Procedural Shape Generation for Multi-dimensional Data Visualization

David S. Ebert [a], Randall M. Rohrer [b], Christopher D. Shaw [c], Pradyut Panda [a], James M. Kukla [a] and D. Aaron Roberts [d]

[a] *CSEE Department, University of Maryland Baltimore County*
1000 Hilltop Circle, Baltimore, MD 21250, USA
{ebert,panda,jkukla1}@csee.umbc.edu

[b] *Department of EECS, The George Washington University*
Washington, DC, 20052, USA
rohrer@seas.gwu.edu

[c] *Department of Computer Science, University of Regina*
Regina, Saskatchewan, S4S 0A2, Canada
cdshaw@cs.uregina.ca

[d] *NASA Goddard Space Flight Center, Mailstop 692.0*
Greenbelt, MD, 20771, USA
roberts@vayu.gsfc.nasa.gov

Abstract

Visualization of multi-dimensional data is a challenging task. The goal is not the *display* of multiple data dimensions, but *user comprehension* of the multi-dimensional data. This paper explores several techniques for perceptually motivated procedural generation of shapes to increase the comprehension of multi-dimensional data. Our glyph-based system allows the visualization of both regular and irregular grids of volumetric data. A glyph's location, 3D size, color, and opacity encode up to 8 attributes of scalar data per glyph. We have extended the system's capabilities to explore shape variation as a visualization attribute. We use procedural shape generation techniques because they allow flexibility, data abstraction, and freedom from specification of detailed shapes. We have explored three procedural shape generation techniques: fractal detail generation, superquadrics, and implicit surfaces. These techniques allow from 1 to 14 additional data dimensions to be visualized using glyph shape.

Preprint submitted to Elsevier Preprint *15 December 1999*

1 Introduction

The simultaneous visualization of multi-dimensional data is a difficult task. The goal is not only the display of multi-dimensional data, but the *comprehensible display* of multi-dimensional data. Glyph, or iconic, visualization is an attempt to encode more information in a comprehensible format, allowing multiple values to be encoded in the parameters of the glyphs [10]. The shape, color, transparency, orientation, etc., of the glyph can be used to visualize data values. Glyph rendering [10,11] is an extension to the use of glyphs and icons in numerous fields, including cartography, logic, and pictorial information systems.

In previous work, we explored the usefulness of stereo-viewing and two-handed interaction to increase the perceptual cues in glyph-based visualization. The Stereoscopic Field Analyzer (SFA) [4] allows the visualization of both regular and irregular grids of volumetric data. SFA combines glyph-based volume rendering with a two-handed minimally-immersive interaction metaphor to provide interactive visualization, manipulation, and exploration of multivariate, volumetric data. SFA uses a glyph's location, 3D size, color and opacity to encode up to 8 attributes of scalar data per glyph. These attributes are used when a vector visualization is not appropriate, such as when displaying temperature and pressure at each glyph. We are extending this work to combine glyph rendering with other visually salient features to increase the number of data dimensions simultaneously viewable.

2 Background

Our use of glyph attributes for visualization is based on human perceptual abilities. Color, transparency, position, and size are perceptually significant visualization attributes that are commonly used in visualization systems. Shape is a more challenging visualization attribute because three-dimensional shape perception is not as well understood as color, size, and spatialization perception. Most evidence suggests that shape variation is a valuable perceptual attribute that we can harness for visualization. Experiments show that humans can pre-attentively perceive three-dimensional shape [8]. Cleveland [3] cites experimental evidence that shows the most accurate method to visually decode a quantitative variable in 2D is to display position along a scale. This is followed in decreasing order of accuracy by interval length, slope angle, area, volume, and color. Bertin offers a similar hierarchy in his treatise on thematic cartography [2].

Our visualization system already utilizes glyph position in 3D, 3D scale (corresponding to Cleveland's length, area and volume) and color. Slope angle is a difficult dimension to use in an interactive system because of arbitrary orientation of the data volume. Therefore, the next opportunity for encoding a scalar value is shape.

One of the most difficult problems in glyph visualization is the design of meaningful glyphs. Glyph shape variation must be able to convey changes in associated data values in a comprehensible manner [10]. This difficulty is sometimes avoided by adopting a single base shape and scaling it non-uniformly in 3 dimensions. However, the lack of a more general shape interpolation method has precluded the use of shape beyond the signification of categorical values [2]. This paper describes three techniques we have explored for the procedural generation of glyph shapes for glyph-based volumetric visualization [7].

3 Perceptually-based Mapping of Shape Attributes

Glyph shape is a valuable visualization component because of the human visual system's pre-attentive ability to discern shape. Shapes can be distinguished at the pre-attentive stage [8] using curvature information of the 2D silhouette contour and, for 3D objects, curvature information from surface shading. Unlike an arbitrary collection of icons, curvature has a visual order, since a surface of higher curvature looks more jagged than a surface of low curvature. Therefore, generating glyph shapes by maintaining control of their curvature will maintain a visual order. This allows us to generate a range of glyphs which interpolate between extremes of curvature, thereby allowing the user to read scalar values from the glyph's shape. Pre-attentive shape recognition allows quick analysis of shapes and provides useful dimensions for comprehensible visualization.

