all objects in HoloSketch have a rotational center defined by the center of their minimal containing bounding volume.

Fine motor control is even harder in 3D than in 2D; this is because the wand is being held in 3D. With no support at the elbow, there is considerable fine-motion jitter. Much of this problem can be solved by providing elbow support; providing wrist support can eliminate virtually all remaining jitter. But we found that adding a ten-X reduction mode (meta key) delivers even finer control and is less cumbersome than physical supports.

Standard practice ends up with the user holding the 3D wand in one hand while the other hovers over the control, meta, and shift keys. Even while wearing stereo glasses these keys can easily be found by sight or touch.

Grouping. Another fundamental editing operation is *grouping*. Grouping plays a key role in the semantics of the HoloSketch system. The simple grouping operation groups all objects currently selected (including other groups). As in 2D systems, the grouped object can be manipulated as a *single* object; it can be positioned, oriented, or scaled (discussed below) as any single primitive can. The orientational center of the group is computed as the center of the bounding volume of the sum of the elements. The perceptual behavior seems fairly natural.

Grouping Stacks Hierarchically. Groups of grouped primitives behave as expected. Objects can also be “ungrouped.” Other forms of grouping are used for animation operations and are covered in the next subsection.

Scaling. Scale change is another useful operation. HoloSketch supports a uniform scale change operation that can be applied to any primitive or grouped object; objects expand/contract in size about their center of rotation (once again, this feels fairly natural). The Z distance of the wand controls the scaling. Nonuniform scale changes at present are confined only to primitives that support them and not to groups of primitives. These resize operations are a lot like the initial creation operation, allowing the aspect ratio of rectangular solids to be changed, etc.

Attributes. While most editing operations are controlled by waving the wand, all objects also have property sheets where many of their parameters can be changed numerically. For example, text can have its scale set in point size (or other physical units) by typing in the desired number into a field of the property sheet. The property sheet is accessed by first selecting the 3D object desired and is then invoked via the attributes button of the fade-up menu. This invocation (temporally) replaces the workplace by the 3D property sheet. Finally, the property sheets are true 3D objects; fields to be changed are first selected by the 3D wand tip (*not* the 2D mouse).

5.6 Animating Operations

By its very nature, a virtual reality system performs real-time animation. If this is done properly, a user may perceive the displayed objects as “not doing

anything, just hanging in space,” while in reality the computer is madly recomputing the images dozens of times per second in response to every slight movement of the user’s head. So long as we are paying for this real-time display capability, many dynamic animation effects (movement, rotation, etc.) can be included for little extra computational cost.

There are a number of systems for controlling real-time animation, including 3D widgets [Elliot et al. 1994; Snibbe et al. 1992; Zeleznik et al. 1993]. However, as HoloSketch’s goal was to produce a system that could be used by novices (and with hardly any typing), an even-simpler interface needed to be found for simple animation actions.

This led to the development of the concept of *elemental animation objects*. These are iconic tokens that when isolated remain inert but when grouped with other primitives apply some form of dynamic effect.

Rotation about an Axis: Rotors. One of the easiest animation actions is simple rotation. HoloSketch supports this through the elemental-animation object “rotor.” A rotor is created by selecting rotor creation mode and clicking the left wand button at the point in space where the rotor is desired. Rotors appear as an axis circled by an arrow (seen in the upper right of Figure 2) and can be repositioned and oriented as with any other object. Rotors do their work when a set of objects is grouped that includes a rotor. After grouping, the set of objects will rotate about an axis in space defined by the rotor (which is no longer visible). This whirling group is selectable, positionable, orientable, and scalable as before and will continue rotating.

A second form of rotor oscillates between two angular extremes; this is a good way to get a butterfly’s wings to flap. The angle of oscillation, as well as the frequency, is controllable via property sheet.

The power of rotors is that you specify their position and orientation via the same mechanism used for geometry. This is also where dynamic head-tracking offers another advantage: one can rapidly align a rotor axis with geometry to be rotated with sufficient accuracy for most purposes (e.g., without requiring numeric entry).

Solid Movie Looping: Temporal Grouping. For hundreds of years, 2D images have been animated by laboriously hand drawing sequential frames for display. This same technique can be used to animate 3D drawings, via HoloSketch’s temporal-group elemental animation operation. First a series of objects are drawn, each slightly different. After selecting all the objects, temporal grouping causes a different single member of the group to be displayed in each successive frame in time. Rather than sculpting thousands of individual 3D objects, the intent is that short object motion loops can be added to otherwise static objects (hair waving in breeze, tail twitching).

