doi: 10.32131/jes.6.25

Journal of the Earth Simulator, Volume 6, October 2006, 25–34

Visualizations of Large-scale Data Generated by the Earth Simulator

Fumiaki Araki[1]*, Hitoshi Uehara[1], Nobuaki Ohno[1], Shintaro Kawahara[1],
Mikito Furuichi[1] and Akira Kageyama[1]

[1] *The Earth Simulator Center, Japan Agency for Marine-Earth Science and Technology (JAMSTEC),
3173-25, Showa-Machi, Yokohama 236-0001, Japan*

(Received June 23, 2006; Revised manuscript accepted August 3, 2006)

Abstract Recent research and development on visualization methods and tools carried out by Advanced Perception Research Group of the Earth Simulator Center are reported. Focuses are placed on two software programs developed, (1) MovieMaker, a parallel rendering software for generating sequential image files from large scale simulation data of the order of 1TB, and (2) VFIVE, a virtual reality (VR) visualization software for a CAVE-type VR system. The designs and implementations of these programs are described with sample snapshot images applied to simulation data produced by the Earth Simulator. A couple of other visualization methods for three-dimensional vector fields investigated in the group are also presented.

Keywords: Large-scale data visualization, Earth Simulator, volume rendering, parallel processing, virtual reality, visualization techniques, animation

1. INTRODUCTION

Data visualization is an indispensable and valuable step in the simulation research. The raw output data become perceptible only through the visualized images. However, in ultra-high resolution simulations, data visualization often falls into difficult situation. This situation becomes serious in the era of the Earth Simulator (ES). Considering the ever-lasting development of computer hardwares, this would become much more serious in the coming era.

The difficulty in the large-scale data visualization can be classified into two categories; (1) the difficulty due to the data size itself, and (2) the difficulty caused by an intrinsic complexity of the data's spatial and temporal structure.

The first difficulty comes from the simple fact that, because of their extremely large size, the data cannot be visualized on the simulation researchers' PC of ordinary use. Let us, for example, consider a typical size of output data produced by a simulation on the Earth Simulator. The data has spatial three-dimensional (3D) and time-varying structures, and contains several variables for several hundred time-steps. Even a single variable for only one time-step has a few giga bytes (GB), implying that the total output data amounts to the order of tera bytes (TB). Such large data can be hardly handled by commercially available visualization tools on PC. Even if it can be done by some means, it would take impractically long time. A straightforward way to resolve the data size problem is to drastically reduce the amount of the data by sacrificing the temporal and spatial resolution. The reduced data could be quickly transferred and easily analyzed by the standard visualization tools. However, the data reduction is not always an effective approach, because it may lose some important features related to phenomena included in the original data. To resolve the large data size problem, a new visualization tool is strongly required that can handle a large-size data of the TB-scale without data reduction. Since such data has hundreds of time-steps, it would be wise and appropriate to represent the data in terms of a movie (sequential images corresponding to each time-step of the original data). The tools, however, must rely on visualization technologies with a high calculation cost, such as volume rendering and isosurface generating, and yet execution must be finished within a reasonable time.

The second difficulty lies in how one can extract and grasp hidden featuring characteristics of highly-complicated structures in a high-resolution 3D simulation data. The difficulty becomes more prominent for higher resolution of simulation data. Therefore, it is very necessary to incorporate the advanced technologies into the traditional

* ***Corresponding author****: Dr. Fumiaki Araki, The Earth Simulator Center, Japan Agency for Marine-Earth Science and Technology (JAMSTEC), 3173-25, Showa-Machi, Yokohama 236-0001, Japan. email: arakif@jamstec.go.jp*

visualization process in order to make an intuitive comprehension of 3D complex structures. In the meantime, the recent progress of the virtual reality (VR) system enables a viewer to literally "enter" into a virtual 3D space. The viewer can "walk" in the virtual space and "interact" with any object virtually existing there. Thus, by applying VR technology in scientific visualization, the simulation data can be analyzed in a more interactive and intuitive manner.

The intrinsic complexity hidden in the data is particularly important and significant in analyzing no-scalar field variables. For example, analysis of a vector variable needs simultaneous representation of not only the intensity but also the direction. So far the Line Integral Convolution (LIC) method has been often used, where the vector field is expressed by a set of many short streamlines. Although the original LIC method is developed under the 2D texture-based modeling, we can expect to comprehend the complicated structure of the 3D vector field if this method is extended to 3D texture.

In the Earth Simulator Center, Advanced Perception Research Group (APRG) focuses primarily on the above-mentioned two problems for the large-scale data visualization. In this paper, we present our recent research and development of the visualization software tools. For resolving the first problem of handling the extremely large dynamic data, say, TB data, we developed a software tool, named as "MovieMaker" (in Section 2). In challenging the second problem, namely, in elucidating the intrinsic complexity hidden in the massive data, we developed the CAVE-type VR hardware environment, named as "BRAVE" and our original VR visualization software tool, "VFIVE" (in Section 3). In Section 4, we report two kinds of effective visualization methods for vector fields. One is a 3D extension of original LIC method and the other is a kind of the Oriented LIC method, "Comet-Like Tracer". Main products of our visualization methods and tools are usually movies. In order to convey the quality of our visualizations, some sample snapshots of our movies are shown in Section 5. Section 6 is the summary.

2. "MovieMaker"

Most of commercially available software programs for the scientific visualization cannot deal with large-scale data exceeding a few GB for one snapshot. Of course, there are such programs that can handle larger data. However they take impractically long time to complete the processing. The volume rendering and the isosurface are, though very powerful and therefore popular, especially time consuming. If it takes as long as 10 minutes to make one snapshot image, it is practically impossible to make a movie for a time-varying simulation that requires at least a thousand snapshots, since 10 min × 1000 ~ 1 week. There are commercially available tools, but they are powerless for the TB-scale data visualization.

This situation has motivated us to develop our own software tool, "MovieMaker". The goal of MovieMaker is to produce a movie from 1TB simulation data within 12 hours, or one night [1]. We confirmed that the MovieMaker can perform 3D visualization from 1 GB snapshot data within approximately 40 seconds. This means that the MovieMaker can process one thousand snapshot (up to 1 TB) data by about 11 hours.

