

SQUIDS: Interactive Deformation of Superquadrics for Model Matching in Virtual Environments

James Sinnott and Toby Howard

Advanced Interfaces Group
Department of Computer Science
University of Manchester
Manchester M13 9PL
sinnottj@cs.man.ac.uk

Abstract

This paper presents work-in-progress on SQUIDS, a system for the interactive creation and deformation of superquadric shapes in virtual environments. SQUIDS is one component of a prototype system for semi-automatically creating realistic virtual environments, based on image data captured from real-world scenes.

Keywords: Superquadrics, interactive deformation, virtual environments, model matching.

1 Introduction

The work described in this paper forms part of an ongoing research project called REVEAL [AIG 1999]. REVEAL (Reconstruction from Video of Environments with Accurate Lighting) is concerned with the development of techniques for the semi-automatic construction of virtual environments (VEs) that faithfully correspond to real-world scenes. The goal is to reconstruct VEs that are geometrically accurate, have correct global illumination, and are fully interactive. The reconstructed environments should not only look the same as the original scenes, but the objects within them should also behave as in the real world.

The ability to build environments that accurately model real-world scenes has a wide range of potential applications, including architectural design, simulation and training, and media applications such as virtual studios or film sets. One partner in the REVEAL project is Greater Manchester Police, and a specific deliverable of the project is to demonstrate that the techniques under

development are suitable for accurately creating reconstructions of scenes of crime.

Previous work in the Advanced Interfaces Group [Howard et al. 2000], [Murta et al. 1998] has shown that it is possible to manually construct virtual environments that model real-world scenes with a high level of accuracy. Constructing such environments is, however, an extremely labour-intensive and time-consuming process. If such environments are to be used successfully in real-world applications, techniques must be developed to speed up and simplify the construction process. The aim of the REVEAL project is to develop such techniques.

In particular, on-going research is focussing on the automatic extraction of scene geometry by processing video images, to generate an unstructured polygon mesh. This paper describes SQUIDS (SuperQUadrics with Interactive DeformationS), which is one component of a system concerned with processing the polygon mesh to identify discrete objects. These may then be structured to permit articulation and other behaviour. The central problem to be solved is how to convert a static

environment, that a user can walk through and observe, into one in which they can pick up objects, move them around, and interact with them as in real life.

The remainder of the paper is structured as follows. Section 2 describes a proposed model matching system, based on deformable superquadrics. Section 3 presents a brief formal definition of superquadrics, Section 4 discusses deformation of superquadrics and Section 5 describes the SQUIDS system itself. Section 6 concludes the paper by looking at areas for future work.

2 Model matching with superquadrics

The ultimate aim of this work is to develop a 3D scene understanding system. The input to this system will be a 3D polygon mesh that has been reconstructed from image sequences captured from a real-world scene. The task of the system will be to locate and identify, within this polygon mesh, the various objects that make up the scene.

It is not expected that the system will be fully automated, but that it will take an interactive, user-driven approach. It is anticipated that a number of different model matching techniques will be used, although the primary approach will be to fit models based on deformable superquadrics to the 3D data.

Superquadrics [Barr 1981] are a family of simple parametric models that can represent a wide range of 3D forms with just a small number of controlling parameters. Figure 1 shows some examples of superellipsoids, one member of the superquadric family. Parametric deformations are used to further increase the range and complexity of forms that can be modelled. Complex objects can be modelled as collections of simpler parts, using arrangements of deformed superquadrics. The simplicity and flexibility of shape representation of deformable superquadrics make them particularly suited to model matching [Bardinet et al. 1995], [Solina et al. 1990], [Terzopoulos et al. 1991].

The model matching process will take place within a virtual environment (VE). An important aspect of this is that it allows for the possibility of working in true stereoscopic 3D, using a headset to generate a fully immersive environment.

