Metaphors for visualization

Alex Pang and Michael Clifton

Computer and Information Sciences Board
University of California, Santa Cruz
California, 95064, USA

Abstract

This paper investigates the use of several common objects such as spray cans, flashlights, carving knives, and others as metaphors for visualization. The motivation behind this work is to provide an intuitive and natural 3D interface such that users will view these objects as tools during the visualization process. To help achieve this goal, the selection of the object tools are derived from expressions commonly used during visualization. For example, "let's paint this (iso)surface red"; "cut away the front part of that volume"; "look at it from this angle"; "you get similar effects from an x-ray photo"; and "let's explore this data set". In this paper, we also extend the metaphor of cutting planes to allow users to carve non-planar cross-sectional cuts through their data sets. This extension will directly benefit applications such as medical visualization where one might want to generate curved coronal cross-sections of lumbar spines; and in oceanography where one might want to compare numerical model output against data obtained along non-planar ship tracks.

1 Introduction

The goal of visualization is to present data in such a way as to facilitate its understanding. The amount and ease in which insight is gained from the data set is an indication of the success of the visualization process. Equally as important as the development of powerful visualization methods is the development of techniques and interface methods that make these powerful tools available to the users. The tools should be made accessible in an unencumbered, direct, and natural way – first to gain acceptance from the users, and second to promote unimpeded investigation of their data without having to think about the visualization methods. This is not to say that the users do not need to be aware of the limitations of the methods, but rather the interface should be so natural to use that it is almost invisible to the users. In this case, the measure of an interface's success is its apparent absence from the user's thought process while investigating their data set.

Our approach at achieving such an interface is to take advantage of common adages or metaphors used in discussions when examining data sets. These metaphors are then translated to common day tools and devices for the users as interaction handles. Thus, users are provided with visualization tools to directly manipulate when examining their data. These tools are identified from colloquialisms such as: "let's paint this (iso)surface red"; "cut away the front part of that volume"; "look at it from this angle"; "you get similar effects from an x-ray photo"; and "let's explore this data set". Clearly, from the visualization users' point of view, they are not thinking in terms of the feature extraction or rendering algorithm but rather the physical manipulation needed to get the desired effect. We can enhance this train of thought by providing users with virtual 3D widgets that they can directly manipulate and which closely correspond to their mental model of manipulating their data sets. In the above examples, the virtual 3D widgets might respectively correspond to a spray can or the low tech paint brush; butcher knife or carving knife; a camera; a portable x-ray camera or perhaps a space age laser gun of some sort; exploration may conjure groping around in the dark. So, we can provide a virtual flashlight.

Since these metaphors are, in almost all cases, 3D objects, it is natural to implement these with virtual reality input devices (gloves and trackers, at a minimum). The 3D widgets that correspond to these objects differ from those employed in most applications of virtual reality technology to scientific visualization in their means of interaction. The goal is to design the interaction with the 3D widgets in such a way that the user will not have to consciously think about how to interact with the widget. This frees up the mental process of the user to concentrate on data analyses instead of keeping track of how to interact with the widget in order to get to the data. This objective is also in line with the human-computer software interface recommendations by [1] and the scenarios of use approach by [2]. The added level of indirection that users have to deal with is surprisingly common in application of virtual reality technology to scientific visualization applications. For example, one of the 3D widgets [3, 4] presented was a floating rectangle where the corners, edges and the entire rectangle were grabbable. The position, size and orientation of this rectangle was then controlled by grabbing on to the different parts of the rectangle and manipulating them. While the different components of the rectangle are highlighted to alert the user that they are grabbable, the user still has to think in terms of grabbing corners or edges of the rectangle and manipulating those to get to the data. Thus, there is an added level of interaction for the user to deal with when trying to visualize their data sets. In contrast, the 3D widgets in this paper rely on the users' experience with common everyday objects. Thus, control and manipulation of these 3D widgets can be achieved in a natural and unobtrusive way. A simple example, based on [5], is the use of a treadmill/bicycle metaphor to allow the user to walk/bike around the data set. In this scenario, the bike handles are used to control the direction of travel and view while the leg movements control the speed. This type of metaphor allows the user to feel very comfortable and concentrate on looking at the data as opposed to consciously interacting with the widgets to get at the data.