Color Oscillation. Sometimes the only animation required is to change the color of an object in a repetitive fashion (blinking lights). A color oscillator is an elemental animation object, created similar to a rotor, that, when grouped with other objects, causes the objects to oscillate between the two colors of the color oscillator. The 3D icon for the blinker is three spheres: two fixed with a

third bouncing between them (upper rightmost fade-up menu icon, Figure 2). Color oscillators are an example of HoloSketch objects composed of multiple subobjects: in this case the left and right spheres. Either of the oscillator colors may be modified by selecting the appropriate subobject and then changing the current color in the usual way. The frequency of blinking can be modified by editing the property sheet of a color oscillator.

Scaler. This elemental animation object provides continuous (uniform) change between two sizes for all elements of a group. The group center is the location of the scalar object, *not* the center of the bounding volume. This allows an object placed atop another to maintain contact while shrinking/growing.

Shifter. This elemental animation object provides continuous (linear path) movement between two points.

Flight Path. HoloSketch allows any line segment (wire, polywire, free-form wire) to be changed into a “flight path” elemental animation object. When grouped with a set of other objects, it causes the entire collection to go flying down the path in space (position *and/or* orientation) established by the wire. Other animation timing-control functions (beyond the scope of this article) allow the action of a flight path to be delayed until a desired point in time.

5.7 Environment Operations

Aspects of the 3D environment are also controllable. The current background color can be set similar to how object colors are set. Light sources are controlled by instantiating a 3D handle to them. The handle (a 3D icon looking like a lamp) can then be positioned, oriented, and colored like any other object, but the changes apply to the light source itself and persist after the handle object itself is deleted. Note that this is similar to the Brown 3D Widgets [Snibbe et al. 1992; Zeleznik et al. 1993].

5.8 Viewing Control

Immediate control over viewpoint is *always* controlled by the position of the user’s head. To look around an object, the user simply moves his or her head. To examine a particular object, the user first selects it and then rotates it by rotating the wand in one of the movement modes.

The above viewing control keeps fixed the relationship between the plane of the CRT surface and a corresponding plane of the virtual world (initially the xy plane of model space). However, it is also desirable to control this relationship directly between the real and virtual worlds. This is where the global positioning, orienting, and scaling changes come into effect. When this control mode is selected from the fade-up menu, depressing the wand side button allows rotation and movement of the entire virtual universe as if it were an object. This is one mechanism for the user to move through the virtual world. In this mode, depressing the left wand button changes the scale of the virtual world with respect to the physical world of the CRT. With world building, frequent changes in position and scale of the user occur as one flits about the

virtual world, adding and modifying objects. To facilitate such view changes, HoloSketch supports the concept of “marking” a position/orientation/scale within the virtual universe, so that it can be easily returned to later. At present this uses the same keyboard commands as the Unix vi text editor for marking and returning to positions.

5.9 Units, Snap Grid

One of the things that is different about constructing virtual worlds is that there is a whole world out there, not just isolated objects, and objects and landmarks come in many different scales. This means that much more attention must be paid to the issue of scale. The sizes of objects must be configurable to specific physical units, independent of any viewing-scale changes. Internally, HoloSketch represents all object measurements in units of centimeters. But for editing purposes, other unit systems may be chosen, from microns to light years. Numerical measurements may be entered in these user-defined units. By being precise about units, the same virtual worlds can be used with displays of vastly different scale: 20-inch CRTs and 10-foot projection screens.

HoloSketch also supports a 3D snap grid, where the snap ruling is specified in terms of user units. When snap is enabled, most of the object construction, positioning, and scaling is quantized to integer multiples of the snap grid size.

A different snap operation was selected for rotations. In pure rotation mode, when an object is rotated to within 15 degrees of any 90 degree multiple of its natural coordinate frame, it “snaps” to the 90 degree-aligned orientation.

5.10 Other Operations

A number of other familiar operations are also supported and work in the obvious manner. These operations include: Cut, Duplicate, Undo, Align, Save, Load, and Import. The save files contain mathematical parameterization of the objects and preserve all grouping and elemental-animation information. The file format is either plain text or compact binary and can easily be emailed.

