VOIR: Virtual Reality Visualization Software for Large-Scale Simulations

Nobuaki OHNO and Akira KAGEYAMA[1)]

Graduate School of Information Science, University of Hyogo, Kobe 651-2197, Japan
[1)]Graduate School of System Informatics, Kobe University, Kobe 657-8501, Japan

(Received 24 March 2024 / Accepted 8 June 2024)

Recent technological advances have significantly improved the quality of virtual reality (VR) provided by head-mounted display (HMD) systems. The immersive experiences provided by contemporary HMDs provide an effective environment for scientific research, especially for three-dimensional data visualization. Plasma physics has decades of history of using VR technologies for visualizations with particular emphasis on room-sized, large-scale display systems called CAVE. Modern HMDs are a cost-effective alternative to CAVEs. This study focuses on replicating the CAVE's VR software, VFIVE, developed at the National Institute for Fusion Science. The newly developed software for HMDs, VOIR, is built on Unity with the OpenXR plugin. It allows for interactive analysis of 3D scalar and vector fields in VR. One of the key features of VOIR is the ability to enlarge and analyze specific regions of large-scale simulation data in real time.

Keywords: scientific visualization, virtual reality, head-mounted display

DOI: 10.1585/pfr.19.1401024

1. Introduction

Scientific data visualization is one of the critical application fields of Virtual Reality (VR) [1–3]. Plasma physics has a long history of actively applying VR technology to the visualization [4]. Among them, National Institute for Fusion Science (NIFS) was one of the earliest facilities that introduced a CAVE-type VR system for visualization of simulation data [5]. CAVE is a room-sized VR system with immersive displays on the floor and walls [6]. A general-purpose VR visualization program for CAVEs was developed at NIFS after several years of experiences of the development of VR visualization programs.

The software, VFIVE [7], was also installed and used in other CAVE systems in Japan [8,9] for various purposes of 3-dimensional, interactive visualizations in CAVEs. For example, a helical structure of electrical current in a geodynamo simulation, one of the largest geophysical simulation at that time, was first found by VFIVE [10].

The performance of VR devices are mainly determined by the following two factors: (1) Resolution and field of view: Image resolution and wide field of view provide clearer and realistic images, enhancing the immersion of the VR experience; (2) Tracking accuracy: The tracking technology of a VR device is crucial for its ability to precisely follow the user's movements and to enable real-time interaction.

While a CAVE provides high quality VR environment, CAVE is also known to be expensive and requires a large space for installation.

author's e-mail: ohno@gsis.u-hyogo.ac.jp

Recently, the performance of head-mounted displays (HMDs) has drastically improved. They are overwhelmingly more cost-effective than CAVEs. The rise of modern HMDs suggests the possibility of utilizing software assets that are developed for CAVE-type VR systems such as VFIVE.

This study aims to achieve essentially the same functionality in modern HMD devices as was achieved with the VFIVE in the CAVE. The developed software is called VOIR (Visualization Operating with Immersive Reality).

VOIR enables the user to interactively analyze 3D scalar and vector fields in VR space. One of the features of VOIR is a recursive region-of-interest extraction function, which is indispensable for the visualization of large-scale simulation data. A portion of the simulation region specified by the user is enlarged in real time. Repeating this procedure allows a portion of the selected region to be further enlarged for more detailed analysis.

VOIR is built on Unity and its OpenXR plug-in. Unity [11] is a game engine and we use it here as a middleware for VR applications. OpenXR is an open standard API for VR and Augmented Reality (AR) devices. Therefore, VOIR runs on various types of recent HMD devices.

The HMD systems have been used in the scientific visualization [12–14]. A similar approach as VOIR based on the Unity framework can be found in [15]. Visualization library based on VTK for Unity is also available [16]. Other studies on the application of HMDs to scientific research are also underway, including gesture control by hand [17], CAVELib emulator [18], on-the-fly data exploration [19]. For a recent review of VR visualization, including HMDs, refer to [20].

2. HMD Visualization with VOIR

Using the Unity game engine, one can easily display prescribed objects defined in, say, FBX format [21] in the VR space. It is also possible to display user-defined polygons. We use OpenXR, a Unity plug-in for output to VR devices.