MovieMaker incorporates two different kinds of visualization methods; the polygon-based visualization and the volume-based visualization. In the polygon-based visualization, we make up 3D objects from simulated data by assembling small polygons. An example of the polygon-based visualization is the isosurface generation of a scalar field. In the volume-based visualization, in contrast, we generate a visual image by tracing virtual rays starting from one view point. The color of a pixel of

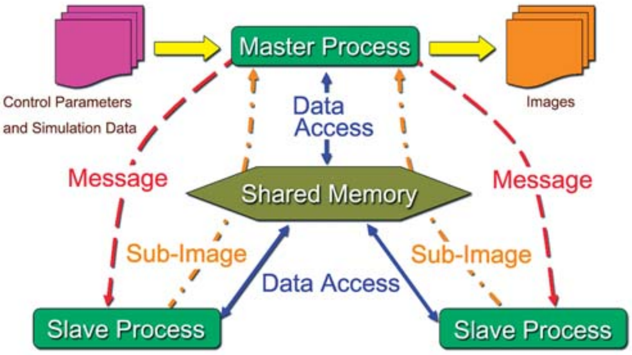

Fig. 1 The Design of MovieMaker by master/slave mode.

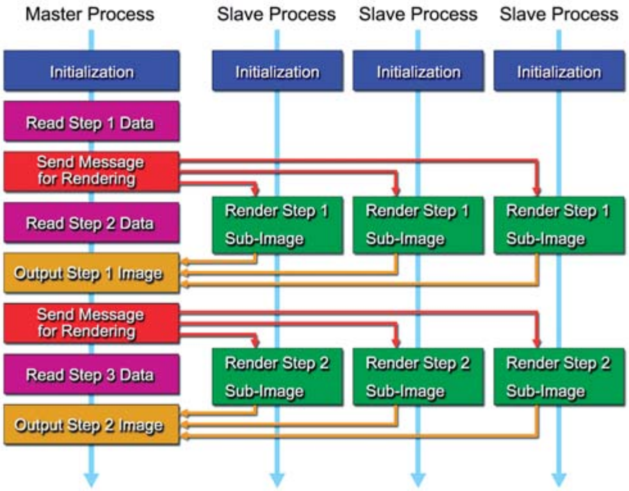

Fig. 2 The Flowchart of MovieMaker.

the image is calculated by integrating the data values along the ray path that runs through the pixel. A typical example of the volume visualization is the volume rendering.

We have designed MovieMaker as a master/slave parallel rendering program for the shared-memory architecture. As shown in Fig. 1, the master process and slave processes share the simulation data stored in the shared memory area. The master process performs the following tasks; (1) to read a configuration file, (2) to read the simulation data into the shared-memory area, and (3) to control the slave processes, keeping the optimal load balance.

The slave processes perform rendering tasks according to commands sent from the master process and then return partial-images back to the master via shared memory. The inter-process communications are performed with Message Passing Interface (MPI). The rendering tasks allocated to all the slave processes are dynamically controlled by the master: When a slave process finishes a rendering task, the master detects it and immediately assigns a new task to the process.

In addition to the dynamic load balancing described above, we have also implemented the task overlapping method for efficient parallel processing. Fig. 2 shows the flowchart of MovieMaker. While the slave processes are working for rendering tasks for one specific time step of the simulation data, the master process reads the data of the next time step.

MovieMaker is implemented in C++ with OpenGL. The present version of MovieMaker has three visualization methods; (1) volume rendering, (2) isosurfaces, and (3) streamlines. The volume rendering and the isosurfaces are implemented based on the basic algorithms of the ray casting [2] and the marching-cubes [3], respectively. The isosurfaces and streamlines are rendered by the off-screen rendering function of OpenGL. These three visualization methods can be used separately or simultaneously. The fine-tuning of each visualization method is possible by controlling visualization parameters specified by a configuration file.

Figs. 3 and 4 show sample images of the volume rendering applied to the mass fraction of the material ejected from the volcanic vent on the high-resolution eruption cloud simulation such as the eruption at Mt. Pinatubo in June 1991 [4, 5] by MovieMaker. The distribution of the eruption cloud is represented in Fig. 3. The intensity of the mass fraction and structure in the eruption cloud is appeared by the volume rendering with colors and opacity in Fig. 4.

