論文

没入型バーチャルリアリティ環境のためのVR Jugglerに基づく可視化ソフトウェア†

目野　大輔*・陰山　聡*・政田　洋平*

Visualization Software with VR Juggler for Immersive Virtual Reality Environment

Daisuke Meno*, Akira Kageyama* and Yohei Masada*

Abstract To analyze large scale 3-dimensional data, the modern virtual reality (VR) technology will play important roles in future simulation studies. Various VR visualization programs have been developed to date, including our original VR visualization software VFIVE. However, almost all of the previous VR software are based on CAVElib, which is a de facto standard commercial library for VR environments. To replace the basic API from CAVElib, we are developing a new visualization software based on VR Juggler which is an open source free software library. Our program design and implemented visualization methods are reported in this paper.

Key words Visualization, Virtual Reality, Immersive Virtual Environment

1. Introduction

To analyze three-dimensional (3-D) numerical data produced by large scale computer simulations, researchers today are required to find a new way to visualize their 3-D data with an ecient way. The virtual reality (VR) tech-nology with immersive display systems provides such an innovation for the 3-D data visualization.

The immersive VR systems, that are also called CAVE systems, were developed by Electronic Visualization Laboratory at University of Illinois, Chicago in early 1990's[1]. The classical CAVE system consists of a room-type screens, on which stereo images are projected, and a head/hand tracking system by which the view point of the stereo images are automatically adjusted to the viewer's head position.

Various visualization programs have been developed to date for the visualization in CAVE's VR environment. One of them is our original VR visualization software named "VFIVE"[2]. The purpose of VFIVE is to make it possi-ble to perform a fully immersive and interactive visualiza-tion of 3-D data in a CAVE's room. Through continuous development and improvements for more than a decade, VFIVE has become a practically useful tool for scientific visualization. For example, we have recently found a new phenomenon (a helical structure of electric current) in a supercomputer simulation data on geomagnetic field[3] by the VFIVE visualization in a CAVE. We believe that VFIVE is useful for many researchers in various fields, un-der the present situation that there are many CAVE sys-tems all over the world. We have made the source code of VFIVE being freely available[4]. However, popularly of VFIVE falls a little bit short of our expectation. One of the reasons would be a purely economical one; VFIVE is based on CAVElib which is a commercially available API for the interface to the CAVE hardware.

To make our visualization software being "fully free", we have decided to replace the basic API in our visual-iza-tion software, substituting a new free API instead of CAVElib. There are several free (or open source) APIs for

* Kobe University, Graduate School of System Informatics
† 2012年11月12日受付

CAVE-type VR systems, such as Open CABIN, CoVE, Vrui. Among them, we have decided to use VR Juggler[5]. VR Juggler is a platform for virtual reality application development. It allows a user to develop and run a VR ap-plication on wide variety of VR systems. An important feature of VR Juggler is that it is scalable from simple desktop systems like PCs to complex multi-screen CAVE systems running on high-end work stations. In contrast to some other freely available APIs, VR Juggler is actively maintained and upgraded. Instead of just replacing API, we have decided to develop a new visualization software for CAVEs from the scratch based on VR Juggler.

The development is performed on two CAVE systems in Kobe University, one is pCAVE (one screen system with head/wand tracking) and Section 1-CAVE, which is a newly in-stalled 4-screens, rectangular-shaped CAVE system with the size of 7.8 $m \times 3\ m \times 3\ m$[6,7].

2. OpenGL/CAVE Application Basics

For the overview, we briefly describe the basic archi-tecture of general OpenGL programs for CAVE system and other multiple display devices. We use VR Juggler as the management library [(4) in Fig.1]. It is an inter-face between the I/O devices [(2) and (3) in Fig.1] and visualization processes [(1) in Fig.1]. The input includes viewer's tracked data (head/wand positions and angles), wand buttons, joysticks, and the keyboard. The output includes projectors (CAVE screens), LCD monitors and speakers. The management library also takes care of the synchronization of multiple stereoscopic images when mul-tiple screens are used in CAVE system.

