Scalable Self-Orienting Surfaces: A Compact, Texture-Enhanced Representation for Interactive Visualization of 3D Vector Fields

Greg Schussman Kwan-Liu Ma
University of California, Davis*

Abstract

This paper presents a study of field line visualization techniques. To address both the computational and perceptual issues in visualizing large scale, complex, dense field line data commonly found in many scientific applications, a new texture-based field line representation which we call self-orienting surfaces *is introduced. This scalable representation facilitates hardware-accelerated rendering and incorporation of various perceptually-effective techniques, resulting in intuitive visualization and interpretation of the data under study. An electromagnetic data set obtained from accelerator modeling and a fluid flow data set from aerodynamics modeling are used for evaluation and demonstration of the techniques.*

1 Introduction

A large number of scientific applications call for the visualization of vector fields to convey not only the field strength but also directional information. Some examples include vector fields obtained from modelings of complex chemical reactions, air flow surrounding an aircraft, fluid flow in a car engine, and blood flow in a human heart. A convenient and intuitive way for visualizing vector fields is to plot streamlines, streamtubes, or streamribbons. Newer approaches based on texture synthesis [5, 24, 25] can produce realistic visualization mimicking what scientists observe in traditional experimental flow imaging. The remaining challenges in visualizing vector fields include unambiguously displaying densely clustered field lines and adequately displaying vector topology information [13]. This paper addresses the first challenge and presents the results of our comprehensive study on using a new hardware-accelerated, textured line-based visualization technique.

Accurate prediction of complex natural phenomena (such as vortex behavior behind airplanes, or electromagnetic field propagation in particle accelerator structures) often requires large scale physics-based simulation on parallel computers. These simulations tend to produce very large sets of complex time-varying data. Our work was motivated by the need of simultaneous interactive visualization of large three-dimensional electric and magnetic field data generated from accelerator simulations. The goal is to display a dense collection of intertwined lines in a way that shows clear spatial relationships between them, with unambiguous global and local details.

We introduce a new representation which we call Scalable Self-Orienting Surfaces (SOS) for field line data that supports perceptually effective visualization techniques in an interactive, efficient and convenient way. This work takes into account existing research on perceptual cues and preattentive processes, applying these insights to the challenge at hand. The result applies to other kinds of vector fields as well (e.g. fluid flow), and to other line-based data (e.g. particle paths).

The remainder of this paper is organized as follows. Section 2 provides a review of existing research on visual perception cues and preattentive processes. This provides the context for the description of the SOS representation in Section 3. Perceptually effective techniques involving illumination, atmospheric effects, and selective de-emphasis of data are applied to the SOS representation in Section 4.

2 Fundamental Visual Perception Issues

To design 3D rendering techniques leading to effective visualizations, we must take into account how the human visual system processes information. Some of the well studied theories in cognitive science and visual perception which define the nature of our responses to various visual stimuli like depth, color, and textures form the basis of our study.

2.1 Cue Theory

Assorted visual cues can help or hinder accurate 3D interpretation of 2D images. Wanger et al. point out that the

*Center for Image Processing and Integrated Computing (CIPIC), Department of Computer Science, University of California, Davis, CA 95616-8562, USA; Email: {schussma|ma}@cs.ucdavis.edu

close correspondence of our perceptions to physical reality can be undermined when the visual information we rely on is missing or ambiguous [29].

Wanger's work summarizes cue theory, and determines relative effectiveness of well known cues. There are two primary cues. Convergence/accommodation is considered a weak cue, and involves the angle of convergence between the two eyes when focused on an object, as well as the focus of the eye's lens. Binocular disparity is the powerful cue of stereo vision, arising from differences between two retinal images. Wanger identifies this cue as the one which allows us to make our finest depth judgments.

Even without the compelling effect of stereo viewing, pictorial cues assist in perceiving depth. This secondary class of cues includes perspective, texture, shading and shadow, motion, and reference frames. Perspective is the proportional contracting of objects with increasing depth. Texture gradients provide information about orientation and relief of flat or curved surfaces. Likewise, shading refers to gradients of illumination, providing information about surface shape and orientation. Shadows cast from one object onto another provide spatial position information.

Texture, shading and shadow, and reference frames are effective visual cues. Wanger conducted user studies involving perspective, shadow, and motion. He concluded that shadow and perspective are important cues for sensing object location, that perspective, motion, and shadow are important cues for sensing object orientation, and that all of these cues are important for simultaneously sensing an object's size and location.

2.2 Preattentive Processes

Certain visual features are processed preattentively. That is, the visual system automatically processes them quickly and accurately without the need for concentrated attention [7, 17, 27, 30].

Preattentive processing enables quick searching for objects with a unique visual feature, easy identification of the boundaries between groups of objects with common features, easy tracking of motion of groups of objects, and forming rapid estimates of the number of objects with a specific visual feature. This processing takes place in less than 0.2 seconds, and is independent of the number of objects viewed. Healey [12] showed that these tasks can be performed repeatedly at a rate of 5 times per second.

3 Field Line Representations

There are many different representations for field lines. Streamlines [10] are curves which are, at every point along their length, tangent to an instantaneous vector field. Streamlines were drawn as lines to show the structure of a vector field. Work has been done to render streamlines in ways that improve perception of their structure, or provide information in addition to their structure. These techniques include stream polygons [22], streamribbons and streamtubes [6], stream surfaces [14], and streamballs [4]. Additional work has been done for efficient computation and rendering of streamribbons and streamtubes [28].

Stream ribbons are essentially ribbons that lie along streamlines and twist to indicate rotation within the vector field.

