

# LOW-COST HIGH-PERFORMANCE SCIENTIFIC VISUALIZATION

*The authors discuss the development of a low-cost stereoscopic visualization system using commonly available components. The system is used to improve understanding about the field-line structure and associated dynamics, confinement, and geometry of spheromak plasma. Such a system might interest research groups doing remote large-scale computing.*

Present-day scientific computation requires routine analysis of large 3D (or even higher) data sets. Often, complex geometry is an integral component of the problem's complexity, so it is only natural to apply 3D visual analysis in combination with more traditional analysis and diagnostic techniques. Recent advances in computer technology have pushed conventional desktop computers to the point where scientific visualization is feasible at a level that previously required expensive hardware systems and associated specialized software. These technological advances provide new opportunities for individual scientists to do substantial scientific visualization at their desks without the cost of a specialized visualization laboratory. One aspect of high-end visualization is the ability to create stereoscopic (see the sidebar) 3D projections.

We built and installed the visualization system described in this article at a modest cost of approximately \$15,000. Our goal was to obtain reasonable graphics capability that would still be

accessible to individual, application-side research groups. Our basic setup includes a Linux workstation, two video projectors, a dual-head graphics card, the Open-DX visualization software package, polarized glasses, polarizing filters, and a polarization-preserving lenticular projection screen. (A similar hardware setup is discussed elsewhere.<sup>1</sup>) Here, we build such a system and demonstrate its utility in computational plasma physics applications. In addition, we discuss the necessary development within Open-DX required to generate 3D vector field lines, volume rendering, and animation.

One of our guiding principles in developing this system was using low-cost components. This is possible because of the explosion of open source software, which provides free operating systems and visualization solutions. Also, the PC market's competitive nature has driven the hardware cost-performance ratio to the point of making the PC a reasonable low-cost solution. All equipment discussed is off-the-shelf and readily available over the Internet. In fact, many researchers might already have most of this equipment.

## The hardware setup

Our low-cost approach's primary merit is its flexibility. We present the specific hardware with which we implemented our system, but you can

## Stereoscopic Vision

Stereoscopy is the creation of separate images generated at the perspective of the left and right human eye. This gives the illusion of depth and three-dimensionality. The stereoscopic effect can be created in a number of ways. Anaglyphs are the common red-blue separated images that we can view with a pair of red-blue glasses. However, anaglyphs work best with black and white images, because intrinsic color defeats the red-blue filtering. Infrared synchronized shutter glasses allow the left-eye, right-eye images to be quickly alternated (generally at 60 Hz) and are commonly used in the scientific community. Some of these applications are immersive, such as the popular CAVE<sup>1</sup> environment and the Immersadesk. The drawback is that CAVE and CAVE-like environments are relatively expensive and largely inaccessible to scientists at remote sites.<sup>2</sup> Shutter glasses are also available for use with a single computer monitor and are a popular choice among computer

gamers.<sup>3</sup> A third option, which we implement here, is polarized filters to separate the stereo information intended for the left and right eyes. This method is effective for presenting color stereo information to large audiences because of the relative inexpensiveness of polarized glasses compared to their shutterglass counterparts.

### References

1. C. Cruz-Neira et al., "Surround-Screen Projection-Based Virtual Reality: The Design and Implementation of the CAVE," *Proc. Siggraph*, ACM Press, New York, 1993, pp. 135–142.
2. A. van Dam et al., "Immersive VR for Scientific Visualization: A Progress Report," *IEEE Computer Graphics & Applications*, vol. 20, no. 6, Nov./Dec. 2000, pp. 26–53.
3. C. Bungert, "Basics of Stereoscopic 3D," [www.stereo3d.com/quick.htm#shutter](http://www.stereo3d.com/quick.htm#shutter) (current 10 Apr. 2001).

make a working system with comparable equipment that might be more readily available. Our visualization system is an 800-MHz Pentium III PC running Linux with 512 Mbytes of RAM and controlling two off-the-shelf digital projectors. Visualization of 3D vector fields is very computationally intensive, so we need a system with a fast processor and a large internal memory for ideal performance.

Our graphics card is a Matrox G400+ dual-head accelerated graphics port card with an Xfree86 Graphics driver. The dual-head card provides two RGB outputs (to create separate output for the left- and right-eye images). An alternative is to use two single-head PCI cards.<sup>1</sup> In either case, we need a multihead X server. Numerous PCI and dual-head AGP cards are available and sufficiently powerful for visualization.<sup>2</sup> Our primary caution in selecting a graphics system is to insure that driver support (commercial or otherwise) is available.

