Minimally-immersive Flow Visualization

David S. Ebert Purdue University [*]
Christopher D. Shaw Georgia Tech. [†]

November 22, 2000

Abstract

This paper describes a minimally immersive interactive system for flow visualization of multivariate volumetric data. The system, SFA, uses perceptually-motivated rendering to increase the quantity and clarity of information perceived. Proprioception, stereopsis, perceptually-motivated shape visualization, and three-dimensional interaction are combined in SFA to allow the three-dimensional volumetric visualization, manipulation, navigation, and analysis of multivariate, time-varying flow data.

Index Terms: SFA – Stereoscopic Field Analyzer, Flow Visualization, Two-Handed Interaction, 3D Volumetric Interaction, Desktop Virtual Environments, Glyph Rendering, Superquadric surfaces,

1 Introduction

The simultaneous visualization of multi-dimensional flow data is a difficult task. The goal is not only the display of multi-dimensional data, but the *comprehensible display* of the flow data. Glyph, or iconic, visualization allows multiple data values to be encoded in the parameters of the icons [14]. The shape, color, 3D size, transparency, and orientation of the glyph can be used to visualize data values. Glyph rendering [14, 15, 9] is an extension to the use of glyphs and icons in numerous fields, including cartography, logic, semiotics, and pictorial information systems.

[*]School of Electrical and Computer Engineering, 1285 EE Building, West Lafayette, IN, 47907-1285, USA. Phone: 765-494-3536, Fax: 765-494-3544, Email: ebertd@purdue.edu

[†]GVU Center, College of Computing, Georgia Institute of Technology, 801 Atlantic Drive, Atlanta, GA, 30332-0280, USA. Phone: 404-894-6328, Fax: 404-894-2970, Email: cdshaw@cc.gatech.edu

This paper describes our system, SFA (Stereoscopic Field Analyzer), for the interactive display of volumetric data which utilizes the capabilities of the human perception system to increase the information perceived from the visualization system. As described in [7], the system provides a minimally immersive interactive visualization tool to increase the understanding of volumetric data while being affordable on desktop workstations. We have extended this work to develop a perceptually-motivated visualization system that harnesses the power of the human perceptual system to increase the quantity of information conveyed from our system. Our system takes advantage of the priority structure of human visual perception [6, 3], stereopsis, motion, and proprioception (the brain's unconscious awareness of the sense of body in space) to create meaningful visualizations from scientific data. SFA uses a glyph's location, 3D size, color, orientation, and opacity to encode up to 8 attributes of flow data per glyph.

We have developed new techniques for automatic glyph shape generation that allow perceptualization of data through shape variation. Location, color, size, orientation, and opacity are more significant perceptual cues than shape [6]; however, shape variation can also be effectively used to convey related scalar flow parameters, especially in an interactive system.

Several researchers have examined the use of virtual reality environments for visualization. Examples include Bryson and Levit's Virtual Windtunnel [4], UNC-Chapel Hill's work on the topics of exploratory molecular visualization [2] and real-time exploration of atomic scale phenomena [20].

Hinckley et al. [10] built a two-handed 3D neurosurgical visualization system where the user sits in front of a high-resolution monitor and manipulates the scene using two 3D magnetic trackers. The left hand holds the brain model represented by a tracker mounted inside a doll's head, and the right hand manipulates a cutting plane represented by a flat plate connected to a second tracker. The user moves the cutting plane to interactively generate cutaway views of the brain. New users immediately understand how the manipulation works and need essentially no training to use the system.

Most previous techniques for interactive visualization of volumetric flow data use 2D interfaces or immersive head mounted displays. Our approach to interactive visualization combines glyph-based volume rendering with a perceptually-motivated minimally-immersive interaction metaphor to provide interactive visualization, manipulation, and exploration of multivariate, volumetric data. Careful data value mapping to glyph rendering attributes allows better comprehension of the multivariate data than can be achieved using isosurfaces or direct volume rendering. The rendering techniques used in our system are briefly described first, followed by a description of our procedural glyph generation. We then discuss the two-handed metaphor for interaction and show the results of applying these new techniques to flow visualization. Finally, we describe future directions for research in this area.

2 Rendering Within SFA

SFA allows the visualization of both regular and irregular grids of volumetric data through the use of a glyph at each grid point [14, 15]. SFA also provides the ability to render more than one set of volumetric data using the same grid. It has many of the advantages of volume rendering, while avoiding the limitations and difficulties of iso-surface rendering. An additional feature of SFA is the ability to interactively visualize and manipulate time-varying multivariate volumetric data.

