Interactive Lens Visualization Techniques

Christopher D. Shaw, James A. Hall
U. of Regina [*]

David S. Ebert
U. of Maryland Baltimore County [†]

D. Aaron Roberts
NASA Goddard [‡]

Abstract

This paper describes new techniques for minimally immersive visualization of 3D scalar and vector fields, and visualization of document corpora. In our glyph-based visualization system, the user interacts with the 3D volume of glyphs using a pair of button-enhanced 3D position and orientation trackers. The user may also examine the volume using an interactive lens, which is a rectangle that slices through the 3D volume and displays scalar information on its surface. A lens allows the display of scalar data in the 3D volume using a contour diagram, and a texture-based volume rendering.

Keywords: Volumetric Data, Glyphs, Two-Handed Interfaces, Interactive Volume Rendering, Contour Diagrams, Stereoscopic Field Analyzer SFA, Seed Fill, Over Blending

1 Introduction

Visualization of complex 3D scientific data is a difficult task. The goal of a visualization system is to comprehensibly display multidimensional data such that at each grid point of a scientific simulation, multiple dimensions of the simulation are simultaneously visible and comprehensible. Glyph visualization [16, 17, 10] allows the display of multiple data values located at a grid point. Multiple data values can be encoded in the glyph's visual parameters [16], such as 3D size, orientation, color, transparency, and shape. Glyph rendering draws on the use of visual icons in the fields of cartography, statistics, and pictorial information systems.

SFA (Stereoscopic Field Analyzer) [8] is a system we have built for the interactive display of volumetric data. The capabilities of the user's perception system are exploited by SFA to increase the information perceived. SFA presents the user with a desktop VR visualization tool to increase the understanding of volumetric data while being affordable on desktop workstations. SFA takes advantage of the priority structure [6, 2] of human visual perception, stereopsis, motion, and proprioception (the brain's unconscious awareness of the sense of body in space) to create meaningful visualizations from scientific data.

We have developed new techniques for interactive inquiry and display of spatial data. In addition to the visual glyph attributes, SFA can display an arbitrary 2D slice through the data with an interactive technique called the *Contour Plate*. The Contour Plate samples the scalar values at the location of the plate, and displays a 2D contour diagram of these scalar values. As the user moves the plate through the volume, the contour diagram is updated to reflect the new scalar values sampled by the plate.

We have also developed a variant of the Contour Plate called the *X-Ray Lens*, which places a volume-rendered X-Ray of the 3D volume on the plate. In this case, the plate acts like a lens through which the user can see an X-Ray of the volume. The volume rendering is produced by taking multiple slices through a 3D texture that has been filled with one of the scalar values in the volume [5].

2 Background

This section briefly reviews research on the use of 3D input devices to provide the user with high spatial bandwidth in interacting with and visualizing scientific data.

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 [1] and real-time exploration of atomic scale phenomena [19].

Hinckley et al. [13] 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.

Clifton and Pang [7] built a system that uses a CyberGlove to control a series of cutting planes and surfaces. The user sits in front of a monitor and cuts the volume using hand postures to state the geometry of the cutting surface. The cutting surface is subdivided into smaller and smaller triangles depending on the amount of variation in the volume. On the surface is drawn a choropleth contour diagram that is computed by coloring each triangle vertex according to the scalar value at its location. This system used only regular grids.

There have been several previous approaches to the interactive display of volumetric data. Many research systems [4, 5, 12, 14] and several commercial systems, including IBM DX, AVS, and FAST, allow interactive viewing of data using traditional 2D interfaces. Several systems also allow stereoscopic visualization [9, 15] while still using two-dimensional interaction techniques.

Many 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-

[*] Department of Computer Science, University of Regina, Regina, Saskatchewan, Canada S4S 0A2, Phone: (306) 585-4071, email: cdshaw@cs.uregina.ca, hallj@cs.uregina.ca

[†] Computer Science and Electrical Engineering Department, University of Maryland Baltimore County, 1000 Hilltop Circle, Baltimore, MD 21250, Phone: (410)455-3541, email: ebert@cs.umbc.edu

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

immersive interaction metaphor to provide interactive visualization, manipulation, and exploration of multivariate, volumetric data.