Our use of glyphs is related to the idea of marks as the most primitive component that can encode useful information [2]. Senay points out that shape, size, texture, orientation, transparency, hue, saturation, brightness, and transparency are retinal properties of marks that can encode information [13,14]. Bertin has studied the use of marks for two-dimensional thematic maps and gives examples of how shape can be misused in the rendering of these maps [2]. In his examples, shapes are used to represent purely categorical data and, for this reason, he uses a small collection of distinct icons such as star, cross, square, circle, triangle, and so on. Because each individual shape does not have any inherent meaning, the reader is forced to continually look up the shape's meaning in the map legend. The main difficulty is that a collection of arbitrary icons does not have any necessary visual order, and so any assignment of shape to meaning is equivalent.

4 Procedural Shape Visualization

We have explored three different procedural techniques for the generation of glyph shape: superquadrics, fractal detail, and implicit surfaces. All three techniques use a procedural approach for glyph design to solve the complex problem of meaningful glyph design. Procedural shape generation techniques provide flexibility, data abstraction, and freedom from specification of detailed shapes [5]. Procedural techniques allow the shape to be controlled by high-level control parameters. The user changes the glyph shape from a more directorial, indirect aspect, where he or she is unburdened from the full explicit specification of detailed shapes. Our goal for glyph design was to allow the automatic mapping of data to shape in a comprehensible, easily controllable manner.

4.1 Fractal shape detail

One simple procedural technique for shape visualization is to generate distorted versions of a basic shape, such as a cube, where the amount of deviation from the basic shape is proportional to the data dimension being visualized: low data values will map to the base shape and high data values will map to very perturbed shapes. Displacement mapping is used to create the deviation from the base shape. We used a fractional Brownian motion (fBM) [6] turbulence function to create the displacement from the original surface. The main idea with this technique is to add a high frequency component to the shape while not distracting from the perception of the low frequency base shape perception. This addition of high frequency information does not detract from the overall spatial pattern of the data, which can occur from the generation of non-related shapes. Figure 1 shows the results of this technique applied to a solar wind simulation where X, Y and Z flow values are mapped to Color, Shape, and Opacity. In this case, rougher values indicate a larger Y component.

4.2 Procedural shape visualization using superquadrics

Superquadrics are a natural tool for automatic shape visualization that can allow from one to two data dimensions to be visualized with shape. Superquadrics, first introduced to computer graphics by Barr [1], are extensions of quadric surfaces where the trigonometric terms are each raised to exponents. Superquadrics come in four main families: hyperboloid of one sheet, hyperboloid of two sheets, ellipsoid, and toroid. For our initial implementation we have chosen superellipses due to their familiarity, but the system can

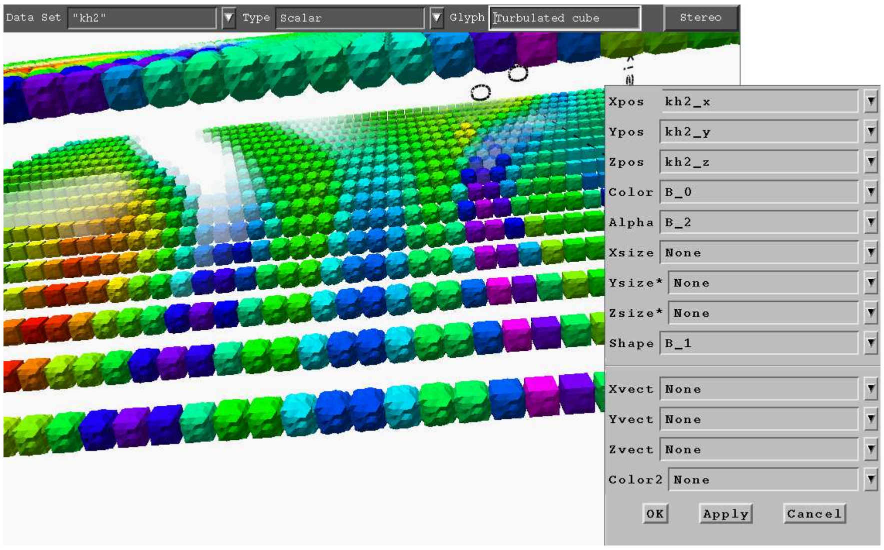

Fig. 1. Example fractal displacement of a base cube shape. The data values are a solar wind data set, with values ranging from 0 (smooth cube) to 1 (fuzzy cube) for the Y component of the flow vector. The Y values are strongest in the center. One slice out of the 12 in the dataset is shown. At the back of the slice, the Z component quickly drops to 0 in the transparent zone.

be easily extended to use other types of superquadrics as well as combinations of types. For example, supertoroids could be used for negative values and superellipsoids for positive values.