We have used both commercial and open source graphics drivers. We currently use the Xfree86 4.0.2 driver for the Matrox G400+. Because of the dual outputs, we are unable to use the hardware rendering capability of the card and must use software rendering. Software rendering is generally slower than hardware, and this affects our ability to interactively manipulate images at an acceptable rate. We are currently exploring distributing the rendering processes over a Linux cluster to improve performance.

The two video outputs feed two separate digi-



**Figure 1.** A picture showing one of the authors (Samuel T. Jones) wearing the inexpensive polarized glasses and using the visualization system. To the right are the two LCD projectors (with polarized filters) used for the right and left stereoscopic images.

tal projectors. We used ViewSonic PJ1060 projectors with reasonable success; however, we have not extensively compared the numerous other available projectors. Important characteristics determine appropriate projector choice. The PJ1060 is an XGA-standard (1,024 × 786) projector, which is a relatively high resolution. The PJ1060 has a high luminosity rating of 2,000 ANSI lumens, which is important because the polarized filters block approximately 50 percent of the outgoing light. In addition, LCD projec-

Figure 2. A picture showing the basic setup, including the lenticular screen, the two LCD projectors, and the Linux workstation (on the right).



tors inherently produce a polarized light source. This causes a larger reduction in luminosity from the polarizing filter than we would expect from an unpolarized light source. Initial investigations of digital mirror-type projectors indicated that these projectors did not produce a polarized light source. However, the net filtered luminosity compared with LCD projectors did not make them competitive (their competitive advantage is that these devices are typically smaller).

We placed the two projectors (see Figure 1) one atop the other and adjusted them so that their areas of projection overlap. Placing polarized filters over their lenses at  $45^\circ$  and  $135^\circ$  angles differentiates the right-eye and left-eye projectors. This particular orientation matches the orientation of existing stereoscopic glasses and is an accepted stereoscopic standard. We superimpose the images from the two projectors onto a silver lenticular screen (see Figure 2), which preserves the polarization information. We can then view the resulting image through stereoscopic glasses with polarized filters matching the  $45^\circ$  to  $135^\circ$  orientation of the projector filters. The left lens blocks the orthogonally polarized

light from the right projector while allowing the left projector image to pass, and vice versa. This creates the desired stereo effect.

### The software setup

Obviously, we need visualization software to create the two images. We currently use Open-DX, an open source version of IBM Data Explorer. As open source software, DX is free and the source code is available for customization.<sup>3</sup> We could use other visualization software tools as well, such as AVS-Express and VTK. Clearly, individual choice is key, and it depends on software availability and past visualization experience. Open-DX is flexible in terms of the data formats that it accepts. It also handles nonuniform grids, general curvilinear coordinates, and other complex data sets, which was a minimum criterion for our choice of visualization software. DX is a graphical programming language, employing modules and connection lines instead of command statements. This visual style accentuates the data-flow environment on which DX is based. As a graphical development tool, DX (and several other visualization software packages) can create intuitive graphical interfaces to make the graphics more interactive. This interaction lets visual elements be manipulated without modifying the previously developed network.

To create stereo images in DX, we first render one image from a camera perspective provided by the user interface. We then generate a second rendering with a camera offset in the horizontal direction to simulate the spatial difference of the two eyes (approximately 6.5 cm).<sup>4</sup> We use a DX macro available from the Cornell University Center to make this spatial orientation transformation.<sup>5</sup> This macro works well for generating still images and creating

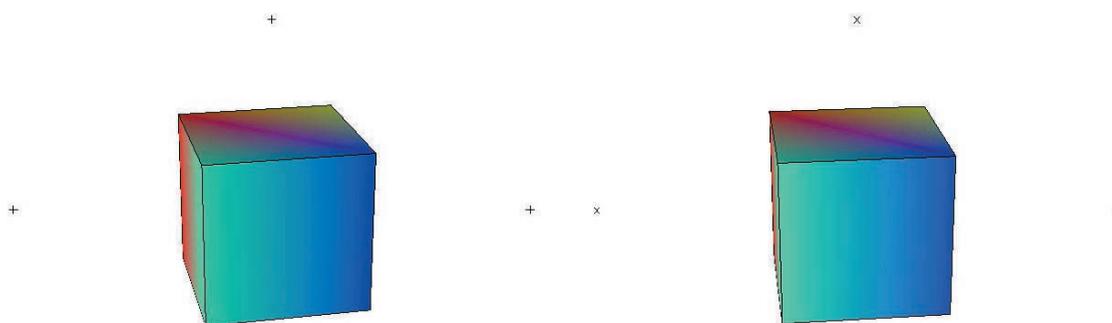


Figure 3. Still test images used for alignment and to develop stereo MPEG animations. The slight tilt angle between the two images produces the stereoscopic illusion that the cube is suspended in mid-air. The '+'s and 'x's are a convenient way to initially align the two projectors.