2 Previous/Ongoing work

There are several notable application of virtual reality interfaces to aid scientific visualization. One that has received significant amounts of attention is the virtual wind tunnel experiment [6]. This is an immersive environment where the user can move about and examine flow fields by simply positioning a rake that generates streamlines. The naturalness of directly positioning a tool in the neighborhood of the data where visualization is needed, coupled with the interactive rates that this is achieved are the key ingredients of success in their work. The utility of these virtual reality interfaces can be greatly enhanced when combined with the power of using metaphors that facilitate their use. The recognition of gestures associated with some of these metaphors have also been investigated by [7].

The interactions associated with the metaphors that we are proposing will build upon and extend the conceptual design of spray rendering [8, 9]. Spray rendering uses the metaphor of providing the users with a shelf of spray cans, each of which can generate a different way of visualizing a data set. For example, a spray can might generate an iso-surface in a scalar field, while another spray can might generate streamlines in a flow field. An attractive feature of spray rendering is that most users have at one time or another operated an aerosol can. Hence, learning how to use the system is quite intuitive.

Internally, spray rendering combines particle systems [10] and behavioral animation [11]. Particle systems were originally designed for modeling objects that were difficult to model using traditional techniques but have also been used in visualization applications such as particle aging and tracing [12]. In these applications the particles were usually passive and were simply advected by the flow field. We incorporate behavioral animation strategies with these particles so that they may interact with other particles as well as with the data field that they are traveling through. These smart particles (sparts) have an endowed intelligence which can be broken down into roughly two parts: targets and behaviors. Targets are features, usually in the data set, that the spart is programmed to seek out. As these targets are found, the sparts are made visible according to how their behaviors are defined. For example, the marching cube [13] algorithm may be modified so that several independent sparts are looking for the iso-surface rather than have the marching cube algorithm process the entire volume data. In this case, the iso-surface sparts would be seeking for surfaces of the specified threshold value. Once these surfaces are found, the spart would leave a polygon which will then be sent to the renderer. It is also possible for the spart to leave behind a non-visible marker for communication with other cooperating sparts. One example where sparts work in pairs is the flow tracking ribbon sparts [14]. Using this framework, we have been able to modify several existing visualization techniques and also introduce some new ones [15], as well as extend it to a collaborative environment where multiple geographically distributed users can share spray cans and visualizations within a shared virtual workspace [16].

3 Approach

We are investigating the effectiveness of several metaphors for visualization that allow users to directly manipulate and generate visualization through interactions with VR devices. These metaphors include but are not limited to: spray cans, flashlights, and kitchen knives. Based on the chosen metaphor, the operations are then mapped to the VR devices so that the appropriate hand gestures and postures are interpreted properly. Outlined below are some of the preliminary postures and gestures associated with each metaphor and how they adjust the relevant parameters. The same postures and gestures would apply to the left hand as well as to the right hand. Note that while most VR implementations try to achieve immersion, this is not a requirement for the initial phase of this work. The same metaphors may later be applied to immersive or augmented environments.

3.1 Spray can

The figures below use some objects as props notably a $2 (dish washing) "cyber" glove to illustrate the different postures and gestures associated with the use of each metaphorical object.

It is important to note that this metaphor allows the user to pretend that s/he is holding the can (as in Fig. 2). This hand posture can then be mapped to control the different parameters of the virtual spray can.

Default posture: the can is held with the index finger on the nozzle ready to fire.

Pointing posture: the direction of spray is calculated based on the index finger orientation. It is roughly 45 degrees down from the first flexion joint of the index finger.

Firing gesture: the amount of index finger flexion is mapped to the number of sparts that are being sprayed. The sparts are sprayed continuously while the index finger is flexed.

Cone adjustment gesture: nozzle size can be increase/decreased by the amount of abduction between the thumb and the index finger.

Figure 1: Using the spray can.

Figure 2: Hand posture for holding can.

