

Virtual reality visualization by CAVE with VFIVE and VTK

N. OHNO, A. KAGEYAMA and K. KUSANO

The Earth Simulator Center, Japan Agency for Marine-Earth Science and Technology
3173-25 Showa-machi, Kanazawa-ku, Yokohama, 236-0001 Japan
(n_ohno@jamstec.go.jp)

(Received 17 August 2005 and accepted 15 December 2005)

Abstract. The CAVE-type virtual reality (VR) system was introduced for scientific visualization of large-scale data in the plasma simulation community about a decade ago. Since then, we have been developing a VR visualization software, VFIVE, for general CAVE systems. Recently, we have integrated an open-source visualization library, the Visualization Toolkit (VTK), into VFIVE. Various visualization methods of VTK can be incorporated and used interactively in VFIVE.

1. Introduction

Simulation science is supported by two key technologies: computation and visualization. While computer technology still keeps growing exponentially, its counterpart, visualization technology, does not. Therefore, the gap between the two key technologies is exponentially growing. It is becoming harder for simulation scientists to understand the output data by means of the traditional visualization technology based on graphic workstations or PCs. We believe that the modern virtual reality (VR) technology provides the solution to this technological imbalance. Among various VR systems developed to date, the CAVE-type VR system is especially suited to our purpose, i.e. the visualization of complicated three-dimensional simulation data. The CAVE is a room-sized, cubic-shaped VR system developed at the University of Illinois, Chicago (Cruz-Neira et al. 1993). We installed a CAVE system ‘BRAVE’ which is shown in Fig. 1(a) in our facility in 2003. Stereo images are projected onto its $3 \times 3 \text{ m}^2$ screens of the walls and the floor. The viewer stands in the CAVE’s room wearing the liquid crystal shutter glasses shown in Fig. 1(b). A magnetic sensor is attached to the glasses so that the projected images are automatically adjusted to the viewer’s position in real time. The stereo images for the four screens (three walls and one floor) are generated by four DLP projectors. Since they are projected seamlessly on their borders, everything looks natural from the viewer inside the CAVE and feels a deep immersion in the VR world. The viewer interacts with the simulation data through a portable controller, the ‘wand’, shown in Fig. 1(c). Another magnetic sensor is attached to the wand to detect its position and direction.

2. VFIVE and VTK

VFIVE is our original virtual reality visualization software for CAVE systems (Kageyama et al. 1999, 2000). The major visualization methods of VFIVE are

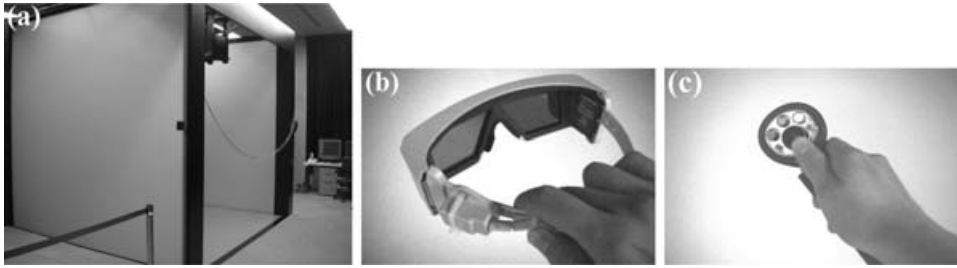


Figure 1. (a) Our CAVE system, 'BRAVE'. (b) Liquid crystal shutter glasses. (c) A portable controller, the 'wand'.

Table 1. Major visualization methods of VFIVE

For scalar data	For vector data
Isosurface	Field Lines
Color Contour	Particle Tracer
Local Slice Plane	Vector Allows
Contour Lines (VTK)	Spotlighted Particles
	Stream Surfaces (VTK)
	Stream Tubes (VTK)

listed in Table 1. VFIVE is a fully interactive visualization tool; we can control the isosurface level by the vertical motion of the wand in the CAVE room; we can start new tracer particles from the tip of the wand by pressing a wand button; and a spotlight is emitted by the wand and thousands of tracer particles are flying in the cone-shaped light. In the original version, VFIVE has only basic visualization methods such as isosurface and field lines. In order to add a new visualization method to VFIVE, one had to write a subset of computer graphics (CG) software with OpenGL and CAVELib. In contrast to usual CG programs, a CAVE application programmer does not have to write the projection part of the software since the CAVELib automatically takes care of that part. The programmer writes the *display process* part and the *application computation process* part. The former is to draw images on the screens and the latter is for calculations, e.g., the orbits of tracer particles. Therefore, simulation researchers who want a new visualization method in VFIVE for their own analysis need a fairly good knowledge and experience of programming with OpenGL and CAVELib, but it is clearly overwhelming for most simulation researchers. This has been a hurdle for VFIVE to become an actively used research tool in the simulation science. To resolve this problem, we have made a major upgrade of VFIVE by incorporating a general visualization library, the Visualization Toolkit (VTK).

VTK (Shroeder et al. 2002) is an open-source, freely available visualization software including many sophisticated visualization methods for scalar, vector, and tensor fields. They are implemented as a C++ class library. The usage is very simple: choose the modules in the library and combine them by the so-called 'VTK pipeline', then you will get a full set of high-performance visualization program without a detailed knowledge of the visualization algorithm and OpenGL. It is an attractive idea for CAVE users to integrate VTK into CAVE applications (Leigh et al. 1998).