Given the reconstructed data of a scene, it is envisaged that the system will operate as follows: first, the user will interpret a set of points from the polygonal scene data, deciding what object the

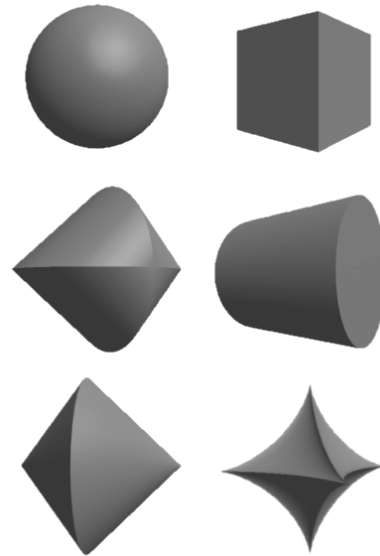


Figure 1 Examples of superellipsoids, one member of the superquadric family

points represent. Next, the user selects an appropriate object model from a database and suggests an approximate initial fit of the model to the data points. The system then automatically refines the fit, recovering the precise parameters of the object from the data.

In order to provide the initial fit to the data, the user will need to be able to specify a set of starting parameters for the model. These starting parameters will include the position, orientation, dimensions and any other controlling parameters of the model. If models based on deformable superquadrics are used, then the user will also need to specify the parameters of any appropriate deformations.

The proposed approach to model matching establishes a need to investigate simple and intuitive methods for applying deformations to superquadrics. In particular, some form of *deformation interface* between the user and the superquadric is required. By manipulating this interface, the user should be able to alter the controlling parameters of a particular deformation, and then have the deformation applied to the superquadric. A number of requirements can be identified for such an interface:

- It must be simple to use in a virtual environment.
- It should operate in real-time; when the user alters the controlling parameters, the modelled shape should respond in interactively.

- It must be intuitive: there should be an obvious mapping between the operation of the interface and the resulting deformation. The user should be able to predict the results of his actions.

The simplest approach would be for the user to enter the values of the deformation parameters explicitly, via the keyboard. This form of interface, however, would not be at all intuitive. It would be very difficult, for example, to choose appropriate changes in the parameter values so as to vary the effect of a deformation in a predictable and desirable manner.

A better approach would be to use a mouse-operated graphical interface, for example using slider bars to control the value of the deformation parameters. This would provide a much more intuitive and predictable way of specifying the desired level of deformation.

A significant problem with both of these approaches, however, is that the user is manipulating an interface that is entirely separate from the object undergoing the deformation. This disconnection degrades the mapping between the operation of the interface and the resulting effect of the deformation.

The problem becomes particularly apparent when the model matching process is taking place within a fully immersive virtual environment. In this case, the user may be viewing the environment through a stereo headset, making operation of a keyboard or standard mouse very difficult.

It would be much more appropriate to allow the user to apply deformations to the models *directly*, within the virtual environment. The deformation interface should be connected as closely as possible to the object undergoing the deformation. The mapping between manipulation of the interface and the resulting deformation should be such that the user can rely almost entirely on the visual effect of the deformation to guide his actions, with the actual parameter values of secondary importance.

SQUIDS has been developed with the aim of providing such an interface.

3 Superquadrics

Superquadrics are a family of parametric solids derived from the basic quadric surfaces and solids. Extra flexibility in shape representation is achieved by raising each trigonometric term in the quadric equations to an exponent. These exponents control the relative roundness and squareness along the major axes of the surfaces. By altering the value of the exponents, superquadrics can represent a wide

range of forms: spheres, cylinders, parallelepipeds, pinched stars and shapes in between.

There are four forms of superquadric: superellipsoids, supertoroids and superhyperboloids of 1 or 2 sheets. Only superellipsoids, however, will be used for model matching, as the other three forms of superquadric do not provide a useful set of shapes for modelling real-world objects. Hence, only superellipsoids will be described here.

