EMBODIED MODELLING TOOLS IN A 3D ENVIRONMENT

Luis Narváez Porras narvaez@mx1.ibm.com

Isaac Rudomín Goldberg rudomin@campus.cem.itesm.mx

Department of Computer Science.
ITESM Campus Estado de México, Carretera Lago de Guadalupe Km 3.5,
Atizapán de Zaragoza, C.P. 52926, Estado de México,
México

ABSTRACT

Most 3D modelling tools available today operate using interfaces based on traditional static menus, which implies that the user must go through a long training period before he can use the modelling tool in a productive manner. In this paper we explore the definition and use of new embodied modelling tools within a 3D environment. Each embodied tool works in its own virtual 2D space that senses the action and movements of the "mouse" and translates them into operations over the elements of a 3D scene. We develop embodied modelling tools covering the most common operations of a traditional modelling program, including creating, colouring, lighting and editing objects, as well as showing the internal hierarchy of the scene. Lastly, we conclude by describing the characteristics of the modelling tools that have been shown to be useful in improving interaction.

Keywords: computer graphics, human-computer interaction, 3D modelling, user interface design, transparency, sketching.

1. INTRODUCTION

At the present time there are many graphics applications and techniques for creating, modifying and controlling objects within a 3D environment. Interface with the user, however, is still lacking, because the prevailing interaction paradigm was developed for 2D applications, and is not natural for 3D. Application designers must explore new interfaces that allow the end user the creation and modification of 3D objects in a more natural way, where instead of using the traditional keyboard and conventional overloaded menus on screen, editing tools become unobtrusive, self-explanatory, and part of the 3D environment. If this happens, the tools will be immediately useful for user, who will be able to avoid spending a lot of time learning the location of menu entries or keyboard shortcuts.

The flexibility and success of any modeller is based on the friendliness of the interface, and the practicality and flexibility of the tools it contains. It is important to recall that the end user for 3D modelling applications may not be a specialist, yet even for the untrained user the interface must be intuitive. Functionality is not only measured in terms of programming performance, but also by the quality of interaction. In the end, the goal is that as a result of introducing new concepts and interaction techniques, the end user will be able to concentrate in the design and development of his ideas instead of the operation of the modelling program itself.

2. RELATED WORK

Most current windowing systems and graphical user interfaces (GUIs) emerged as an answer to the needs of 2D applications. These windowing systems and related GUIs impose a certain style to the interaction, by using menus. This style is not really natural for 3D. Given the fact that 3D graphics libraries are available on most programming platforms, new GUIs can and must be developed that provide a better fit to interaction within 3D scenes. Several researchers have been working in this direction. One example is Kandogan et. al. [Kando97], who study and create a complete multi-

window operations with depth. They do not focus on 3D scenes.

Effective schemes, which exploit new techniques and concepts, have been proposed by other researchers. Harrison [Harri96] introduces the concept of transparent menus, reflecting the increasing interest in building new ways to interact with 3D models. Kurtenbach [Kurte97] designs a new GUI based on two multi-sensor tablets, two mice and transparent UI components that reflect the way an artist would work with pencil and paper.

Virtual reality technology has the potential of giving the user the opportunity of working in 3D space directly, Stoakley et al [Stoak95] explore a new interface called WIN (Worlds In Miniature), a 3D scene with a hand-held miniature copy of the virtual environment. The key idea in virtual environments is to transport the user into the 3D scene, so that for modelling the objects may be directly manipulated by the user.

3DStudio and Maya [Chris96] represent the state of the art in commercial 3D modelling applications and interface. Still, the interface relies on traditional 2D GUI widgets like buttons, arrows, menu panes, tool bars, multiple windows, etc. that reduce the size of the 3D scene workspace. The buttons appear on the bottom and the top of the windowing system, around the scene as a separate environment. In Maya, the use of the Hotbox and Marking Menus, as well as tools like Artisan and PaintEffects, are interesting steps in the direction of tools that operate within the 3D environment.

Today, there are new I/O input devices, like the Rockin' Mouse [Blakr97] that improve on the traditional mouse, in order to manage four degrees-of-freedom thus allowing the user to manipulate objects in 3D environments. The Responsive Workbench [Krüge95] builds a virtual 3D scene over a table that allows user to manipulate objects, the user wears binocular stereo displays and gloves in both hands over the table. The Virtual Tricorder [Wloka95] proposes the metaphor of using a single tool immerse in the virtual scene using a special hardware device. Unfortunately these new devices are not standard, so they are not practical yet for widespread use.

