Using Deformations to Explore 3D Widget Design

Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins,
D. Brookshire Conner and Andries van Dam

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

CR Categories

H.5.2 [Information Interfaces and Presentation]: User Interfaces; Interaction Styles; Theory and Methods; I.3.5 [Computer Graphics]: Computational Geometry and Object Modeling; Curve, Surface, Solid, and Object Representations; I.3.6 [Computer Graphics]: Methodology and Techniques; Interaction Techniques

1 A 3D Interface for Deformations

We are developing a framework for creating interactive 3D environments for applications in design, education, and the communication of information and ideas [3]. Our most recent work focuses on providing a useful and powerful interface to such a complex environment. To this end we have developed *3D widgets*, objects that encapsulate 3D geometry and behavior, to control other objects in the scene [2]. We build 3D widgets as first-class objects in our real-time animation system. Because our system allows rapid prototyping of objects, we hope to enlarge today's surprisingly small vocabulary of 3D widgets that includes menus floating in 3D, gestural picking, translation and rotation, cone trees, and perspective walls.

As a way to focus on issues of 3D widget design, we have developed widgets to perform a particular task: applying high-level deformations to 3D objects [1]. The complexity of these operations makes numerical specification or panels of sliders difficult to use, and yet direct manipulation interfaces cannot provide meaningful feedback without fixing most parameters. In this video paper, we show a set of new 3D widgets to control deformations called *racks*.

A simple rack consists of a bar specifying the axis of deformation and some number of handles attached to the bar specifying additional deformation parameters. For example, a taper rack has two additional handles. Moving the ends of the handles towards or away from the axis bar changes the amount of taper of the deformed object; changing the distance between the handles changes the region over which the deformation is applied.

A more complex rack can have multiple handles specifying different deformations. The racks in Figures 1–3 all have handles for twisting (purple), tapering (blue), and bending (red) an object. The deformation range is the region between the twist and taper handles.

2 The Issues in 3D Widget Design

Many of the issues in designing 3D widgets are similar to those in designing a good 2D interface. However, good answers that serve in 2D are not necessarily good answers in more complex 3D environments. For example, two common 2D interface solutions are control panels (e.g., sliders, buttons, and menus) and the direct manipulation of objects (e.g., dragging files in the Macintosh Finder). A control panel for a deformation can require as many as a dozen sliders and thus consumes a great deal of screen real estate; it also artificially differentiates related parameters and does not provide good correlation to its 3D effects. In addition, although these controls may be arranged so that related sliders (say, for the three components of a vector) are logically grouped, users are rarely able to correlate changes in these controls with corresponding changes in the deformed objects.

A 3D direct manipulation interface to control the same deformation might be implemented in which the user clicks and drags the faces of an object. One problem with this approach is that most computer graphics objects look rigid, as though they do not permit deformation. Rigid objects appear to afford six degrees of freedom, three translational and three rotational, and direct manipulation techniques like virtual sphere rotation work very well because they exploit this apparent rigidity. To compensate for the lack of feedback in certain direct manipulation techniques, we can introduce tools that make additional degrees of freedom available; the rack is an example of such a tool. However, the widget's geometry, while possibly suggesting the kinds of actions that can be performed, may at the same time obscure the object of interest or unnecessarily complicate the scene. In addition to showing this tradeoff, we consider the following issues in widget design:

- Self-disclosure

  A widget whose geometry indicates its behavior is called *self-disclosing*. For example, a handle that twists an object can itself be a twisted object, thereby suggesting its action.

- Implicit versus explicit control of parameters

  Widgets usually explicitly control one or more parameters of an action. For example, a handle in a taper rack controls the amount that the object is tapered. A widget may simplify an action by implicitly controlling other parameters. For example, a widget can implicitly control the tesselation of objects by varying the tesselation with the amount of deformation. There is no handle to explicitly control the tesselation.

- Degrees of freedom

  A 3D environment offers multiple degrees of freedom that can hinder as well as help a particular operation. A tool can be made more effective by *removing* unnecessary degrees of freedom with constraints.

- Intended use

  An impressionistic modeling tool for illustration or design requires only that the modeled object “look right.” Tools designed for manufacturing and mechanical design, however, must set parameters to precise values and produce changes that are exactly reproducible.

3 Conclusions

The design space of 3D widgets is extremely large and unexplored. A great deal of experimentation, by both designers and users, is necessary before a common, useful, and powerful collection of 3D metaphors and widgets becomes as widespread as those of the 2D windows, icons, and mouse interfaces of today. By allowing the same power for widget development as it does for application development, our system provides the rapid prototyping environment necessary to explore this vast design space — a variant of a deformation widget can be built within a few hours, a completely new widget within days. We can now begin to look critically at 3D interface design and develop productive environments that use the power of 3D effectively.

In addition, we are now in the process of using our widgets to build more widgets, leading to a complete 3D interface construction set that will itself be a 3D interface. For example, the deformed handles of some of the widgets presented in the video paper were made with these very same widgets. We will also continue designing widgets, extending our existing designs to include animation, complex rendering techniques, and even simulation methods in order to explore, for example, physically based widgets.

Acknowledgments

This work was supported in part by the NSF/DARPA Science and Technology Center for Computer Graphics and Scientific Visualization. We also gratefully acknowledge the sponsorship of IBM, NCR, Sun Microsystems, Hewlett Packard and Digital Equipment Corporation. We also thank the members of the Brown University Graphics Group for their help and support, especially Bob Zeleznik, Nate Huang, and our visitor from Darmstadt, Frank Graf.

References

[1] Alan H. Barr. Global and local deformations of solid primitives. In *Proceedings of the ACM SIGGRAPH, Computer Graphics*, volume 18(3), pages 21–30, July 1984.

[2] D. Brookshire Conner, Scott S. Snibbe, Kenneth P. Herndon, Daniel C. Robbins, Robert C. Zeleznik, and Andries van Dam. Three-dimensional widgets. In *Proceedings of the 1992 Symposium on Interactive 3D Graphics*, pages 183–188, 1992.

[3] 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. In *Proceedings of the ACM SIGGRAPH, Computer Graphics*, volume 25(4), pages 105–112, July 1991.

Figure 1: The starting configuration for a pink cube and a rack with twist (purple), taper (blue), and bend (red) handles.

Figure 2: The taper handle is translated downward, tapering the cube. The deformation range is the region between the twist and taper handles.

Figure 3: A pocketknife is twisted and bent with a rack.