The stream polygon technique consists of a sequence of polygons normal to the curve tangent which are drawn with deformation to indicate local rotation, strain, and shear. The envelope of these swept polygons forms a streamtube.

The streamball technique is based on metaballs, which were originally Blinn's blobby objects [3]. As such, they are based on an influence function. An alternate streamball approach for the discrete ball case is to set the radii proportional to a field parameter. Either way, this prevents uniform metaball radii, resulting in a size vs. depth ambiguity similar to that of stream tubes.

3.1 Scalable Self-Orienting Surfaces

We propose a new representation for field lines, the Scalable Self-Orienting Surface (SOS), which serves as consistent geometry for the perspective depth cue, and as a platform for texturing and lighting effects which provide other information and cues.

The basic SOS ribbon has constant width (in world space) and lies along a curve in space. The surface twists axially about the curve in a view-dependent way to minimize the angle between the surface normal vector and the corresponding viewing vector (that is, the vector from the ribbon's surface to the point from which the ribbon is observed).

The SOS ribbons are constructed from a sequence of points along a curve, an associated sequence of tangent vectors, and a viewing position. First, a viewing vector and a sideways offset vector are computed for each point in the sequence. Then a triangle strip is generated.

A viewing vector is constructed by subtracting that point's position from the viewing position, as is illustrated in Figure 1.

A sideways vector is constructed by normalizing the cross product of the point's viewing vector and it's tangent vector. The resulting vector is perpendicular to both the viewing direction and the surface normal. It is then scaled to provide positive and negative sideways offsets from the original associated point. These new offset points then serve as vertices for a triangle strip. This process for generating triangle strips is illustrated in Figure 2. The sideways vectors can be scaled according to a user's preferences. Varia-

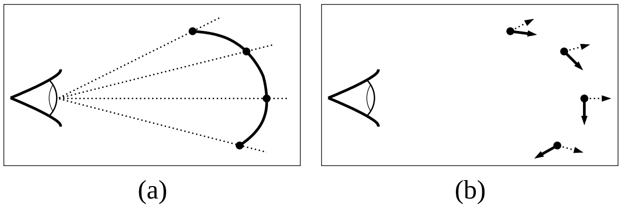


Figure 1. From the curve and viewing rays (a), the curve tangent vectors and viewing vectors (solid and dotted arrows, respectively) are constructed (b).

tions on the basic idea include scaling the sideways vectors based on local field properties, and having dynamic and/or view dependent textures.

The SOS representation is more flexible than sprites or billboards which have been common practice in the computer gaming industry [8]. Although sprites were originally rectangular textures to be rendered as 2D objects by 2D graphics hardware (before commodity 3D graphics hardware was available), they continue to be used with 3D graphics hardware today. Sometimes these sprites are also called impostors. A billboard is an extension of a sprite, where the rigid, textured polygon is free to rotate about a point or a line axis in a view dependent way in order to face an observer. The SOS geometry (unlike that of sprites or billboards) is not rigid, which allows it to rotate axially about a curve in space rather than being limited to a straight line.

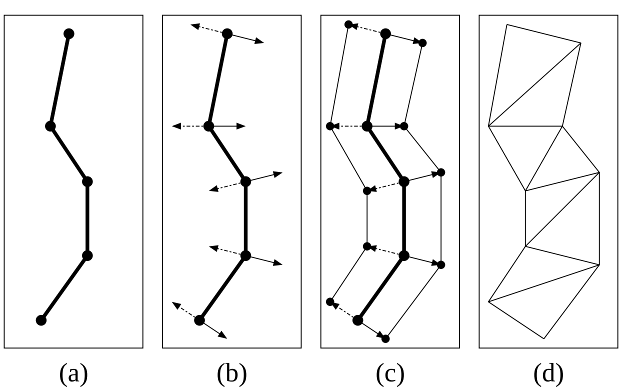


Figure 2. A sequence of points (a) is converted to a triangle strip by adding positive and negative sideways offset vectors (b) to produce new points (c) which become vertices for a triangle strip (d).

3.2 Computational and Perceptual Issues

Although our approach is motivated by the need to visualize simulations of electric and magnetic vector fields, it is applicable to other vector field data (e.g. flow fields), or path data. The primary concern is to remove ambiguity from the rendered images.

In this paper, our discussion assumes images produced with perspective projection. We also restrict our attention to single images, as opposed to animated sequences. Although shading, shadow, and animation or user controlled interactive changing of the view can help disambiguate the images, independent single images should still present as much unambiguous information as possible.

Streamribbons oriented according to twists in a flow field produce images containing potential ambiguity with respect to depth cues. There are two physically plausible reasons for a rendered ribbon to appear to have a thin spot. The first is that the ribbon could be indicating a twist in the flow field. The second is that the ribbon could be changing in depth, moving away from the observer. Because the SOS representation has, by default, a consistent width in world space and is always oriented toward the observer, the projected width in a final image provides an unambiguous perspective depth cue.

The cross section width of streamtubes can vary according to converging or diverging flow. This again produces a depth cue ambiguity: a section of the tube could appear larger due to a local spreading out of the vector field in world space, due to changes in depth even if the tube does not change size in world space, or due to an intermediate combination of the two. Stream surfaces can converge or diverge in the same way that streamtubes do, and so suffer from the same visual ambiguity.

An alternate form of streamtubes maintains a constant radius for the tube along its entire length. This permits good shading and perspective cues. The difficulty is that convincing tubes require 5 or 6 locally parallel triangle strips, which would require moving excessive amounts of geometric data to the rendering pipeline. Specular highlights provide particularly good cues, but require even finer tessellation of tubes in order to accurately capture specular highlight edges. Our approach uses a single triangle strip, yet can achieve a round appearance via hardware accelerated bump-mapping available on current graphics hardware such as the nVidia GeForce 3.