simple animations. For more complex data sets, the lag between rendering the left- and right-eye images becomes distracting and dilutes the stereo illusion. In this case, it is preferable to pre-render the images and compile them into an animation using the Berkeley MPEG encoder<sup>6</sup> or similar tools. We generate frames containing the left- and right-eye images so that playback has no synchronization problems. Figure 3 shows a still from a test pattern MPEG that we use to align the projectors.

### Application to 3D nonlinear simulations

One application of our low-cost visualization system and stereo capabilities is the analysis of NIMROD (the *nonideal MHD with rotation open discussion project*) simulation data. NIMROD is a nonlinear magneto-hydrodynamic (MHD) simulation<sup>7</sup> for modeling macroscopic plasma phenomena in toroidal plasmas. It uses a spectral representation in the toroidal direction and a combination of triangular and quadrilateral finite-element meshed regions (called *t*-blocks or *r*-blocks) in the poloidal plane. This grid is periodic in at least one dimension. We developed a simple interface called nimX to let Open-DX easily import the finite-element grid data. Open-DX has difficulty recognizing that data periodicity, such as volume and field-line quantities, should be interpolated across the periodic boundaries. We implemented periodicity by creating a module to patch the connections from one boundary to another. Let's examine the patching algorithm that creates the connections for a periodic data set.

Open-DX employs a data model in which a field is the fundamental object. A field primarily comprises data, positions, and connections. The connections component lists the vertices that define the interpolation element between points.<sup>8</sup> To illustrate this concept, consider the following example in two dimensions. Figure 4 shows an evenly spaced 5 × 3 grid. The *x,y* coordinates are such that vertex #0 is at [0, 0], vertex #1 at point [0, 1], and so on. Vertices 0, 1, 3, and 4 define a cell that we label #0. So we have a list of 15 (= *n* × *m*) vertex positions and a list of 8 (= *n* - 1 × *m* - 1) cell vertices:

Positions	Connections
0: [0 0]	0: [0 1 3 4]
1: [0 1]	1: [1 2 4 5]
2: [0 2]	2: [3 4 6 7]
:	:

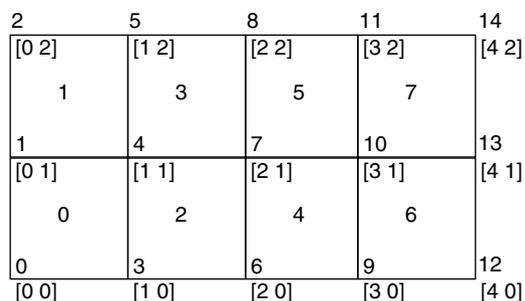


Figure 4. An array grid and connections using the Open-DX grid data structure. The numbers in brackets are the Cartesian coordinates of each vertex. The scalar at the upper right of each vertex is the vertex identification number, and the number on each cell face is the cell identification.

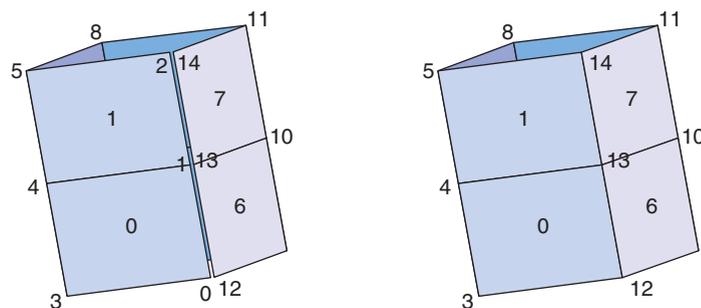


Figure 5. Grid connections for domains with periodic boundary conditions. Grid and connections after (a) vertices are coordinate-transformed and after (b) redundant vertices (0, 1, 2) are eliminated. The elimination of these vertices connects cell 0 to cell 6 and cell 1 to cell 7. This completes the grid's periodicity.