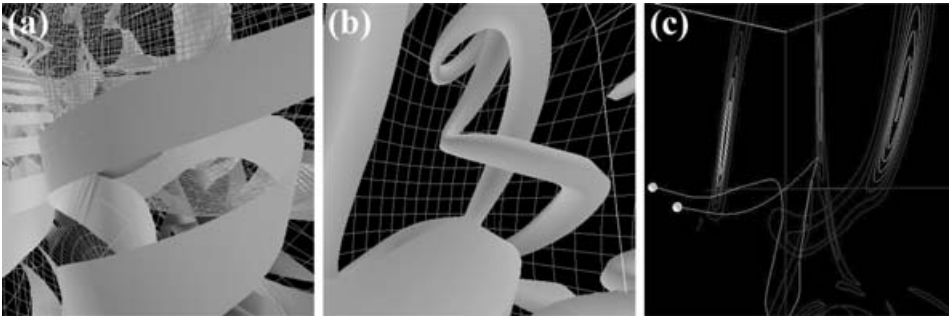


Figure 2. (a) Stream surfaces of the flow velocity of a geodynamo simulation. (b) Stream tubes of the flow velocity of a geodynamo simulation. (c) Contour lines of the magnitude of the electric current and field lines of the flow velocity of a Solar coronal simulation.

3. Integrating VTK into VFIVE

Many visualization methods of VTK are based on polygon-based rendering. This means that each visualization module converts the input field data to a set of numerous numbers of discrete polygons. The polygonal data consists of the vertex positions, vertex colors, and the normal vectors. In order to combine VTK with VFIVE, we need to show the VTK polygons in the CAVE's three-dimensional virtual space. However, to use VTK with the CAVELib is not easy due to the design mismatch between these two libraries.

To use VTK in the CAVE, Rajlich (1998) created a software 'vtkActorToPF' to extract the polygonal data of VTK and render them using IRIS Performer (or OpenGL Performer). Hall (1999) created 'vtk2CAVE', in which the *display processes* access the vtkActor copied in shared memory and draw its polygonal data with OpenGL. Shamonin (2002) created a library 'VtkCave', which makes it possible to create a CAVE application with VTK.

We took another approach to combine VTK into our CAVE program, VFIVE. Basically, we have followed the idea of 'vtkActorToPF', but we did not want to use IRIS Performer because VFIVE is written directly on OpenGL, without IRIS Performer. Therefore, we have revised VFIVE in such a way that the *application computation process* can make the polygonal data by using VTK and send them via the hard disk drive to the *display processes*, and the *display processes* can receive the output polygonal data of VTK and draw them with OpenGL. In this way we have succeeded in integrating important polygon-based visualization methods of VTK, for example, contour lines and stream surfaces. It is easy to integrate other VTK modules into VFIVE.

Here we show the power of revised VFIVE enhanced by VTK in the applications to MHD simulations. Figure 2(a) shows a VTK visualization of a geodynamo simulation data by stream surfaces. This method visualizes a three-dimensional vector field by ribbon-like surfaces created by multiple field lines started from a line segment. The colors of the ribbons indicate the magnitudes of the vector field. Figure 2(b) shows another VTK visualization, stream tubes. This method also visualizes a three-dimensional vector field by a field line wrapped by a tube by which more information than a simple one-dimensional field line can be conveyed. The colors of the tube indicate the magnitudes of the vector field and its radius depends on the vector magnitude. Figure 2(c) shows contour lines of the electric

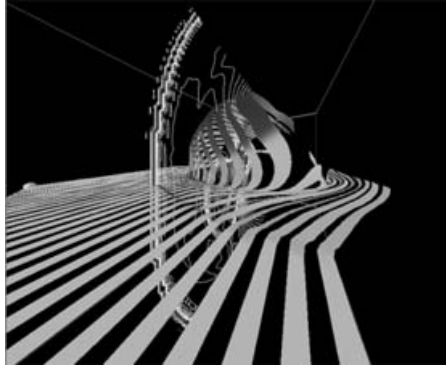


Figure 3. Stream surfaces of the flow velocity and contour lines of the pressure of the Earth's magnetosphere simulation.

current magnitude of a Solar coronal simulation. This is also a VTK visualization method. Figure 3 shows a mixed application of the two VTK visualization methods, stream surfaces and contour lines.

4. Conclusions

We have integrated VTK into our virtual reality software VFIVE for three-dimensional, interactive visualization in the CAVE system. The purpose of this improvement is to reduce the hurdle to add a new visualization function into VFIVE. VTK was chosen because of its rich and sophisticated visualization methods.

We believe that the development of an easy-to-use and easy-to-extend software for VR visualization in the CAVE, such as our VFIVE, is important for simulation science in the future.

Acknowledgement

We would like to thank Professor T. Ogino for providing us with the simulation data of the Earth's magnetosphere.

References

- Cruz-Neira, C., Sandin, D. J. and DeFanti, T. A. 1993 *Proceedings of the SIGGRAPH '93 Computer Graphics Conference*. New York: ACM Press, p. 135.
- Hall, M. 1999 'vtk2CAVE' Translator. <http://www.cave.vt.edu/people/shore/vtk/vtk2CAVE/>.
- Kageyama, A., Tamura, Y. and Sato, T. 1999 *Trans. Virtual Reality Soc. Japan* **4**, 717.
- Kageyama, A., Tamura, Y. and Sato, T. 2000 *Progress Theor. Phys. Suppl.* **138**, 664.
- Leigh, J., Rajlich, P., Stein, R., Johnson, A. E., DeFanti, T. A. 1998 *Proceedings of IEEE Visualization 98* Research Triangle Park, North Carolina.
- Rajlich, P. 1998 'vtkActorToPF' translator. <http://brighton.ncsa.uiuc.edu/~prajlich/vtkActorToPF/>.
- Shamonin, D. 2002 'VtkCave' library. <http://staff.science.uva.nl/~dshamoni/myprojects/VtkCave.html>.
- Shroeder, W., Martin, K. and Lorensen, B. 2002 *The Visualization Toolkit, 3rd edn*. New York: Kitware.