6. IMPLEMENTATION

A current technology trend is that moderately priced 3D graphics-rendering hardware is getting faster *faster* than general-purpose processors are improving. An undeniable consequence of this fact is that the number of CPU instructions that can be executed per graphics primitive rendered is steadily going *down*, and the trend is likely to continue for some time to come [Deering et al. 1994]. Working the other direction is the tendency for software systems to bloat with age; adding features but never rethinking their fundamental architecture. This is why the transition to virtual reality displays will be hard for many existing application systems.

The system architecture of HoloSketch, though, was designed with these trends in mind. HoloSketch's approach is to perform incremental compilation of geometry as it is created and edited. Static stacks of position and orientation matrices are composed and cached as they are edited; indeed, in all cases where the geometry is simple, pretransformed *geometry* is cached as well. The inverses of these cached matrices are also cached; this speeds the selection operation. Each object class has its own optimized selection routine, to ensure both rapid response as well as well-defined picking volumes. Other optimizations include merging multiple primitives upon grouping (multiple independent line segments merged into polypolylines).

To make the natural-feeling orientation semantics work, all matrices are internally represented by three separate components: a position, an orientation represented by a quaternion, and a uniform scale factor. These combine to produce a (cacheable) 4×4 general modeling matrix.

The virtual reality head and wand tracking, and calibrated viewing matrix generation, is provided by a low-level VR package, with direct support and optimization for the ultrasonic tracker used. Special predictive filtering algorithms *completely eliminate* visually perceptible lag for scenes where the geometry is simple enough to allow a 57Hz frame rate.

The complete system and features as described in this article are currently up and running. HoloSketch is meant as an *experimental* test bed; we add new features and primitives as we think of them and try them out on users.

7. RESULTS

It is surprisingly quick and easy to create complex forms in three dimensions using HoloSketch. Figures 5–7 are moderately complex scales. Simple “creatures” have been sketched in as little as 90 seconds. All these figures were created exclusively with the wand.

Most first-time users comment on the freedom that the system gives; there are few artificial constraints. The most-common mistake (other than confusing modes) is forgetting to look *around* the things that they are creating. We are so used to head motion *not* making any difference when interacting with computers that most people hold their heads artificially still. In the real world something that looks mostly straight from one angle is trivially shown to be off-kilter when looked at from a different vantage point; but it takes time to learn that now the computer world works that way too.

7.1 Applications

HoloSketch was created as a general-purpose 3D sketching and simple animation system; there was no one single niche application intended. Rather, a range of related uses are envisioned. These include:

- Quick 3D sketching for a single user.
- Quick way to show/email 3D sketch to others. Can also print views as 2D images.
- Can export/import geometry to/from other packages for other uses.

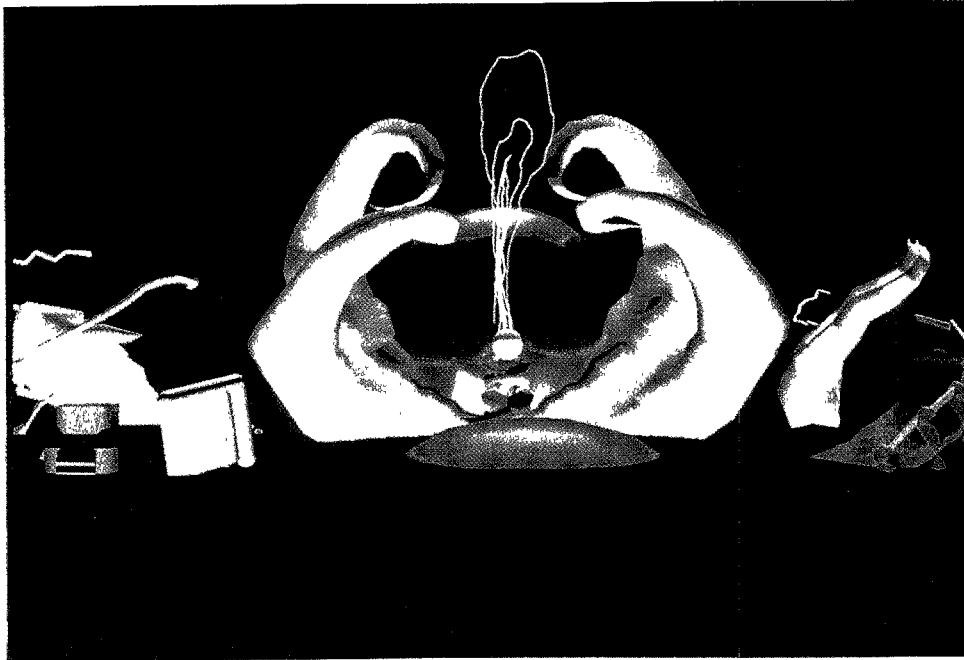


Fig. 5. A complete scene created using HoloSketch. The loop spark in the center is motion animated. (Artist: Halle L. McConlogue).