2.1 Software architecture

Figure 1 shows the flow chart of VOIR. In the beginning, VOIR reads a configuration file that specifies the simulation data to be visualized; the data size, type, and paths in the file tree. VOIR accepts data defined in rectangular coordinates; uniform coordinates or rectilinear coordinates. The configuration file is written in the plain text. An example of the configuration file is shown in Listing 1. The data information of the configuration file is used to estimate the amount of memory allocation to load the input data.

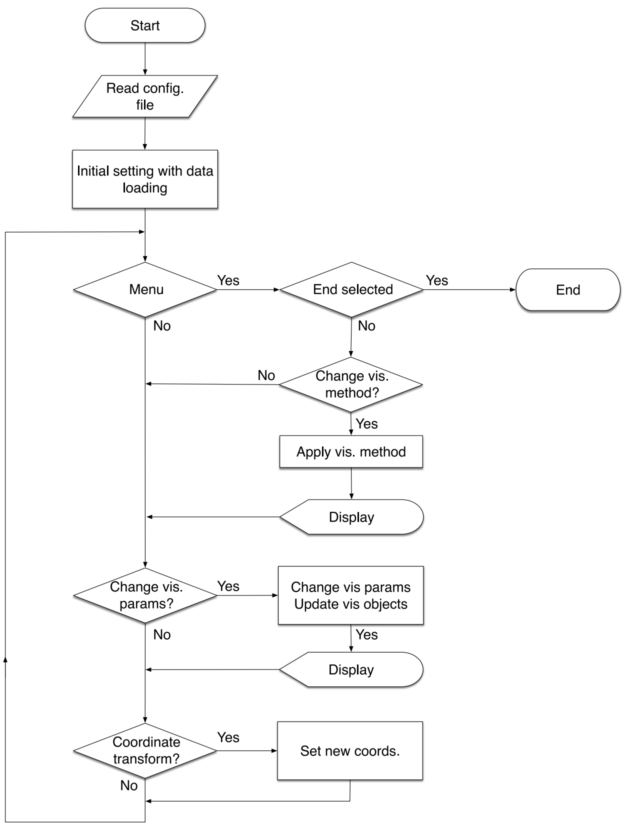

Fig. 1 Flow chart of VOIR.

Listing 1 An example of the configuration file of VOIR.

```
GRIDSIZE      800  800  800
NVEC          1
NSCAL         2
UNIFORM
SINGLEPRECISION
DX  0.00125  0.00125  0.00125
CORNER  −0.5  −0.5  −0.5
VECT0_LABEL  Velocity
VECT0X    c:\Users\ohno\turbdata\hvelx
VECT0Y    c:\Users\ohno\turbdata\hvely
VECT0Z    c:\Users\ohno\turbdata\hvelz
SCAL0_LABEL  Pressure
SCAL0     c:\Users\ohno\turbdata\hpressure
SCAL1_LABEL  Enstrophy
SCAL1     c:\Users\ohno\turbdata\henstrophy
```

Figure 2 (a) shows an image of a user wearing an HMD device while visualizing a simulation data with VOIR. A function of VOIR is selected by shooting through a panel in the menu with the laser beam emitted from the controller. Figure 2 (b) is a sample snapshot of the user's field of view. A menu panel is selected by the laser beam. A submenu, if any, appears on the right hand side of the shot panel. Figure 2 (c) shows a sample submenu panels, which are to select the applied visualization method in this case.

The menu system is constructed with rectangular objects and text objects that are implemented in Unity. The collision detect feature of Unity is used for the panel selection with the beam.

Every visualization result is constructed as a set of point, line, or polygon objects with Unity. In some visualization methods, a visualization object is constructed and shown in real time. For example, in "Particle tracer" visualization method, ball-like objects fly in the VR space with