3.1 Superellipsoids

The following position vector \underline{x} defines a superellipsoid surface:

$$\underline{x}(\eta, \omega) = \begin{bmatrix} a_1 \cos^{\epsilon_1} \eta \cos^{\epsilon_2} \omega \\ a_2 \cos^{\epsilon_1} \eta \sin^{\epsilon_2} \omega \\ a_3 \sin^{\epsilon_1} \eta \end{bmatrix} \quad \begin{matrix} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \omega < \pi \end{matrix}$$

The vector \underline{x} sweeps out a closed surface as the two independent parameters, angles η and ω , change in the given intervals. Parameters a_1 , a_2 and a_3 define the superquadric scaling in the x, y and z directions, respectively. ϵ_1 is the squareness parameter in the north-south direction, while ϵ_2 is the squareness parameter in the east-west direction. By varying the value of the squareness parameters, a wide range of shapes can be produced. Figure 2 illustrates a number of the shapes that can be generated by varying the value of each of the squareness parameters.

Superquadrics have well-behaved inside-outside

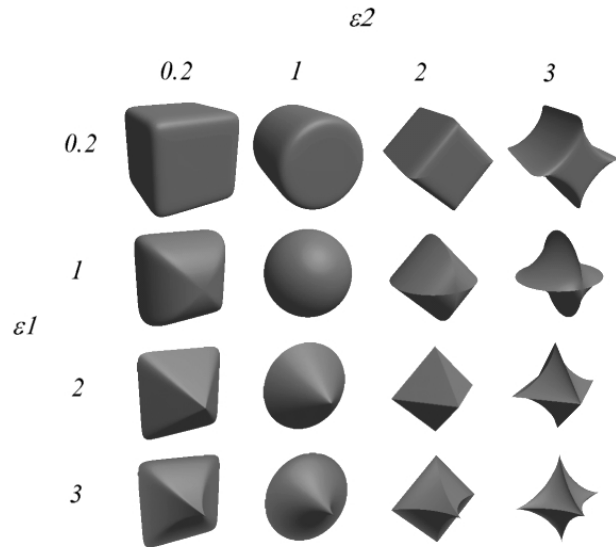


Figure 2 Various superellipsoid shapes generated by altering the shape parameters ϵ_1 and ϵ_2 from 0.2 to 3

functions, which are used to determine whether an arbitrary point falls inside, outside or on a superquadric's surface. Such inside-outside functions make superquadrics particularly suited to the problem of fitting a surface to a set of 3D points.

4 Deformation of superquadrics

Although superquadrics can model a relatively wide range of shapes, they are too restrictive for modelling the complex forms that might occur in a real-world scene. A particular restriction is that superquadrics can only model shapes that are symmetric.

Many researchers have proposed the use of *parametric deformations* to improve the representational range of superquadrics. By deforming a superquadric in some way (for example bending or tapering it), it can be made to model shapes that would not otherwise be possible. Of particular interest are deformations that are used in the manufacture of objects, or that allow the modelling of complex natural forms [Solina et al. 1990].

There are two categories of deformation: global and local. A global deformation affects an object as a whole; that is to say, given a set of points on the surface of an object, the *same* deformation is applied to each of the points. This may be contrasted with a *local* deformation, in which the deformation is only applied to a set of points corresponding to a sub-region of the object's surface.

At the time of writing, three global parametric deformations have been implemented in SQUIDS, global linear tapering, global axial twisting and global linear bending. These deformations are based on the work by Barr [1984].

4.1 Global linear tapering

A tapering deformation can be thought of as a differential scaling. An object is tapered by differentially changing the length of two components of each of its vertices (y and z , say), whilst leaving the third component (x) unaltered.

Global linear tapering along the x -axis is given by the equations:

$$\begin{aligned} X &= x \\ Y &= f_y(x)y \\ Z &= f_z(x)z \end{aligned}$$

Where, X, Y, Z are the components of the surface vector \underline{X} of the deformed object, x, y, z are the components of the original surface vector \underline{x} and f_y, f_z

are the tapering functions in the y - and z -axes. For linear tapering along the x -axis, the tapering functions used are [Solina et al. 1990]:

$$\begin{aligned} f_y(x) &= \frac{t_y}{a_1}x + 1 & -1 \leq t_y \leq 1 \\ f_z(x) &= \frac{t_z}{a_1}x + 1 & -1 \leq t_z \leq 1 \end{aligned}$$

Where a_1 is the superquadric scaling in the x direction. The functions f_y, f_z increase (or decrease) linearly as x increases, depending on the value of the tapering parameters t_y and t_z . Tapering along the y - or z -axes is achieved in the same manner as for the x -axis.