- Live, animated technical presentations.
- Authoring tool for 4D illustrations for documentation/books/training.
- As a shared (networked) real-time 3D whiteboard. This mode has been successfully demonstrated between two participants over a T1 link 40 miles apart.

HoloSketch now runs on several different display configurations. While the interactive editing operation has been optimized for the direct-view 20-inch CRT stereo head-tracked display, once a universe has been created, it can be viewed in any of several other display configurations, including:

- Stereo Head-Mounted Display,
- The Virtual Portal, a three-rear-screen projection display system [Deering 1993],
- VREX stereo LCD projection display panel, and
- Nonstereo CRT displays.

HoloSketch has been used to author several animations/presentations in several of these display configurations. A fanciful animation of the information superhighway was created with HoloSketch for display in eight simultaneous HMDs at *SIGGRAPH '94*. Several technical talks on virtual reality and stereo have been created in HoloSketch and presented live using the

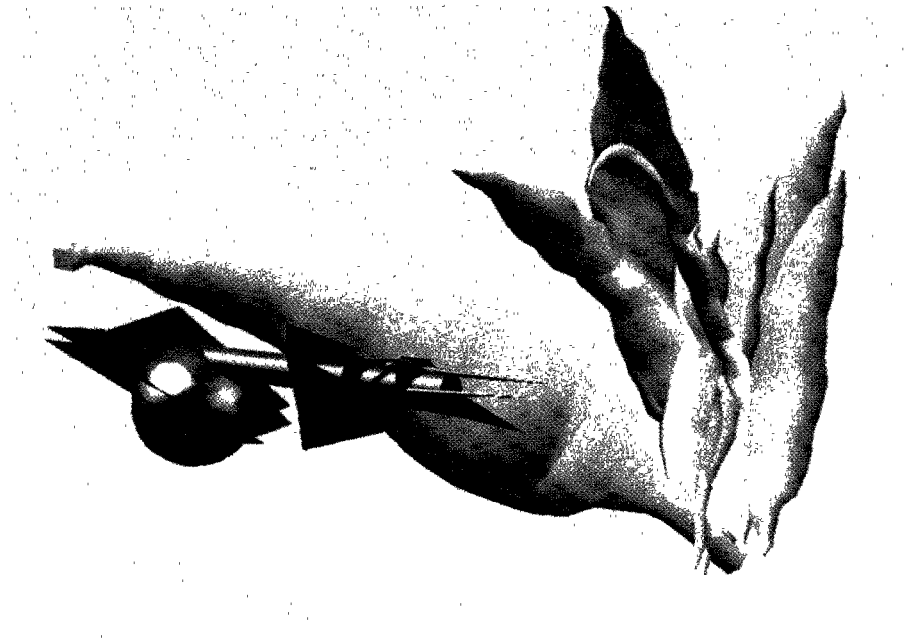


Fig. 6. Two objects created with HoloSketch. The plant was created entirely of the variable-thickness toothpaste. (Artist: Hallie L. McConlogue).

VREX stereo display. All the new worlds shown with the Virtual Portal are authored with HoloSketch.

7.2 User Trial: HoloSketch versus Artist

HoloSketch was designed with the noncomputer scientist user in mind. To test our success at this, we engaged a computer/traditional artist for a month of trials using the system. While not as effective as a broadbased user usability study, it was an appropriate compromise given the initial state of development of the system. This section presents a brief informal overview of the results of this experiment.

The artist started cold, having no prior experience with HoloSketch (though she had used traditional 3D modeling systems) and without any documentation. Within a few days complex objects started appearing, along with comments about the user interface. The artist immediately liked the immediacy of HoloSketch as a 3D medium and the ability to manipulate objects directly in 3D space. Some of the interaction mechanisms took some getting used to, though after the first few days this was no longer perceived as an important issue. The lesson here is, probably no matter how simple or natural-feeling you make the interface, when making the radical change to 3D VR, some things will take more than five minutes for users to internalize.