In the case the of superellipsoids, the trigonometric terms are assigned exponents as follows:

$$\underline{x}(\eta, \omega) = \begin{bmatrix} a_1 \quad \cos^{\epsilon 1} \eta \quad \cos^{\epsilon 2} \omega \\ a_2 \quad \cos^{\epsilon 1} \eta \quad \cos^{\epsilon 2} \omega \\ a_3 \quad \sin^{\epsilon 1} \eta \end{bmatrix}, \quad \begin{matrix} -\pi/2 \leq \eta \leq \pi/2 \\ -\pi \leq \omega < pi \end{matrix}$$

These exponents allow continuous control over the characteristics (the concavity or convexity) of the shape in the two major planes which intersect to form the shape, allowing a very simple, intuitive, abstract schema of shape specification. For example, $\epsilon 1 < 1$ and $\epsilon 2 < 1$ produces cuboid shapes, $\epsilon 1 < 1$ and $\epsilon 2 \sim 1$ produces cylindroid shapes, $\epsilon 1 > 2$ or $\epsilon 2 > 2$ produces pinched shapes while $\epsilon 1 = 2$ or $\epsilon 2 = 2$ produces faceted shapes. As can be seen in Figure 2, varying the exponents achieves smooth, understandable transitions in shape. Therefore, mapping data values to the exponents provides not only a continuous, automatic control over the shape's overall flavor, but a comprehensible shape mapping as well.

To produce understandable, intuitive shapes, we rely on the ability of su-

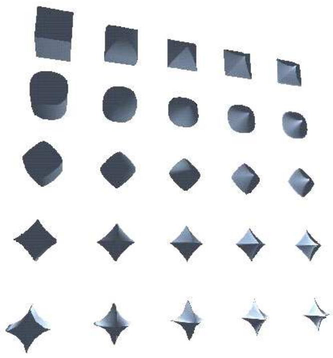

Fig. 2. Example superquadric shapes created by varying each exponent from 0 to 4.

perquadrics to create graphically distinct [13,14], yet related shapes. We encode two data dimensions in glyph shape in a manner that allows the easy separation of the shape characteristics.

4.3 Experimental Evaluation

We conducted a user test to find the set of superellipsoid parameters that most people would be able to distinguish. The central question being addressed is: can a JND (Just-Noticeable Difference) scale be established so the user can pre-attentively identify a meaningful difference among superellipsoids, or a pattern of superellipsoids. The experimental approach was based on the Weber-Fechner theory of Just-Noticeable Differences. This theory, established in the early 1900's, focuses on the absolute threshold: how much a stimulus must change in order for the observer to sense a difference. The difference detected is the difference threshold. JND scales have been established for many factors, such as intensity of tone (1/10), and pressure (1/7). We propose to establish an incremented scale for superellipsoids that are noticeable difference from one another.

The experimental setup was a computerized JND task in which a target superellipsoid and an alterable superellipsoid were presented to the participant (figure 3). The participant was instructed to match the alterable figure to the target figure in shape. The participants used the mouse to control a slider scale directly below both figures. They dragged the slider either left or right along the scale to modify the shape of the alterable superellipsoid.

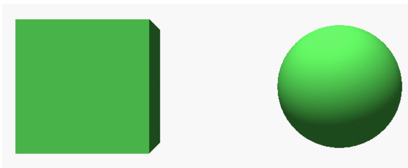

Fig. 3. Example of a target superquadric (right) and adjustable superquadric (left).

The computerized JND task consisted of three blocks of tests. The blocks were divided into three sub-blocks, each containing twenty trials. The first sub-block and the third sub-block, horizontal layout or vertical layouts, were counterbalanced. The horizontal layout (figure 3) contained two objects side by side, and the left object was the target. The instructions for the vertical layout were similar, with the target superellipsoid above the alterable figure. Each individual matching task had to be performed within a 10-second time limit; all times were recorded. Consequentially, if the participant had not completed the match within 10 seconds, a pop-up alarm box stating that time had elapsed would appear, trial was recorded but not statistically analyzed.

Fifteen women and sixteen men, between the ages of 18 and 48 years old, participated in the study. All participants were post-secondary university students. The average age of the males who participated in the experiment was 24.13 years (SD=4.79); the average age of the females was 24.40 years (SD= 7.07).

4.3.1 Targets

Twenty target objects were presented in random order, and the participant was asked to match the target with the adjustable superellipsoid. We used the same 20 targets across all trials so that statistical analysis could be performed per target. The size of the target and alterable shape measured 195 pixels in each dimension, presented on a 20 inch diagonal SGI monitor refreshed at 1280 pixels horizontal by 1024 pixels vertical. There are 1280 pixels displayed along a horizontal range of 350mm, yielding $350/1280 = .2734$ mm/pixel. Most shapes had a horizontal and vertical measure of $195 \times .2734 = 53$ mm or 2.1 inches. The cuboid shapes tended to have a larger apparent area because the near face of the cube is expanded due to perspective, at its largest 215 pixels, 59mm or 2.3 inches.