3. Design of New Visualization Software

Our program design is summarized in Fig.2. The main function creates two kinds of objects; a VR Juggler kernel and a visualization framework. The visualization framework consists of (1) visualization methods, (2) user in-terface, and (3) data set that is to be visualized. The VR Juggler kernel takes care of the interface with CAVE hardware, and controls the visualization framework through user defined functions such as init func, draw func, and so on.

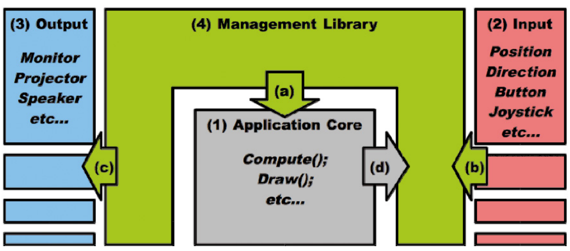


Fig.1 Basic architecture of OpenGL/CAVE application.

In the development of visualization programs [part (1) in Fig.1], we follow the adopted style of VR Juggler's cod-ing. We have developed several visualization methods in our program. They are inherited from a superclass named "VisualizationMethod".

As for the visualization methods, we have implemented two basic methods so far. One is called Particle Tracer that is for the vector field visualization. As the name suggests, Particle Tracer shows motions and trajectories of test (or massless) particles in a specified vector field (Fig.3). When one presses a wand button, a new tracer particle is released from the wand tip, and then the particle flies following the equation of motion $dx_i = dt = a_i\ (x_1;\ x_2;\ x_3)$, ($i$ = 1, 2, 3), where x_i is the i-th coordinate of the particle position, and a_i is the i-th component of the target vector field. The equation of motion is numerically integrated in real time. Repeated

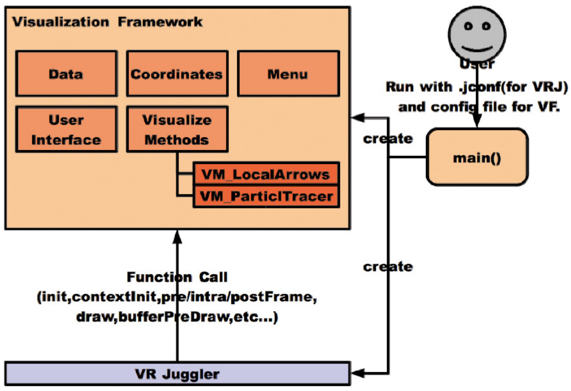


Fig.2 Software architecture.

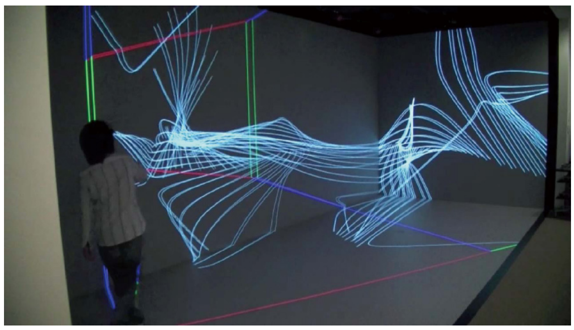


Fig.3 A snapshot of the visualization method Particle Tracer that is implemented in the present software.

clickings of the wand button generate a sequence of parti-cles. The trajectory of each particle is shown by a curve.

Another visualization method implemented in the code is called Local Arrows, which is also for the vector field vi-su-alization. In this visualization method, a bunch of short ar-rows, each of which denotes the direction and amplitude of the vector field at the location, are shown in CAVE's VR space around the hand (or the wand). See Fig.4. Since these arrows always appear around the wand, one can intu-itively understand the spatial variation of the vector field by observing the change of each arrow's length (that de-notes the vector amplitude there) and the vector direction by mov-ing the wand.

For the user interface, we have developed a kind of vir-tual touch screens in VR space. In this interface, rectan-gu-lar 2-D screens appear in front of the user that can be select-ed (or executed) by "touching" them with the wand. The touch is judged by the distance between the screen center and the wand position.