Because of the need to resample the space, standard voxel-based techniques for visualization are not appropriate when the sample points occur on an irregular grid, as is common in flow visualization. Certain assumptions must be made about the underlying physical properties of the space being rendered that may not be true for the particular data set in question, potentially resulting in a meaningless or misleading visualization.

2.1 User Control

Within the basic framework of SFA, the user may easily control the visualization in numerous ways, including selection of data sets mapped to glyph color, opacity, size, shape, and orientation. To avoid the clutter induced by displaying insignificant data, the user may select the data ranges to be displayed, which may be a collection of disjoint intervals.

As a further aid in reducing visual clutter, the user may select maximum and minimum grid extents to display in each dimension. The user may choose to concentrate on a particular subset of the volume, culling the unwanted outlying glyphs. Similarly, the user may choose to display only every n^{th} grid point along each dimension. These two controls serve to reduce the number of glyphs drawn, thereby reducing the amount of time taken to draw the scene.

To create a good 3D impression, SFA renders the scene using either opaque or translucent glyphs, and performing standard Z-buffer hidden-surface removal. The scene may be drawn in stereo using a pair of Liquid Crystal Shutter Glasses. We use the standard parallel camera model to render the 3D scene, which is identical in all respects except that it must be rendered twice, once for each eye. Stereopsis is a very important visual cue for 3D depth perception in most users.

3 Shape Visualization

Cleveland [6] 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 [3]. However, these orderings of visual effectiveness are based on two-dimensional visualization systems. Some recent work has been performed on three-dimensional perception of shape, indicating that shading based on shape is perceptually significant [12].

SFA employs glyph position in 3D, 3D scale (corresponding to Cleveland's length, area and volume) and color, and in the vector-based flow visualization, slope angle. The next opportunity for encoding values 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 [14]. 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 [3]. We have chosen the procedural generation of glyph shapes using superquadrics [1] because superquadrics offer the required shape interpolation mechanism.

3.1 Procedural Shape Visualization Using Superquadrics

Because of the need for meaningful glyph design and the complexity of the problem, we opted for a procedural approach, which allows flexibility, data abstraction, and freedom from specification of detailed shapes [8]. Our goal for glyph design was to allow the automatic mapping of data to shape in a comprehensible, easily controllable manner. Superquadrics are a natural choice to satisfy this goal. 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 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:

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

These exponents allow continuous control over the characteristics (in some sense the "roundness" or "pointiness") 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, as Barr states, $\epsilon 1 < 1$ 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 1, 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. By using superquadrics, we can provide the appropriate shape visual cues for discerning data dimensions mapped to glyph shape while not distracting from the cognition of global data patterns.

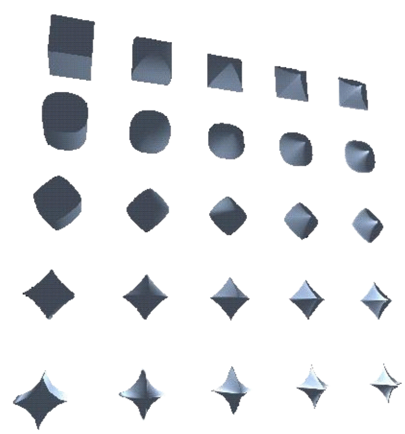

Figure 1: Example superquadric shapes created by varying each exponent from 0 to 4.

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 [13] using curvature information of the 2D silhouette contour and, for 3D objects, curvature information from surface shading [12]. 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 [3]. Senay points out that shape, size, texture, orientation, transparency, hue, saturation, brightness, and transparency are retinal properties of marks that can encode information [16, 17]. To produce understandable, intuitive shapes, we are relying on the ability of superquadrics to create graphically distinct, yet related shapes. We are encoding two data dimensions to glyph shape in a manner that allows the easy separation of the shape characteristics.

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 were 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, outlyers, anomalies. We chose to map either one independent variable to both glyph exponents or two related variables to each glyph exponent to ensure the understandability of the shapes.

4 Two-Handed Minimally-Immersive Interface

SFA uses a pair of 3D magnetic trackers and stereo glasses to provide a minimally-immersive desktop visualization system. The user sits in front of a graphics console that has a screen, keyboard, mouse, and two 3D sensors (see Figure 2). Each 3D sensor, or *Bat*, has three buttons glued onto the surface, as shown in Figure 3. The user interacts with the system by manipulating the two trackers, and pressing the tracker buttons to invoke operations.