In our research, we decided to create our interfaces based on the common standard mouse. This is the approach explored by SKETCH [Zeles96] that proposes a complete system for creation and edition of 3D scenes, by defining a 2D interface that uses a conventional mouse as the principal I/O device that is used to draw a 2D sketch which is interpreted as actually constructing a 3D scene. TEDDY [Igara99] proposes a pen-based system for creation of 3D freeform models from 2D sketches.

In this paper we will explore a set of embodied modelling tools within a 3D environment. All tools use multiple two-dimensional properties to go back and forth from 2D working areas (embedded in the 3D scene) to 3D operations over the objects of a 3D scene. The idea is to allow users to edit the scene avoiding the barriers posed by conventional 2D GUIs. We also explore the idea of using 2D sketches for creating 3D models. Our prototype implementation is a 3D modeller called MOD3D [Narva99], which has been programmed using VRML and Java. The intention is to have a proof_of_concept modeller rather than to compete with commercial modellers on any particular tool.

3. DESIGN AND OBJETIVES

Our research focus is on techniques for creating modelling tools with the following goals:

1. **Ease of use**
   The interface is intended for use by persons without computer skills and non-professionals. All tools only have two states: active / inactive. The user can swap the state by clicking on the tools with the mouse.

2. **Tools work on an imaginary plane**
   The tools can only be moved on an imaginary 2D plane in front of the scene, allowing the user to move them so they are not obtrusive, yet remain available no matter what is the viewpoint of the user.

3. **Increase the amount of screen-space available for the 3D scene**
   Each tool can be minimised to an icon, thereby maximising the proportion of the screen to be used for rendering the actual scene, and making the tools less obtrusive.

4. **Transparent working areas**
   Transparency avoids obstructing the 3D scene and is useful in achieving integration of the tool space and the workspace.

5. **Direct object manipulation**
   Each object in the scene has special sensors that perceive any interaction when the mouse is over and makes contact with any visible part of the object.

The set of embodied modelling tools can be divided into two classes:

- **Scene Control**
  Objective: Control of movements through 3D space and determining the position and orientation of the objects.

- **Scene Edition**
  Objective: Create, modify, delete and manage the geometric objects of the scene.

Each tool has particular objectives and special operation modes. Table 1 shows the complete list of embodied modelling tools and their particular functions. We will go through of these tools in detail.

MODELLING TOOL	FUNCTION
1. Drawing Pencil	Create 3D objects.
2. Editor	Edition of geometric properties: a) Position b) Orientation c) Scale d) Deletion e) Undo f) Groups
3. Hierarchical tree	Present a graphical organisation of the scene in as a tree.
4. Colour palette	Change colour of objects.
5. Lights	Create shadows for better visualisation.

Functions of embodied modelling tools.
Table 1

4. DRAWING PENCIL

This tool (actually a tool set) was designed as a movable interface that allows the user to draw traces with the help of a pencil over a 2D transparent surface, much in the way artists write and draw with a marker on a (transparent) blackboard. The drawing pencil tool includes several elements. Each one performs a simple but effective operation. The tool can be moved over an imaginary plane in front of the scene. The interface has two states:

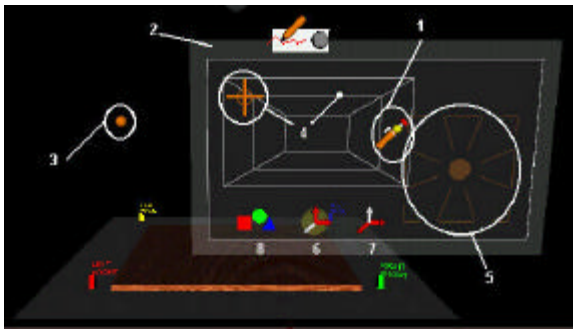
a) Active: All elements are present. The elements are ready for working within the tool.

b) Inactive: No elements are shown. The trace is not present but is not erased, remains hidden.

The user can swap the state of the interface with the help of the mouse. Table 2 lists the elements of the drawing tool and Fig. 1 shows the interface.