3.2 Flashlight

While the shape of the spray can has remained pretty universal, the flashlight has undergone more shape changes. Let's look at the traditional flashlight and the more modern design with a separate handle. Both require the user to hold the flashlight or the handle with 4 fingers clasp around it and the thumb poised over the on/off switch and pointing in the direction of the flashlight.

Pointing posture: controlled by thumb direction.

Firing gesture: on/off and the amount of sparts (or photons) to shoot is controlled by the thumb flexion.

Parameter adjustment postures: Since we are mapping non-flashlight parameters, such as beam shape and spread angle, the mapping here is not as natural. One possibility is to control beam shape and amount of spread angle by the amount of finger flexion.

Figure 3: Using the flashlight.

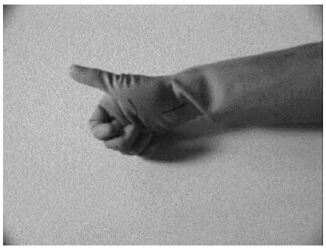

Figure 4: Posture for holding flashlight.

3.3 Knife

This metaphor allows the user to slice and dice their data sets in a natural and direct way. However, it also presents some interesting issues such as segmentation of hand gestures as pointed out in [17]. It also requires us to modify our local contouring algorithm to hand-carved surfaces. Since the hand posture for holding a knife is quite similar to holding a flashlight, we modified the hand posture to a karate chop instead. This way, the user can pretend that his/her hand is the knife itself with the cutting edge extending from base of the palm to the end of the small finger. Unwanted pieces of the sliced data may be successively removed by simply flicking the thumb. There are two ways of using the knife:

Traditional: This includes planar slicing, sweeping and rotation of the knife. In this case, the palm defines the cutting plane. The direction of the chop is defined by the forward movement with the cutting edge as the leading edge. A new cut is initiated when the user first moves the hand back and then moving forward again with the cutting edge. The cutting plane can be swept around the volume when the hand movement is normal to the cutting plane. Wrist movements control plane orientations.

Carving: Non-planar cross-sections can be defined in two ways:

1. Non-planar 3D ruled surfaces can be defined by simply moving the cutting edge along some curve while keeping the 4 fingers straight. This may be applied to taking curved coronal slices of cat-scan data or to curved slices around structures in CFD data.

2. Non-planar 3D curved surfaces can also be generated in a similar manner by allowing the user to flex the 4 fingers. The amount of finger flexion would define the amount of surface curvature.

Figure 5: Using the "Ginzu" knife.

Figure 6: "The Chop". Hand posture to represent virtual knife.

3.4 Helmet/Eyeball-in-hand

The three metaphors listed above allow the user to manipulate the data set directly. We also need a mechanism for the user to navigate around the data set. There are several options here. Since our application is scientific visualization as opposed to creating virtual worlds, "flying" around the data set is not a necessity. What the user needs is a way to view the data from different positions and orientations; perhaps from within the data set as well. The metaphor we chose to adopt here is the eyeball-in-hand as proposed by [18]. This can be simply modified to an eyeball or flashlight mounted on a spelunker's helmet when we move to an immersive environment. In the meantime, the user uses one hand to manipulate the data and the other hand to control our virtual (tennis ball) eyeball.

Gaze posture: the hand position is mapped to the eye position; while the finger direction is mapped to the gaze direction. The middle finger provides the up vector.

New view: can be obtained by simply moving the hand and adjusting the wrist to the new gaze direction.

Note that aside from the technical aspects of gesture recognition, one must also be sensitive to the social implications of certain hand postures and gestures. For example, the okay (thumb forming a circle with the index finger) or the good luck (thumbs up) symbol may mean different things to different culture; much like the American way of waving goodbye may be construed to mean a beckoning

Figure 7: Holding the eyeball.

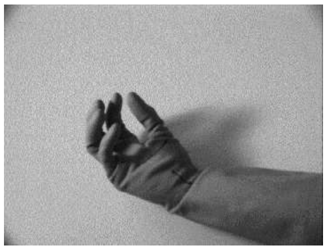

Figure 8: Hand posture for holding virtual eyeball.

gesture in other cultures [19]. Since one of the broader goals of spray rendering is collaborative visualization among scientists who are geographically distributed and potentially with different cultural background, it is important to take these subtleties into consideration.

