3D Widgets for Exploratory Scientific Visualization

Kenneth P. Herndon and Tom Meyer

Brown University
Department of Computer Science
Providence, RI 02912
(401) 863-7693; {kph,twm}@cs.brown.edu

1 Introduction

Computational fluid dynamics (CFD) techniques are used to simulate flows of fluids like air or water around such objects as airplanes and automobiles. These techniques usually generate very large amounts of numerical data which are difficult to understand without using graphical scientific visualization techniques. There are a number of commercial scientific visualization applications available today which allow scientists to control visualization tools via textual and/or 2D user interfaces. However, these user interfaces are often difficult to use. We believe that 3D direct-manipulation techniques for interactively controlling visualization tools will provide opportunities for powerful and useful interfaces with which scientists can more effectively explore their datasets. A few systems have been developed which use these techniques, including [1].

In this paper, we will present a variety of 3D interaction techniques for manipulating parameters of visualization tools used to explore CFD datasets, and discuss in detail various techniques for positioning tools in a 3D scene. We generally call these techniques *3D widgets* [2].

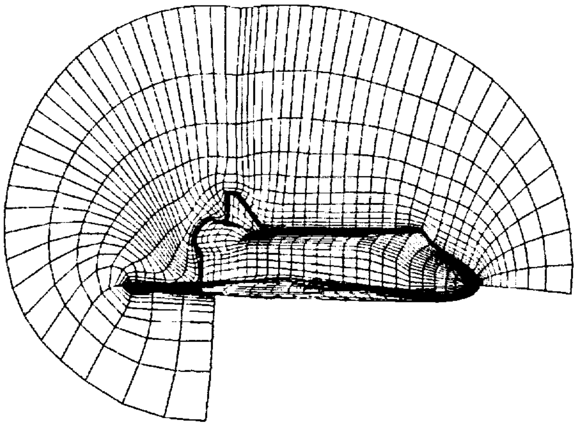

Figure 1: A 3D curvilinear grid for the Space Shuttle.

Our environment, built on top of the UGA system [4], supports both vector and scalar fields. The data may be arranged in a regular grid or it may be deformed using a curvilinear grid to provide more detail in areas with more complex flow. In a typical curvilinear dataset (Figure 1), the computation grid is wrapped around the body of an aircraft and scaled so that there are many more sample points in the boundary region (near the surface) than in other areas.

We are conducting this research project under contract to NASA in order to provide scientists there with more effective tools for exploring CFD datasets. No formal user studies have yet been conducted to verify the general usability of our interfaces.

2 3D widgets for scientific visualization

3D widgets are naturally suited for CFD visualization applications because the data are inherently 3D. Also, several of the visualization techniques commonly used in CFD visualization are based on real-world tools used in actual wind tunnels (*e.g.*, "rakes" of streamlines simulate smoke-emitting rakes). With these metaphors in mind, 3D widgets can be constructed to control parameters of commonly used visualization techniques.

In general, the design of a widget must consider two conflicting requirements: that the widget have adequate geometry to disclose its affordances; and that this geometry not be so complex that it obscures other objects in the scene. When exploring or analyzing a dataset, the visualizations (*i.e.*, streamlines, cutting planes, etc.) of the data are usually the most important elements in the scene. In these kinds of applications, it is crucial that 3D widgets provide only the necessary functionality with a minimum of geometry.

In general, a widget's degrees of freedom should correspond to the type of data it affects (*e.g.*, a widget which produces a scalar value should be constrained to a single degree of freedom, as in a slider or a knob). Also, widgets should provide useful visual feedback for the user's actions (*e.g.*, highlighting when selected).

We have implemented 3D widgets for the following visualization techniques: streamline and particle path; rake of streamlines or particle paths; array of tufts ("hedgehog"); scalar and vector probe; isosurface; and cutting plane.

Each 3D widget provides interactive access to a technique's parameters, such as position, orientation, resolution, etc. When possible, we align a widget's range of motion with the effect it has on a visualization technique. For example, the rake's resolution handle (Figure 2), which determines the spacing and number of streamlines displayed, slides along the bar of the rake; also, the arrow-shaped extent handles of the "hedgehog" are aligned with and move in three orthogonal directions. It is more difficult to create interfaces to some abstract parameters such as the integration step of a streamline.

3 Case study: Positioning widgets

A very common task for scientists is specifying the position of a 3D probe in a dataset (*e.g.*, placing the source of a streamline in a vector field). To demonstrate the range of choices in 3D widget design,

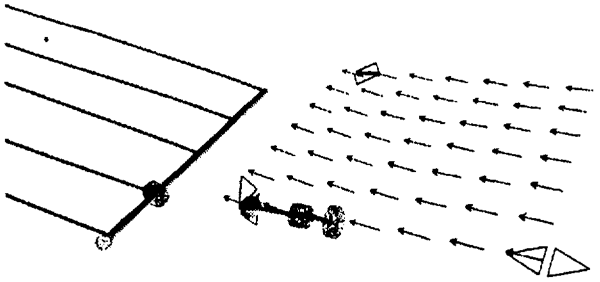

Figure 2: The rake (left) and hedgehog (right) 3D widgets. The cylindrical shapes are sliders which control resolution of streamlines or tufts. The arrow-shaped controls on the hedgehog modify the extent of the array of tufts in each of three dimensions.

we will discuss the designs of several techniques for positioning widgets which we have implemented in our system.