ELEMENT	FUNCTION
1. Pencil	Draw traces
2. Blackboard	Mobile surface where the user draws
3. Point3D	Position of new object
4. Room2D	Reset point3D to initial position
5. Guide	Template for trace
6. Axes X-Y	Motion in plane XY
7. Axes X-Z	Motion in plane XZ
8. Action	Creation of geometric object

Elements of Drawing Pencil tool.
Table 2



Drawing Pencil tool. See Table 2 for list elements.
Figure 1

The blackboard is the base structure used to draw with the pencil. The pencil has two states: active/inactive. It is used to draw traces over the blackboard. In any state the pencil can be moved only in 2D, over the blackboard. In active state a white trace appears following the tip of the pencil, like a pen over paper. The size of the blackboard is not the limit of the drawing surface; the pencil can draw outside it. However there is a limit on the pencil's movement in order to avoid loosing it. Fig. 2 shows examples of different traces.

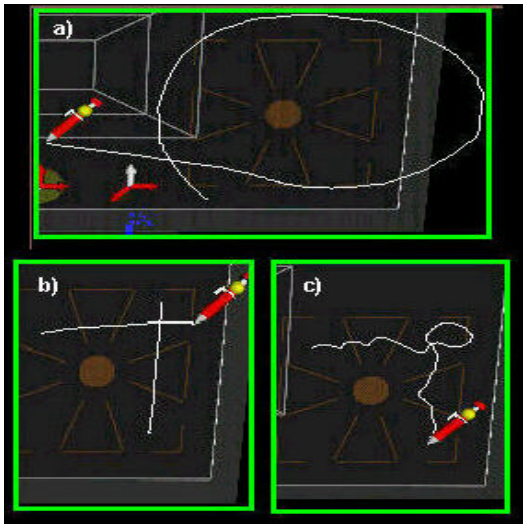
Our research explores the 2D interface as a way to express 3D operations and objects. Other papers such as [Zelez96] and [Igara99] focus on a 2D-sketching interface for designing models in a 3D space. We developed this drawing pencil interface to include a trace analysis algorithm, in order to create 3D objects from a 2D trace. This is explained in more detail in a previous paper [Narva99]. Without going into details here, the traces are analysed depending on the

presence/absence of the trace in each zone of the guide (shown in brown in Fig. 2). As a result of the analysis, the interface creates a 3D object in the 3D scene. For instance, In Fig. 2a) a sphere is created and in 2b) a cube is inserted in the scene.

This tool-set is actually not the first version we developed. Initially, we used the blackboard metaphor as a transparent window covering the whole scene. It turned out that:

- The tool-set was very obtrusive even if it could be iconised. It had to be made movable.
- Adding the guide aided the user in tracing but made the tracing a separate action from adding the 3D object.

In order to improve the interface, we are exploring the possibility of setting the location of the guide according to pencil's position when the user activates it. As a result, the user would use the pencil to create and mark the position of the new object at the same time.



Examples of different traces. a)one trace b) two traces c) irregular trace.
Figure 2

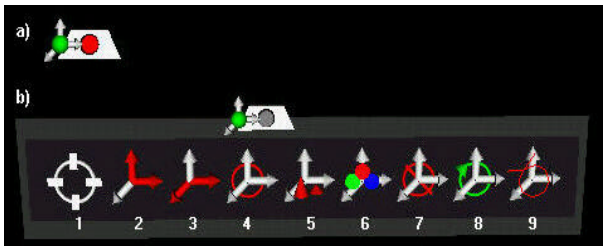
5. EDITOR

This tool was designed to allow changing the geometric properties of the scene's elements by operating directly on the element itself. The user selects the object with the mouse and by dragging it; the computer senses the movements of the mouse and transform this data to new commands and geometric values.

The modelling tool in its active state shows the editing operations available to the user. Table 3 contains the basic elements of the editor tool and their function. As in the colour palette the elements of the tool are only present when it is in active state. Fig. 3 shows the edition modelling tool in 3a) inactive and 3b) in active state.