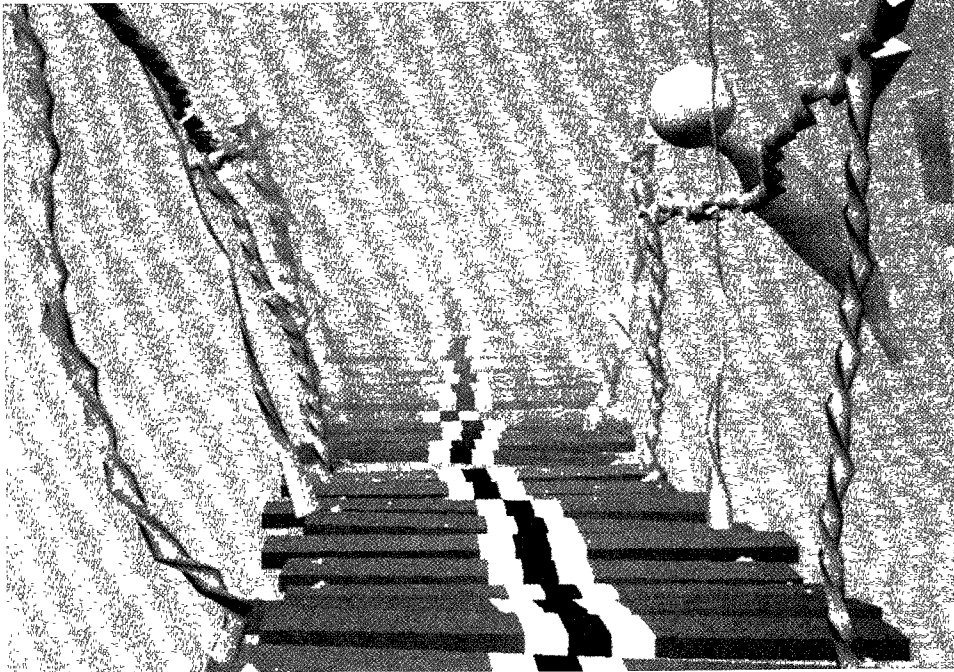


Fig. 7. A rickety-rope bridge created with HoloSketch. This is one section of the fanciful information superhighway animation. (Artist: Hallie L. McConlogue).

As work progressed (the goal was to produce the information superhighway animation for *SIGGRAPH*), more-subtle complaints came in. Holding one's hands in the air for long periods of time was not hard to achieve (with the help of some elbow rests), but making fine adjustments in this position was hard: the 10X reduction mode (which had already been in the works) eliminated this problem, along with the ability to change scale.

The major new mode added was the ability to draw *two-handed*. The idea is to use the 2D *mouse* as a lever to vary dynamically the radius of a toothpaste primitive being simultaneously drawn by the other hand, holding the 3D wand. The coordination involved in using this drawing mode certainly is beyond a novice user but quickly became a primary primitive, especially when creating organic-like forms (see Figures 5–7). The artist liked this primitive because complex shapes could be drawn very rapidly.

Indeed the rapid interaction of the system encouraged the artist many times to create the same object over and over again, to see how simple variations would affect the final outcome. This is how a more-productive system can also lead to higher-quality results.

At the end of the month, the final batch of comments from the artist concerned new features primarily. Overall, she found the interface much more productive than traditional modelers, but to take on some types of projects, a richer and more-complete set of features would be needed.

7.3 Limitations

There are still several technology limitations. The complexity of imagery that can be created is kept low by the capabilities of current rendering hardware (although this is rapidly improving [Deering et al. 1994]). The six-axis tracking hardware is not as robust as would be desired; hybrid trackers combining multiple complementary sensing technologies (such as ultrasound and silicon accelerometers) may eventually stabilize the situation.

The HoloSketch package itself is subject to the same sorts of limitations in what you can draw as any simple 2D system. Without constructive solid geometry (CSG) modeling, there is a limit on what can be built. In the quest to keep things simple the package is artificially constrained in its animation power; physical simulation features (kinematics, constraints, collision detection, more-complex sound operators) would support more-complex animation and virtual worlds.

While much of this article has been concerned with issues of user interface, we have been quite selective in focusing on the novel 3D interface issues. Thus optimal placement of menu buttons, or choice of button colors or names, has been considered secondary for now.

8. FUTURE DIRECTIONS

There are more directions in which HoloSketch could be expanded than we could possibly follow. It is tempting to incorporate a CSG modeler, or constraint-based animation controllers, but we have two paths that we are immediately following up on.