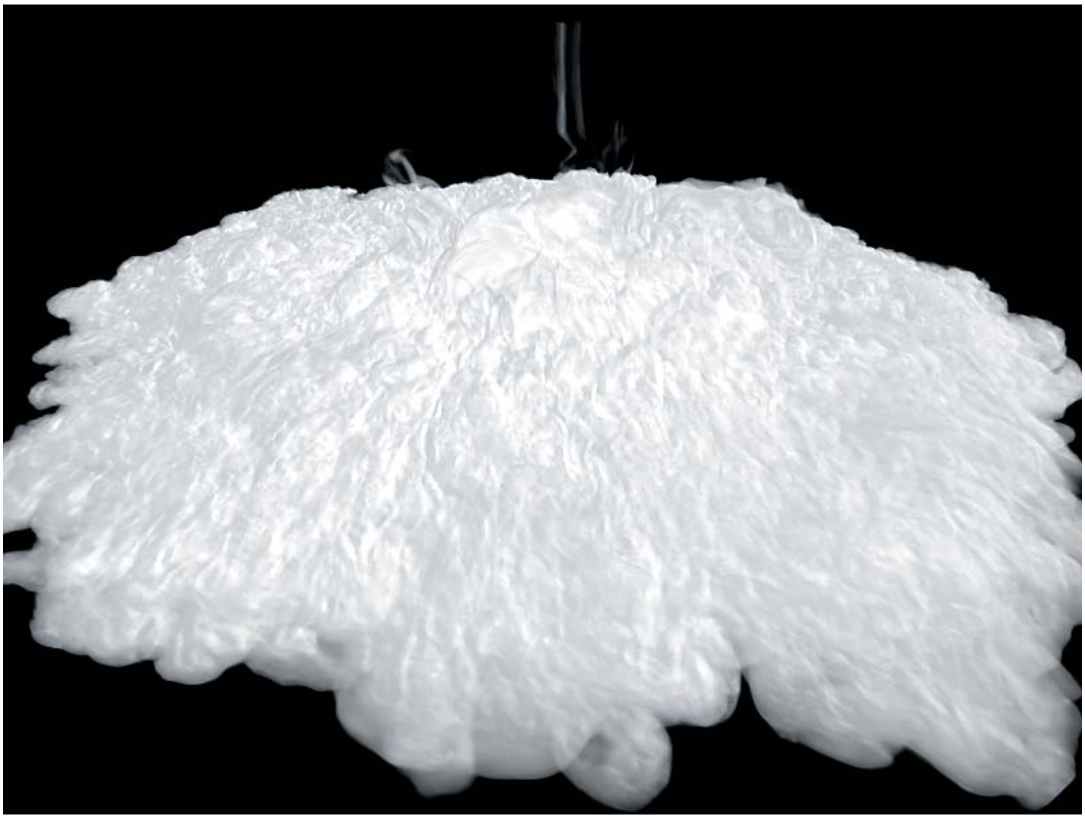

Fig. 3 The mass fraction of the material ejected from the volcanic vent by the volume rendering (grayscale) of MovieMaker.

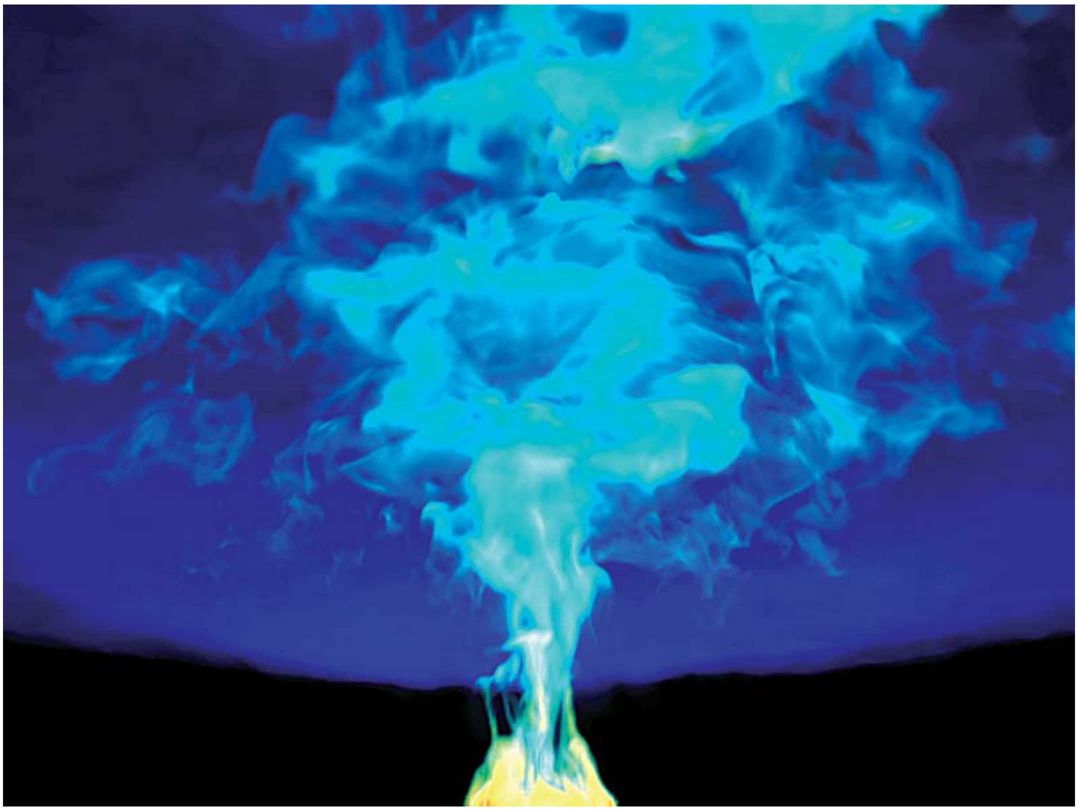

Fig. 4 The volume rendering with colors and opacity of the mass fraction as same as Fig. 3 by MovieMaker.

3. "BRAVE" AND "VFIVE"

In order to analyze three-dimensional (3D) data in 3D space, we installed an advanced virtual reality (VR) system "BRAVE" in 2003 (Fig. 5). BRAVE is a CAVE-type VR system [6] that was developed by University of Illinois, Chicago. The central part of the CAVE is a cubic booth surrounded by four screens (three of which are side walls and the rest is a floor). The CAVE makes a virtual world therein, by projecting stereoscopic images onto the screens. The stereoscopic images surrounding the viewer inside the booth compose a highly immersive sense. In addition, a head tracking system is equipped on the CAVE. This system enables the viewer to walk freely in the CAVE's cubic booth. It always tracks viewer's head position and regenerate the stereo computer graphics on the screens that should be seen as natural as possible from the viewer's eyes position and direction. As for the head tracking system, we have adopted an optical motion capture technology (Fig. 6). The 3D scene in the CAVE's booth is automatically adjusted to the user's view point.

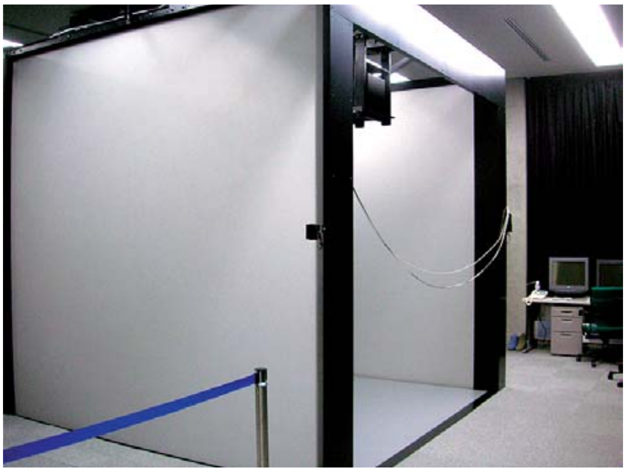

Fig. 5 A CAVE system "BRAVE" installed in the Earth Simulator Center.

Each of the screens is a square with the side of 3 m. Images are projected by four DLP projectors with 2000 ANSI rumens to the screens (the walls and the floor). The user wears a pair of stereo-glasses with a shutter system using the liquid crystal (see the left of Fig. 7). Each image for the right or the left eye is projected to the screens alternately with the frequency of 96Hz. The switchovers between right- and left-shutter of the glasses are synchronized with the alternate right- and left- images projected on the screens by an infrared system with fourteen infrared emitters around the CAVE's booth.