Figure 3 shows some superquadric objects that have been subjected to a tapering deformation.

4.2 Global axial twisting

Just as tapering is simulated by differential scaling, a twist can be approximated by a differential rotation. An object is twisted by rotating two components of each of its vertices whilst leaving the third component unchanged. This deformation is analogous to 'twisting' a deck of cards: each card is rotated slightly more than the card beneath it.

A twist about the x -axis is produced by the equations:

$$\begin{aligned} \theta &= f(x) \\ X &= x \\ Y &= y \cos(\theta) - z \sin(\theta) \\ Z &= y \sin(\theta) + z \cos(\theta) \end{aligned}$$

Where X, Y, Z, x, y and z are defined as for tapering, and f is the twisting function. For a twist about the x -axis, the twisting function used is:

$$f(x) = n\left(\pi + \frac{x}{a_1}\pi\right)$$

Where the twisting parameter n determines the number of complete (i.e. 360°) twists about the x -

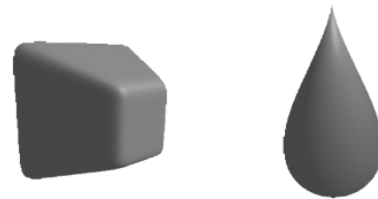


Figure 3 Two tapered superquadric objects

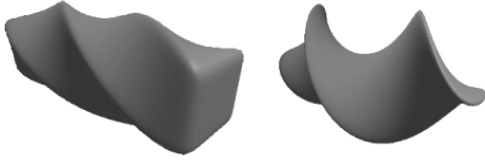


Figure 4 Two twisted superquadric objects

axis. A twist along the y - or z -axes is achieved in the same manner as for the x -axis.

Figure 4 shows some superquadric objects that have been subjected to a twisting deformation.

4.3 Global linear bending

Bending is approximated by simultaneously rotating and translating two components of each vertex around the third. A global linear bend along the x -axis is given by the equations:

$$\begin{aligned}\theta &= ky \\ X &= x \\ Y &= -\sin(\theta)\left(z - \frac{1}{k}\right) \\ Z &= \cos(\theta)\left(z - \frac{1}{k}\right) + \frac{1}{k}\end{aligned}$$

Where X , Y , Z , x , y and z are defined as for tapering, and k is the bending rate parameter. $1/k$ gives the radius of curvature of the bend. A bend along the y - or z -axes is achieved in the same manner as for the x -axis.

Figure 5 shows some superquadric objects that have been subjected to a bending deformation.

5 SQUIDS

The aim of SQUIDS is to provide an interface that allows the user to apply deformations to a superquadric object directly within a virtual environment.

SQUIDS can be used either in a non-immersive VE, viewed on a monitor and operated with a standard mouse, or in a fully immersive VE, viewed

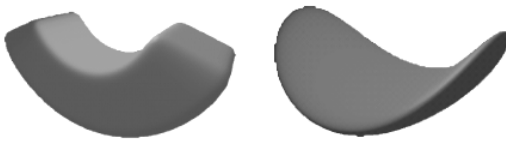


Figure 5 Two bent superquadric objects

through a stereo headset and operated with 3D mice.

SQUIDS is implemented in C and uses the MAVERIK system to create and manage the virtual environment. MAVERIK (Manchester Virtual Environment Interface Kernel) [Cook et al. 1999] [Howard et al. 1999] [Hubbold et al. 1996], is a toolkit for developing single-user virtual environment applications. It has been in development since 1997 and has recently been released publicly, as a part of the Free Software Foundation's GNU project [GNU]. MAVERIK provides the programmer with the tools necessary to create, visualise, manage, navigate and interact with virtual environments.