First, we want to see how more novice users react to the system. Can they use it effectively? Is it really intuitive? Is there some real work that they can use it for? To this end we intend to stabilize, harden, and distribute the HoloSketch system and see what happens.

Second, we intend to extend and strengthen the system's animation controls by extending our animation-scripting language. We also want to explore deformation shape interpolation for motion loops. The goal is to use the system to create virtual environments for our virtual reality research in displays and other applications.

9. CONCLUSIONS

A new virtual reality-based sketching system has been described, showing that the simple 2D sketch-draw paradigm can be extended to 3D. The entire system is controlled through the use of a 3D multilevel fade-up pie menu. The system also supports animation through the generalization of grouping operations when elemental animation objects are grouped with static objects. Complex, animated 3D scenes can be constructed within minutes.

ACKNOWLEDGMENTS

The author would like to thank Michael Cox for help on the writing, Scott R. Nelson for comments on both the system and article, and Hallie McConlogue for the art.

REFERENCES

- BUTTERWORTH, J., DAVIDSON, A., HENCH, S., AND OLANO, T. M. 1992. 3DM: A three dimensional modeler using a head-mounted display. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics* (Cambridge, Mass., Mar. 29–Apr. 1). ACM, New York, 135–138.
- DEERING, M. 1992. High resolution virtual reality. In *Proceedings of SIGGRAPH '92* (Anaheim, Calif., Aug. 1–6). ACM, New York, 195–202.
- DEERING, M. 1993. Making virtual reality more real: Experience with the Virtual Portal. In *Proceedings of Graphics Interface '93* (Toronto, Ontario, May 19–21). Canadian Information Processing Society, 219–226.
- DEERING, M. AND NELSON, S. 1993. Leo: A system for cost effective 3D shaded graphics. In *Proceedings of SIGGRAPH '93* (Chicago, Ill., July 26–31). *Comput. Graph.* 26, 2 (July), 101–108.
- DEERING, M., SCHLAPP, S., AND LAVELLE, M. 1994. FBRAM: A new form of memory optimized for 3D graphics. In *Proceedings of SIGGRAPH '94* (Orlando, Fla., July 24–29). ACM, New York, 167–174.
- ELLIOTT, C., SCHECTER, G., YEUNG, R., AND ABI-EZZI, S. 1994. TBAG: A high level framework for interactive, animated 3D graphics applications. In *Proceedings of SIGGRAPH '94* (Orlando, Fla., July 24–29). ACM, New York, 421–434.
- GAYLEAN, T. AND HUGHES, J. 1991. Sculpting: An interactive volumetric modeling technique. In *Proceedings of SIGGRAPH '91* (Las Vegas, Nev., July 28–Aug. 2). ACM, New York, 267–274.
- HALLIDAY, S. AND GREEN, M. 1994. A geometric modeling and animation system for virtual reality. In *Proceedings of VRST '94* (Singapore, Aug. 23–26). World Scientific, Ltd., Singapore, 71–84.
- HANRAHAN, P. AND HAEBERLI, P. 1990. Direct WYSIWYG painting and texturing on 3D shapes. In *Proceedings of SIGGRAPH '90* (Dallas, Tex., Aug. 6–10). *Comput. Graph.* 24, 4 (Aug.), 215–223.
- LIANG, J. AND GREEN, M. 1993. Geometric modeling using six degrees of freedom input devices. In *Proceedings of the 3rd International Conference on CAD and Computer Graphics* (Beijing, China, Aug. 23–26). 217–222.
- SACHS, E., ROBERS, A., AND STOOPS, D. 1991. 3-Draw: A tool for designing 3D shapes. *IEEE Comput. Graph. Appl.* 11 (Nov.), 18–24.
- SNIBBE, S., HERNDON, K. P., ROBBINS, D. C., BROOKSHIRE CONNER, D., AND VAN DAM, A. 1992. Using deformations to explore 3D widget design. In *Proceedings of SIGGRAPH '92* (Chicago, Ill., July 26–31). *Comput. Graph.* 26, 2 (July), 351–352.
- ZELENIK, R., HERNDON, K. P., ROBBINS, D. C., HUANG, N., MEYER, T., PARKER, N., AND HUGHES, J. F. 1993. An interactive 3D toolkit for constructing 3D widgets. In *Proceedings of SIGGRAPH '93* (Anaheim, Calif., Aug. 1–6). ACM, New York, 81–84.

Received November 1994; revised February 1995; accepted March 1995