This works well until we attempt to work with a periodic set. Say that we want to enforce periodicity in the *x* direction. We could easily perform a coordinate transform to make the grid spatially periodic (see Figure 5a), but no connection remains between vertices 14 and 2, for example. To remedy this, we change the vertices of cell 0 to be [12 13 3 4] and likewise to cell 1. This creates the patch needed to facilitate the correct interpolation of data across the boundary (see Figure 5b). It requires, of course, vertices 0, 1, and 2 to be redundant to vertices 12, 13, and 14, respectively. We accomplish this position transformation in a general way with this logic:



Figure 6. An example of a highly nonaxisymmetric magnetic field line from a nonlinear magneto-hydrodynamic simulation. Note the corkscrew field-line bundle, which would not appear in a Poincaré surface of section diagnostic. The field line (a) alone and (b) with an isosurface measures the strength of the magnetic field,  $|B|$ , showing that there is a “ball” of magnetic field energy.

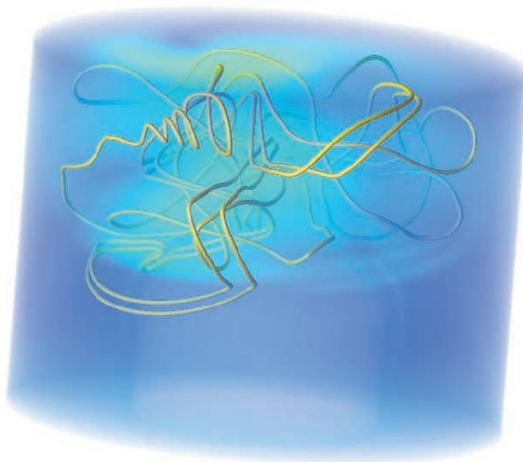


Figure 7. A volume rendering of magnetic field strength,  $|B|$ . Volume rendering is computationally quite demanding and a good test of our low-cost graphics system.

```

for each cell #
  if (vertex #0 mod (dim #1 x dim
    #0) < dim #1)
    then vertex #0 += ((dim #0
      - 1) x dim #1)
    and vertex #1 += ((dim #0
      - 1) x dim #1)
endfor

```

For example, in cell 1, vertex #0 is 1.  $(1 \bmod (3 \times 5) < 3)$  is true, so cell #1 vertex #0 becomes  $1 + ((5 - 1) \times 3) = 13$  and vertex #1 goes from 2 to 14. In general, to enforce a periodicity in the  $n$ th dimension of an  $m$ -dimensional system, you must change appropriate vertices when

$$(\text{vertex \#}0 \bmod (\text{dim \#}n \times \text{dim \#}n + 1 \times \dots \text{dim \#}m)) < (\text{dim \#}n + 1 \times \text{dim \#}n + 2 \times \dots \text{dim \#}m) \text{ with } \text{dim \#}m + 1 = 1.$$

With the simulation data loaded, along with the necessary periodicity patching, DX can produce volume renders, isosurfaces, section slices, and magnetic field-line traces (see Figures 6 and 7). nimX, the DX code we developed for this analysis, is available online along with a gallery of current NIMROD results that we are investigating.<sup>9</sup>

Spheromak field lines are complicated, and although DX does a nice job rendering them, they remain 2D projections of a 3D object. Following the path of the field line can be confusing from this perspective and the structure obscured. The stereo projection system remedies this, allowing us to explore the inherent 3D structure of the lines. Of practical interest is the formation of spheromak plasmas and whether a structure can resemble closed flux surfaces. That is, are there closed surfaces mapped out by magnetic field lines? NIMROD simulations using Poincaré surface sections at a fixed toroidal angle indicate that field lines make few (on the order of 10) passes around the toroidal direction before hitting the wall.<sup>10</sup> However, 3D visualization indicates what we describe as a nonaxisymmetric balling up of the field line, which indicates better confinement than expected from a Poincaré surface of section diagnostic. Keep in mind that the parallel heat conduction is very large, so any field line connecting to the wall still indicates poor confinement.

**W**e have assembled a high-performance stereoscopic visualization system using inexpensive and widely available PC hardware along with open source visualization software tools. Small individual research groups can easily assemble and use such a graphics system for a modest cost and for a wide variety of application-specific tasks.