4 Implementation

We have concentrated our initial efforts at implementing the knife as a metaphor for visualization. The instrumentats used in this work are: an 18 sensor Cyberglove, two 6D Ascension trackers, and Crystal Eyes stereo glasses.

Currently, we have implemented four methods for interacting with the data. First, the user can point the index finger to a position in space to get a reading of the data at that location. Second, cutting planes aligned with the principal planes can be obtained if the user forms a fist and moves it forward/backward, left/right, or up/down. Third, arbitrary cutting planes can be obtained if the user makes a flat posture with the hand as in a karate chop. An arbitrary cutting plane will then follow the position and orientation of the user's hand. Finally, if the user points with both the index and middle fingers, these fingers begin to act as a carving knife. The carving knife can be swept along a curve in space, generating a ruled surface. On this surface will appear a curved cross-section of the data set.

For these various forms of manipulation, the first thing that is needed is a way to recognize which hand posture the user is engaged in. To do this, an 18-value vector containing the joint angles from the glove is recorded. This vector is normalized to the "standard hand" by means of offsets and scales for each joint. This normalization ensures that for users with differently shaped hands, the normalized sensor readings for similar hand positions will have similar values. Next, the dot product is taken between this vector and each of the recognizable postures. The closest match is determined to be the current posture, unless the user's hand does not match any of the postures in which case the user is assumed to not be interacting with the data.

Once the current posture is determined, the system can activate the appropriate tool. For the flat cutting plane, the position and orientation of the plane are determined by measuring the 6D tracker attached to the user's wrist. From this, a plane can be placed in the correct location in the data set. The plane is triangulated adaptively so that the cross-sectional data values at the vertices of any triangle will not vary beyond some user-specified tolerance. Then the triangles are drawn using Gouraud shading hardware, assigning a color to each vertex from a color map that represents the range of values of the data.

The carving knife tool works similarly to the flat cutting plane, except that instead of drawing a single plane that follows the user's hand, a sequence of rectangles are drawn along the sweep of the hand. Each of the rectangles corresponds to the location of the virtual blade at a previous point in time. When the user moves his/her hand one blade's width away from the last rectangle, another rectangle is created to line up with the last. These rectangles are never removed until the user changes to another hand posture, at which time they all disappear.

Another good thing about letting the user manipulate which portion of the data set is of interest is that it prevents the program from having to compute and display the entire data set. In addition, this could also be used to cut down a huge data set into user-controlled local slices, allowing the user to interactively examine data that was previously too large and too slow to draw.

5 Results

To illustrate the results of the knife metaphor, we obtain planar and curved cuts from two different data sets. Each data set is a 20x20x20 regular grid. The first one contains a toroidal shape, while the second one contains a head shape. The shapes are of varying thickness and may contain holes. That is, the models do not use infinitely thin surfaces. Instead, they use walls that have thickness – that makes the head have a thickness yet have a hollow center. The grids also have distance values which refer to the distance from the walls. Yellow maps to zero distance (on either side of the wall), blue maps to negative distances (inside the wall), and red maps to positive distances (outside the wall – which may be inside or outside the object as a whole), with a smooth blend between colors.

6 Conclusion

We have presented some early results in our work on using common objects as metaphors for visualization. In particular, the use of knives to specify cutting planes have been extended to non-planar cuts. Aside from providing this extra degree of freedom in specifying regions of interest, we believe the work shows strong promise of these metaphors to aid the visualizer. Interaction is natural, and the user is not encumbered by having to remember awkward gestures; nor have to indirectly interact through intermediate handles.

References

[1] Gary Bishop et al. Research directions in virtual environments: Report of an NSF invitational workshop. In *Computer Graphics*, pages 153–177, August 1992.

[2] John M. Carroll and Mary Beth Rosson. Putting metaphors to work. In *Graphics Interface '94*, pages 112–119, 1994.

[3] Tom Meyer and Al Globus. Direct manipulation of isosurfaces and cutting planes in virtual environments. Technical report, Dept. of Computer Science, 1993. Technical Report 93-54.