We chose the minimally immersive style because it allows 2D functionality to remain if 3D trackers are not available at the user's workstation. Thus, SFA has a certain amount of interface redundancy to allow for different hardware configurations. We chose not to adopt the immersive style, in which a Head Mounted Display (HMD) is used to display the scene to the user. The nominal benefit of this style is that the HMD gives a large panoramic field of view, but it comes at a cost of cutting the user off from traditional I/O devices like the screen, keyboard, and mouse.

The combination of two-handed interaction and stereo viewing allows us to harness the user's proprioceptive sense to convey information. Magnetic trackers

Figure 2: The two-handed stereo interface to SFA.

Figure 3: Polhemus sensor with attached buttons.

also allow easier manipulation of complex 3D objects than a mouse [21], because the user does not have to mentally break down the 3D task into a sequence of 2D operations. The use of a 3D device allows the user to directly manipulate the objects of interest without intermediate steps. Two 3D devices give the user access to double the spatial bandwidth, since both hands can be employed in parallel to quickly achieve the desired operation.

Each 3D sensor has a distinct role, with the dominant hand being responsible for picking and manipulation, and the less-dominant hand being responsible for context setting of various kinds. For the sake of rhetorical convenience, we will refer to the dominant hand as the right hand and the less-dominant hand as the left, but the system is ambidextrous because the Polhemus trackers are symmetric and can be handled with equal ease by either hand.

We use two sensors because their simultaneous use takes advantage of people's innate proprioceptive knowledge of where their two hands are in space. Guiard

[11] gives psychophysical evidence for the idea that the left and right hands quite often act as elements in a kinematic chain. For right-handed people, the left hand acts as the base link of the chain. The right hand's motions are based on this link, and the right hand finds its spatial references in the results of motion of the left hand. Also, the right and left hands are involved in asymmetric temporal-spatial scales of motion (right hand for high frequency, left hand for low frequency).[1] The Two-Handed interface to SFA uses this natural division of manual labour by assigning the (low-frequency) setting of spatial context to the left hand, and the (high-frequency) selection and picking operations to the right.

This interface is similar to one designed by Shaw for free-form surface modeling [18], and some of the interaction techniques are derived from it. Hinckley et al.'s system [10] offers a similar assignment of hand roles, but there is intentionally only a very limited set of interaction techniques. Other work at the University of Virginia [19] has further explored the idea of two-handed interfaces using an object, or *prop*, attached to each sensor.

4.1 Left Hand Operations

In SFA, the left hand has three tasks to perform:

- Manipulate the position and orientation of the entire scene.
- Select the drawing context from a 3D tracker-based hierarchical menu.
- Select the constraint axis for bounding-box adjustment.

Repositioning the scene is initiated by clicking the left bat button 3 (the button near the wire). Clicking button 3 attaches the volume to the left cursor, and clicking it again leaves the volume in place. This clutching mechanism allows the user to spend a large amount of time moving the workpiece around without the burden of continuous button pressure.

The middle button (button 2) on the left bat pops up a hierarchical Sundial menu [18], as shown in Figure 4, which is a menu controlled by the orientation of the left bat. The menu choices are arrayed on the circular plate of the Sundial, each on its own pie-shaped sector. The desired item is picked by pivoting the *shadow stick* about its base so that the stick's endpoint lies visually in front of the sector. The base of the shadow stick is located at the center of the plate and, when the menu first pops up, the stick is aligned with the line of sight, pointing directly at the user. Around the base of the stick is an inner circle which can be used to indicate *no selection*. The position of the Sundial is fixed rigidly to the left bat, and the dial lies parallel to the projection plane.

[1] For left-handed people, the roles of right and left are reversed.

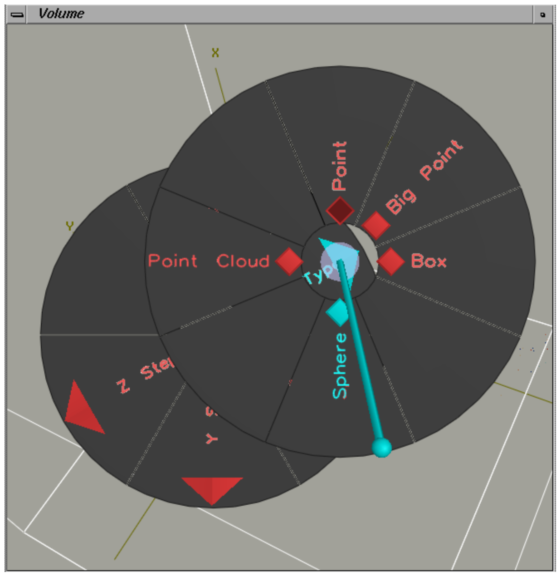

Figure 4: A 3D hierarchical Sundial menu.