To simplify the search space, we used the superellipsoids that were symmetric about all 3 axes. This yields the shapes along the diagonal from the top left to the bottom right in figure 2. This is accomplished by setting the two superellipsoid exponents to the same value at all times. That is, $\epsilon 1 = \epsilon 2$ for all the superellipsoids in this experiment.

To bootstrap the process of selecting a set of targets for each participant to match, we first chose 20 candidate parameters (values of $\epsilon 1 == \epsilon 2$) from a simple linear ramp: $0.0, 0.2, 0.4, ...4.0$. We then performed some trial runs of the experiment using these parameters, and adjusted some parameter values up or down by one or two tenths to reflect our impression that some parameter ranges did not seem to change the shape much.

	Mean			Mean	
Target	Response	Std Dev	Target	Response	Std Dev
0.0	0.0015	0.0061	2.0	2.0565	0.1101
0.3	0.3203	0.0934	2.3	2.3122	0.1792
0.5	0.5304	0.0972	2.5	2.5673	0.1664
0.7	0.6620	0.0738	2.7	2.7471	0.1780
0.8	0.7780	0.1064	2.9	2.9329	0.1745
1.0	1.0089	0.0225	3.0	3.0815	0.1843
1.1	1.1432	0.0570	3.4	3.3760	0.1839
1.2	1.2892	0.1258	3.7	3.5413	0.1993
1.6	1.6162	0.1025	3.9	3.7443	0.1662
1.8	1.8330	0.1166	4.0	3.8064	0.1919

Table 1
Superquadric target parameters with the mean and standard deviation of the response.

4.3.2 Results

Table 1 shows the experimental parameter plus the experimental outcome.

The graph in figure 4 shows the standard deviations that were measured in user responses at each target. The interesting points to note are that the standard deviation at $\epsilon 1 = 0.0$ is very small. This is because the left limit of the adjustment slider was $\epsilon 1 = 0.0$, giving participants a strong indication that they had reached the correct parameter. Also, the shape at $\epsilon 1 = 0.0$ is a box, which is readily recognizeable.

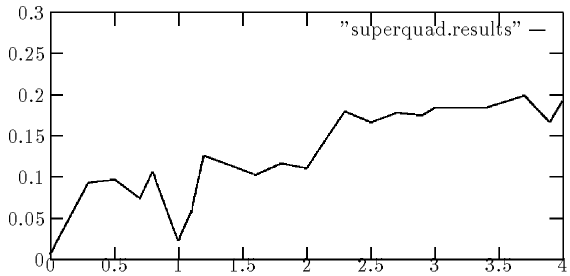

Fig. 4. Plot of Standard Deviations. The horizontal axis is the superquadric parameter value, and the vertical axis shows the experimentally measured standard deviation.

The standard deviation at $\epsilon 1 = 1.0$ is also very small. This value of $\epsilon 1$ yields a sphere, so it seems that participants were quite able to discriminate a sphere from the surrounding "not-quite sphere" shapes.

Aside from $\epsilon 1 = 0.0$ and $\epsilon 1 = 1.0$, the values of $\epsilon 1$ between 0 and 2 tended to yield a standard deviation around 0.1. Beyond $\epsilon 1 = 2.0$, the standard deviation is around 0.18, with a rising trend. Interestingly, although the right limit of the slider is at $\epsilon 1 = 4.0$, there is still a large standard deviation, indicating that participants genuinely did not know that the target shape was equal to the shape at the slider limit.

4.3.3 Estimating Number of JND Shapes

The goal of this study is to determine how many shapes are descriminable, and what the parameter values are of those shapes. This set of shapes would then form a table of shapes that could be selected by a data parameter in a visualzation system. For example, if there were 10 distinct shapes, then given a data parameter in the range [0..1], then values [0..0.1] would select shape 0, values [0.1..0.2] would select shape 1, and so on.

To determine this set of shape parameters, we computed a recursive linear estimation of paramater values. If we repeated the experiment with this new set of target values, participants would never be off in their response by more than 1 shape step. The data indicate that the participant responses are normally distributed about the means. Thus, setting the next estimated $\epsilon 1$ value to be 2 standard deviations away should yield a situation in which 68% of responses for each target should be correct, and almost all responses should be off by only one target.