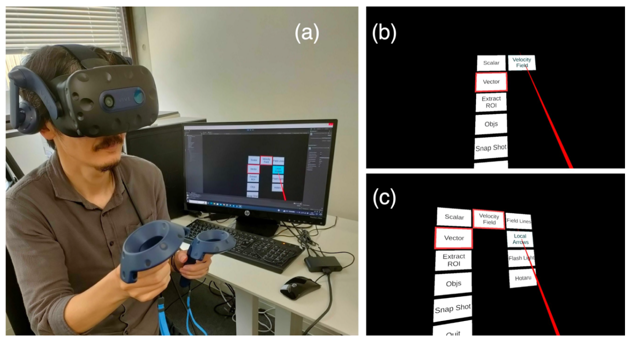

Fig. 2 (a) Overview of VOIR usage with HTC VIVE Pro. This HMD requires devices called "Base Station" to track the headset and controllers. At least two of them must be installed. The headset and controllers receive the infrared light emitted from the devices and determine their positions and orientations. (b) Menu panel in the user's view who is selecting a panel in the second column in the menu to specify the target field to be visualized. In this case "Velocity Field" will be selected. (c) The user is selecting a sub-menu panel in the third column to specify the visualization method to be applied to the target field. In this case "Local Arrows" will be selected to visualize the velocity field.

trailer curves behind them. The trailer curves grow in real time while the numerical integration of the particles motion is performed to visualize the target vector field. The initial condition of a ball, or the seed position, is interactively specified by the user with the tip of the laser beam.

The hand-held controller is used to specify other visualization parameters such as the isosurface level. Moving the controller upward (downward) increases (decreases) the level. OpenXR API is used to retrieve the tracking data of the controller's height.

For translation and rotation of the viewpoint in the VR space, a joystick of the controller is used.

Some parts in visualization algorithms implemented in VOIR are computationally demanding. For example, in the isosurface visualization, we construct myriads of polygons that collectively constitute isosurfaces. We use thread parallelization with `C#` to accelerate the computation. We divide the whole simulation region into multiple slices perpendicular to the z axis and apply the isosurface generation algorithm in each slice in parallel. The users can automatically benefit from the parallelization of VOIR with no special care in a multi-core PC system for their HMD system.

2.2 Implemented visualization methods in VOIR

The user of VOIR specifies the target data to be visualized, the method of the visualization, and the visualization parameters by the GUI and controller.

The user can specify multiple visualization methods/-targets at once. The corresponding visualization results are superimposed with automatic hidden surface removal.

Owing to the tracking functionality of the HMD, the user can intuitively change the viewpoint/viewing direction by just changing the head position/direction. It is also possible to apply coordinate translations by the controller's joystick.

The following visualization methods are available in VOIR:

Isosurface: This is a basic visualization method for scalar fields. A contour surface of a specified field is constructed in real time by a kind of the marching tetrahedra algorithm. Two isosurfaces can be shown at the same time in maximum. The level of the isosurface can be changed interactively by vertically moving the hand-held controller. A three-dimensional cone-shaped icon appears as the indicator of the relative value of the isosurface level. The maximum and minimum values of the whole scalar field is indicated by the upper and lower bounds of the cone. An example of the isosurface visualization with the cone-shaped icon is described in Sec. 3.

Ortho-slice: This is also a visualization method for scalar fields. Color contour and the height plot for a scalar field is shown. The plane is perpendicular to one of the x-, y-, and z-axis of the simulation coordinate system. The slice position of the axis can be changed in real time by moving the controller. Two ortho-slices for each axis, six in total in maximum, are possible to be shown at the same time.

Local-slice: This is yet another slice-based visualization method. Color contour of a specified scalar field is shown on a square plate. The plate, whose center is at the tip of a virtual beam, is always perpendicular to the beam. Observing dynamically changing color contour on the plate, by moving the location of the plate in space, the user can intuitively grasp the three-dimensional distribution of the scalar field.

Stream line: This is a visualization method for vector fields. The seed points are specified by the beam. Releasing a controller's button starts numerical integration of a stream line from the seed point at the tip of the beam. Multiple stream lines, with different colors for different vector fields, are presented.

Arrow glyph: This is also a visualization method for vector fields. Arrow-shaped glyphs appear in a relatively small volume around the tip of the virtual beam. The size and direction of each arrow illustrate the vector fields there. As the user move the virtual beam in the space, the small region in which the arrow glyphs appear follow the motion. Observing the dynamical changes of the local arrows while moving the beam enables the user to intuitively understand the 3D distribution of the target vector field.

"Flash light": This is visualization method for vector fields. A cone-shaped region is "illuminated" by a virtual flash light emitted from the controller. Thousands of tracer particles in the virtually illuminated cone region are presented like wind-blown snow flakes in the night. As the user changes the position and direction of the virtual flash light, he/she can understand the 3D "wind", or the target vector field. This kind of fully interactive and 3D analysis is a characteristic feature of VOIR.

