

# Visualization Software with VR Juggler for Immersive Virtual Reality Environment

Daisuke Meno<sup>1</sup>, Akira Kageyama<sup>1</sup>, and Youhei Masada<sup>1</sup>

<sup>1</sup>Graduate School of System Informatics, Kobe University, Kobe, Japan

## 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 efficient way. The virtual reality (VR) technology 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 possible to perform a fully immersive and interactive visualization 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, under the present situation that there are many CAVE systems 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 visualization software, substituting a new free API instead of CAVELib. There are several free (or open source) APIs for 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 application 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  $\pi$ -CAVE, which is a newly installed 4-screens, rectangular-shaped CAVE system with the size of  $7.8m \times 3m \times 3m$  [6, 7].

## 2 Design of New Visualization Software

The program design is summarized in Fig.1. 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 interface, 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.

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.2). 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 clickings of the wand button generate a sequence of particles. 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 visualization. In this visualization method, a bunch of short arrows, 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.3. Since these arrows always appear around the wand, one can intuitively understand the spatial variation of the vector field by observing the change of each arrow's length (that denotes the vector amplitude there) and the vector direction by moving the wand.

For the user interface, we have developed a kind of virtual touch screens in VR space. In this interface, rectangular 2-D screens appear in front of the user that can be selected (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 categories; (i) visibility control, (ii) calculation control, and (iii) input control. The visibility control is used to enable/disable the visibility of the calculated visualization

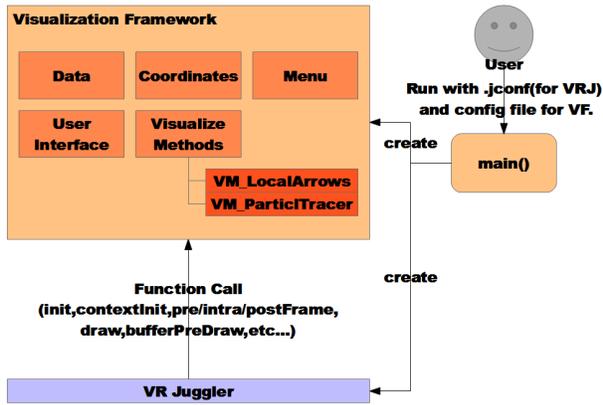


Fig. 1. Software architecture.

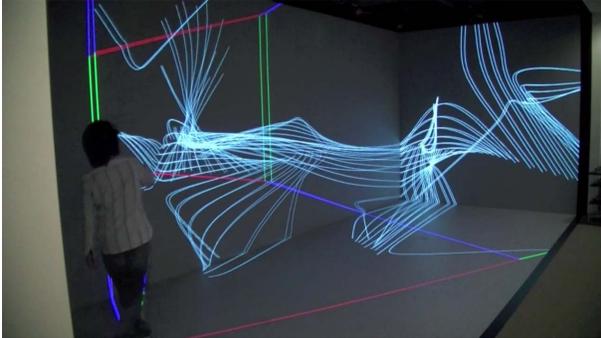


Fig. 2. A snapshot of the visualization method Particle Tracer that is implemented in the present software.

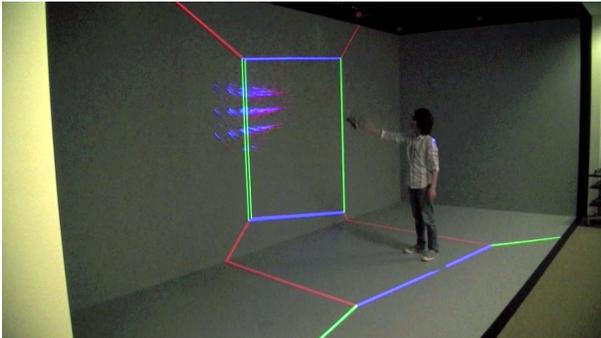


Fig. 3. Another snapshot of CAVE visualization. Local Arrows for vector field are shown around the wand.

objects such as tracer trajectories and arrows. The calculation control is used to specify, for example, if the numerical integration 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.4, visibility of local arrows is on and visibility & calculation state of Particle Tracer are also on.

In our previous visualization software, VFIVE, visibility and calculation states cannot be controlled by the user. In this regard, the present visualization software has more flexibility than VFIVE. By selecting through the virtual touch screens, one can specify any four possible states (on/off  $\times$  on/off) of each visualization objects. Fig.5 shows an example 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.

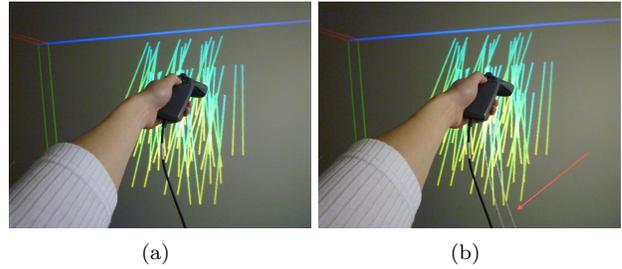


Fig. 4. A combined visualization of Local Arrows and Particle Tracer. (a) Arrows (short lines) denoting 3-D vectors of the target field. (b) A new tracer particle is released from the seeding point when the user presses a wand button.

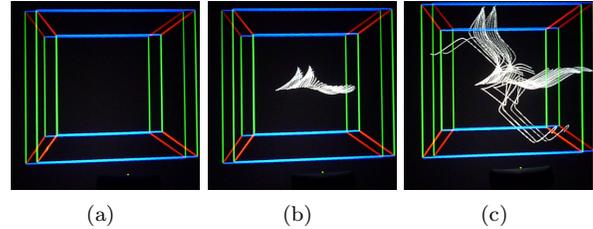


Fig. 5. A sequence of snapshots of Particle Tracer visualization.

### 3 Summary

For interactive, immersive, and therefore efficient visualization of three-dimensional simulation data in an immersive VR environment, or CAVE, we have developed a new visualization software based on VR Juggler API instead of traditional CAVElib. Though the implemented visualization is still limited, we have found that the modern design of VR Juggler enables us to develop a flexible scientific tool in CAVE's VR environment.

### Acknowledgements

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

### References

- [1] Carolina Cruz-Neira, Daniel J. Sandin, Thomas A. DeFanti, Robert V. Kenyon, John C.Hart, "The CAVE: audio visual experience automatic virtual environment", *Magazine of Communications of the ACM*, Vol.35, Issue 6, pp.64-72(1992)
- [2] Akira Kageyama, Nobuaki Ohno, "Interactive 3-dimensional visualization virtual reality software", *Journal of Plasma Fusion Research*, Vol.84, No.11, pp.834-843(2008)
- [3] Akira Kageyama, Takehiro Miyagoshi and Tetsuya Sato, "Formation 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 application development", *Proceedings of IEEE Virtual Reality*, pp.89-96(2001)
- [6] Akira Kageyama, Yuki Yamaura, Daisuke Meno, Youhei Masada, Ken-ichi Yoshizaki and Kouhei Yamada, "Application Launcher for CAVEs", *Proceedings of JSST 2011*
- [7] Akira Kageyama, Youhei Masada, Yuki Yamaura, Daisuke Meno, Kouhei Yamada, "Immersive Visualization in  $\pi$ -CAVE", *Proceedings of DS'11*