The SOS representation cooperates well with texturing, allowing interactive manipulation of parameters for texture-based techniques. Because the SOS is consistently oriented in the same view-dependent way, the texture coordinate for moving across the strip becomes view-independent. Difficulties that can occur with polygonal tubes are avoided this way (e.g. texture twisting around the tube, or glyphs not orienting toward the viewer). Although polygonal tubes can be precomputed in a view-independent way, some useful texturing techniques (e.g. extending the 2D arrow glyphs [16] to 3D streamlines, or applying "cookie cutter" style arrows as in [19]) would require view-dependent texture coordinate computation on the fly. Many of those computations would be wasteful because they would apply to vertices on the side of the tube facing away from the viewer. The computation needed for the polygonal tube representation consists of rotating the texture axially around the tube to face the viewer, taking into account that vertices of the tube can twist axially relative to one another along the length of the tube.

4 Field Line Rendering

Current commodity graphics hardware is optimized for texturing operations on triangle primitives. The SOS representation brings this power to bear on the presentation of line data much more efficiently than the tube representation does while providing richer illumination than is possible with the line representation. Other hardware assisted techniques, especially those based on texture benefit as well. Examples are depth-based attenuation, local or global transparency-based de-emphasis, and scaling texture width according to local field magnitude.

4.1 Illumination

Raskar [21] uses graphics hardware to produce an edge effect similar to haloing for 3D polygonal models. Front facing polygons are drawn filled, and rear facing polygons are drawn as black wireframe where line thickness is expanded in a view-dependent way. Additional edge effects are ridges and valleys. The proposed implementation is to modify the primitive shader stage of the standard hardware graphics pipeline to allow it to automatically generate new polygon primitives. Unfortunately, no readily available graphics hardware supports this.

Line-based rendering approaches include haloing [1], illuminated streamlines [26], or a combination of the two [23]. Haloing has also been considered in volume-based rendering [15], although we restrict our attention to illumination of line-based and polygon-based representations.

Smooth tubes illuminated in the real world would be the ideal representation for field lines at any reasonable distance from an observer. This is because an observer would have the benefit of many perceptual cues: perspective, shading and shadow, texture, motion, and stereo, as well as the benefit of preattentive edge detection for resolving relative depths of intersecting lines.

The crudest approximation would be simple unshaded rendering of a line representation. This provides only the motion cue, which contributes nothing to accurate perception of a static image. This is shown in Color Plate 1a.

An improved approximation is to use a hardware accelerated illumination model for the lines [26], which adds the shading cue, as shown in Color Plate 1b. Added haloing facilitates preattentive processing to rapidly resolve relative depths of crossing lines. The illuminated line technique looks good for thin lines, but becomes an increasingly worse approximation to an illuminated tube as the thickness increases, as is the case for near lines under perspective projection. Under this condition, the approximation is even worse when the light source is not a headlight. Therefore, rather than apply haloed, illuminated lines as a texture on an SOS representation, we apply illumination and hardware accelerated bump-mapping instead. This produces shading that appears exact and is shown in Color Plate 1c, and supports multiple hardware accelerated lights as is demonstrated in Color Plate 1d. One or more of the lights can be interactively manipulated for additional shading and motion cues. With a headlight and a low ambient Phong coefficient, the dark sides of the simulated tube are equivalent in effect to haloing for lines with respect to preattentive processing of crossing lines.

The illuminated polygonal tube representation is compared with the SOS representation in Color Plate 2. Hardware Phong shading is used on the polygonal tube representation in Color Plate 2a, and hardware bump-mapping is used on the SOS representation in Color Plate 2b. The visual effect is similar, yet a tube would typically have five or six times the number of triangles as an SOS ribbon in order to maintain an apparently consistent cross-sectional width in the final projected image.

This gives the SOS representation the advantage of a smaller memory footprint or less computation, since tubes require either the storage of five or six times the number of triangles, or on-the-fly computation of vertices for that same geometry. On-the-fly computation of tube geometry must include measures to avoid potential twisting, which can cause a pinched appearance when one section is rotated axially relative to the next.

Without the benefit of hardware bump-mapping, the shading cue provided by illumination can still be achieved for polygonal tubes and SOS ribbons by resorting to finer tessellation and Gouraud shading of Phong illuminated vertices. The reason for finer tessellation is that the specular highlights, which provide the more detailed portion of the shading cue, have fast, non-linear roll-off of intensity, which linear Gouraud shading does not capture over coarse tessellation. For SOS ribbons, along the cross-section, one can use multiple evenly spaced vertices to adequately capture the shape of specular highlights. For tubes, however, the screen space gap between cross sectional vertices is not linear, resulting in larger gaps toward the middle of the cross section. For this reason, a disproportionately larger number of vertices must be introduced to maintain the maximum gap size similar to that of the SOS ribbons. SOS ribbons adequately capture specular highlights when the number of shaded vertices is increased from two to about six, whereas tubes need an increase from five or six to about 18 to achieve a similar appearance with Gouraud interpolated specular highlights.