Starting at one endpoint of the parameter space (say $\epsilon 1 = 4.0$), we then computed the next Just-Noticeably Different shape by subtracting $2 \times StdDev$. We in fact need to compute the standard deviation as it exists around the next parameter, since the standard deviation is not equal for all parameters:

$$\epsilon_{dest} = StdDev_{dest} + StdDev_{source} + \epsilon_{source} \tag{1}$$

Since $StdDev_{dest}$ is not yet known, we used an altered form of Newton's root-finding method to compute it:

$$\epsilon'_{dest} = StdDev_{source} + StdDev_{source} + \epsilon_{source} \tag{2}$$

ϵ'_{dest} is the first guess at the destination ϵ_{dest}. There will be a pair of target ϵ values named ϵ'_{left} and ϵ'_{right} that bracket ϵ'_{dest}, and these will have their associated standard deviations. The value of ϵ'_{dest} is used to linearly blend between the two standard deviations:

$$\alpha = \frac{\epsilon'_{dest} - \epsilon'_{left}}{\epsilon'_{right} - \epsilon'_{left}} \tag{3}$$

$$StdDev_{dest} = (1 - \alpha) \times StdDev_{left} + \alpha \times StdDev_{right} \tag{4}$$

Starting at $\epsilon 1 = 0.0$	Starting at $\epsilon 1 = 4.0$	Mean	Starting at $\epsilon 1 = 0.0$	Starting at $\epsilon 1 = 4.0$	Mean
0	0.010	0.005	1.037	1.042	1.039
0.016	0.030	0.023	1.133	1.177	1.155
0.042	0.063	0.052	1.318	1.393	1.355
0.086	0.117	0.102	1.545	1.610	1.577
0.161	0.206	0.184	1.763	1.828	1.795
0.289	0.355	0.322	1.987	2.054	2.020
0.466	0.541	0.504	2.255	2.341	2.298
0.641	0.727	0.684	2.583	2.695	2.639
0.825	0.881	0.853	2.928	3.056	2.992
0.966	0.975	0.971	3.286	3.425	3.356
			3.667	3.805	3.736
			4.072	4.000	4.036

Table 2
Estimated superquadric parameters derived from repeatedly adding 2 standard deviations to the current parameter value to generate the new one. One of these standard deviations is computed using a linear interpolation. The third column shows the mean of these two estimates.

Finally, the estimated $StdDev_{dest}$ is used to compute ϵ_{dest}, as shown in equation 1.

The new ϵ_{dest} is then put on the role of ϵ_{source}, and the preceding process is repeated to generate the desired list of perceptually-linear superellipsoid shapes. In situations where standard deviations don't change much, this procedure is reasonably accurate. With changing standard deviations, repeated iterations of equations 2 through 4 could make the estimate more accurate, but since the basis of this estimation procedure is statistically variable data, we did not do this.

Another issue is which endpoint to start at. We computed two parameter tables; one starting at $\epsilon 1 = 0.0$ and working up, and one starting at $\epsilon 1 = 4.0$ working down. Table 2 shows these results.

The results show that there are 22 distinct superellipsoid shapes, with 10 in the range $0.0 < \epsilon 1 < 1.0$, 5 in the range $1.0 < \epsilon 1 < 2.0$, and 7 in the range $2.0 < \epsilon 1 < 4.0$. Figure 5 shows the shapes that arise from this test.

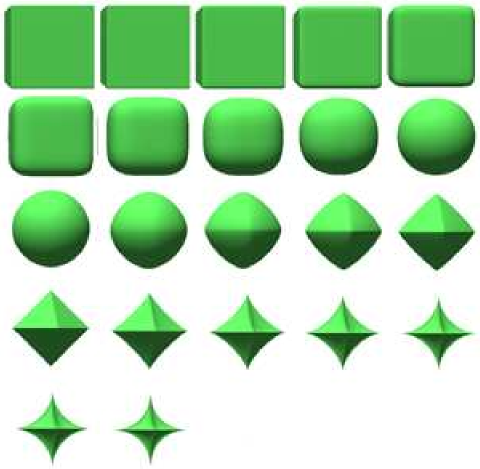

Fig. 5. Estimated superquadric results resulting from the average estimated parameter value in table 2

4.4 Procedural shape visualization with implicit surfaces

Implicit surfaces are a powerful geometric modeling tool which use an underlying density field to represent volumetric information. Implicit techniques provide smooth blending functions for individual implicit field sources [5]. Isosurface generation techniques are then used to create a geometric representation of this volumetric density field. This natural, smooth blending of multiple implicit sources makes implicit surfaces a natural choice for shape visualization.

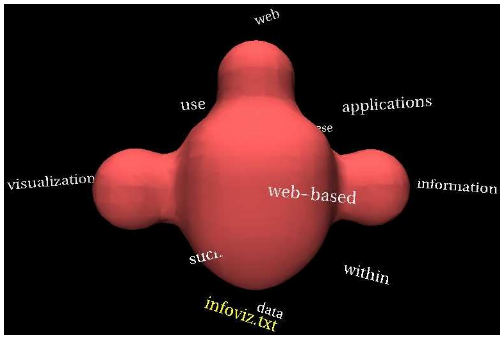

Fig. 6. Example implicit surface shape for a document on "Web-based Information Visualization." The three main bulges correspond to the frequency of the terms "web", "visualization", and "information".