Furthermore, CAVE system can receive interactive operations by the viewer through a portable controller. For interactive control of the visualization, BRAVE is

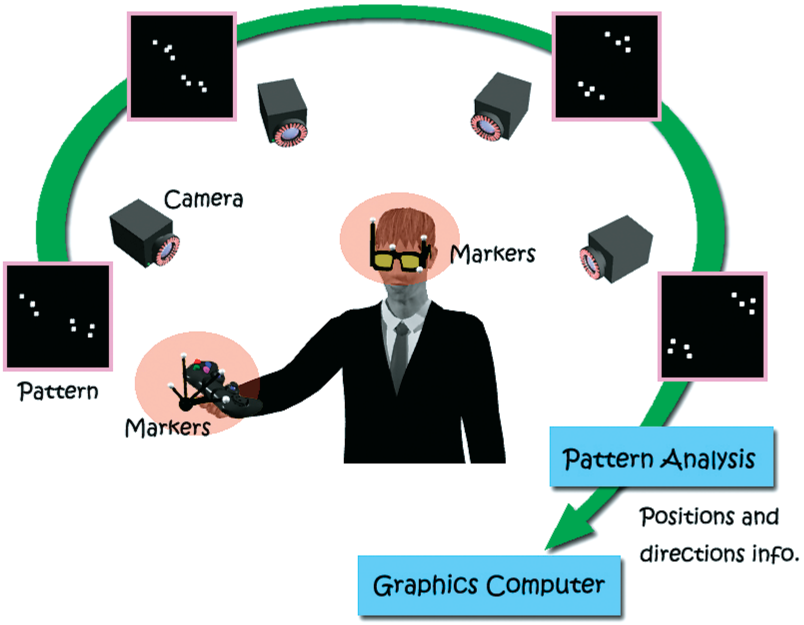

Fig. 6 User's positions and directions are tracked by the high-accuracy optical tracking system with four IR cameras.

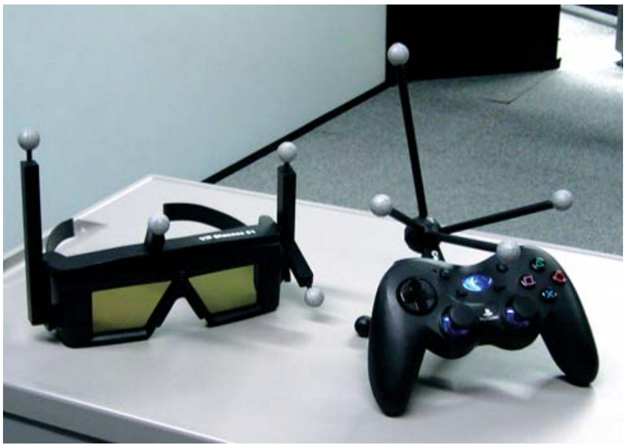

Fig. 7 The liquid crystal shutter glasses and the wireless controller with sets of markers for the optical positioning sensor system.

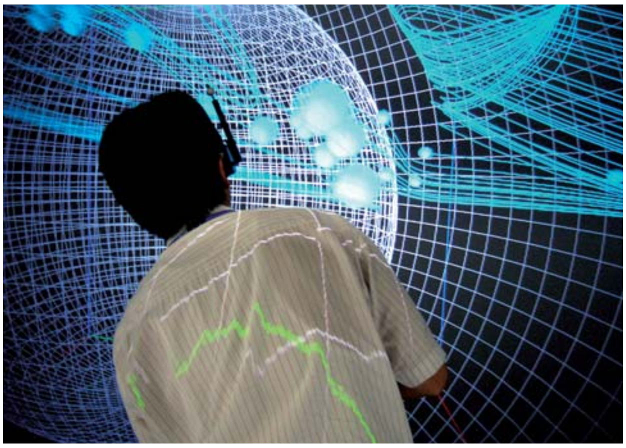

Fig. 8 The interactive data analysis of geodynamo simulation in the CAVE's VR space by VFIVE. "Particle tracer" method is used.

equipped with a wireless handy controller with some buttons and a joy stick (see the right of Fig. 7). It is used to interact with 3D objects in the virtual space. The position and direction of the controller is also detected by the optical sensor system at every moment.

We have been developing an interactive visualization program called "VFIVE" for the CAVE-type VR system since late 1990s [7]. VFIVE is designed to make the best use of the immersive, stereoscopic, and interactive VR environment.

VFIVE is written in C++ with OpenGL and CAVELib. It has a graphical user interface that helps the viewer to intuitively select visualization methods and to control parameters by a virtual laser beam from the virtual menu imaged in front of the viewer.

VFIVE has various visualization methods. In Figs. 8 and

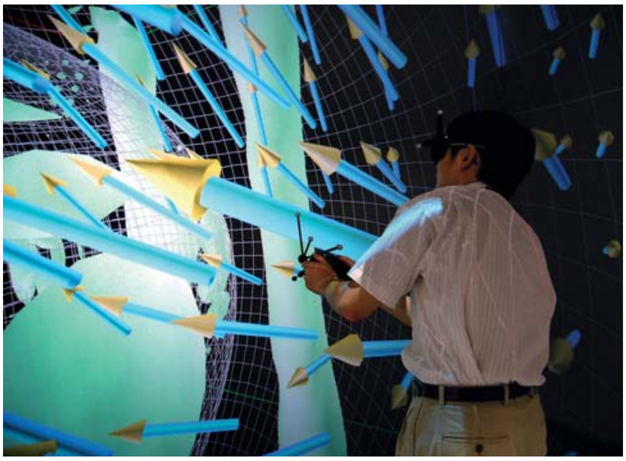

Fig. 9 The interactive data analysis of geodynamo simulation in the CAVE's VR space by VFIVE. "Local arrows" and "isosurface" methods are used.