Table 1 summarizes the relative performance of polygonal tubes, finely tessellated SOS, polygonal tubes in a display list, and hardware bump-mapped SOS. For each column of the table, the scene was identical for each of the rendering styles. Timing was performed on a 1.4 GHz Pentium 4 PC running Red Hat Linux 7.3, and using OpenGL on an nVidia GeForce 3 Ti500. The slowest approach was polygonal tubes computed on-the-fly with care to avoid local axial twisting which would result in a pinched appearance. The finely tessellated SOS was divided into 6 pieces crosswise, and was also (necessarily) computed on-the-fly, but ran an average of 5.90 times faster than the polygonal tubes. Providing there is sufficient memory to hold a display list of polygonal tubes for all field lines, the on-the-fly computations can be eliminated in subsequent renderings, resulting in a speeds that are an average of 2.95 times faster than the finely tessellated SOS. However, with a current commodity graphics board such as the nVidia GeForce 3, hardware accelerated bump-mapping on the SOS representation is even faster still, performing an average of 1.4 times faster. It is important to note that the SOS representation does not require extra the extra memory for a display list, which can be significant. For example 628 megabytes were needed by the display list for the 10,000 line polygonal tube data used in producing the timing results of Table 1. For datasets too large to permit the use of display lists, the hardware bump-mapped SOS representation runs an average of 24.4 times faster than the polygonal tube representation.

	150 lines	800 lines	10k lines
Polygonal tube	0.445	3.001	32.8
Finely tess. SOS	0.077	0.512	5.54
Poly. tube disp. list	0.027	0.173	1.82
Hardware SOS	0.019	0.124	1.28

Table 1. Timing results are given for different line representations: Polygonal tube, finely tessellated SOS, polygonal tube in a display list, and hardware bump-mapped SOS. Hardware bump-mapped SOS runs an average of 1.4 times faster than polygonal tubes in a display list, and an average of 24.4 times faster than polygonal tubes computed on the fly. All times are in seconds.

4.2 Depth-Based Attenuation

Atmospheric effects such as fog or smoke in the real world provide natural depth cues. Light traveling from surfaces to the eye can be attenuated or augmented as a result of light interactions with particles in the air.

In the case of smoke, some light rays never reach the eye because they are absorbed by carbon particles, or (to a lesser extent) scattered off in a different direction. In effect, this attenuates the apparent brightness of distant objects more so than closer ones.

Fog behaves similarly to smoke, but with an additional process called in-scattering. Some light rays that would not ordinarily reach the eye (because they were originally traveling somewhere else) are redirected into the eye via scattering, thereby adding to the total number of rays reaching the eye. Absorption, scattering, and in-scattering combine for the net effect that the color of more distant objects appears desaturated, or “greyed out”. Because closer objects have less intervening atmosphere, they appear to have their natural colors.

Figure 3a shows a plain set of field lines. Desaturation is shown in Figure 3b, and darkening shown in Figure 3c. Figure 3d presents a more compelling image resulting from a combination of darkening and desaturation. To illustrate the effectiveness of fog alone as a depth cue, the images of Figure 3 are rendered with no perceptual cues other than perspective projection.

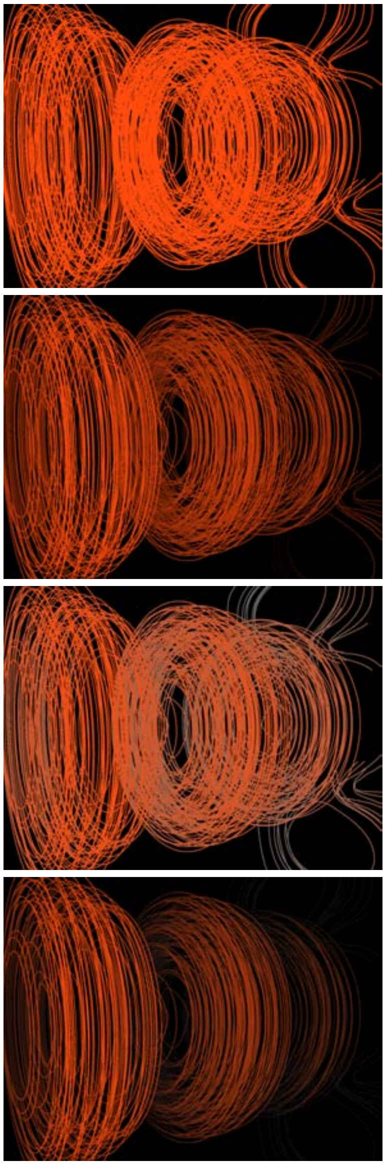
OpenGL provides direct support depth cuing that mimics these atmospheric effects [2]. Desaturation is accomplished with grey fog, and darkening with black fog. A recent OpenGL extension, EXT_fog_coord [20], provides improved control of fog behavior on a per-vertex basis for all geometric primitives.

4.3 De-Emphasis

The purpose of visualization is not so much to see every numerical value in the entire dataset, as it is to see the relationships between the data in that set. When there is too much data to see everything in perfect detail all at once, displaying some of the data in less detail can be preferable to eliminating it entirely. Selective de-emphasis serves this purpose by maintaining a global context while revealing global or local detail that would otherwise be hidden through occlusion or lost in visual clutter.

One can focus attention on a subset of the data to be visualized by emphasizing that subset, or, equivalently, de-emphasizing the rest of the data. Fuhrmann et al. [11] take the former approach, calling it a “magic lens”. We take the latter approach, using transparency with the SOS representation to achieve global or local de-emphasis.