5.1 The deformation interface

The basic principle of the SQUIDS deformation interface is the *deformation node*. A deformation node is an object that corresponds to one of the controlling deformation parameters. For a particular deformation, there will be a separate node corresponding to each of the deformation parameters.

The user can select a deformation node and drag it around within the VE. Each deformation node has a specific path along which it may move. The node's possible range of movement within the VE is restricted so that it can be dragged only along its path. A deformation node's position along its designated path determines the value of its corresponding deformation parameter; by dragging the node along its path, the user alters the value of the deformation parameter. The interface provides real-time feedback – as the user drags a deformation node about, varying the deformation parameter, the superquadric is continuously deformed with the new parameter values. Depending upon the nature of the deformation, there may be positions along the nodes' paths corresponding to the maximum level of deformation possible, and a neutral position corresponding to no deformation.

The next section gives an illustrative example of the deformation interface, using the global linear tapering deformation described in Section 4.1.

5.2 Interactive tapering with SQUIDS

Consider a tapering deformation applied to an object along its x -axis. Two parameters control the tapering deformation: t_y , corresponding to the amount of tapering applied perpendicular to the y -axis, and t_z , corresponding to the amount of tapering applied perpendicular to the z -axis. Figure 6

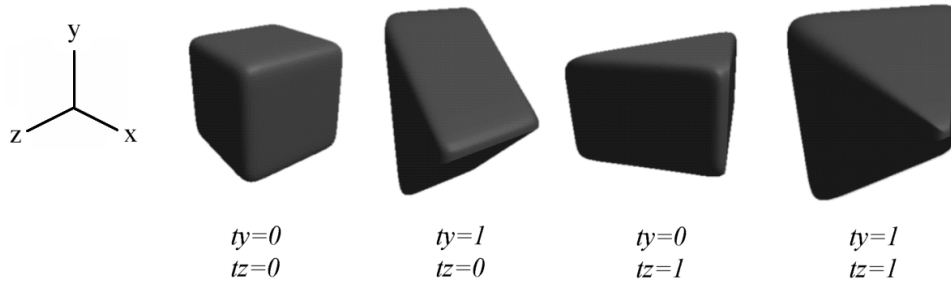


Figure 6 The effect of varying the controlling parameters of a tapering deformation

illustrates the effect of varying these parameters, on a simple cuboid-shaped object. Tapering along the y - or z -axes occurs in exactly the same manner.

For interactive tapering along the x -axis, two deformation nodes are used, one corresponding to t_y , and the other to t_z . Figure 7 illustrates the deformation interface to allow for interactive tapering. The figure shows how the deformation nodes (the small light-coloured cubes) are positioned to allow interactive deformation of a superquadric (large cuboid). Each node has a path, parallel to the axis of deformation, along which it may be moved; these paths are marked out by thin black lines. A bounding box is shown surrounding the superquadric as a visual aid, to ensure the user has a clear idea of the superquadric's orientation during the deformation process.

By dragging either node along its path, the user can vary the level of tapering applied to the superquadric. Tapering can be applied in both the

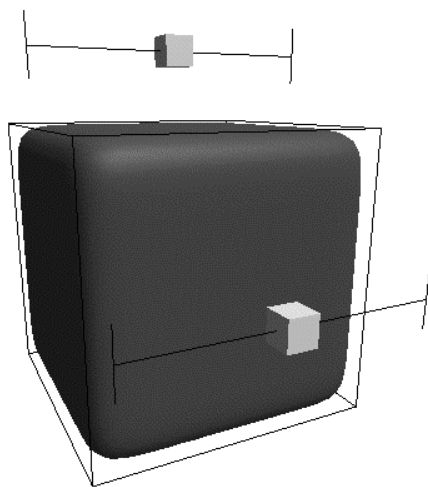


Figure 7 The deformation interface for interactive tapering of a superquadric

positive and negative directions along the x -axis. The end positions of each nodes' path (shown by vertical black bars) indicate the maximum amount of tapering that can be applied in either direction.