ELEMENT	FUNCTION
1. Selector	Selection of objects
2. Axes X-Y	Motion in XY plane
3. Axes X-Z	Motion in XZ plane
4. Rotation	Rotate objects
5. Scale	Increase/Decrease the scale
6. Colour	Change colour of objects
7. Erase	Delete objects
8. Undo	Restore objects
9. Collector	Build groups of objects

Elements of edition modelling tool.
Table 3



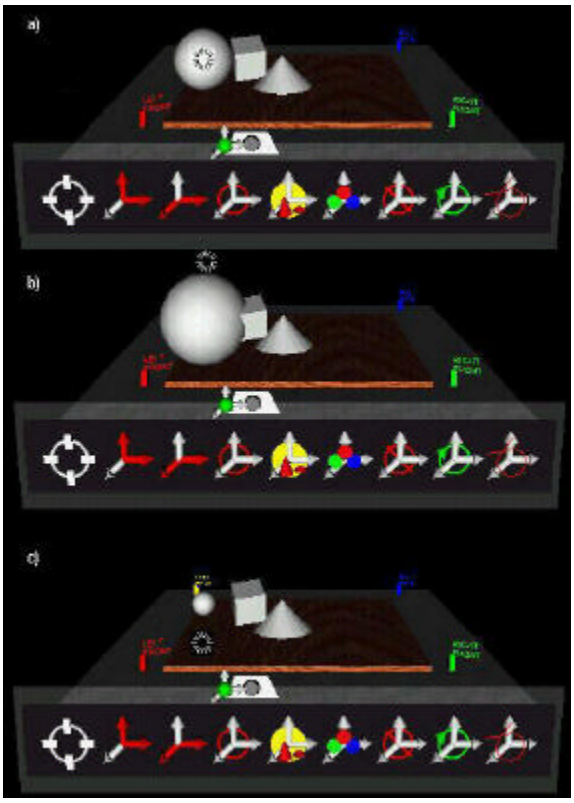
Edition modelling tool a) inactive b) active.
See Table 3 for list of elements.
Figure 3

The edition process involves three phases:

- Phase I: Operation selection.
  The user chooses the icon of the operation that he wants on the edition tool with the mouse.
  Visual Feedback:
  A small yellow circle appears behind the selected icon.
- Phase II: Object selection.
  The user selects the object on the scene directly.
  Visual Feedback:
  The transparency of the object selected changes, so the user can identify it easily.
- Phase III: Modify geometry.
  Change the value of geometric properties by dragging the mouse.
  Visual Feedback:

The object changes its geometric parameter in proportion of the mouse movement, until the user releases the mouse's button.

The direction of the mouse's movement during the phase III depends of the operation selected, for example: if the user decided to edit the object's scale, by dragging the mouse upwards, he increases the scale, while dragging downwards decreases the scale of the object. In Fig. 4 the interaction process for changing the scale of an object is illustrated, the small star represents the mouse's position over the sphere.



Interaction process in scale transform: a)original size
(mouse over object) b) increase (mouse moves up)
c) decrease scale (mouse moves down).
Figure 4

In some cases, it's not necessary to complete the three phases, for example: if the user wants to delete an object, only phase I and II are needed to complete the process. Table 4 lists the type of mouse movements for each edition element.

This tool is essentially a movable, transparent and iconisable graphical menu that allows for directs interaction with the 3D objects. The main advantage is therefore allowing for a familiar type of interaction, but having the menu be unobtrusive yet available.

Working into a 3D space, the user can navigate and select the objects directly. Therefore the role of the mouse is not only a classic "point to" device, but its movements can be express actions or commands. In addition, each object can manage several classes of sensors in order to response in different ways at the same type of 2D mouse's movements.

ELEMENT	DIRECTION	ACTION
Selector	Point to object	Select object
Axes XY	Up Down Right Left	Move in XY plane Coordinate (x,y,z) Up increase y Down decrease y Right increase x Left decrease x
Axes XZ	Up Down Right Left	Move in XZ plane Coordinate (x,y,z) Up increase z Down decrease z Right increase y Left decrease y
Rotation	Up Down Right Left	Rotate the object around its centre
Scale	Up Down	Increase scale Decrease scale
Colour	Point to object	Change colour
Erase	Point to object	Delete object
Undo	No Application	Restore object
Collector	Point to objects	Create group

Relationship between mouse's direction and edition
Table 4

6. HIERARCHICAL TREE

This tool describes the status of the internal organisation of the scene's elements. This status is presented by means of a dynamic graphical tree that contains the relationship of the individual objects and groups. There are three kinds of elements on the tree:

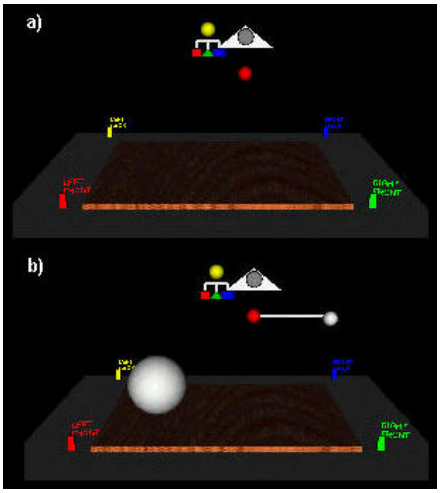
a) Root.
Marks the start of the tree.

b) Individual object node.