Occasionally, it makes sense to simultaneously visualize



(a)

(b)

(c)

(d)

Figure 3. Depth is difficult to discern for plain lines with no depth cues (a). Although darkening (b) or desaturation (c) provide some depth cue, a combination (d) is more effective.

multiple vector fields, as in the case of electric and magnetic fields, or velocity and vorticity fields. When visualizing a dense collection of field lines, some lines can occlude others. Figure 4a illustrates this with electric field lines colored blue, and the magnetic field lines colored orange. The magnetic field lines occlude the electric field lines near the center of the image. To see the inner structure, one can use the traditional method of clipping away some of the data, as in Figure 4b. The SOS representation is interactively scalable, so the width of the lines can be easily adjusted by a user as needed. Figure 4c shows the effect of reducing the width of the field lines. For very dense lines, even at thin widths an unacceptable amount of occlusion can occur. Simply drawing the electric field alone without the magnetic field loses the local and global relationship between the two. A more effective method to reveal the structure of the electric field is to globally de-emphasize the magnetic field, as shown in Figure 4d. This view would not have been possible with clipping because the electric and magnetic fields are so intertwined.

Color Plates 3 and 4 demonstrate how depth cuing combines naturally with illumination from multiple light sources. Color Plate 3 shows electric (blue) and magnetic (orange) field distribution from the time-domain electromagnetic simulation of an RF (Radio Frequency) cavity. The electric field lines are blue, and the magnetic field lines are orange. Color Plate 4 shows a bottom-up view of two wake vortices from an aerodynamics simulation with the velocity field as blue, and the vorticity field as orange. The development of small lateral vortices underneath the two primary vortices is revealed. Additionally, the development of an intricate structure within the vorticity field is readily apparent.

For very dense line data, it can be difficult to unambiguously perceive the structural details in a region in the interior of the dataset. Surrounding lines, when sufficiently dense, can occlude the interior structures. Again, one can cut away the data which is not in the region of interest. While effective, this loses the global context for the current region of interest.

Another approach is to make everything uniformly transparent. In the limit, as line density increases, this becomes equivalent to direct volume rendering, and suffers from similar visual ambiguity arising from loss of fine detail and multiple possible physical interpretations of color in the final image.

By leaving the region of interest opaque and using transparency to de-emphasize the remaining data, the interior structures remain clear, and the global context is not lost. Color Plate 5 shows this effect for a near and a far region of interest in an electric field dataset. Interactively sweeping in this way provides a fast and compelling way to become familiar with a new dataset. One useful option is to leave a fixed region of space as the region of interest, and permit the user to interactively move the data relative to that region. Another option is to always emphasize the closest region, in a view-dependent way. A third option is to leave everything fixed, but interactively change the size or transfer function of the region of interest.

Interactive manipulation of transparency can be easily implemented in OpenGL either through texturing or ma-

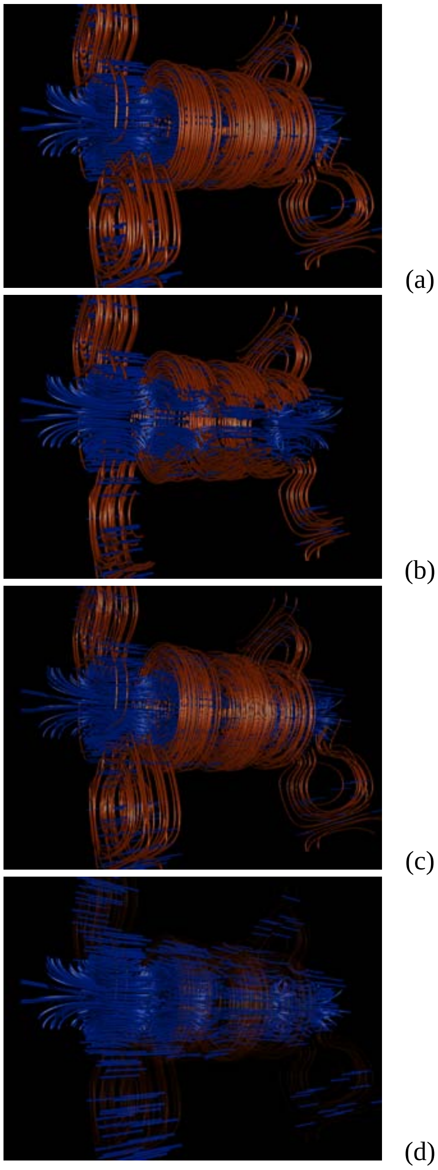


Figure 4. The dense orange magnetic field lines occlude the blue electric field lines (a). Interior structure can be revealed through clipping (b), interactively adjusting SOS width (c). However, for dense, highly intertwined lines, global de-emphasis of the magnetic field provides a view of the electric field not possible with those other methods (d).

terial properties. This effect can be achieved modulating the existing ribbon color with a one-dimensional white texture whose alpha component provides a transparency transfer function. The texture coordinates are issued as as 1.0 for vertices at or near the center of the region of interest, and as progressively smaller values for more distant vertices.

The parameters affecting the appearance of the rendered region of interest can be changed interactively. Moving the region of interest only requires redrawing the scene, issuing different texture coordinates. Changing the transfer function is only a matter of recomputing a small, one-dimensional texture, and redrawing the scene. Interactively sweeping the region of interest from one end of new data to the other provides a compelling effect.

Transparency in complex scenes requires back-to-front compositing for a correct image. Depth sorting is not practical for very large datasets. The SOS representation can utilize the multi-pass order-independent transparency technique [9] supported on current commodity graphics hardware, such as the nVidia GeForce 3 or 4, if rendered with finer tessellation and hardware lighting instead of with bump-mapped texture. This is because all four texture units are used by the order-independent transparency technique, leaving none free for traditional texturing. Since the SOS representation does not change orientation from one pass to the next, it can be placed in a display list. Rendering passes subsequent to the first are simple display list calls, avoiding the computation needed to orient the surface. Compared to the polygonal tube representation, this is a win because the savings in geometry data through the rendering pipeline is multiplied by the number of rendering passes.