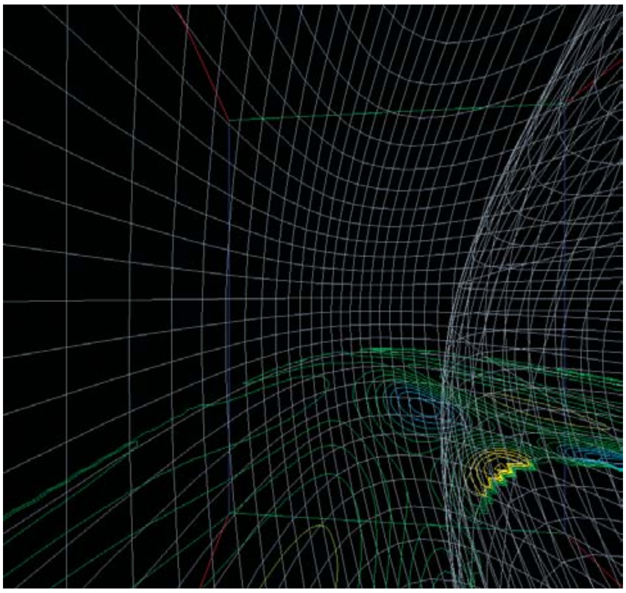

Fig. 10 "Contour Lines" method by VTK, integrated into VFIVE, applied to the vorticity field of the geodynamo simulation.

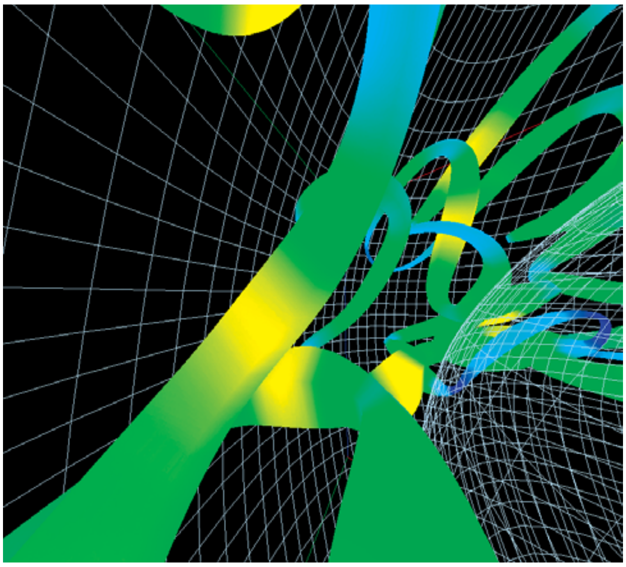

Fig. 11 "Tubes" method by VTK, integrated into VFIVE, applied to the velocity field of the geodynamo simulation.

9, the user analyzes the data of a geodynamo simulation. In Fig. 8, "particle tracer" method is used to analyze the flow velocity field. In Fig. 9, user operates a controller and probes vector fields with "local arrows" method of VFIVE. The vorticity of the fields is also visualized by "isosurface" method, as shown in lightgreen objects in the Fig. 9.

Although VFIVE offers several basic visualization methods, it also allows additions of other visualization routines by linking with external libraries. For this purpose, we integrated Visualization Toolkit (VTK) into the original VFIVE. VTK is a C++ class library of many visualization methods [8]. Figs. 10 and 11 show examples of the VTK visualization methods integrated into VFIVE.

4. ADVANCED VISUALIZATION METHODS FOR VECTOR FIELDS

One of the most challenging task in the scientific visualization is the expression for vector fields. When the vector field contains intrinsic complexity, it is difficult to comprehend the characteristic structure by a simple visualization method such as the arrow or the streamlines. To understand the global nature of the vector field, the Line Integral Convolution (LIC) method has been attracting considerable attention recently. LIC is a texture-based visualization technique for 2D vector data, proposed by Cabral and Leedom [9]. This technique consists of the following three steps. First, we prepare an input image or a texture which can be a random white noise pattern. Secondly, stream lines started from every pixel are traced for the vector field to be analyzed. Finally, each pixel value of the output image is calculated by the following integral along the stream line L, starting from the pixel position $\vec{x}_0$: $I(\vec{x}_0) = \int_L h(l - l_0) F(\vec{x})\, dl$, where F is the input texture, h is a filter function, and l is the length of the streamline.

We extend the LIC method, which was originally used in 2D, to 3D vector fields by making use of the volume rendering technique. A key point of the 3D LIC method is to specify Region Of Interest (ROI) and highlight it by defining a proper transparency. Fig. 12 shows an example of the 3D LIC method showing a simulated typhoon. In Fig. 12, ROI is determined by the wind speed.

Although the 3D LIC method described above is an effective representation method for 3D vector fields, it is a very time consuming process. For the purpose of improving the processing speed, many extended methods are proposed. For example, R. Wegenkittl et al. [10] employed the streamline calculations in order to approximate the convolution result of LIC. We have developed Comet-Like Tracer (CLT) representation. This method also visualizes 3D vector fields through the volume visualization, however we employ a Semi-Lagrangian advection scheme. The starting points for input texture are randomly decided. Then, the pixel value of the texture, which depends on the velocity, are set at each starting point, and advected iteratively to upward by Semi-Lagrangian scheme. This method is faster than LIC. The direction and length of the tail of CLT display orientation and magnitude of the vector field (Fig. 13).