For implicit shape visualization, we map up to 14 data dimensions to uniformly spaced vectors emanating from the center of a sphere. The length of each of these vectors is scaled based on the data value being visualized. At the end of each of these vectors, we place a uniform point source field function. We then generate an isosurface (typically, $F = 0.5$) from the resulting density field to create the glyph shape. This produces "blobby" shapes with bulges in a given direction indicating large data values for that dimension of the data. An example of the a single blobby glyph of this type can be seen in Figure 6.

Since size and spatial location are more significant cues than shape, the importance mapping of data values should be done in a corresponding order. In decreasing order of data importance, data values are mapped to location, size, color, and shape. In our experience, shape is very useful for local area comparisons among glyphs: seeing local patterns, rates of change, outliers, anomalies.

5 Implementation and Results

5.1 Information Visualization Results

We have applied procedurally-generated glyph shapes to the visualization of both scientific and information data. For information visualization, we have chosen an example of the visualization of "thematic" document similarities. Figure 8 shows a visualization of document similarities generated with the Telltale system [9]. The document corpus consists of 1883 articles from *The Wall Street Journal* from September and October 1989. Each glyph in Figure 8 represents a document in the corpus, and the document's X, Y, and Z position, color and shape each represent the similarity of the document to one of the 5 themes.

Document similarity to *gold price*, the *foreign exchange rate of the U.S. dollar*, and *federal reserve* are respectively mapped to the X, Y, and Z axes. The Y axis is visually indicated in Figure 8 by the vertical line, with the X axis going to the right and the Z axis going to the left. The bulk of the documents have very low similarity to all of these 3 themes, so their glyphs are clustered near the origin at the bottom center.

The documents outside this cluster exhibit two spatial patterns: a cluster of 9 documents to the bottom right and a vertical branch on the left. The right cluster indicates the small number of documents in the corpus that discuss both *gold price* and the *foreign exchange rate of the U.S. dollar*. The vertical branch depicts a larger collection of documents that discuss both *foreign exchange rate of the U.S. dollar* and the *federal reserve*.

A fourth attribute, similarity to *stock prices*, is inversely mapped to both superquadric exponents of the glyph shape, with highest similarity creating cuboids, then spheres, diamonds, and stars (lowest). Referring to the square array of sample glyphs in Figure 2, the similarity to *stock prices* maps to glyphs on the diagonal from the upper left to the lower right of Figure 2, with upper left indicating high similarity, and lower right indicating low similarity.

In Figure 8 the larger, rounder shapes along the vertical branch exhibit some significant relationship to *stock prices* while the more numerous star-shaped glyphs do not. Clearly the vertical branch contains articles relating *foreign exchange*, *federal reserve* and *stock prices*.

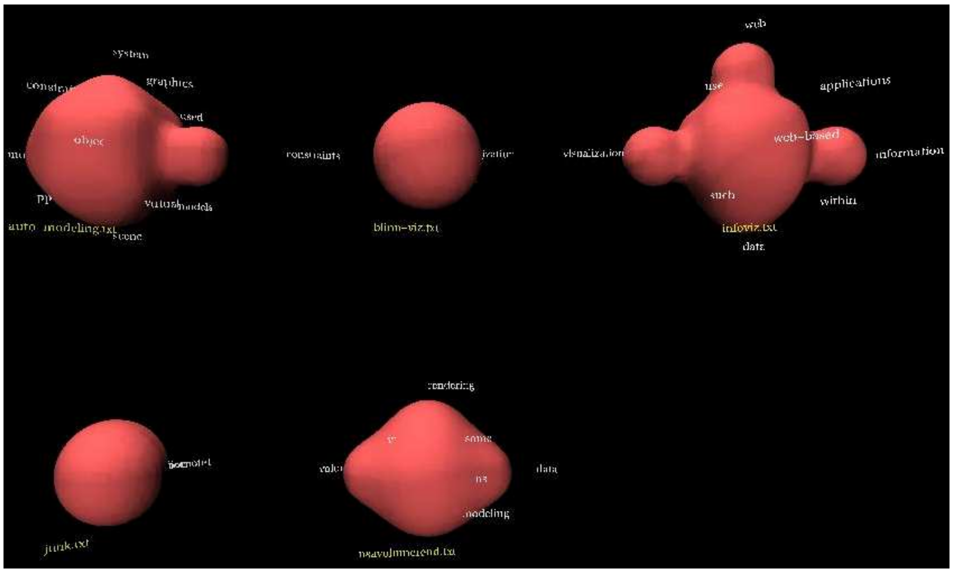

Fig. 7. Multiple documents term frequency visualized as implicit surface shapes. The document in the upper left and the upper right both have a high frequency of the term “information” (bulge to the right). The upper right blob is the same as that in figure 6

Glyph color is mapped inversely to similarity to *Manuel Noriega*. Most of the documents fall in the turquoise and purple range, indicating no significant relationship. However, the documents in the orange, red, and yellow-green range represent documents with a significant relationship to *Manuel Noriega*. Many of these documents mention the effect of the coup attempt against *Manuel Noriega* and its effect on *foreign exchange rate of the U.S. dollar* (vertical axis). The fact that these orange, red, and yellow-green documents are not in either of the branches indicates that these articles did not relate heavily to either *federal reserve* or *gold price*, and their star shape indicates no relationship with *stock prices*.