4.4 Magnitude Indication

Often the magnitude of a vector field is of interest. In the case of electromagnetic fields, by convention of physics, field line density is proportional to field magnitude. Although the same is not true of streamline density in flow fields, this visual effect is still equally meaningful.

After choosing streamlines, one could seed additional streamlines nearby. Alternatively, an SOS can be scaled extra wide, and have multiple lines mapped onto it via texture. This texture can be scaled as well, but in relation to the local field magnitude. Regions of high field strength appear as fine dense lines, and regions of low field intensity appear as coarse sparse lines as shown in Figure 5.

The effect is achieved with a simple 1D white texture with semitransparent ends. This is used to modulate the underlying blue or orange color. The texture coordinates are scaled according to the local field strength, and the texture wrap mode is set to repeat.

5 Conclusion and Future Work

We have introduced a new scalable representation for field lines in support of 3D vector field visualization. The SOS representation facilitates hardware acceleration of rendering techniques which are perceptually effective because

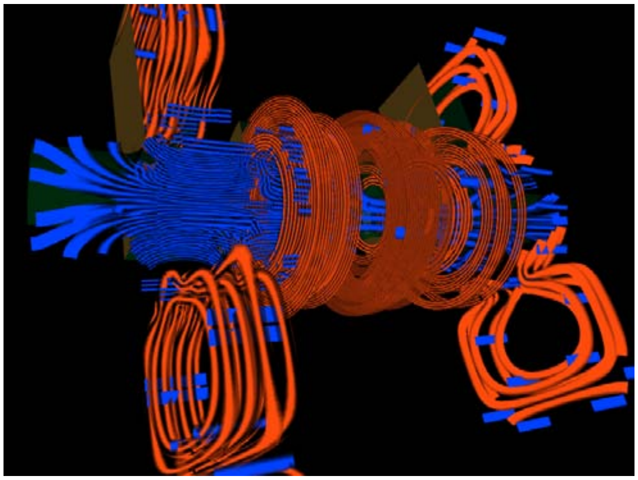


Figure 5. In regions of high intensity field, the lines texture is compressed to produce fine, dense lines.

they provide the cues upon which human visual perception relies. It is efficient both in terms of memory requirement and amount of computation needed to map local field properties or view dependent information onto field lines. We have demonstrated the technique by simultaneous visualization of velocity and vorticity fields from an aerodynamic fluid flow simulation, and electric and magnetic fields from particle accelerator simulation. The result is that even complex, large-scale datasets can be visualized interactively, providing a clear and unambiguous understanding of the internal structural relationships.

The scalable SOS representation has a rich set of possibilities for future work. Multi-texturing can be used for annotating the lines, and for providing the texture depth cue. Static or textures could be used to indicate direction and provide continuity information. An example combination would be color coded arrow glyphs with even spacing along an SOS ribbon, which would indicate direction and local magnitude of the field, as well as providing the texture depth cue.

It may be desirable to trade off the perspective depth cue in favor of local scaling the SOS width according to local field properties. For example, scaling width is sometimes done for streamtubes. Texture applied to the SOS representation can also be varied according local field properties. A LIC texture could be applied. Local velocity magnitude could be indicated with color or texture.

Alternative methods of de-emphasis could be texture with blurred edges to achieve semantic depth of field (SDOF) [18], or by exploiting the flexibility of the per-vertex OpenGL fog coordinate extension.

If future graphics pipelines support the introduction of new polygonal primitives within the pipeline, as Raskar suggests [21], the SOS representation could be accomplished entirely within graphics hardware.

6 Acknowledgments

This work has been sponsored by the NSF PECASE, NSF LSSDSV, and DOE SciDAC. The authors are grateful to Dr. Charlie Zheng at Kansas State University for providing the wake vortices data sets, and the Advanced Computations Department at the Stanford Linear Accelerator Center for providing the time domain electromagnetic field solver datasets.

References

[1] A. Appel, F. J. Rohlf, and A. J. Stein. The haloed line effect for hidden line elimination. volume 13, pages 151–157, August 1979.

[2] OpenGL ARB. *OpenGL Programming Guide*. Addison Wesley, New York, NY, 1993.

[3] James F. Blinn. A generalization of algebraic surface drawing. *ACM Transactions on Graphics*, 1(3):235–256, July 1982.

[4] M. Brill, H. Hagen, H.-C. Rodrian, W. Djatschin, and S. V. Klimenko. Streamball techniques for flow vizualization. In R. Daniel Bergeron and Arie E. Kaufman, editors, *Proceedings of the Conference on Visualization*, pages 225–231, Los Alamitos, CA, USA, October 1994. IEEE Computer Society Press.

[5] Brian Cabral and Leith (Casey) Leedom. Imaging vector fields using line integral convolution. In James T. Kajiya, editor, *Computer Graphics (SIGGRAPH '93 Proceedings)*, volume 27, pages 263–272, August 1993.

[6] Dave Darmofal and Robert Haimes. Visualization of 3-D vector fields: Variations on a stream (AIAA Paper 92-0074). In *Proc. AIAA 30th Aerospace Science Meeting and Exhibit*, Reno, NV, January 1992.

[7] John Duncan and Glyn W. Humphreys. Visual search and stimulus similarity. *Psychological Review*, 96(3):433–458, July 1989.

[8] David H. Eberly. *3D Game Engine Design*. Morgan Kaufmann, San Francisco, CA, 2001.

[9] Cass Everitt. Interactive order-independent transparency. *nVidia white paper*, 2001.

[10] Robert W. Fox and Alan T. McDonald. *Introduction to Fluid Mechanics*. John Wiley & Sons, Inc, New York, 1973.