2.1 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 the two 3D sensors (see figure 1). Each 3D sensor, or *Bat*, has three buttons glued onto the surface, as shown in figure 2. The user interacts with the system by manipulating the two trackers, and pressing the tracker buttons to invoke operations.

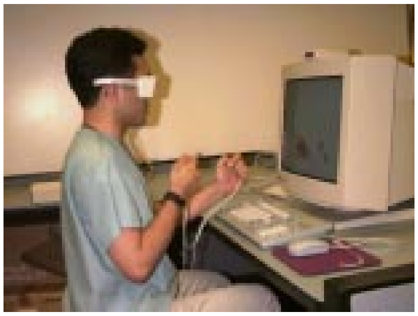

Figure 1: A user using the two-handed stereo interface to SFA.

In the following sections, the new interactive plate visualizations are briefly described. We then show the results of applying these new techniques the visualization of space physics data, plus some preliminary results in the field of Information visualization. Finally, we describe future directions for research in this area.

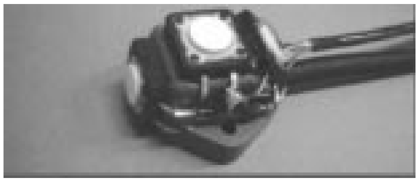

Figure 2: Polhemus sensor with attached buttons.

3 Plate Visualization

The main rendering method in SFA is Glyph visualization, where each data point in the 3D volume is represented by a polyhedral glyph, such as a cube. SFA provides two additional techniques to visualize data attributes. Instead of modifying an attribute of a glyph (eg: shape, size, color), these techniques present the data in a different non-glyph format simultaneously with the glyphs. This allows quick visual recognition of data set features in addition to the glyph visualization. The two methods available are the *Contour Plate*, and the *X-Ray Lens*.

Both methods use a 2D rectangle called a *plate* upon which is drawn one of the two display techniques. The plate is placed at an arbitrary position and orientation in the 3D volume to sample the scalar values near the plate. The user can manipulate the size, position, and orientation of the plate using the right bat.

3.1 Plate Control

All operations on plates are performed using the right bat button 2 (the middle button in figure 2). The functions executed by pressing and releasing this button change depending on the state of the current plate. SFA starts with no plates present, so when right button 2 is pressed, a new plate appears at the right cursor. This new plate immediately becomes the *Current* plate and is placed in the *Move* state, allowing it to be moved by the right bat. The new plate remains in the *Move* state until the user releases the button. This allows the user to sweep the plate through the display volume and gain a visual understanding of the space.

Up to 8 plates can be placed in the scene simultaneously, so each plate has an associated thumbnail window, located on the right side of the main window, stacked vertically. Plate thumbnails show just the plate, the volume borders, and the axis labeling, leaving out other plates and glyphs. The thumbnail provides a Sundial menu to control the plate's attributes, as well as to change the current state of the plate.

When the user moves the right cursor into a plate thumbnail, the thumbnail highlights, and the user may select one of 5 plate operations from a partial sundial menu by pressing right button 2.

The five menu options on this sundial are:

- Add a *new* plate.
- *Move* the current plate.
- *Resize* the current plate in its plane.
- *Delete* the current plate.
- *Change* the current plate's visualization method.

By default, the *current plate* is the most recently created plate. To change the current plate, the user moves the right cursor into the desired thumbnail and presses right button 2, popping up the desired plate's Sundial menu.

The *Change* operation immediately causes the current plate to switch to the selected visualization method. The other menu options place the current plate into a state that determines how the user can manipulate the current plate.

4 2D Contouring

The 2D contouring technique takes all the glyphs within a certain distance of the plate and draws a contour line diagram on the plate to visualize a dataset attribute. These contour lines are then drawn upon the front face of the plate. The presence of the plate object at the 3D location in the volume gives the user a good indication where the data being contoured is from. This is a similar idea to Clifton and Pang's cutting planes [7], except that SFA offers different rendering and control of the planes.

One of the desired features of 2D contouring in SFA is the ability to move the plate in real time and have the contour lines update in real time. This allows the user to quickly check multiple areas of the 3D dataset. Real-time update also gives the user a much greater understanding of how data in different areas of the dataset relate to each other.