User-defined objects: It is possible to display three-dimensional objects specified by the STL (Standard Triangulated Language) format.

2.3 Region-Of-Interest

A ROI (Region-Of-Interest) feature in tandem with automatic control of LOD (Level-Of-Detail) is implemented in VOIR to realize interactive visualization of large-scale and highly-complex data; see Fig. 3. We took advantage of our experience in implementing the ROI with the LOD function in VFIVE for CAVE-type devices, implementing this time the equivalent function for HMDs [22]. Here we illustrate the feature in case of the isosur-

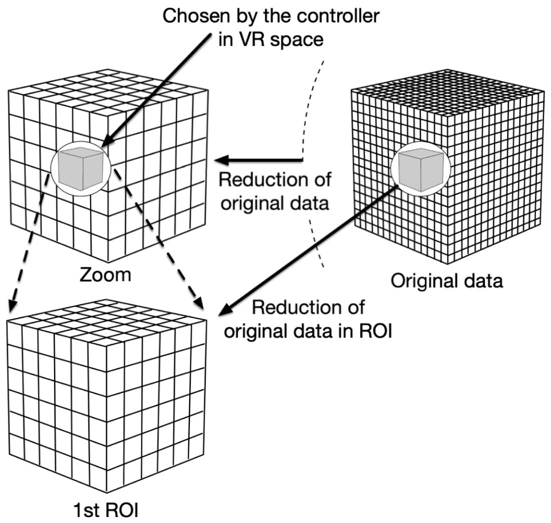

Fig. 3 The concept of Region-Of-interest (ROI) with automatic control of Level-Of-Detail (LOD).

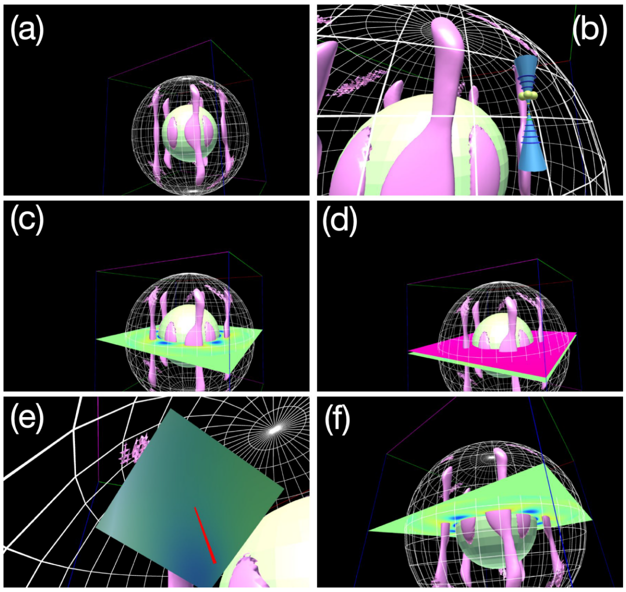

Fig. 4 VOIR visualization of a spherical thermal convection simulation.

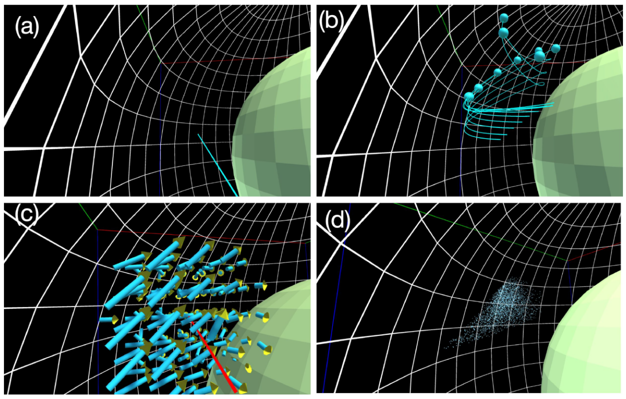

Fig. 5 Visualization methods of VOIR for vector fields.

face visualization method. Similar procedure is applied to other visualization methods, too.