The touch screens are grouped to three different cat-ego-ries; (i) visibility control, (ii) calculation control, and (iii) input control. The visibility control is used to en-able/dis-able the visibility of the calculated visualization objects such as tracer trajectories and arrows. The cal-culation con-trol is used to specify, for example, if the nu-merical inte-gration of the tracer particle is on or off. The input control is used to specify/change the input devices such as joysticks and button allocations on the wand.

Visibility (on/off) and calculation state (also on/off)are two major states in almost all visualization method in CAVEs. In Fig.5, visibility of local arrows is on and visi-bility & calculation state of Particle Tracer are also on.

In our previous visualization software, VFIVE, visibil-ity and calculation states cannot be controlled by the user. In this regard, the present visualization software has more flex-ibility than VFIVE. By selecting through the virtual touch screens, one can specify any four possible states (on/off × on/off) of each visualization objects. Fig.6 shows an exam-ple in which the calculation state of Particle Tracer is off in the beginning. After the user has specified several seed points by the wand, he/she changes the calculation state to on. It is observed that every particles start moving in the vector field at once.

We use Visualization Library[8]. Visualization Library (VL) is a modern visualization library written in C++ and OpenGL. VL has many basic rendering methods (march-ing

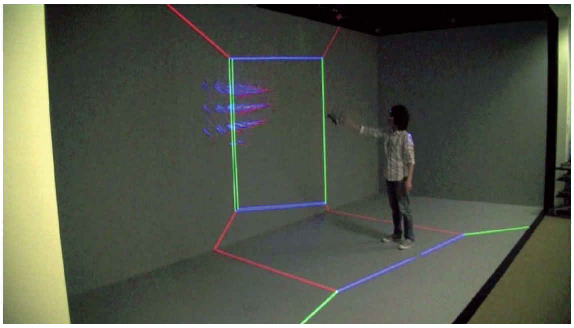
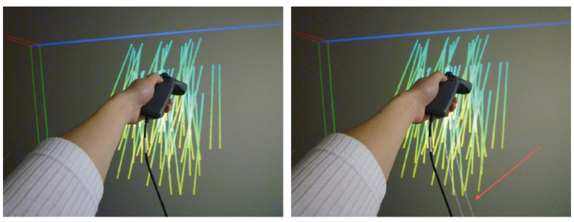
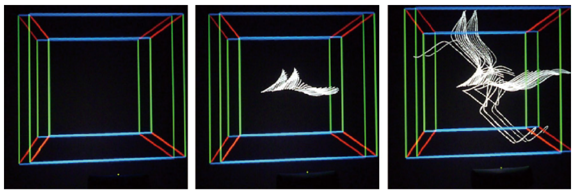


Fig.4 Another snapshot of CAVE visualization. Local Arrows for vector field are shown around the wand.



(a) (b)

Fig.5 A combined visualization of Local Arrows and Par-ticle Tracer. (a) Arrows (short lines) denoting 3-D vectors of the target field. (b) A new tracer particle is released from the seeding point when the user presses a wand but-ton.



(a) (b) (c)

Fig.6 A sequence of snapshots of Particle Tracer visual-ization.

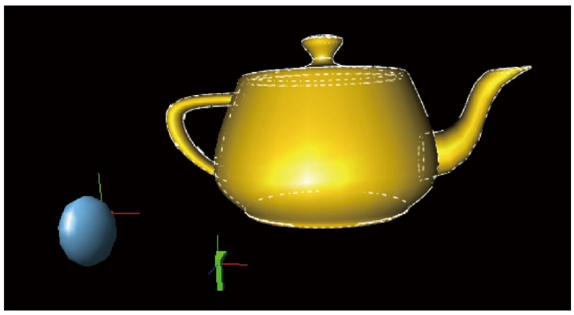


Fig.7 The Visualization Library's edge rendering in a VR Juggler environment. The teapot is yellow. Its edges are white lines. And hidden lines are white dotted lines. The Blue sphere is a simulated eyes of VR Juggler. The green object is a simulated wand of VR Juggler.