This is a 3D version of the pie menu [5] which encodes the position of the 2D cursor by projecting the 3D position of the endpoint of the shadow stick onto the 2D Sundial plate. This main menu in SFA contains a submenu to select the glyph type, a submenu for each of X, Y, and Z step interval selection, and cursor control menu that scales the scene up or down and reorigins the bats to the center of the workspace.

The Sundial menu is provided to allow ease in selection while both hands are occupied manipulating the bats. If only mouse-based menus were provided, then the user would have to continually pick up and put down the right bat in order to interact with the scene.

4.2 Right Hand Operations

The right hand has two tasks to perform:

- Select 3D volume bounding boxes.
- Pick a glyph to print out its value.

The user may decide to concentrate on a subvolume of the 3D space, and temporarily stop drawing the glyphs outside this subvolume. To select this subvolume, the user presses button 1 (furthest from the wire) on the right bat to place one corner of the subvolume bounding box, drags the right bat to the opposite corner, then releases the button. While the button is pressed, the bounding box is drawn as a semitransparent box, allowing the user to see which glyphs will be enclosed by the bounding box as it is dragged out. The bounding box disappears when the button is released, and SFA draws only the glyphs that were within the bounding box. The user may restate the bounding volume at any time by sweeping out a new bounding box.

4.2.1 Multiple Subvolumes

SFA actually makes up to 8 bounding boxes available, and displays the glyphs in the union of these boxes. Each bounding box can be individually resized, reshaped, or deleted as the user sees fit. This bounding box control is provided by a menu of thumbnails in a column on the right of the main window (Figure 5). Each thumbnail always shows its respective semitransparent bounding box lying in the main volume. Glyphs are not drawn in the thumbnails to allow quick rendering. The currently-selected bounding box has its thumbnail highlighted, and the user can restate the currently-selected bounding box by sweeping out a new box in the main window. When the user moves the right cursor into a thumbnail, the thumbnail highlights, and the user may select one of 4 bounding box operations from a partial sundial menu by pressing right button 1. Thumbnail intersection is actually a 2D operation where the projected location of the cursor is intersected with the little thumbnail window, allowing users to more easily select the thumbnail.

The 4 bounding box operations available from the thumbnail are:

- *Restate* the current bounding box.
- Add a *new* bounding box.
- *Delete* the current bounding box.
- *Edit* the current bounding box.

The *restate* operation is the default if right button 1 is pressed in the main window. The *new* bounding box operation will add a bounding box to the current set when right button 1 is used to sweep out a bounding box. When the sweep-out starts, a new thumbnail is added to show the new bounding box. *Delete* simply removes the current bounding box from the list, and its corresponding thumbnail disappears.

The *edit* operation allows constrained editing of the current bounding box. When edit is selected, the semitransparent bounding box is drawn in the main window, with one of its faces highlighted to indicate that it is to be moved. This

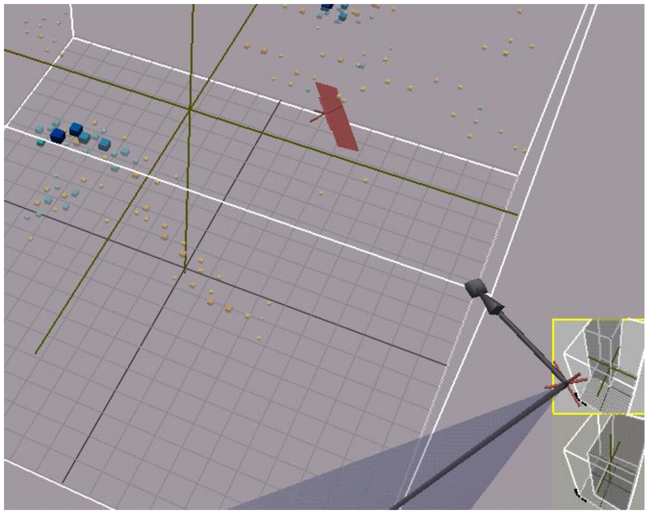

Figure 5: SFA with two subset bounding boxes active. Each bounding box is displayed in its respective thumbnail image on the lower right. The right cursor is currently in the upper thumbnail, which is highlighted by a surrounding yellow line. To make the thumbnails visually comprehensible, the image has been cropped to remove the left and upper parts of the main window.

highlighting is determined by a combination of information from the left and right bats. The orientation of the left bat is used to state the axis along which the bounding box will be edited. The right bat's position with respect to the bounding box's center determines which face is to be moved. The closest face to the right cursor along the constraint axis is the one highlighted for editing. Thus, while the left bat is geometrically stating the constraint axis, the right bat can be selecting a face and starting the edit operation.