We have also used the implicit surface technique for text-based information visualization. The input data for this visualization was word frequency from a collection of text documents [12]. The frequency of the 14 most frequent words were mapped to vector length in positioning the density field sources: documents with common word usage exhibit similar shapes. The results of this process can be seen in Figure 7. From quick examination of this figure, documents with similar topics and word usage are easily distinguishable.

5.2 Scientific Visualization Results

We have used this system to examine several scientific visualization data sets. Figure 9 shows the visualization of a magnetohydrodynamics simulation of the solar wind in the distant heliosphere (20 times the distance of the Earth to the Sun). The simulation data is a $64 \times 64 \times 64$ grid containing the vector vorticity and velocity for the simulation.

Opacity is used to represent vorticity in the j direction, so that the 6 vortex tubes (only 4 are visible) represent zones in space where this vorticity is somewhat larger than zero. Glyph shape is based inversely on the velocity in the j direction. Positive velocities are displayed as larger, rounder to cuboid shapes and negative velocities are displayed as spiky, star-like shapes. Zero velocity is represented by the diamond shape. The overall columnar pattern of the data is not disturbed by the introduction of the shape mapping, but the velocity variation can still be seen as we traverse the lengths of the tubes. In this case, values close to zero in terms of j vorticity (still fluid) have been masked out.

Figure 10 is a visualization of the same magnetohydrodynamics data, but with the opacity, color, and glyph shape all mapped to the j component of vorticity. Negative vorticity components produce concave shapes (blue stars), while positive values produce convex shapes (orange cuboids and ellipsoids). The use of this data mapping clearly shows three tubes with negative j vorticity and three tubes with positive j vorticity.

6 Conclusions

We have developed several new techniques for intuitive, comprehensible creation of glyph shapes. These techniques are based on procedural shape generation and increase the number of dimensions of data that can be comprehensibly visualized in a glyph-based visualization system. Superquadric functions, fractal surface displacement, and implicit surfaces have been shown to be useful techniques for the automatic generation of glyph shapes and the visualization of multi-dimensional data. Superquadrics are a natural technique for shape visualization of one to two data dimensions. Fractal surface displacement seems useful for small categorizations (4 to 6 easily discernible shapes) of a single data dimension and can be added to the other two techniques as a secondary shape cue. Implicit surface techniques show great promise in visualizing between eight and twenty data dimensions with shape. All of these procedural shapes allow the intuitive understanding of data variation among glyphs, while preserving the global data patterns. We have shown the value of these techniques for both multi-dimensional information and scientific visualization.

References

[1] A. Barr. Superquadrics and angle-preserving transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.

[2] J. Bertin. Semiology of graphics, 1983.

[3] William S. Cleveland. *The Elements of Graphing Data.* Wadsworth Advanced Books and Software, Monterey, Ca., 1985.

[4] David Ebert, Chris Shaw, Amen Zwa, and Cindy Starr. Two-handed interactive stereoscopic visualization. *Proceedings IEEE Visualization '96*, October 1996.

[5] David S. Ebert. Advanced geometric modeling. In Allen Tucker Jr., editor, *The Computer Science and Engineering Handbook*, chapter 56. CRC Press, 1997.

[6] David S. Ebert, F. Kenton Musgrave, Darwyn Peachey, Ken Perlin, and Steven Worley. *Texturing and Modeling: A Procedural Approach, Second Edition.* AP Professional, 1998.

[7] J. D. Foley and C. F. McMath. Dynamic process visualization. *IEEE Computer Graphics and Applications*, 6(3):16–25, March 1986.

[8] Andrew Parker, Chris Cristou, Bruce Cumming, Elizabeth Johnson, Michael Hawken, and Andrew Zisserman. The Analysis of 3D Shape: Psychological principles and neural mechanisms. In Glyn Humphreys, editor, *Understanding Vision*, chapter 8. Blackwell, 1992.

[9] Claudia E. Pearce and Charles Nicholas. TELLTALE: Experiments in a Dynamic Hypertext Environment for Degraded and Multilingual Data. *Journal of the American Society for Information Science (JASIS)*, 47, April 1996.

[10] Frank J. Post, Theo van Walsum, Frits H. Post, and Deborah Silver. Iconic techniques for feature visualization. In *Proceedings Visualization '95*, pages 288–295, October 1995.

[11] W. Ribarsky, E. Ayers, J. Eble, and S. Mukherja. Glyphmaker: creating customized visualizations of complex data. *IEEE Computer*, 27(7):57–64, July 1994.

[12] Randall M. Rohrer, David S. Ebert, and John L. Sibert. The Shape of Shakespeare: Visualizing Text using Implict Surfaces. In *Proceedings Information Visualization 1998*, pages 121–129. IEEE Press, 1998.