This need for real-time update requires that glyphs near the contour plate must be efficiently selected. As with all SFA operations, the user can manually speed this process up by creating subsets to eliminate glyphs from the scene. Glyphs eliminated by subsetting are not considered for contouring, regardless of their location with respect to the plate. All remaining glyphs are then passed to a cache building phase.

Our algorithm for contour generation takes three steps:

1. Glyph selection and caching
2. 3D projection and seed fill
3. 2D Contour generation

4.1 Glyph Selection and Caching

Because an infinitely thin slice is unlikely to intersect any datapoints in the volume, the contour plate takes a rectangular plate-aligned volume of the dataset and samples the glyphs in this volume. This volume is simply a rectangular prism created by extruding the plate along the plate's normal vector. Data points that fall within this volume are cached for further processing.

The contour generation algorithm in step 3 requires a rectangular grid of points, so the contour plate is divided into a 64×64 grid of *cells*. Each cell must be assigned a scalar value.

The first step for each glyph being considered is to convert the glyph's location to the plate's coordinate system. A quick comparison of the X and Y coordinates of the glyph against the plate's coordinate boundaries accepts or rejects the glyph for further consideration.

Next, the glyphs are bucket sorted in Z distance from the plate. Each glyph is thus assigned to one of the Z buckets based on its distance from the plate. The bucket extent in Z is equal to the cell extent in X or Y, whichever is larger. All of the 2D cells on the plate are initialized to an empty value. Next, all of the glyphs that are in the cells immediately surrounding the plate are put on a queue. These glyphs are within one cell thickness of the plate. In figure 3, glyphs *a* and *b* are within one cell thickness of the plate.

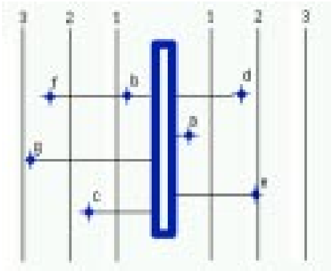

Figure 3: A contour plate and surrounding glyphs. The glyphs are bucket sorted by distance from the plate.

A modified 3D 4-connected seed fill algorithm [11] then commences. While the next item on the queue is within the current Z bucket, it is taken off the queue, and its value is used to fill its corresponding 2D cell. The 4 surrounding cells are put on the tail of the queue to be processed with the next Z bucket. When the current Z bucket is exhausted, glyphs from the next Z bucket (both above and below the plate) are then checked. In figure 3, glyphs *c, d,* and *e* would be checked next. The X and Y coordinates of the glyph (in the plate coordinate system) are converted to cell indices. If the cell location of that glyph has not yet been filled by a glyph at a slice nearer the plate, its value is placed into the cell, and that cell's 4 neighbors are placed on the queue.

The four-connected seed fill algorithm fills up the holes in the cells. Each cell in the queue is used to fill values to the north, west, south and east of it, assuming those cells have not already been filled.

As shown in figure 4, where two cells would propagate their contents to the same unfilled cell, the mean of the contributing cells is used as the result of the seed fill. This prevents set ordering from changing the final outcome, since a seed filled value at the same level as the current bucket being considered would not be included in the mean calculation. To fill all holes, we repeatedly call the seed fill algorithm until the queue is empty.

The purpose of this algorithm is to correctly handle situations where there is an irregular grid of glyphs, or there is no grid at all. In our information visualization applications, glyphs are placed at arbitrary locations, with glyph densities ranging from zero to hundreds of glyphs per cubic inch. The bucket sorting step helps speed up the rendering in these situations.

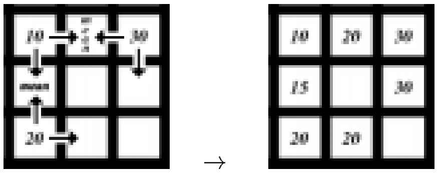

Figure 4: The 4-connected seed fill. Each cell propagates its contents to its north, south, east and west neighbors. Cells with multiple contributors get the mean value of their contributors.

4.2 Contouring Algorithm

Once all the cells have been filled, the cells are passed off to a standard contouring algorithm called CONREC. CONREC subdivides each bucket into four triangular sections and computes which triangular sections need to have line segments drawn in them, based on the surrounding values plus a series of contouring levels [3].