To effect the edit, the user presses right button 1, drags the highlighted face along the constraint axis (perpendicular to the face), and releases right button 1. In order to avoid any inadvertent changes in snap axis, the constraint axis remains fixed while the edit operation takes place, so the left hand does not need to hold a steady orientation during editing.

The axis that is chosen by the left bat is the canonical axis closest to the bat's Z vector. Both cursors change their shape to reflect constrained edit mode, with the left cursor changing to a short pole through a square, and the right cursor adding a short pole that snaps to the current constraint axis. The left cursor does not snap but remains tied to the left bat, because the canonical axis closest to the bat's Z vector (the pole axis in the left cursor) determines the constraint axis. The purpose of constrained editing is to allow the user fine control over bounding box shape without completely restating it from scratch. The user perceives an investment of effort in stating a bounding box, and editing allows the user to preserve that investment.

4.2.2 Glyph Selection

To select a glyph, the user orients a probe into the volume, and the closest glyph has its value printed and optionally passed through a socket connection to software to allow closer examination of this data point. A probe, represented by a narrow cylindrical shaft, is attached to the right cursor, and the user controls the position and orientation of the probe with the right bat. The distance from this probe is computed for each control point using a specialized distance metric called the *probe metric* [7], and the grid point generating the smallest probe metric value is the one picked. The probe metric is designed to allow easy selection of objects that are small and/or far away.

Because the user is selecting among many objects, the probe axis is drawn to some arbitrary length, and a translucent cone is drawn. The probe is the cone axis, and a spotlight is attached to the cursor that has identical visual geometry to the translucent cone. As the user sweeps the probe about the scene, the objects that fall within the cone intersect visually with the cone wall and are highlighted by the spotlight. This is purely a visual cue to help the user locate the cone in 3D visual space.

5 Results and Conclusions

SFA is a perceptually-motivated, minimally-immersive visualization system that allows multi-dimensional flow data to be visualized, analyzed, and manipulated easier than traditional visualization systems.

In terms of rendering, the glyph approach does not suffer the initial costs of isosurface rendering or voxel-based volume rendering and, at the same time, offers the capability of viewing the entire space. Thus, SFA allows the user to immediately examine and explore the data.

SFA has been used to visualize many scientific data sets, including gas plasma vortex data and other magnetofluid simulations. Figure 6 shows a 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 $33 \times 33 \times 33$ grid containing the vector vorticity for the simulation.

The power of shape visualization can be seen in Figure 7, which is another magnetohydrodynamics simulation of the solar wind in the distant heliosphere. In this simulation, the 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 8 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). Using this data mapping clearly shows three tubes with negative j vorticity and three tubes with positive j vorticity.

SFA's users (scientists at NASA) have commented on the ease of use of the two-handed system, and have been enthusiastic about its use for exploring complex fluid dynamics simulations. One of the main benefits of SFA is the ability to quickly understand the contents of the 3D space because of the use of perceptually-based visualization techniques. Perceptually-motivated glyph design allows multi-valued flow data to be *comprehensibly* visualized. The two handed interface harnesses the user's proprioceptive sense of the 3D space, which is impossible in other approaches. In combination with stereoscopic display, a powerful 3D impression can be given to the user. Our system, therefore, allows volumetric visualization, manipulation, navigation, and analysis of multivariate, time-varying volumetric data, increasing the quantity and clarity of the information conveyed from the visualization system.

6 Future Extensions

Several extensions can be made to the current system, including further optimization of the rendering algorithms to increase the performance of the system and the use of texture memory to provide annotations to glyphs and help distinguish glyph types.

7 Acknowledgments

This project was supported by the National Science Foundation under grant CCR-9409243 and by NASA under grant NAG 5-2893.

References

[1] A. Barr. Superquadrics and Angle-Preserving Transformations. *IEEE Computer Graphics and Applications*, 1(1):11–23, 1981.

[2] Lawrence D. Bergman, Jane S. Richardson, David C. Richardson, and Frederick P. Brooks, Jr. VIEW – an Exploratory Molecular Visualization System with User-Definable Interaction Sequences. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 117–126, August 1993.

[3] J. Bertin. Semiology of Graphics, University of Wisconsin Press, 1983.