When a node is at the centre of its path, the corresponding deformation parameter has no effect on the tapering deformation; if both nodes are in their central neutral position, no tapering is applied at all.

Figures 8 and 9 illustrate the process of interactively tapering a superquadric with SQUIDS. The figures show the deformation nodes being progressively moved along their paths, and the resulting effect on the superquadric. The top row of Figure 8 shows the different levels of tapering that result from moving one of the deformation nodes while leaving the other in its neutral position. Conversely, the bottom row shows the effect of moving the second node while leaving the first in its neutral position. Figure 9 shows the effect of moving both nodes simultaneously to the same positions along their respective paths.

Tapering can be interactively applied in the same way along the y - or z -axes as well. The positions of the deformation nodes, and the paths along which they may be moved, change to reflect the axis along which the tapering is applied.

6 Future work

Further to the three global deformations described in Section 4, it is planned that other forms of deformation will be used. In particular, local deformations, such as the free-form deformations proposed in [Sederberg et al. 1986], are to be investigated. For any further deformations that are used, appropriate techniques for interactively applying those deformations to superquadrics must be devised.

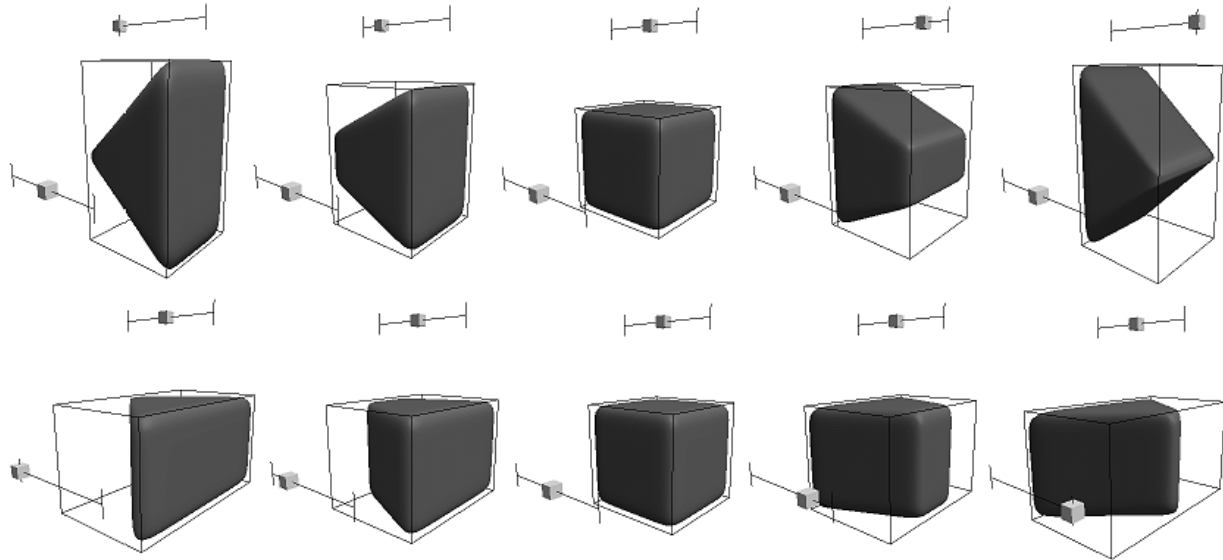


Figure 8 Interactively tapering a superquadric using one deformation node

The work described in this paper represents only a small part of an ongoing research project. As such, the short-term development of SQUIDS will be restricted to the requirements of the overall project aims. However, there is potential for future development of SQUIDS into a stand-alone tool for interactively deforming general 3D objects. In particular, SQUIDS is not restricted for use with superquadrics. The system has been designed around the deformations, rather than the specific objects to which the deformations are applied, and as such could be used in conjunction with any geometric model that can be deformed in this way.