[11] A. Fuhrmann and E. Gröller. Real-time techniques for 3D flow visualization. In *IEEE Visualization '98 (VIS '98)*, pages 305–312, Washington - Brussels - Tokyo, October 1998. IEEE.

[12] Christopher G. Healey, Kellogg S. Booth, and James T. Enns. Visualizing real-time multivariate data using preattentive processing. *ACM Transactions on Modeling and Computer Simulation*, 5(3):190–221, July 1995.

[13] J. Helman and L. Hesselink. Representation and display of vector field topology in fluid flow data sets. *Computer*, 22(8):27–36, August 1989.

[14] J.P.M. Hultquist. Constructing stream surfaces in steady 3d vector fields. In *Proceedings of Visualization '92*, pages 171–178, Boston, MA, October 1992. (also RNR Tech Report 92-025).

[15] Victoria Interrante and Chester Grosch. Strategies for effectively visualizing 3D flow with volume LIC (color plate S. 568). In Roni Yagel and Hans Hagen, editors, *Proceedings of the 8th Annual IEEE Conference on Visualization (VISU-97)*, pages 421–424, Los Alamitos, October 19–24 1997. IEEE Computer Society Press.

[16] Bruno Jobard and Wilfrid Lefer. Creating evenly-spaced streamlines of arbitrary density. In Wilfrid Lefer and Michel Grave, editors, *Visualization in Scientific Computing*, Focus on Computer Graphics, pages 43–55. Springer-Wien-New-York, 1997. Proc. of Eighth Eurographics Workshop on Visualization in Scientific Computing, Boulogne sur Mer, France, Apr 28-30, 1997.

[17] B. Julèsz. A brief outline of the texton theory of human vision. *Trends in Neuroscience*, 7(2):41–45, 1984.

[18] Robert Kosara, Silvia Miksch, and Helwig Hauser. Semantic depth of field. In *IEEE Symposium on Information Visualization 2001 (InfoVis 2001)*, San Diego, CA, USA, October22–23 2001.

[19] Helwig Löffelmann, Lukas Mroz, and Eduard Gröller. Hierarchical streamarrows for the visualization of dynamical systems. Research paper, Institute of Computer Graphics, Vienna University of Technology, 1997.

[20] NVIDIA Corporation. *NVIDIA OpenGL Extension Specifications*. http://developer.nvidia.com/view.asp?IO=nvidia_opengl_specs, Santa Clara, CA, 2002.

[21] Ramesh Raskar. Hardware support for non-photorealistic rendering. *2001 SIGGRAPH / Eurographics Workshop on Graphics Hardware*, August 2001.

[22] William Schroeder, C. R. Volpe, and W. E. Lorensen. The stream polygon: A technique for 3D vector field visualization. In *Visualization '91*, pages 126–132, 1991.

[23] Greg Schussman, Kwan-Liu Ma, David Schissel, and Todd Evans. Visualizing DIII-D tokamek magnetic field lines. In T. Ertl, B. Hamann, and A. Varshney, editors, *Proceedings Visualization 2000*, pages 501–504. IEEE Computer Society Technical Committee on Computer Graphics, 2000.

[24] H. W. Shen, C. R. Johnson, and K. L. Ma. Global and local vector field visualization using enhanced line integral convolution. In *Symposium on Volume Visualization*, pages 63–70, 1996.

[25] Han-Wei Shen and David L. Kao. UFLIC: A line integral convolution algorithm for visualizing unsteady flows (color plate S. 556). In Roni Yagel and Hans Hagen, editors, *Proceedings of the 8th Annual IEEE Conference on Visualization (VISU-97)*, pages 317–322, Los Alamitos, October 19–24 1997. IEEE Computer Society Press.

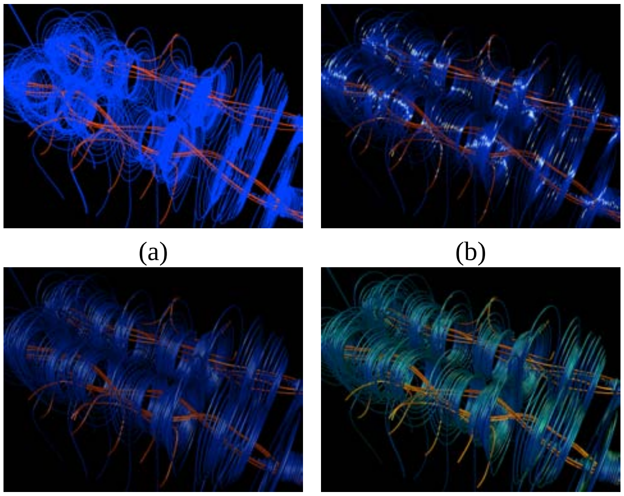
[26] Detlev Stalling, Malte Zöckler, and Hans-Christian Hege. Fast Display of Illuminated Field Lines. *IEEE Transactions on Visualization and Computer Graphics*, 3(2):118–128, April 1997.

[27] Anne Treisman. Preattentive processing in vision. *Computer Vision, Graphics, and Image Processing*, 31(2):156–177, August 1985.

[28] Shyh-Kuang Ueng, Christopher Sikorski, and Kwan-Liu Ma. Efficient Streamline, Streamribbon, and Streamtube Constructions on Unstructured Grids. *IEEE Transactions on Visualization and Computer Graphics*, 2(2):100–110, June 1996.

[29] Leonard R. Wanger, James A. Ferwerda, and Donald P. Greenberg. Perceiving spatial relationships in computer-generated images. *IEEE Computer Graphics and Applications*, 12(3):44–58, May 1992.