[4] Robert C. Zeleznik et al. An interactive 3d toolkit for constructing 3d widgets. In *Computer Graphics*, pages 81–84, 1993.

[5] Henry Fuchs. Siggraph '91 Demo, 1991.

[6] Steve Bryson and Creon Levit. The Virtual Windtunnel: An environment for the exploration of three-dimensional unsteady flows. In *Proceedings: Visualization '91*, pages 17 – 24. IEEE Computer Society, 1991.

[7] David J. Sturman. *Whole-hand Input*. PhD thesis, Massachusetts Institute of Technology, 1992.

[8] Alex Pang, Naim Alper, Jeff Furman, and Jiahua Wang. Design issues of spray rendering. In *Proceedings: Compugraphics '93*, pages 58 – 67, 1993.

[9] Alex Pang. Spray rendering. *IEEE Computer Graphics and Applications*, 14(5):57 – 63, 1994.

[10] William Reeves. Particle systems: A technique for modelling a class of fuzzy objects. In *Computer Graphics*, pages 359–376, 1983.

[11] Craig Reynolds. Flocks, herds, and schools: A distributed behavioral model. In *Computer Graphics*, pages 25–34, 1987.

[12] Johan Stolk and Jarke J. van Wijk. Surface-particles for 3d flow visualization. In *Proceedings Second Eurographics Workshop on Visualization in Scientific Computing*, 1991.

[13] William Lorensen and Harvey Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *Computer Graphics*, pages 163–169, 1987.

[14] J. P. M. Hultquist. Constructing stream surfaces in steady 3d vector fields. In *Proceedings: Visualization '92*, pages 171 – 177. IEEE Computer Society, 1992.

[15] Alex Pang and Naim Alper. Bump mapped vector fields. In *SPIE & IS&T Conference Proceedings on Electronic Imaging: Visual Data Exploration and Analysis*. SPIE, Feb 1995.

[16] Alex Pang, Craig M. Wittenbrink, and Tom Goodman. CSpray: A collaborative scientific visualization application. In *Proceedings SPIE IS &T's Conference Proceedings on Electronic Imaging: Multimedia Computing and Networking*, Feb 1995.

[17] Thomas Baudel and Michel Beaudouin-Lafon. Charade: Remote control of objects using free-hand gestures. *Communications of the ACM*, 36(7):28–35, July 1993.

[18] Colin Ware and Steven Osborne. Exploration and virtual camera control in virtual three dimensional environments. In *Computer Graphics*, pages 175–183, March 1990.

[19] Roger E. Axtell. *Gestures: The Do's and Taboos of Body Language Around the World*. John Wiley & Sons, 1991.

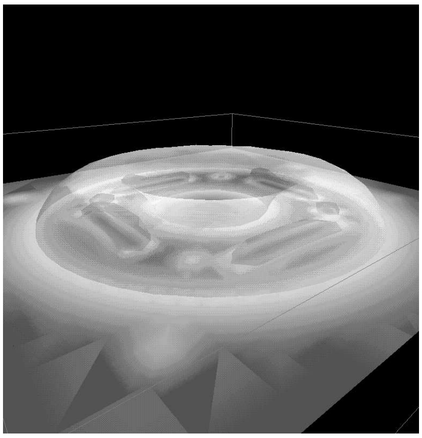

Figure 9: Horizontal planar cut through torus.

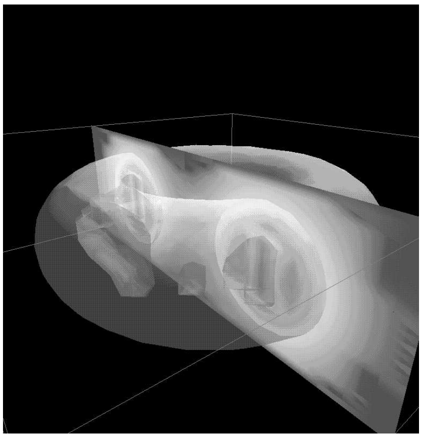

Figure 10: Arbitrary planar cut.