[4] Steve Bryson and Creon Levit. The Virtual Wind Tunnel. *IEEE Computer Graphics and Applications*, 12(4):25–34, July 1992.

[5] Jack Callahan, Don Hopkins, Mark Weiser, and Ben Shneiderman. An Empirical Comparison of Pie vs. Linear Menus. In *Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems*, pages 95–100. ACM SIGCHI, Washington, DC, 1988.

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

[7] David Ebert, Chris Shaw, Amen Zwa, and Cindy Starr. Two-Handed Interactive Stereoscopic Visualization. *Proceedings IEEE Visualization '96*, October 1996.

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

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

[10] John C. Goble, Ken Hinckley, Randy Pausch, John W. Snell, and Neal F. Kassel. Two-Handed Spatial Interface Tools for Neurosurgical Planning. *Computer*, 28(7):20–26, 1995.

[11] Yves Guiard. Asymmetric Division of Labor in Human Skilled Bimanual Action: The Kinematic Chain as a Model. *The Journal of Motor Behavior*, 19(4):486–517, 1987.

[12] Victoria Interrante, Penny Rheingans, James Ferwerda, Rich Gossweiler, and Toms Filsinger. Principles of Visual Perception and its Applications in Computer Graphics. In *SIGGRAPH 97 Course Notes, No. 33*. ACM SIGGRAPH, August 1997.

[13] Andrew J Parker, Chris Christou, Bruce G Cumming, Elizabeth B Johnston, Michael J Hawken, and Andrew Zisserman. The Analysis of 3D shape: Psychophysical Principles and Neural Mechanisms. In Glyn W Humphreys, editor, *Understanding Vision*, chapter 8. Blackwell, 1992.

[14] 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.

[15] W. Ribarsky, E. Ayers, J. Eble, and S. Mukherjea. Glyphmaker: Creating Customized Visualizations of Complex Data. *IEEE Computer*, 27(7):57–64, July 1994.

[16] H. Senay and E. Ignatius. A Knowledge-based System for Visualization Design. *IEEE Computer Graphics and Applications*, 14(6):36–47, November 1994.

[17] H. Senay and E. Ignatius. Rules and Principles of Scientific Data Visualization. *ACM SIGGRAPH HyperVis Project, http://homer.cs.gsu.edu/classes/percept/visrules.htm*, 1996.

[18] Chris Shaw and Mark Green. THRED: A Two-Handed Design System. *Multimedia Systems Journal*, 3(6):to appear, November 1995.

[19] Richard Stoakley, Matthew J. Conway, and Randy Pausch. Virtual Reality on a WIM: Interactive Worlds in Miniature. In *Proceedings of ACM CHI'95 Conference on Human Factors in Computing Systems*, volume 1, pages 265–272, Denver, Colorado, May 7-11 1995.

[20] Russell M. Taylor, II, Warren Robinett, Vernon L. Chi, Frederick P. Brooks, Jr., William V. Wright, R. Stanley Williams, and Eric J. Snyder. The Nanomanipulator: A Virtual Reality Interface for a Scanning Tunnelling Microscope. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 127–134, August 1993.

[21] Colin Ware and Danny R. Jessome. Using the Bat: A Six-Dimensional Mouse for Object Placement. *IEEE Computer Graphics and Applications*, 8(6):65–70, November 1988.