Miniaturised versions of the scene objects are placed on the tree, copying the colour and type of geometric primitive: cone, cube, sphere, etc.

c) Group node.
A special node (a cube inserted into a sphere) represents the existence of a group in the scene. A group is a set of objects that the end user collects and defines for his particular needs.

Initially the tree is empty. When the user creates new objects, they are also inserted at the root of the graphic tree. Fig. 5a) shows the scene without objects and in 5b) the user creates an object, so one node is inserted into the tree. By default, the tree grows to the right of the root node. A long and white cylinder represents each level of the tree. As was the case with the tools previously described, the graphical tree can be present as an icon or it can be expanded showing its elements.



Graphical trees a) empty b) one object.
Figure 5

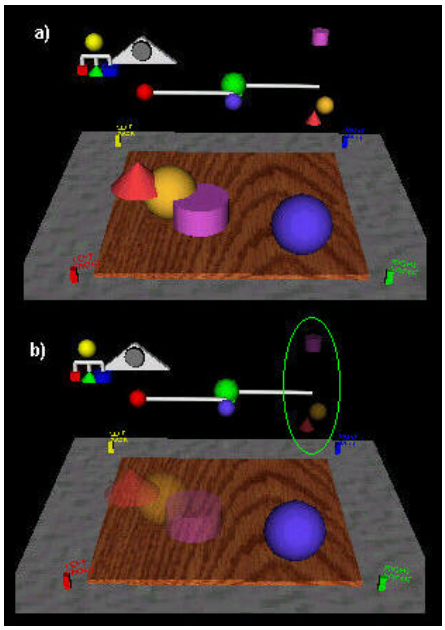
The node, which represents a group, has two states:

a) Open: All elements of the group are shown.
Colour: green group node.

b) Closed: No elements of the group are shown.
Colour: red group node.

Each node of the tree has a sensor in order to define a dynamic behaviour, so the user can select the object directly from the scene or he can select the same object using the graphical structure by selecting the node of the tree. Fig. 6 illustrates this case; in 6a) the original scene with one group (cone, sphere and cylinder) and one individual object (blue sphere) and in Fig. 6b) the user selected the group. The user decides which group is active or not active with the help of the mouse. The transparency of the elements in the tree and in the scene changes when selected by the user, therefore, the user can visualise the object and then apply some edition operation over the set of selected objects (groups or individual elements). The objects will return to their original transparency after either the user deselects them or the edition operation is completed.

This tool is nice because it is a structural view of the scene, but it is within the scene, making it available, yet unobtrusive because it can be iconised or opened partially as needed. It is a familiar tool but it is more accessible because it is within reach in the 3D workspace.



Graphical tree: a) original scene b) select the group.
Figure 6

A colour and light tool were also added for completeness, but will not be discussed further.

7. EVALUATION AND CONCLUSION

We included a tutorial section and quick reference guide to the modeller in order to teach how to use the system. We selected a small group of 15 people technical untrained and invited them to test it. The range of age was 12 to 40. After testing each interface the volunteers filled an evaluation form. The main results can be summarised as follows:

- The Editor panel is the easiest interface to use.

- The Drawing pencil is useful to create objects because the pencil's trace has not to be exact, so, the user can draw in a friendly way.

- At the beginning, the Hierarchical tree is the most difficult interface to understand, but when the users collect objects into groups they begin to use it immediately

- As a result of the user's survey in the colour palette one of the easier interfaces to understand, there is a suggestion to add a special section to be able of merging colours. We are considering make a new interface tool based on HSV model.

- The idea of including 2D autonomous tools in a 3D scene was accepted as natural and useful.

The embodied modelling tools explored in the research described in this paper give the user who is immersed in a 3D space a more intuitive interface, since it is a 2D interface embedded in a 3D world. Each tool has specialised functions that translate 2D mouse movements into 3D operations. Using this approach we improve the user's attention to his artwork, by integrating the modelling tools with the task space. This allows us to appreciate a dynamic interaction between the user and the scene.