Once the contour line segments have been calculated, we indicate the scalar value that the contour represents by color. We do not write numbers alongside each line because of visual clutter. The same color table that is active for glyph coloring is used to generate contour colors. The entire scalar range of the set (not just the values projected onto the plate at the current time) is segmented into 16 sections. As the contour values switch from one side of a section to another, a contour line is drawn. As the crossover levels increase, so does the index into the color ramp.

By using this method with an appropriate color ramp, the user can immediately tell the features being drawn using the contour lines on the plate. It also provides an effective mechanism for comparing one portion of the set to other values in the set, since the color selected is based on the entire range of the set. This allows the user to quickly locate patterns in the data using the contours, even when the plate is being moved around.

4.3 2D Contouring Results and Future Improvements

Figure 5 shows an example of the Contour Plate. Contouring provides an excellent alternative method for the user to visualize the data set. Patterns within the attribute being visualized are easily recognizable, as are abnormally valued glyphs, which stand out in the contour. The use of color ramps to select the color of the contour lines being drawn aids in the user's understanding of data being visualized on the plate. Color also shows how that data relates to other datapoints in the main set.

Another use of the contour plate is to display a different attribute than what the glyphs are showing. The presence of the contour lines outside the glyph locations makes good use of the unused space between the glyphs.

5 X-Ray Lens Visualization

When a plate is in the X-Ray Plate visualization mode, the plate becomes pseudo-transparent. The enclosing wireframe box of the volume is drawn, along with the blue cylinder plate border. This enables the user to see the space that the X-Ray is rendered in. This ability to locate the plate in 3D space is essential to understanding the scene drawn through the lens. Through the lens, the user is presented with a real-time volume-rendered image of one scalar value in the data set. The volume rendering obscures the glyphs that would otherwise be visible behind the plate.

The volume rendering is produced using back-to-front polygonal slices through a 3D texture volume [5]. This texture is applied using *over blending*, where slices through the 3D texture are drawn back to front, using the alpha value in the texture to blend in previously rendered slices with each new slice. This blending multiplies the incoming slice's luminance value by its alpha value, and adds this to one minus the alpha value times the value of the slices already rendered. SFA uses a luminance and alpha texture, using 8 bits for both luminance and alpha.

Since each plate can be arbitrarily positioned and oriented within the volume, the texture-mapped slices through the volume must be determined using the current eyepoint and plate location. From the eyepoint, rays are projected through each of the four corners of the plate. At fixed distances along each of these rays, new coordinates are generated for each texture slice. This method of slicing continues until all four coordinates generated fall outside of the set volume. This list of generated slices is then rendered back to front, over blending the luminance and alpha texture.

5.1 Building the Texture

To build the 3D texture, the 3D volume data is quantized into a $64 \times 64 \times 64$ voxel array. Typically, this texture size provides decent visual accuracy while still fitting into the four megabyte texture memory found on SGI workstations with dedicated texturing hardware. For greater visual accuracy, the number of buckets per axis could be increased. On machines with dedicated texturing hardware, using a smaller texture typically does not buy much extra performance, while exceeding the texture memory substantially lowers performance due to texture memory swapping.

For each glyph within the 3D volume, its 3D voxel location is computed, and its scalar value is added to that texel. In addition, we then distribute the glyph's value to the 26 voxels surrounding the main voxel, multiplied by a factor corresponding to the Gaussian distribution based on the distance from the glyph's location to the center of each voxel.

This resampling process creates a 3D volume of spherical *spots*, each of which corresponds to a glyph. Because SFA uses a white background, the texture scale is inverted, so that the maximum scalar value corresponds to an opaque black spot (luminance = 0, $\alpha = 1$). This resampling scheme allows each glyph to be sampled into the texture volume independently. Glyphs that occur close to each other in the voxel space will naturally blend together.

One desirable outcome of this scheme is that the glyph spacing and the spot spacing is the same. An advantage to glyph rendering is that there is usually sufficient amount of space around each glyph to see the other glyphs that are farther away in depth. In scenes of high depth complexity, this allows the user to view the glyph values throughout the volume, without the foreground glyphs obscuring those in the background. The X-Ray texture has a similar transparency property, because the spots typically have enough space around them to allow spots in the background to be visible.