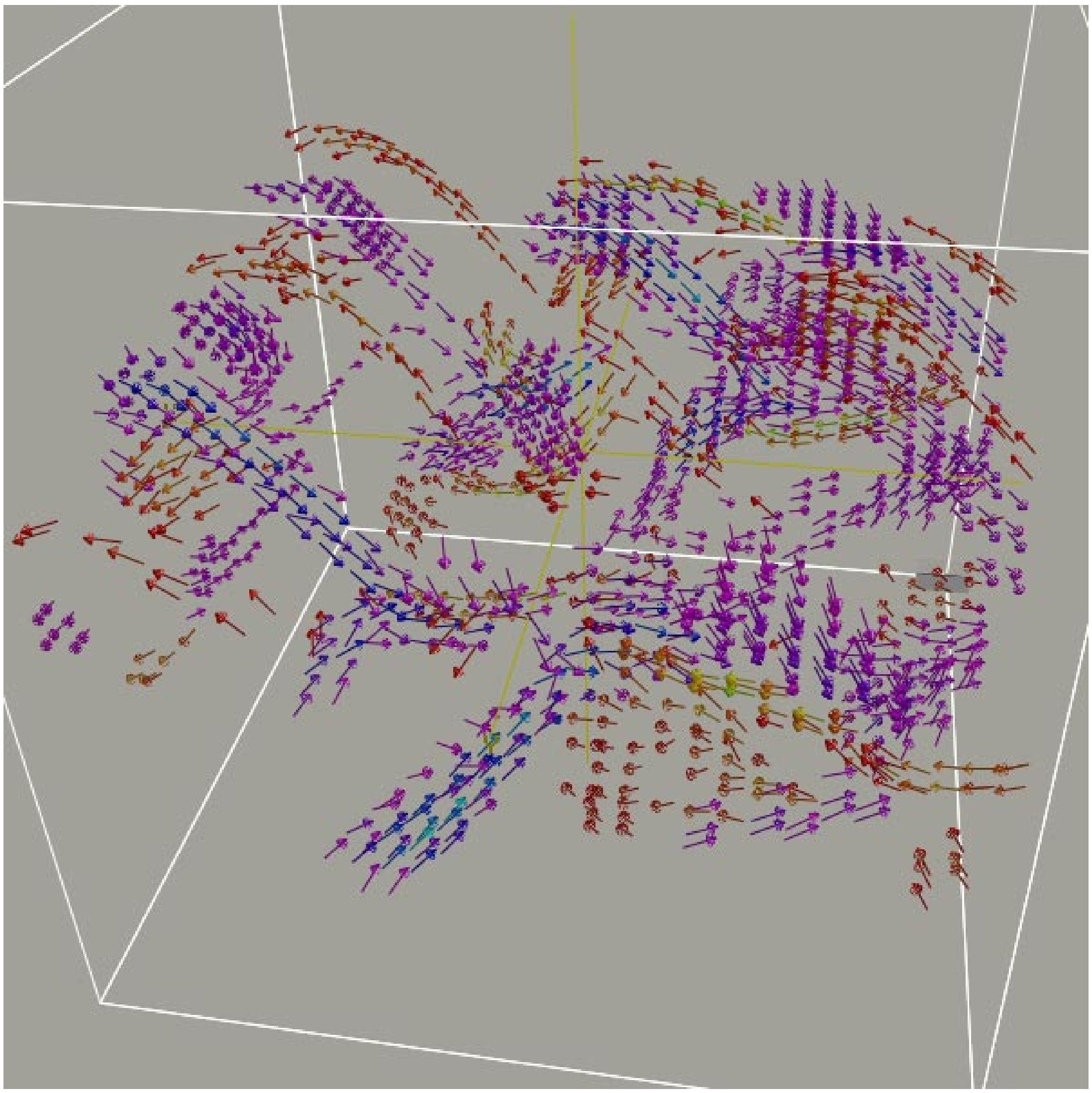

Figure 6: Visualization of a magnetohydrodynamics simulation of the solar wind in the distant heliosphere.