Traditional 2D GUIs were not planned for 3D graphic operations; thus graphics interfaces usually concentrate all the options and buttons in one big window like in word processors and worksheets. It is our belief that the new 3D interfaces must be divided into specific classes, in order to maximise the screen-space available for the artwork and to take advantage of virtual environments that allow us the opportunity to "live" in the 3D space. The tools we explored in this paper are movable, can be iconised and are transparent, all with the effect of making the tools available at all times yet unobtrusive.

Each of the 3D modelling tools has been designed to be autonomous from the others. This means that each one has its internal processes, rules and features that make sense for a particular set of geometric values (scale, size, etc) and allow us to work with other elements: colour, light, audio, etc. as required by each.

VRML and Java present a robust and quick development environment for graphics prototypes for testing these new kinds of modelling tools.

8. FUTURE WORK

We must refine all the tools to make them more intuitive and designs new evaluation tests in order to evaluate the interface objectively for groups of user of different abilities. We also need to improve the hierarchical tree in order to manage large set of objects in scene and include links into the 2D interface to explain the VRML syntax for tutorial proposes. Some other possible work is listed:

a) Create an interface to edit text within 3D space.
b) Create 3D interface for audio and video control.
c) Show drawing guide based on the pencil's position when the user actives the pencil.
d) Add a HSV model tool to colour palette.

REFERENCES

[Balak97] Ravin Balakrishnan, Thomas Baudel, Gordon Kurtenbach, George Fitzmaurice: The Rockin'Mouse: Integral 3D Manipulation on a Plane. *Proceeding of CHI' 97*, Atlanta Georgia, 1997.

[Chris96] Steve Christon, Deion Green, Bod Gundu, Carla Sharkey, *Learning Maya 2,* Alias Wavefront Education, 1996.

[Dougl96] Sarah Douglas and Ted Kirkpatrick. *Do color models really make a difference ?.* Proceedings of CHI'96, New York, NY, 1996.

[Foley90] Foley, J., van Dam, A., Feiner, S. and Hughes, J. *Computer Graphics: Principles and Practice, 2nd ed. Addison Wesley,* Reading, MA, 1990.

[Harri96] Beverly L. Harrison, Kim J. Vicente. An experimental evaluation of transparent menu usage. *Proceedings of CHI'96,* New York, NY,1996.

[Hearn97] Hearn, D., Baker, M. Pauline,. *Computer Graphics C version, Prentice Hall,* 572-573, 1997.

[Kando97] Eser Kandogan, and Ben Shneiderman. Elastic Windows: Evaluation of Multi-Window Operations. *Proceedings of CHI-97,* Atlanta Georgia, 1997.

[Igara99] Igarashi, T., Matsuoka, S., Tanaka, H. Teddy: A Sketching Interface for 3D

Freeform Design, *ACM SIGGRAPH 99 Conference Proceedings*, 1999.

[Krüge95] Wolfgang Krüger, Christina-A.Bohn, Bernd Fröhlich and Gerold Wesche. The responsive workbench: A virtual work environment, *IEEE Computer*, pages 42-48, July 1995.

[Kurte97] Gordon Kuterbach, George Fitzmaurice, Thomas Baudel, Bill Buxton. The Design of a GUI Paradigm based on Tablets, Two-hands, and Transparency. *Proceedings of CHI'97*, Atlanta Georgia,1997.

[Narva99] Luis Narvaez, Isaac Rudomín. MOD3D Modelador 3D basado en trazos 2D. *ENC'99, Segundo Encuentro Nacional de Computación*, Pachuca, Hidalgo. 1999. ISBN: 968-6254-46-3

[Stoak95] Richard Stoakley, MatthewJ.Conway, Randy Pausch: Virtual Reality on a WIM: Interactive Worlds in Miniature. *Proceedings of CHI'95*, Denver Colorado, 1995.

[Zelez96] R. Zeleznik, J. Hughes: SKETCH: An Interface for Sketching 3D Scenes, *Computer Graphics (SIGGRAPH '96 Proceedings)*, pp.163-170, August 1996.

[Wloka95] Matthias M. Wloka and Eliot Greenfield. The Virtual Tricorder: A uniform interface for virtual reality, *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 39-40, 1995.