Firstly, VOIR generates a low-resolution data with grid size N^3 from the original simulation data, whose grid size is, say, $M_1 \times M_2 \times M_3$, where $M_i > N$. A default value of $N = 150$. The tri-linear interpolations are applied to reduce the grid size. This data, which is the coarsest, is used to comprehend the overall structure of the data.

Secondly, when the user specifies a small region (ROI) in the whole simulation space by the controller, VOIR samples the local ROI out of the original data in such a way that the grid size in the ROI is again N^3, the same size as the previous one.

Finally, this ROI extraction with the automatic LOD control can be applied recursively up to the finest resolution that is the same as the original one of the simulation.

3. Application Examples

Figures 4 and 5 are sample snapshots of the VOIR visualization applied to fluid simulations.

In Fig. 4, thermal convection in a rotating spherical shell is analyzed in terms of the axial vorticity, ω_z. The resulting isosurfaces are shown as bar-like pink objects. In panel (b), the user gets closer to the simulation region, or the meshed sphere. The user interactively changes the isosurface level by moving the hand-held controller vertically. The level is visually presented with a pair of cones; the upper cone for positive levels and the lower cone for negative ones. In panel (c), additional visualization method, "Orthoslice", is selected from submenu and the resulting planes are superimposed. In this case, the distribution of the axial ω_z is visualized in a slice plane perpendicular to the z-axis. In panel (d), the position of the slice plane can be controlled by the hand-held controller. A temporal plane (red) is presented to indicate the slice position while the user is moving the controller vertically with a button being pressed. The slice position is fixed when the button is released. The slice axis is specified by a submenu. In panel (e), "Local slice" is used to analyze the target field. A regular square plane that appears at the beam tip illustrates the local distribution of the field there. As the user changes the direction of the controller, the square plane follows the beam motion. In panels (e) and (f), the user is located inside or close to the simulation boundary or the spherical mesh. VOIR provides highly immersive experiences in the visualization.

In the following, we are to describe a typical sequence of procedure to analyze a vector field in the VR space with VOIR. Figure 5 shows visualization of the flow vector field in the same simulation as in Fig. 4. Figures 5 (a) and (b) are for "Field lines" method. The beam tip specifies the seed point of a field line. A new tracing of the field line with a sphere on its tip is invoked every time a controller button is

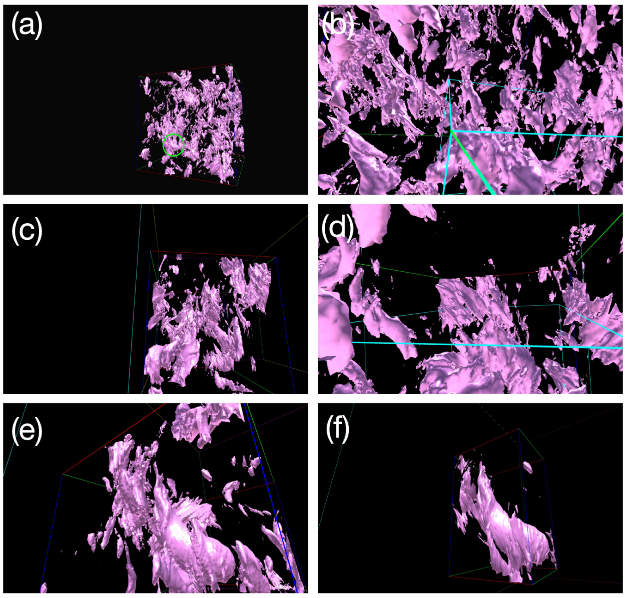

Fig. 6 (a) Overall view of the simulation. The user has decided to focus on a specific region, i.e., an ROI (circled in green). (b) Using the virtual beam, the user "crops" a rectangular ROI, depicted by light blue lines. (c) The cropped ROI is automatically enlarged. (d) For more detailed visualization, the user chooses a sub-region of the previously selected ROI with the beam. (e) The second level of the ROI is again enlarged. According to the LOD feature of VOIR, the spatial resolution of the data used for isosurface rendering is kept constant. (f) The user applies the ROI cropping yet again to analyze the ROI in detail.

released. As the numerical integration by the Runge-Kutta method goes on, the spheres fly in the simulation space with their trails as field lines. Figure 5 (c) shows "Local arrows" visualization method. As the user moves the beam, tens of arrow glyphs smoothly track the tip of the beam. As the user repeatedly moves the tip back and forth around a particular point in space, the length and direction of the arrows change in a characteristic pattern. The observation of the dynamic change of the pattern provides a chance to intuitively grasp complex structures of the spatial distribution of a three-dimensional vector field. The combination of hand movement and vision is a prominent feature of VR visualization with VOIR. Figure 5 (d) shows "Flash light" visualization method. Thousands of tracer particles appear inside a conical region with the user's controller at its apex. This is like using a virtual flashlight to illuminate countless tracer particles floating in the night sky.