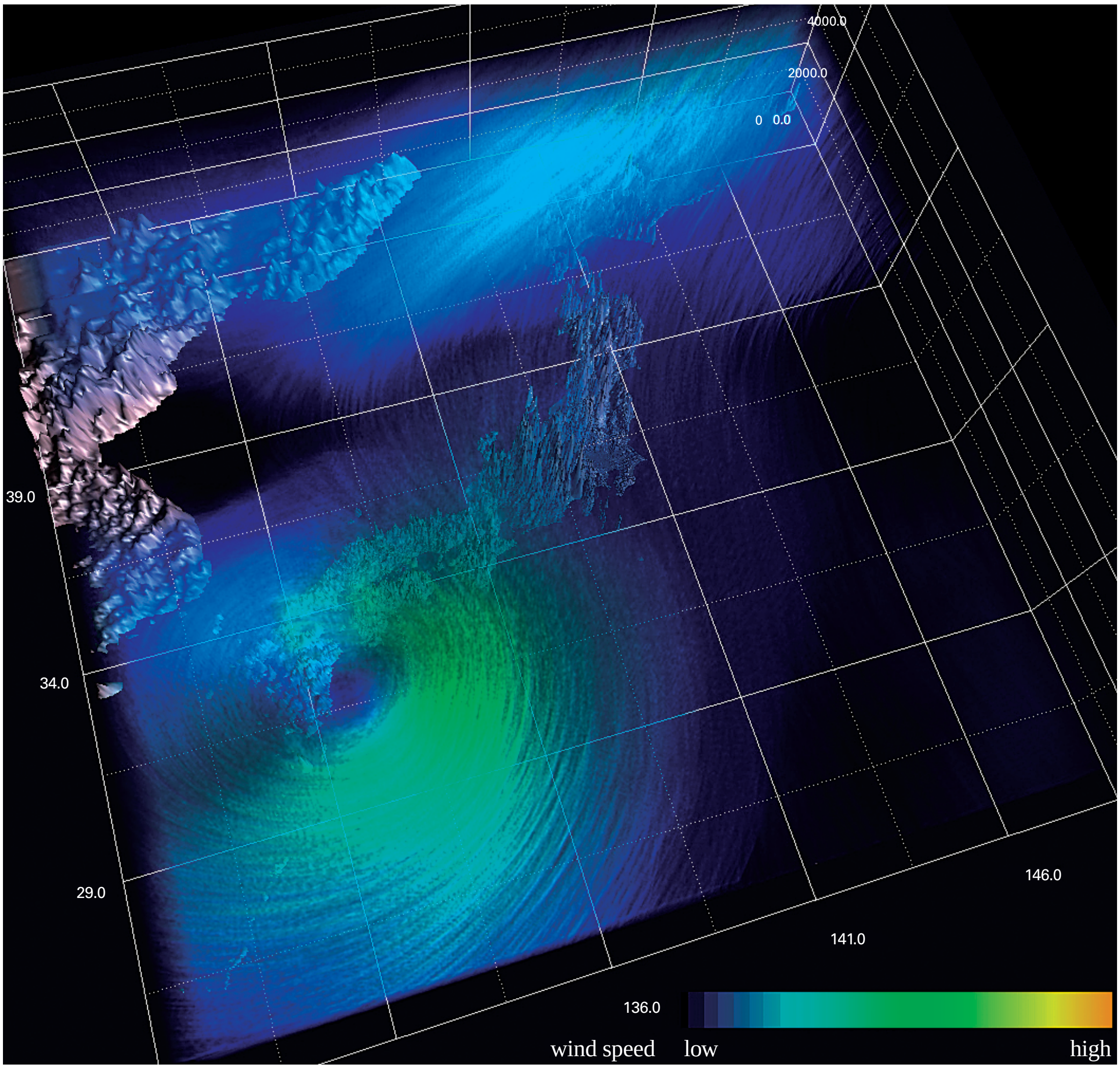

Fig. 12 The wind velocity of a simulated typhoon visualized by the 3D Line Integral Convolution (LIC) method.

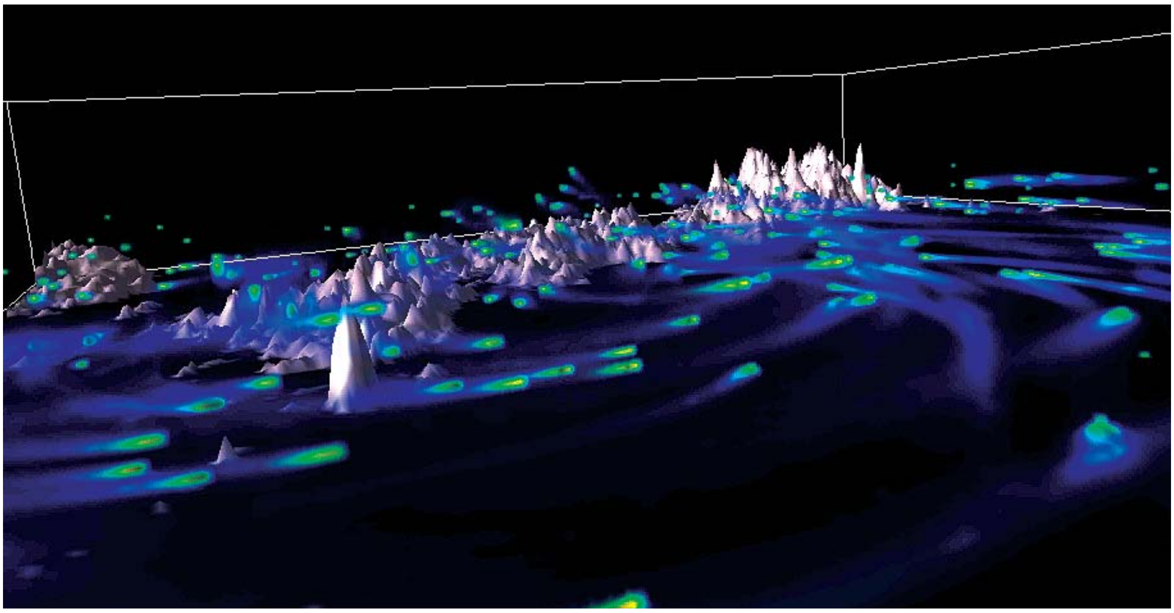

Fig. 13 The wind velocity of a simulated typhoon visualized by the Comet-Like Tracer (CLT) method.

5. VISUALIZATION EXAMPLES (MOVIES)

In this section we show some snapshots taken from visualization movies produced by methods and tools described so far in this paper.

5.1 Phytoplankton concentration in the northwestern North Pacific (Fig. 14)

Snapshot of simulated phytoplankton concentration (pmol L-1) in the northwestern North Pacific. The marine ecosystem simulation [11] is performed using a simplified ecosystem model embedded in the OFES [12]. Distributions of phytoplankton concentration are represented by the volume rendering of MovieMaker.

5.2 Hawaiian Lee Countercurrent (Fig. 15)

The Hawaiian Lee Countercurrent is a far-reaching narrow eastward current to the west of the Hawaiian Islands. Fig. 15 shows eastward current speed simulated by OFES and vector field of surface wind stress (observed by the QuikSCAT satellite) that is the driving force of the ocean current, visualized by the volume rendering and 2D LIC methods, respectively. The figure captures the 3D pathway of the current along the band-like distribution of wind stress convergence [13].