7 Acknowledgements

It is a pleasure to acknowledge the support of our

colleagues in the Advanced Interfaces Group: Jon Cook, Simon Gibson, Roger Hubbard, Alan Murta, Steve Pettifer and Adrian West. We are also greatly indebted to D.I. David Heap of Greater Manchester Police for his long-standing enthusiasm and advice. The REVEAL project is co-funded by EPSRC (GR/M14531) and Greater Manchester Police.

8 References

- AIG 1999, The Advanced Interfaces Group, <http://www.cs.man.ac.uk/aig/reveal/>
- Bardinet E, Cohen LD, Ayache N (1995) *A parametric deformable model to fit unstructured 3D data*. Research Report 2617, INRIA, Sophia-Antipolis, July 1995
- Barr AH (1981) *Superquadrics and angle preserving transformations*. IEEE Transactions on

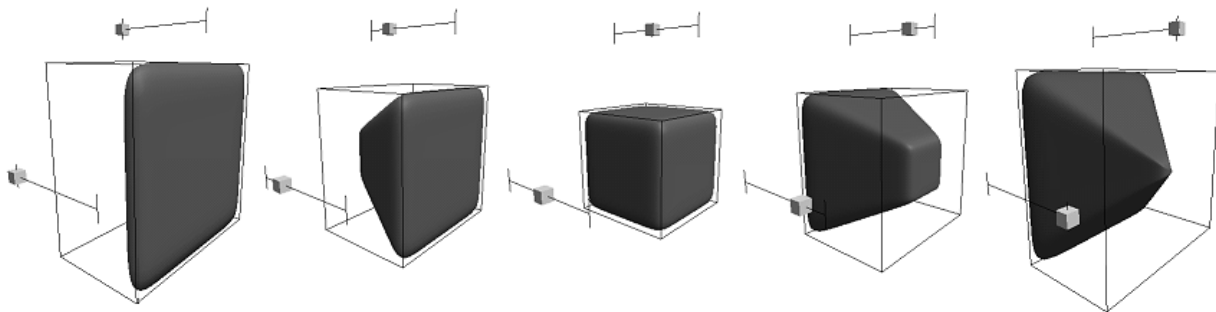


Figure 9 Interactively tapering a superquadric using both deformation nodes simultaneously

- Computer Graphics and Applications, **1**(1), 11-23, January 1981
- Barr AH (1984) *Global and local deformations of solid primitives*. Computer Graphics, **18**(3), 21-30, 1984
- Cook J, Howard T (editors) (1999) *MAVERIK programmer's guide V5.0*. The Advanced Interfaces Group, Department of Computer Science, University of Manchester, 1999
- GNU, The Free Software Foundation, <http://www.gnu.org>
- Howard T, Hubbard R, Murta A (1999) *GNU MAVERIK: A virtual reality system for research and teaching*. Proc. Computer Graphics & Visualisation in Education 99, 85-91, 1999
- Hubbard R, Dongbo X, Gibson S (1996) *MAVERIK: the Manchester virtual environment interface kernel*. In Virtual Environments and Scientific Visualisation 1996
- Murta AD, Gibson S, Howard TLJ, Hubbard RJ, West AJ (1998) *Modelling and rendering for scene of crime reconstruction: a case study*. Proc. Eurographics UK, 196-173, 1998
- Howard TLJ, Gibson S, Murta AD (2000) *Virtual environments for scene of crime reconstruction and analysis*. Proc. Visual Data Exploration & Analysis VII, SPIE 2000, to appear
- Solina F, Bajcsy R (1990) *Recovery of parametric models from range images: the case for superquadrics with global deformations*. IEEE Transactions on Pattern Analysis and Machine Intelligence, **12**(2), 131-147, February 1990
- Sederberg TW, Parry SR (1986) *Free-form deformation of solid geometric models*. Computer Graphics (proceedings SIGGRAPH'86), **20**(4), 151-160, August 1986
- Terzopoulos D, Metaxas D (1991) *Dynamic 3D models with local and global deformations: deformable superquadrics*. IEEE Transactions on Pattern Analysis and Machine Intelligence, **13**(7), 703-714, July 1991