Now we describe a typical exploration of the data utilizing the ROI feature of VOIR. Figures 6 (a) to (f) show a snapshot sequence of visualization of a fluid turbulence simulation [23]. The scalar field data of the enstrophy density q_ω on the cartesian grid of the size $M_1 \times M_2 \times M_3 = 1000^3$ is analyzed.

In the beginning, an isosurface visualization of q_ω is viewed from a distance from the simulation space, presenting an overall view of the simulation at a given time step; see panel (a). The spatial resolution of the data processed by VOIR is the coarsest in this initial stage, i.e., $N^3 = 150^3$.

In the next step, the user manipulates the controller's stick to enter the simulation space. Looking around the whole VR space, the user locates an ROI before approaching there. Then the user specifies the ROI by making use of VOIR's menu. A short beam appears from the controller's tip. By moving the tip of the beam in the VR space (while holding down the button), the user can specify a diagonal of a box-shaped ROI; see the blue lines in panel (b). VOIR cuts out the data in the ROI from the original simulation data. Because the extracted ROI is a part of the whole simulation region, the spatial resolution of the data in this ROI is higher than the original grid with $N^3 = 150^3$. The isosurface visualization with the same level in the previous step, shown in panel (a) is automatically applied under the higher spatial resolution. The construction of the isosurface takes just a couple of seconds in this case. It is as if the body has shrunk all at once in that time.

The user can now see the 3D structure of the complex geometry of the isosurface in more detail. Then, the user looks around again to further explore the ROI in this level of detail. Having moved to another spatial location, the user again uses the controller to specify another ROI, or a small box in this level (panel (d)). The procedure is thus recursive. The user can go down to a smaller and smaller scale by repeated application of the ROI selections; see panels (e) and (f).

4. Summary

We have developed an application program VOIR for immersive visualizations in VR space provided by modern HMD devices. We have implemented in VOIR basic visualization methods such as isosurface, field-lines, orthoslices, arrow glyphs, and others. The visualization parameters of those methods can be interactively controlled in real time via input devices of the HMD systems.

Because VOIR is developed on Unity and OpenXR, one can execute VOIR on different types of HMDs without changing the source code.

VOIR allows VR visualizations with modern HMD systems. As long as the memory of the PC system permits, it can visualize simulation data of up to $O(1000^3)$ grid points owing to the ROI feature with automatic control of LOD.

The target users of VOIR are simulation researchers who routinely analyze scalar and vector fields of large and complex 3D data. It is straightforward to download and compile the VOIR's source code.

VOIR's code and documents are available at GitHub [24].

Acknowledgments

This work was supported by JSPS KAKENHI Grant Numbers 22K18703 and 22H03603. We thank Prof. H. Miura for providing the turbulence data used in Fig. 6.

[1] G. Evagorou and T. Heinis, Visual exploration and interaction with scientific data in virtual reality, In Evagorou2018VisualEA (2018).

[2] C. Kwon, Verification of the possibility and effectiveness of experiential learning using HMD-based immersive VR technologies, Virtual Real. **23**(1), 101 (March 2019).

[3] P. Atsikpasi and E. Fokides, A scoping review of the educational uses of 6DoF HMDs, Virtual Real. **26**(1), 205 (March 2022).

[4] G. Foss, A. Solis, S. Bhadsavle, W. Horton and L. Leonard, Plasma simulation data through the hololens, In Proceedings of the Practice and Experience on Advanced Research Computing, number Article 105 in PEARC '18, pages 1-2, New York, NY, USA, (July 2018), Association for Computing Machinery.