5.3 Wind condition in the city (Figs. 16 and 17)

Snapshots of animations of a wind condition simulation near Ohtemachi, Tokyo. In Fig. 16, distribution of the wind speed is represented by the volume rendering in MovieMaker. In Fig. 17, directions and intensities of the wind are indicated by the arrows. The height data of the land and buildings are generated from the aerial laser survey data by Geographical Survey Institute, Japan, after some data processing of filtering and interpolation.

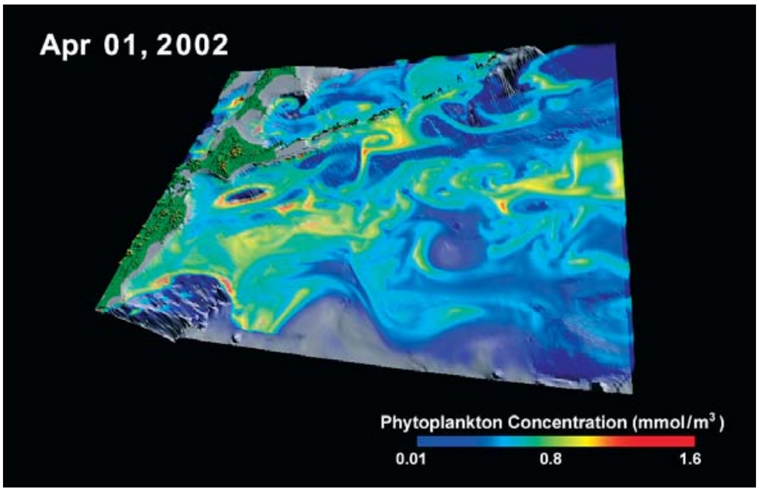

Fig. 14 3D view of simulated phytoplankton concentration in the Pacific Ocean close to Japan.

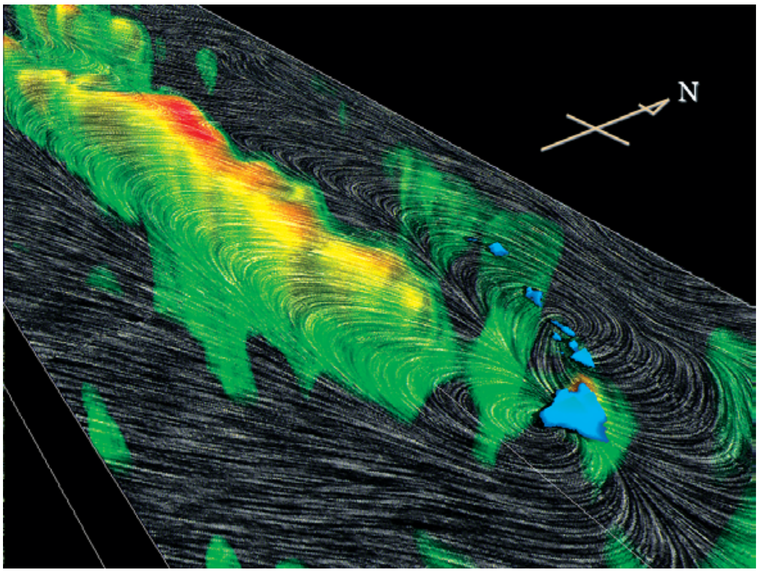

Fig. 15 3D view of simulated eastward velocity and wind stress field west of the Hawaiian island.

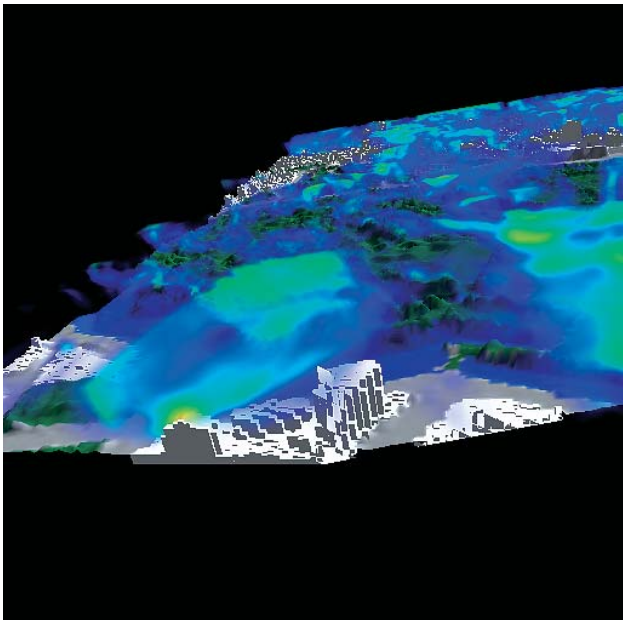

Fig. 16 The wind speed distribution of wind condition simulation near Ohtematch, Tokyo, visualized by the volume rendering in MovieMaker.

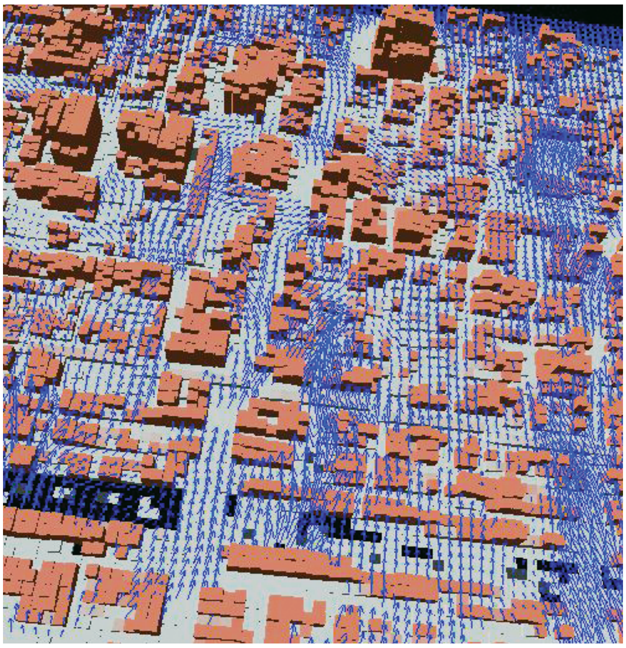

Fig. 17 The wind velocity of wind condition simulation near Ohtematch, Tokyo.