Another potential sampling method would be to do a trilinear interpolation of surrounding glyphs at each texel. This would yield a dense volume of opacities that would be difficult to visually correspond to the polygonal glyph display. Furthermore, the foreground opacity would obscure background values.

5.2 3D Texture Results

Figure 7 shows an example of the X-Ray Lens. This solar wind visualization shows velocity as 3 separate scalar values. X is displayed as glyph color, Y is glyph size, and Z is shown in the X-Ray. The top "slice" of glyphs shows large Y values except at the left and right near the back. The zone of small Y values on the right is crescent shaped. X (color) indicates 3 cyan/blue bands down the center from front to back, (blue indicates large values).

The center region of the next slice below is an X-Ray. The X-Ray starts at the second row of small green glyphs, and extends to the back of the volume. This shows moderately large Z values at the back left (where Y = 0), and a denser stripe on the center left corresponding to the long cyan stripe in X. On the right of this slice is a distinct triangular of very low (white) Z values. This white triangle bounds a dense crescent shaped region corresponding to the crescent shaped region of small Y values at the back right. The next two slices below show progressively smaller Z values near the front of the volume. The strong linear feature that is parallel to the X axis in these slices is a sampling artifact.

Due to the small size of the standard texturing memory, the texture does not always place the luminance spots directly over the glyph. This is because the texture has only 6 bits of spatial resolution in each axis. However, this positional inaccuracy serves a perceptual purpose in that the glyphs and the texture spots do not overwrite each other.

The X-Ray Lens provides a visual reinterpretation of the data being visualized. When glyphs are closely spaced, either directly through the data or due to the perspective projection, the luminance spots begin to blur together, and it can be difficult to interpret the value of single glyphs. However, the X-Ray yields an overview of the data in the dataset that would otherwise be unavailable.

One difficulty with this technique is that it is rather slow on machines without dedicated texture-mapping hardware, but as machines improve in speed, this difficulty will subside. Similarly, small texture memories allow only monochrome textures. A bigger texture memory will allow more visual accuracy in superimposing the texture onto the glyphs, and will allow for the addition of extra features within the texture, such as color, which will increase the utility of this technique.

Figure 6 shows the X-Ray Lens used in an information visualization context [18]. In this case, each glyph (a little square in figure 6) represents a document in a collection. The location of each 3D glyph is determined by the relevance match to 3 search terms. That is, term 1 determines X position, term 2 determines Y, and so on. Color in this case represents a fourth search term, while the X-Ray spots show date. Most documents have minimal relevance to the first 3 search terms, so they are clustered at the origin. The lens is focusing on the upper left branch of documents, which shows that there are 3 or 4 documents with old dates, while most are near the middle of the time period for this collection. Near the bottom center of the lens are two or three more recent documents (dark spots). Notice that both color and size are left available for mapping to search terms, while the information of minor importance can be viewed in the X-Ray lens.

6 Conclusions

SFA is a perceptually motivated, minimally-immersive visualization system that allows multi-dimensional data to be visualized, analyzed, and manipulated easier than traditional visualization systems. The plate visualization technique shows promising results in the display of both primary and secondary data attributes of 3D spatial data. The contouring system is primarily intended for 3D scientific data, and it makes excellent use of free space about the glyphs.

Contouring in 3D is itself a well-known technique. SFA simply packages the contouring visualization in a highly-interactive two-handed interaction system that allows rapid interaction with the 3D space.

The X-Ray technique shows promise in the display of both primary and secondary attributes in both scientific and information visualization applications. Our experience is that it is good for displaying secondary scalar variables while the user is actively investigating primary variables.

SFA allows the user to map any variable to any display modality, such as glyph size, color, orientation, transparency, and shape. For example, the user can investigate one scalar by mapping it to glyph size. Later the user can remap this variable to the X-Ray, freeing up glyph size for a new primary variable. The X-Ray provides an overview of the former primary variable while the user is investigating the new primary. The X-Ray thus provides important visual context, and allows the user to observe visual correlations between primary and secondary variables that would otherwise be unavailable if all of the other primary visual display techniques are in use.

7 Acknowledgements

This project was supported by NSERC Canada under grant OGP0183941, and by NASA under grant NAG 5-2893.