[5] A. Kageyama, T. Hayashi, R. Horiuchi, K. Watanabe and T. Sato, Data visualization by a virtual reality system, In Proceedings of 16th International Conference on the Numerical Simulation of Plasmas (ICNSP1998), pages 12-19 (1998).

[6] C. Cruz-Neira, D.J. Sandin and T.A. De Fanti, Surround-Screen Projection-Based virtual reality: The design and implementation of the CAVE, In Proceedings of the 20th annual conference on Computer graphics and interactive techniques, pages 135-142 (1993).

[7] A. Kageyama, Y. Tamura and T. Sato, Visualization of vector field by virtual reality, Progr. Theoret. Phys. **138**, 665 (April 2000).

[8] A. Kageyama, N. Ohno, S. Kawahara, K. Kashiyama and H. Ohtani, Immersive VR Visualizations by VFIVE PART 2: Applications, Int. J. Model. Simul. Sci. Comput. **04(supp01)**, 1340004 (August 2013).

[9] A. Kageyama and N. Ohno, Immersive VR Visualizations by VFIVE PART 1: Development, Int. J. Model. Simul. Sci. Comput. **04(supp01)**, 1340003 (August 2013).

[10] A. Kageyama, T. Miyagoshi and T. Sato, Formation of current coils in geodynamo simulations, Nature **454(7208)**, 1106 (August 2008).

[11] N. Alejandro Borromeo, *Hands-On Unity 2021 Game Development: Create, customize, and optimize your own professional games from scratch with Unity 2021, 2nd Edition* (Packt Publishing, 2 edition, August 2021).

[12] S. Marks, J.E. Estevez and A.M. Connor, Towards the holodeck: Fully immersive virtual reality visualization of scientific and engineering data, In Proceedings of the 29th International Conference on Image and Vision Computing New Zealand, IVCNZ '14, pages 42-47, New York, NY, USA, (November 2014), Association for Computing Machinery.

[13] K. Nagao, Y. Ye, C. Wang, I. Fujishiro and K.-L. Ma, Enabling interactive scientific data visualization and analysis with see-through hmds and a large tiled display, In 2016 Workshop on Immersive Analytics (IA), pages 1-6, IEEE (March 2016).

[14] S. Orlando, M. Miceli, U. Lo Cicero and S. Ustamujic, Virtual reality for the analysis and visualization of scientific numerical models, J. Italian Astronomical Society **94**(1), 12 (2023).

[15] H. Miyachi and S. Kawahara, Development of VR visualization framework with game engine, Transaction of the Japan Society for Simulation Technology (in Japanese) **12**, 59 (2020).

[16] kitware. ActiViz. https://www.kitware.eu/activiz/, 2023. Accessed: 2023-NA-NA.

[17] Y. Tamura, H. Nakamura and S. Fujiwara, An intuitive interface for visualizing numerical data in a Head-Mounted display with gesture control, Plasma Fusion Res. **11**, 2406060 (2016).

[18] S. Kawahara and A. Kageyama, Development of CAVELib compatible library for HMD-type VR devices, J. Adv. Simul. Sci. Eng. **6**(1), 234 (2019).

[19] B.K. Horton, R.K. Kalia, E. Moen, A. Nakano, K. Nomura, M. Qian, P. Vashishta and A. Hafreager, Game-Engine-Assisted research platform for scientific computing (GEARS) in virtual reality, SoftwareX **9**, 112 (January 2019).

[20] E. Hilal Korkut and E. Surer, Visualization in virtual reality: a systematic review, arXiv 2203.07616 (March 2022).

[21] Unity Documentation. Unity user manual 2022.3 / model file formats. https://docs.unity3d.com/2021.3/Documentation/Manual/3D-formats.html. Accessed: 2024-2-21.

[22] N. Ohno and A. Kageyama, Region-of-interest visualization by CAVE VR system with automatic control of level-of-detail, Comput. Phys. Commun. **181**(4), 720 (April 2010).

[23] H. Miura and K. Araki, Structure transitions induced by the hall term in homogeneous and isotropic magnetohydrodynamic turbulence, Phys. Plasmas **21**(7), 072313 (July 2014).

[24] N. Ohno, VOIR: Interactive visualization software for HMD, https://github.com/vizlab-ohno/voir. Accessed: 2024-3-21.