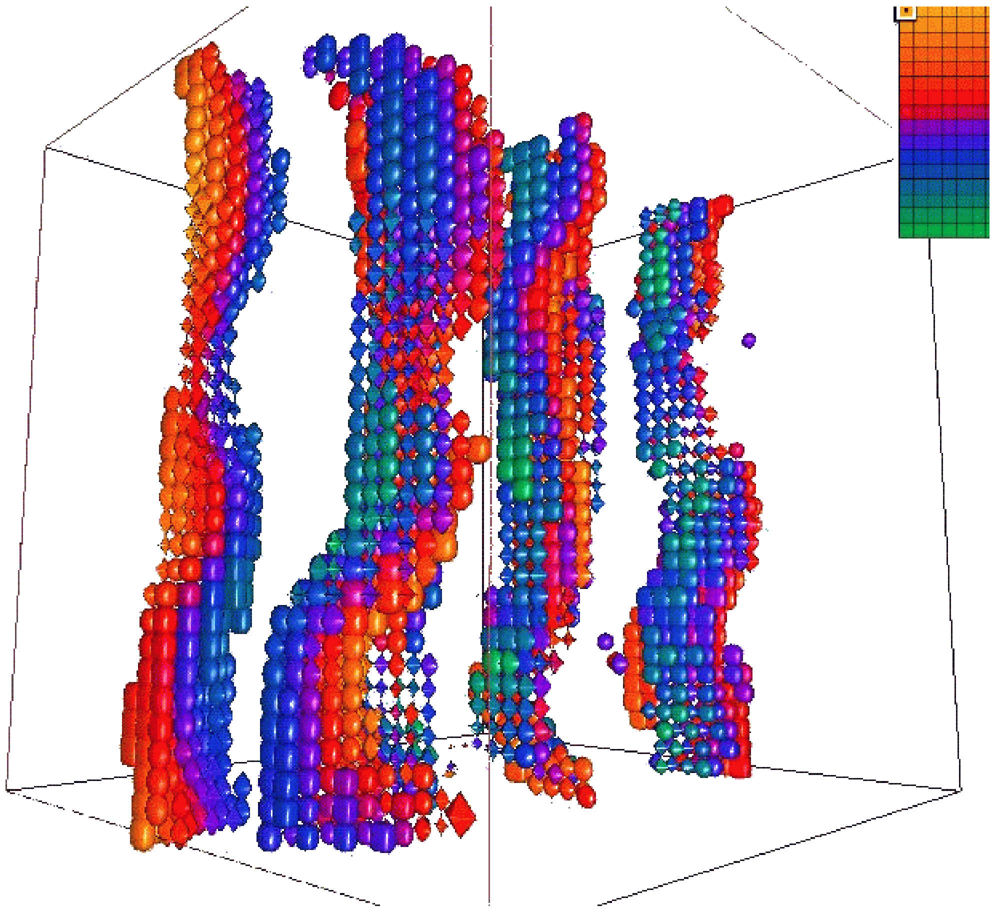

Figure 7: Visualization of a magnetohydrodynamics simulation of the solar wind in the distant heliosphere showing both velocity components and vorticity components of 6 vortex tubes.

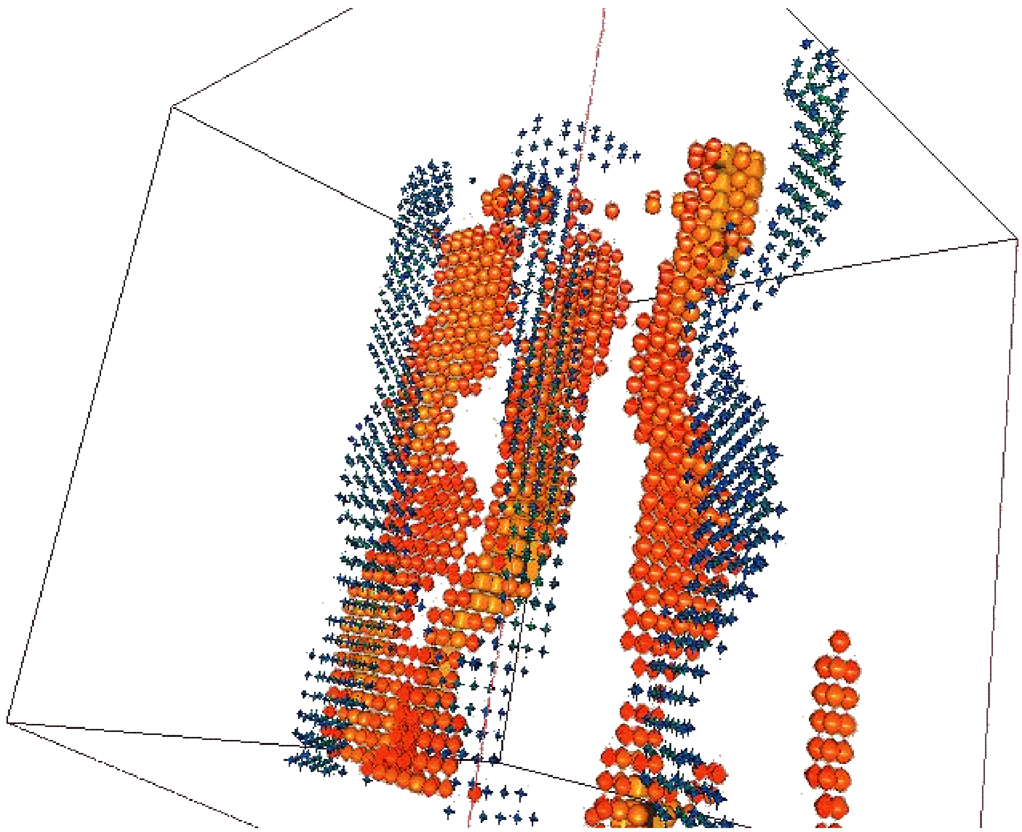

Figure 8: Visualization of a magnetohydrodynamics simulation of the solar wind in the distant heliosphere displaying 3 vortex tubes with positive j vorticity (cuboids and ellipsoids) and 3 vortex tubes with negative j vorticity (stars).