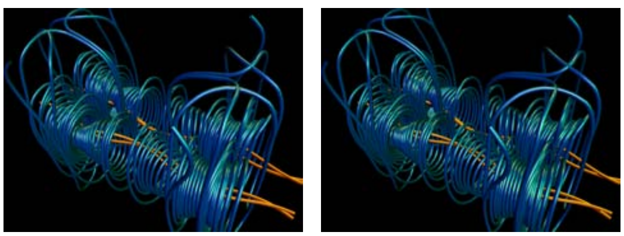
[30] J. M. Wolfe. Guided search 2.0: A revised model of visual search. *Psychonomic Bulletin and Review*, 1(2):202–238, 1994.



(a) (b)

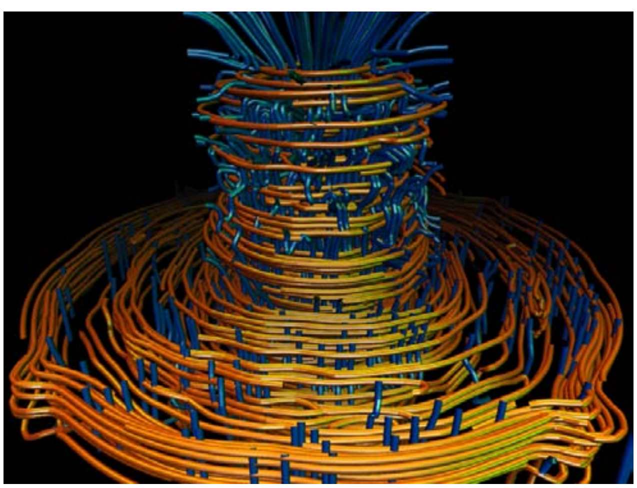
(c) (d)

Color Plate 1. Comparison of visual cues for different rendering techniques: plain lines (a), illuminated lines (b), SOS with a single light (c), and SOS with multiple light sources (d).

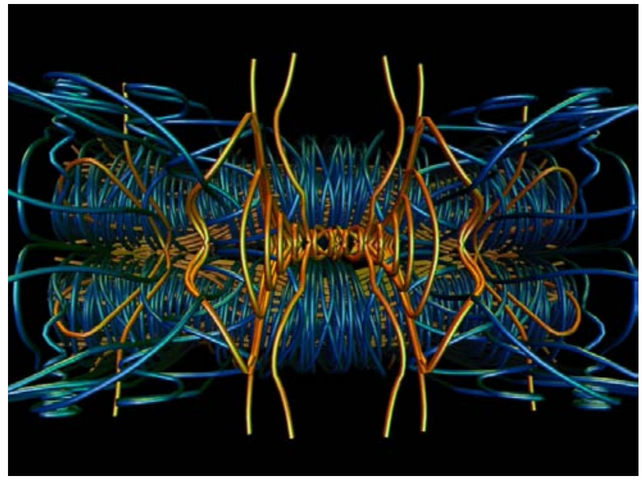


(a) (b)

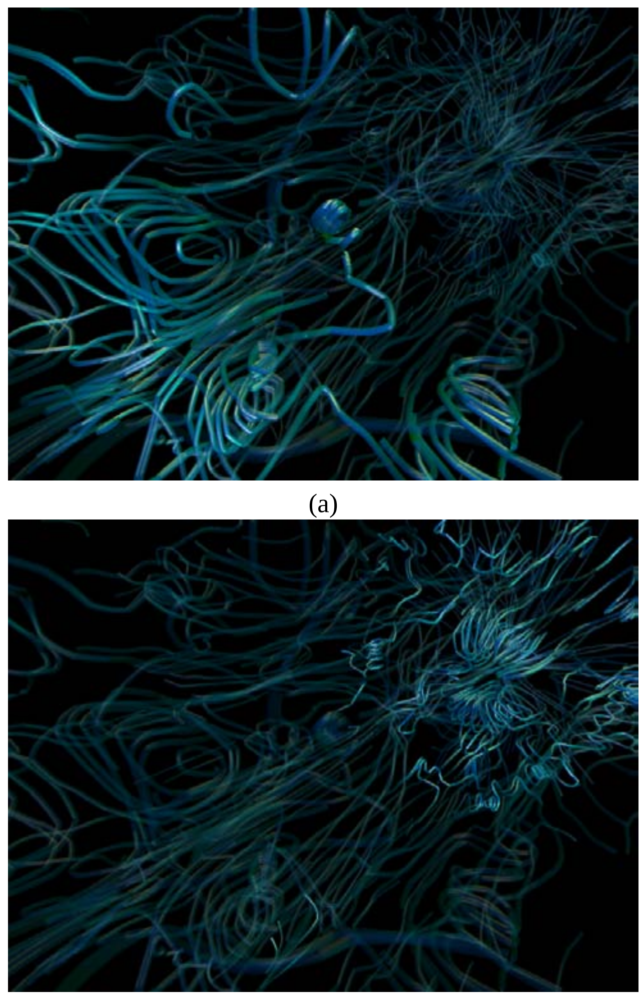
Color Plate 2. SOS lines (a) look almost identical to Polygonal Tubes (b), yet use considerably fewer triangles.



Color Plate 3. Electric (blue) and magnetic (orange) field lines in an RF cavity. The bumped regions in the magnetic field indicate where power is entering the structure via input ports.



Color Plate 4. Velocity (blue) and vorticity (orange) streamlines from an aerodynamics simulation of wake vortices from an aircraft during landing.



(a)

(b)

Color Plate 5. De-emphasis of all but Near (a) or far (b) via transparency. Without transparency, the far structure would be occluded.