cubes, GPU raycast, sliced textures). Transparent polygons are sorted correctly. Edge enhancement (silhou-ettes, creas-es, hidden lines) is one of important functions implemented in VL. (Fig.7) Hidden lines will help us to rec-ognize a whole 3-D shape of objects occluded by their poly-gons. VL also provides various 3-D model loaders (COL-LADA, 3DS, OBJ, etc). They will enrich our software by rendering realistic 3-D models relating with the simulated or observed data. We apply C++ STL map data structure to VL compo-nents. We can obtain an unique integer value which thread is VR Juggler rendering. The key of map is the unique val-ue. The above are implemented on test code written with VR Juggler and VL, not on our software.

4. Summary

For interactive, immersive, and therefore efficient visu-alization of three-dimensional simulation data in an im-mer-sive VR environment, or CAVE, we have developed a new visualization software based on VR Juggler API in-stead of traditional CAVElib. Though the implemented visualization is still limited, we have found that the mod-ern design of VR Juggler enables us to develop a flexible scientific tool in CAVE’s VR environment. We will adopt modern useful li-braries.

Acknowledgements

This research was supported by Okawa Foundation and KAKENHI (23340128).

References

1) Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. De-Fanti, Robert V. Kenyon, John C.Hart, “The CAVE: audio visual ex-perience automatic virtual environment”, *Mag-azine of Com-munications of the ACM*, Vol.35, Issue 6, pp.64-72 (1992)
2) Akira Kageyama, Nobuaki Ohno, “Interactive 3-dimensional visualization virtual reality software”, *Jour-nal of Plasma Fu-sion Research*, Vol.84, No.11, pp.834-843 (2008)
3) Akira Kageyama, Takehiro Miyagoshi and Tetsuya Sato, “For-mation of current coils in geodynamo simulations”, *Nature*, vol.454, pp.1106-1109 (2008)
4) http://www.jamstec.go.jp/esc/research/Perception/vfive.en.html
5) Allen Bierbaum, Christopher Just, Patrick Hartling, Kevin Meinert, Albert Baker, Carolina Cruz-Neira, “VR Juggler: a virtual platform for virtual reality applica-tion development”, *Proceedings of IEEE Virtual Reality*, pp.89-96 (2001)
6) Akira Kageyama, Yuki Yamaura, Daisuke Meno, Youhei Masa-da, Ken-ichi Yoshizaki and Kouhei Yamada, “Appli-cation Launcher for CAVEs”, *Proceedings of JSST 2011*
7) Akira Kageyama, Youhei Masada, Yuki Yamaura, Daisuke Meno, Kouhei Yamada, “Immersive Visualization in π-CAVE”, *Proceedings of DS’11*
8) http://www.visualizationlibrary.org

Author’s Introduction

Daisuke Meno

Daisuke Meno is a master of engineering. He is the research associate of the Graduate School of System Informatics, Kobe University, Japan. His research interests include scientific visualization, immersive virtual reality.

Daisuke Meno was born in Osaka Prefecture, Japan, on April 20, in 1987. He received the bachelor and master degrees from Kobe University, in 2011 and 2013. He works as a limited-term assistant professor at the Integrated Research Center of Kobe University at Port Island, Kobe, Japan. His current research is a development of a seamless software system between supercomputers and an immersive virtual environment.

Akira Kageyama (Member)

Akira Kageyama is a professor at Kobe University. His research interests include magnetic field generations in nature, scientific visualization, and computational science in general. He was a group leader at Earth Simulator Center of JAMSTEC. He holds Ph.D. and M.S. degrees in physics from Hiroshima University.

Yohei Masada

Youhei Masada is an assistant professor of Kobe University. He received the M.S. degree in physics from Osaka University in 2005, and the Ph.D. degree in physics from Kyoto University in 2008. He worked as a postdoctoral research fellow in Academia Sinica Institute of Astronomy and Astrophysics, Taiwan (from 2008 to 2009), and in Hinode Science Project, National Astronomical Observatory of Japan (from 2009 to 2010) . His research fields are astrophysics, especially solar physics, and computational science.