[13] H. Senay and E. Ignatius. A knowledge-based system for visualization design. *IEEE Computer Graphics and Applications*, 14(6):36–47, November 1994.

[14] H. Senay and E. Ignatius. Rules and principles of scientific data visualization. *ACM SIGGRAPH HyperVis Project, http://ironduke.cs.gsu.edu/classes/hypervis/percept/visrules.htm*, 1996.

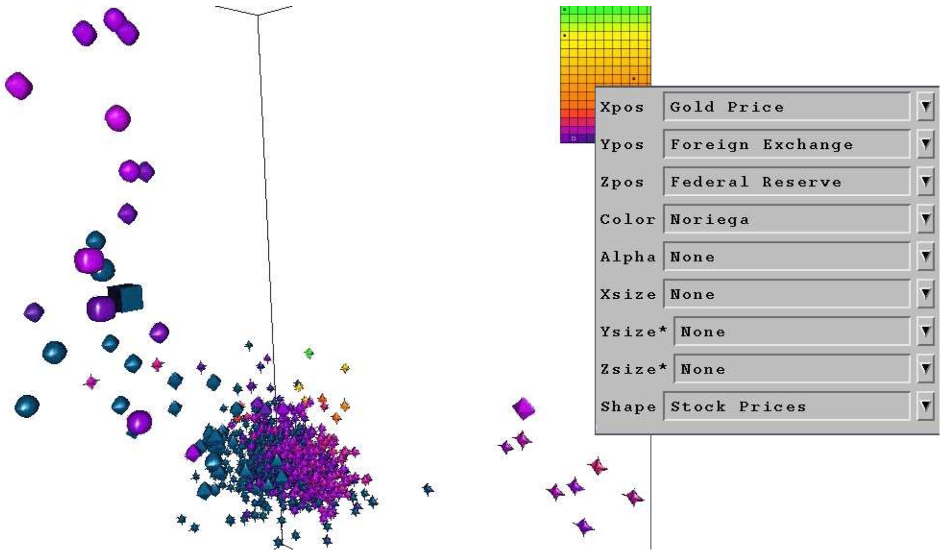

Fig. 8. Three-dimensional visualization of 1833 documents' relationship to gold price, foreign exchange, the federal reserve, stock prices, and Manuel Noriega.

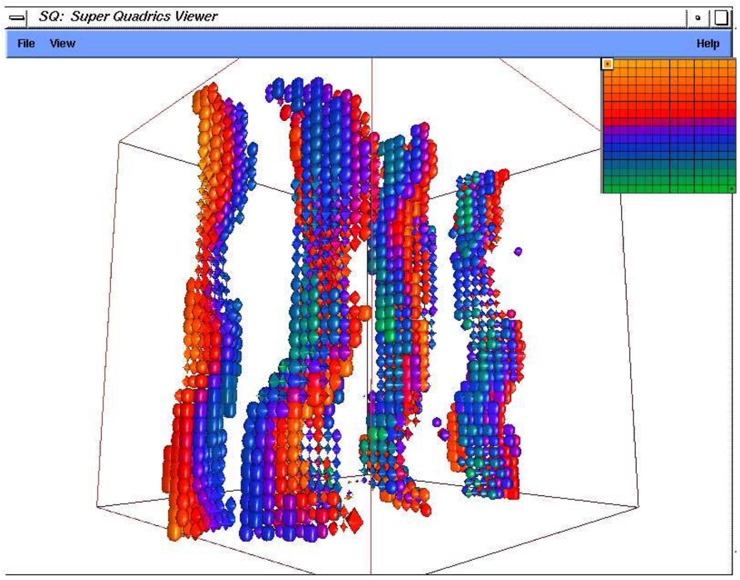

Fig. 9. Visualization of a magnetohydrodynamics simulation of the solar wind in the distant heliosphere showing both velocity components and vorticity components of 6 vortex tubes.

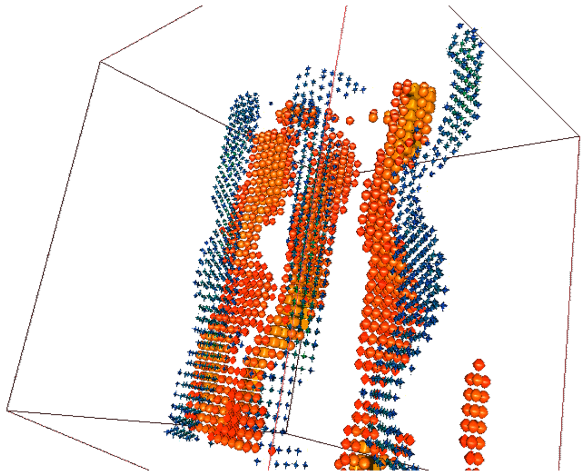

Fig. 10. Visualization of a magnetohydrodynamics simulation of the solar wind in the distant heliosphere displaying 3 vortex tubes with positive j vorticity (cuboids and ellipsoids) and 3 vortex tubes with negative j vorticity (stars).