References

[1] 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.

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

[3] Paul Bourke. CONREC: A Contouring Subroutine. *Byte*, pages 143–150, June 1987.

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

[5] Brian Cabral, Nancy Cam, and Jim Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In Arie Kaufman and Wolfgang Krueger, editors, *1994 Symposium on Volume Visualization*, pages 91–98. ACM SIGGRAPH, October 1994. ISBN 0-89791-741-3.

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

[7] Michael Clifton and Alex Pang. Cutting Planes and Beyond. *Computers & Graphics*, 21(5):563–575, 1997. ISSN 0097-8493.

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

[9] Elliot K. Fishman, Brian S. Kuszyk, David G. Heath, Luomin Gao, and Brian Cabral. Surgical planning for liver resection. *IEEE Computer*, 29:64–72, 1995.

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

[11] James D. Foley, Andries van Dam, Steven K. Feiner, and John F. Hughes. *Computer Graphics, Principles and Practice, Second Edition*. Addison-Wesley, Reading, Massachusetts, 1990. Overview of research to date.

[12] Allen Van Gelder and Jane Wilhelms. Interactive animated visualization of flow fields. *1992 Workshop on Volume Visualization*, pages 47–54, 1992.

[13] Ken Hinckley, Randy Pausch, Dennis Proffitt, and Neal Kassell. Two-Handed Virtual Manipulation. *ACM Transactions on Computer-Human Interaction*, 3(5):260–302, September 1998.

[14] Nelson Max, Roger Crawfis, and Barry Becker. New techniques in 3D scalar and vector field visualization. In *First Pacific Conference on Computer Graphics and Applications*. Korean Information Science Society, Korean Computer Graphics Society, August 1993.

[15] T. V. Papathomas, J. A. Schiavone, and B. Julesz. Applications of computer graphics to the visualization of meteorological data. In John Dill, editor, *Computer Graphics (SIGGRAPH '88 Proceedings)*, volume 22, pages 327–334, August 1988.

[16] 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.

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

[18] Christopher D Shaw, James M Kukla, Ian Soboroff, David S Ebert, Charles K Nicholas, Amen Zwa, Ethan L Miller, and D Aaron Roberts. Interactive Volumetric Information Visualization for Document Corpus Management. *International Journal on Digital Libraries*, 2(2), 1999.

[19] 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.

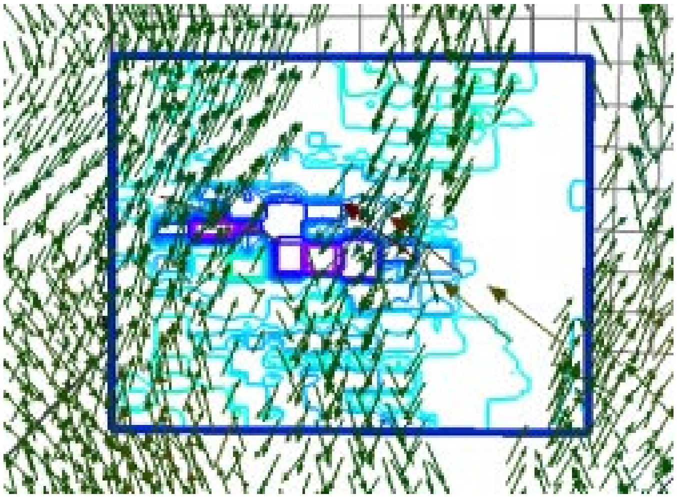

Figure 5: A Contour plate. A vector dataset has arrow glyphs showing upward and downward fluid flow in a pattern of vortex tubes. The contour plate slices through the central tube and left tubes and visualizes the X (horizontal) magnitude of the vector. The purple contour rings in the center and left of the plate show large X magntudes at the plate's location. The corresponding glyphs are behind the plate.