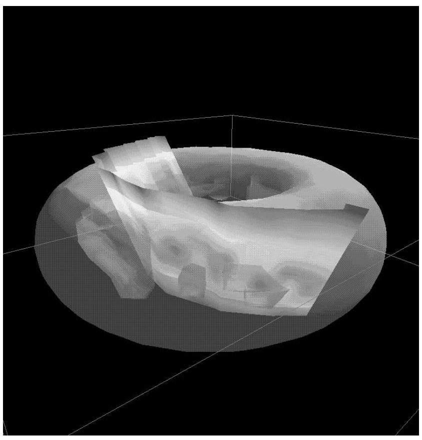

Figure 11: Curved cut through torus.

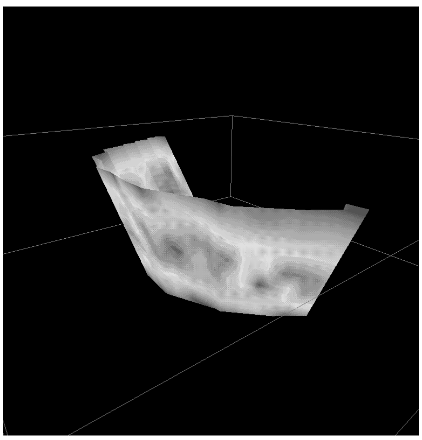

Figure 12: Curved cut alone.

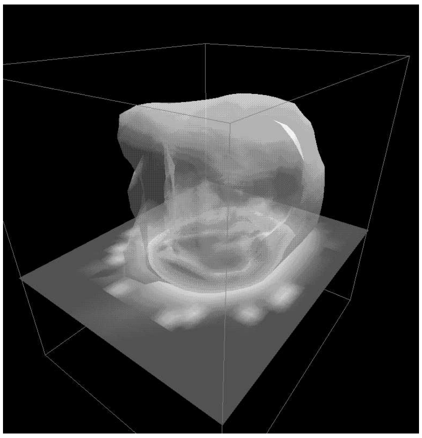

Figure 13: Horizontal planar cut through head.

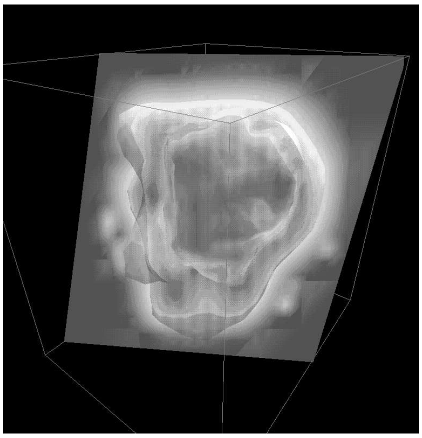

Figure 14: Arbitrary planar cut.

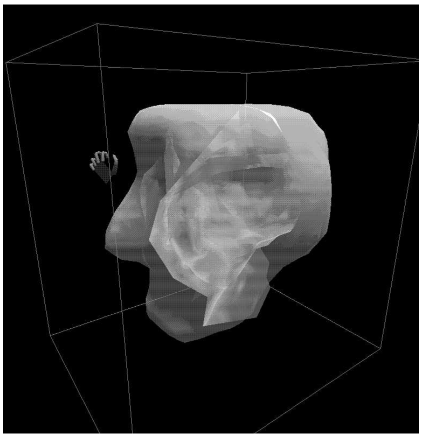

Figure 15: Sweeping curved cut through head.

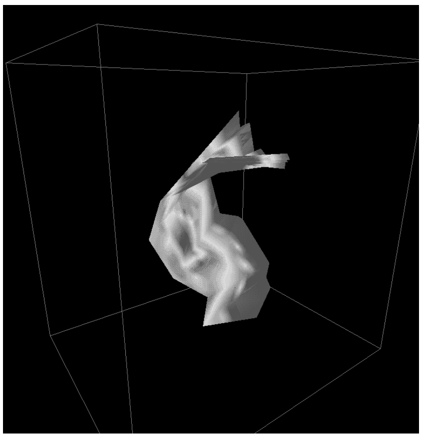

Figure 16: Curved cut alone.