5.4 Season's 10th typhoon in Japan (Fig. 18)

A snapshot of an animation of season's 10th typhoon near Japan in 2003. The simulation data is provided from a non-hydrostatic atmospheric model of Multiscale Simulation Research Group, Earth Simulator Center. The cloud of typhoon is visualized by the spherical volume rendering of MovieMaker.

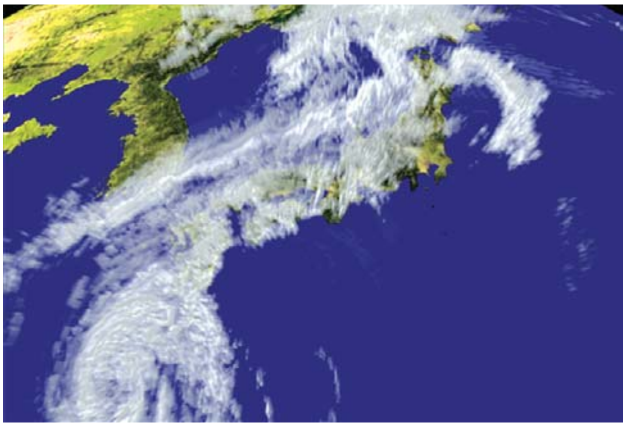

Fig. 18 Cloud water distribution of the simulation result of season's 10th typhoon near Japan in 2003.

6. SUMMARY

To visualize large scale simulation data, we have developed two visualization software programs, MovieMaker and VFIVE. MovieMaker is designed as a master/slave parallel rendering program for the shared-memory architecture. The volume rendering, isosurface and streamlines are available in MovieMaker. The data-read time is hidden in the rendering time by applying the task overlapping method. It turns out that the MovieMaker can complete the process of making one movie by 12 hours for the data of 1 TB.

VFIVE is a virtual reality (VR) visualization program for the CAVE-type system. It enables us to interactively analyze highly-complicated structures in the virtual space. VFIVE has various interactive visualization methods such as "particle tracer", "local arrows", "isosurface", etc. Recently, we successfully incorporated the Visualization Toolkit (VTK) to VFIVE.

We also developed visualization methods for vector fields such as 3D LIC and Comet-Like Tracer. By combining these tools and methods, we are making various visualization movies for large-scale simulation data produced by ES, so that it can greatly contribute to extract hidden treasures in massive data produced by a large-scale simulation.

Acknowledgment

We thank Drs. Y. Suzuki (Fig. 3 and 4), Y. Sasai (Fig. 14), H. Sasaki (Fig. 15), K. Takahashi (Figs. 12, 13, 16, 17 and 18), for providing the simulation data and for fruitful discussions. We thank Dr. M. Kameyama for giving valuable comments on our work. We also thank Prof. T. Sato, the director-general of ESC, for encouragement of our work.

(This article is reviewed by Dr. Tetsuya Sato.)

References

[1] H. Uehara, S. Kawahara, N. Ohno, M. Furuichi, F. Araki, and A. Kageyama, MovieMaker: High Performance Movie-Making System for High-Resolution Data, *Joint Meeting for Earth and Planetary Science CDROM-proceedings*, J031–005, 2005.

[2] M. Levoy, Display of Surfaces from Volume Data, *IEEE Computer Graphics and Application*, Vol. **8**, No. 5, 1988.

[3] W. E. Lorensen and H. E. Cline, Marching Cubes: A High Resolution 3D Surface Construction Algorithm, *Computer Graphics* (Proceedings of SIGGRAPH '87), Vol.**21**, No.4, pp.163–169, 1987.

[4] Y. J. Suzuki, T. Koyaguchi, M. Ogawa, and I. Hachisu, A numerical study of turbulent mixing in eruption clouds using a three-dimensional fluid dynamics model, *J. Geophys. Res.*, Vol.**110**, B08201, 2005.

[5] H. Uehara and Y. Suzuki, Visualization for Large Scale Eruption Clouds Simulations, *Proc. of Japan Geoscience Union Meeting 2006*, J240–003, 2006 (in Japanese).

[6] C. Cruz-Neira, D. J. Sandin, and T. A. DeFanti, Surround-screen projection-based virtual reality: The design and implementation of the CAVE, *Proc. of the Computer Graphics International Conference*, pp.135–142, 1993.

[7] A. Kageyama, Y. Tamura, and T. Sato, Visualization of Vector Field by Virtual Reality, *Progress Theor. Phys. Suppl.*, Vol.**138**, pp.665–673, 2000.

[8] W. Schroeder, K. Martin, and B. Lorensen, *The Visualization Toolkit*, Third Edition, Kitware Inc, Now York, 2002.

[9] B. Carbral and C. Leedom, Imaging Vector Fields Using Line Integral Convolution, *Computer Graphics Proceedings '93*, ACM SIGGRAPH, pp.263–270, 1993.

[10] R. Wegenkittl, E.Gröller, and W. Purgathofer, Animating Flow Field: Rendering of Oriented Line Integral Convolution, *Proc. of Computer Animation '97*, pp.15–21, 1997.

[11] Y. Sasai, A. Ishida, H. Sasaki, S. Kawahara, H. Uehara, and Y. Yamanaka, A global eddy-resolving coupled physical-biological model : Physical influences on a marine ecosystem in the North Pacific, *Proc. of Open International Conference on Modeling and Simulation - OICMS 2005*, pp.129–138, 2005.

[12] Y. Masumoto, H. Sasaki, T. Kagimoto, N. Komori, A. Ishida, Y. Sasai, T. Miyama, T. Motoi, H. Mitsudera, K. Takahashi, H. Sakuma, and T. Yamagata, A fifty-year eddy-resolving simulation of the world ocean: Preliminary outcomes of OFES (OGCM for the Earth Simulator), *J. Earth Simulator*, Vol.**1**, pp.35–56, 2004.

[13] H. Sasaki and M. Nonaka, Far-reaching Hawaiian Lee Countercurrent driven by wind-stress curl induced by warm SST band along the current, *Geophys. Res. Lett.*, Vol.**33** (12), L13602, doi:10.1029/2006GL026540, 2006.