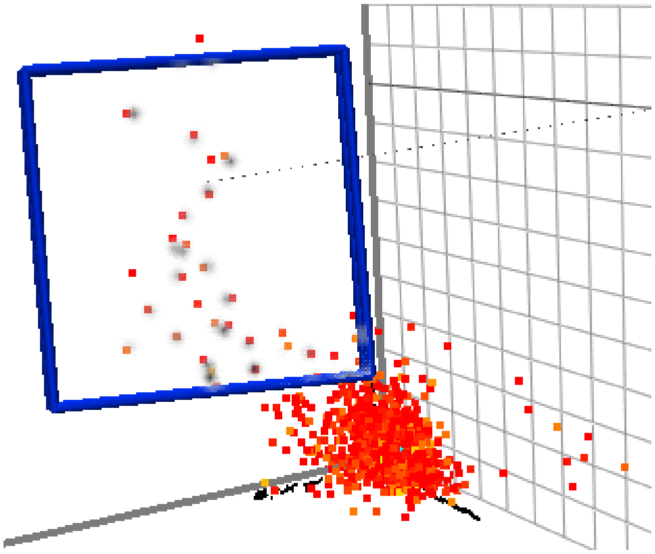

Figure 6: X-Ray Lens for Information Visualization. Each little square is a glyph that represents a document. The document glyph's location is determined by similarity to certain themes, with greater similarity indicated by greater distance from the origin. The vertical axis represents "US Dollar", the axis to the left represents "Federal Reserve", and the axis to the right represents "Stock Prices". The X-Ray lens is focusing on the upper left branch of documents, and the grey X-Ray spots represent age, with newest being the darkest spots. Within the lens are 4 documents with old dates, while most are near the middle of the time period for this collection. Near the bottom center of the lens are three more recent documents (dark spots).

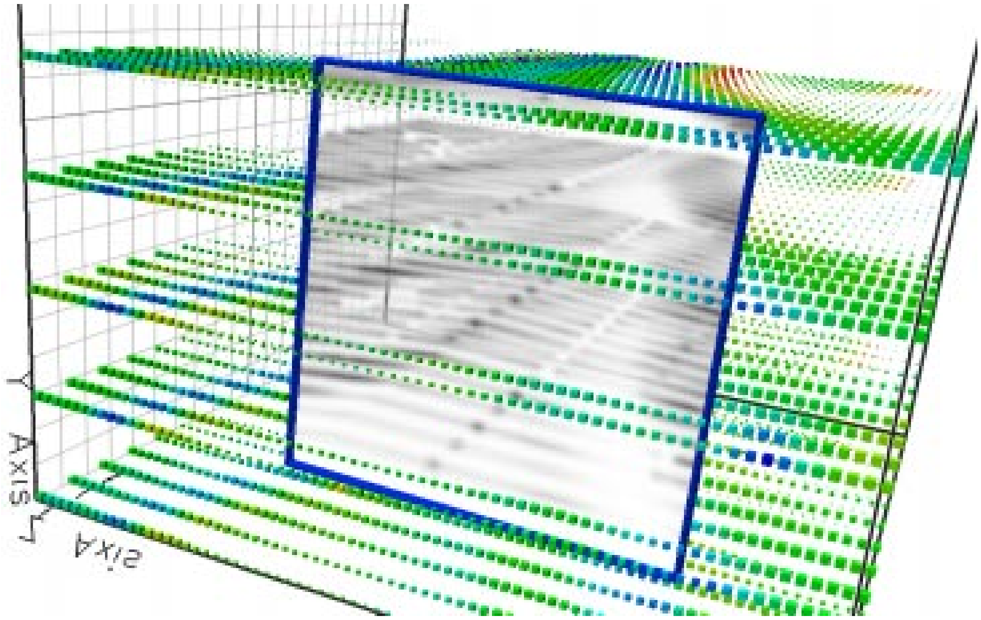

Figure 7: An X-Ray lens. This solar wind visualization shows velocity as 3 separate scalar values. X is glyph color, Y is glyph size, and Z is shown in the X-Ray. The top "slice" of glyphs shows large Y values except at the left and right near the back. The small Y zone on the right is crescent shaped. X (color) indicates 3 cyan/blue bands down the center from front to back, (blue indicates large values). The center region of the next slice below is an X-Ray. This shows moderately large Z values at the back left (where Y = 0), and a denser stripe on the center left corresponding to the long cyan stripe. On the right of this slice is a distinct triangular of very low (white) values. This white triangle bounds a dense crescent shaped region corresponding to the crescent shaped region of small Y values at the back right.