The polarization method we use for stereography is more cost effective than shutterglass technology, but the polarized glasses are sensitive to the angle of viewing and a modest head tilt (approximately 15° or more) degrades the stereo effect. The stereo glasses also require the aforementioned silver lenticular screen while the shutter technology works for a fast-refresh-rate computer monitor. Furthermore, there is no way to incorporate head-tracking methods with the polarized glasses, which is an important element of a truly immersive visualization experience. There are obviously many trade-offs between price and performance that people should consider before choosing hardware components and implementing such a visualization environment.

The system described here performs well. The PC can play 1,280 × 480 MPEG at 20 frames per second, which lets us display two 640 × 480 full-screen images. DX does not animate at this same rate. Working without hardware rendering, the colored cubes, the sliceplane, and the isosurface can be animated in real time at approximately 2.5 frames per second at 2 × 640 × 480 resolution. The field lines require more processing and animate at 0.4 frames per second—approximately one frame every two seconds. The volume rendering takes too long for interactivity, which is why we developed the capability of pre-rendered stereographic MPEG animations for such applications. We are also exploring alternates to DX such as VTK (the Visualization Tool Kit), which might allow us quicker interaction with our stereo graphics. An additional option that we will explore in the near future is running the system on Microsoft Windows, where graphics drivers are almost always available. ❏

### Acknowledgments

We thank Carl R. Sovinec at Los Alamos National Laboratory and Bruce I. Cohen at Lawrence Livermore National Laboratory for supplying NIMROD simulation data and Viktor Przebinda for computing support. This work is an active part of the US Department of Energy, Office of Fusion Energy Sciences NIMROD Computational MHD Project.

### References

1. D. Bennett et al., "A Low Cost Commodity Based System for Group Viewing of 3D Images," *Proc. Visualization Development Environments*, Princeton Plasma Physics Library, Princeton, N.J., 2000, pp. 145–149.
2. D. Salvator, "Where's 3D Now?" *PC Magazine*, vol. 20, no. 1, Jan. 2001, pp. 170–181.
3. "Open-DX: Specification," [www.research.ibm.com/dx](http://www.research.ibm.com/dx) (current 10 Apr. 2001).
4. A. Klien, "How to Make 3D Pictures by Computer," [www.stereoscopy.com/3d-info/index.html](http://www.stereoscopy.com/3d-info/index.html) (current 10 Apr. 2001).
5. "Making a Stereo Camera," [www.tc.cornell.edu/Services/Vis/dx/DXTIPS/stereo.html](http://www.tc.cornell.edu/Services/Vis/dx/DXTIPS/stereo.html) (current 10 Apr. 2001).
6. "Berkley MPEG Tools," <http://bmrc.berkeley.edu/frame/research/mpeg> (current 10 Apr. 2001).
7. A.H. Glasser et al., "The NIMROD Code: A New Approach to Numerical Plasma Physics," *Plasma Physics and Controlled Fusion*, vol. 41, no. 3, Mar. 1999, p. A747.
8. *IBM Visualization Data Explorer: Users Guide 3.1.4*, IBM, Yorktown Heights, New York, 1997, pp. 21–22.
9. S. Jones, "nimX: 3D Visualization Support for NIMROD," <http://samjones.colorado.edu/research/nimx> (current 10 Apr. 2001).
10. C.R. Sovinec et al., "Formation and Sustainment of Spheromaks in the Resistive MHD Model," *Physics of Plasmas*, vol. 8, no. 2, Feb. 2001, pp. 475–490.

**Samuel T. Jones** is a graduate student at the University of Colorado, Boulder. His research interests include scientific visualization and Alfvénic phenomena in the Earth's magnetosphere. He received his BS in physics from Brigham Young University and his MS in physics from the University of Colorado, Boulder. Contact him at the Dept. of Physics, Univ. of Colorado, Boulder, CO 80309; [samuel.jones@colorado.edu](mailto:samuel.jones@colorado.edu).

**Scott E. Parker** is an assistant professor in the Department of Physics at the University of Colorado, Boulder. His research interests include large-scale kinetic simulation of low-frequency plasma physics. He received his PhD in engineering science at the University of California, Berkeley. Contact him at the Dept. of Physics, Univ. of Colorado, Boulder, CO 80309.

**Charlson C. Kim** is a physics doctoral student at the University of Colorado, Boulder. His research area is computational plasma physics. Currently, he is studying kinetic particle effects on magnetohydrodynamic instabilities in fusion devices. He received his BS in physics and applied mathematics from the University of California, Los Angeles. Contact him at the Dept. of Physics, Univ. of Colorado, Boulder, CO 80309.