The default positioning technique is direct-manipulation screen-aligned translation. To move objects in three-space, we have added our "interactive shadow" [3] widgets to this environment. A combination of screen-space translation and interactive shadows allows the user to easily place an object in a 3D scene without having to move the camera. The "shadow" widgets also provide useful depth cues for 3D widgets and other objects in the scene.

However, it is easy for a geometry in the scene (*e.g.*, the Shuttle fuselage) to hide the "shadow" widgets and render them unusable. Another technique, called "object handles", attaches three objects (in our case, simple line segments) to the selected object and aligns them with the world coordinate system. These widgets provide much of the same functionality as the "interactive shadows", but do not provide any depth cues.

Each of these techniques use features of Cartesian coordinate space to position objects. While they are useful techniques in many situations, problems arise when using them to explore curvilinear datasets because the data was structured based on the geometry of the objects being modeled (as in the Space Shuttle), and it is often useful to move objects relative to this geometry. We have extended the handle metaphor to accomodate these situations.

"Grid-aligned handles" are especially useful for curvilinear datasets which are specially fitted to a physical model like an airfoil. As shown in Figure 3, the handles trace out nearby computation grid lines. When a handle is dragged, the selected object is constrained to move along the grid line. Using this technique, it is straightforward to translate objects along complex surfaces whose geometry is reflected in the computation grid, such as the leading edge of an airfoil. Furthermore, because these handles display the nearby structure of the grid, users can possibly gain a better understanding of the dataset as they explore it with these widgets.

It can also be useful to work with interaction techniques based on the data being visualized. For example, the vector probe widget in Figure 3 consists of a grey spherical sample point, an arrow which represents the direction of flow at that point, and a disk which represents the plane perpendicular to the vector. By dragging the arrow component, the sample point can be moved along the streamline formed by the flow through that point. The disk is used to move the sample point perpendicular to the flow, allowing the user to explore nearby streamlines in the flow field.

We can use this same general probe widget to visualize scalar data. In this case, the vector component displays the gradient of a scalar field. Pulling the vector changes the value at which the isosurface is computed; translating the disk moves the sample point along the isosurface.

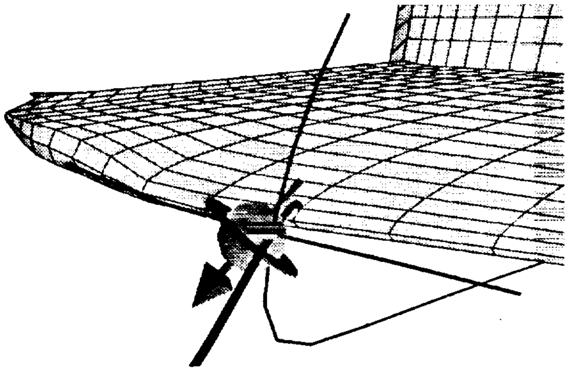

Figure 3: A vector probe widget and three grid-aligned object handles. The three lines extending through the probe widget are the handle widget and serve both as a Frenet frame for the point in the computation grid closest to the probe and as constrained translation widgets. The thicker grid handle extends outward from the surface of the wing.

4 Future Work

We are continuing to explore techniques to further simplify the graphical representations of our widgets without impeding their functionality. Additionally, the widgets described in this paper were rapidly protoptyped to explore the design space. After we have done user studies with different widget designs, we would like to redesign our tools so that they have a more consistent interface. The general probe widget, which can be used as an interface to a vector or scalar probe, a streamline, or an isosurface, is a step in this direction.

Acknowledgments

This work was supported primarily by NASA Ames. Support is also provided by the NSF/ARPA Science and Technology Center for Computer Graphics and Scientific Visualization, by ONR Contract N00014-91-J-4052, ARPA Order 8225, and by the sponsorship of IBM, NCR, Sun Microsystems, Hewlett Packard, and Digital Equipment Corporation. We thank Steve Bryson, Andries van Dam, and the members of the Brown University Graphics Group, especially Jeremy Katz and Lars Bishop, for their help and support.

References

[1] Steve Bryson and Creon Levitt. The virtual windtunnel: An environment for the exploration of three-dimensional unsteady flows. In *Visualization '91*, pages 17–24, 1991.

[2] D. Brookshire Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. Three-dimensional widgets. *Computer Graphics (1992 Symposium on Interactive 3D Graphics)*, 25(2):183–188, March 1992.

[3] Kenneth P. Herndon, Robert C. Zeleznik, Daniel C. Robbins, D. Brookshire Conner, Scott S. Snibbe, and Andries van Dam. Interactive shadows. *1992 UIST Proceedings*, pages 1–6, November 1992.

[4] Robert C. Zeleznik, D. Brookshire Conner, Matthias M. Wloka, Daniel G. Aliaga, Nathan T. Huang, Philip M. Hubbard, Brian Knep, Henry Kaufman, John F. Hughes, and Andries van Dam. An object-oriented framework for the integration of interactive animation techniques. *Computer Graphics (SIGGRAPH '91 Proceedings)*, 25(4):105–112, July 1991.