In-Situ Visualization with Point Clouds

Nobuaki Ohno[1*] and Hiroaki Ohtani[2,3]

[1]Graduate School of Simulation Studies, University of Hyogo, Kobe, Japan

[2]Department of Helical Plasma Research, National Institute for Fusion Science, Toki, Japan

[3]Department of Fusion Science, SOKENDAI (The Graduate University for Advanced Studies), Toki, Japan

ABSTRACT

We have been developing an in-situ visualization tool for PIC (Particle In Cell) simulations. We have incorporated a new function that generates point clouds for interactive visualization of isosurfaces, slices, arrow glyphs, and streamlines. The point clouds can be used to reconstruct the visualization objects such as isosurfaces. They can be observed from anywhere interactively.

Keywords: In-situ visualization, point cloud.

Index Terms: **•Human-centered computing**~**Scientific visualization** •Computing methodologies~Point-based models •Computing methodologies~Massively parallel and high-performance simulations

1 INTRODUCTION

Today's highly increased computational capability of supercomputers makes it possible to carry out large-scale computer simulations. These simulations can give us ample important clues to solve problems. However, the immense size of simulation results can cause difficulties in many ways when researchers visualize and analyze them. For example, storing simulation results requires large space on storage systems. Transferring them from supercomputer sites to researchers' offices requires a large amount of time. Visualizing unprocessed huge data on a PC is almost impossible.

Hence, in-situ visualization that is free from those problems is drawing much attention recently. In the technique of in-situ visualization, visualization is performed with a computer simulation on the same computer simultaneously, and not raw data but image files are output. A simple way is uniting a visualization code with a simulation one. The visualization code can read the data stored in the memory directly. Moreover, because the visualization code runs on the same computer, enough memory size and CPU powers to visualize the data are available. There are free tools such as ParaView and VisIt. Both tools use the Visualization Toolkit (VTK). Some recent research on in-situ visualization has used those tools. In-situ visualization seems to be a promising technique, however, there are some disadvantages to be improved. Generally, computer simulations are carried out as batch jobs so that in-situ visualization is not interactive by nature. "Interactive in-situ visualization" is an active field of research at present. A method using hundreds of movies for adding interactivity is shown [1]. An image based approach requiring much imagery is also proposed [2]. There is another approach for an interactive in-situ visualization. In Ref [3] a method to use information such as depth saved by in-situ is introduced.

* email address: ohno@sim.u-hyogo.ac.jp

We believe that the computational resource, CPU time, is precious for simulation researchers. The CPU time consumed by visualization code is not negligible. Thus, we take a similar approach as Ref [3] which does not draw a great number of images. We have developed a function that generates point clouds instead of images, and have incorporated it into our in-situ visualization tool VISMO. VISMO and the new function are described in the following sections.

2 VISMO

We have been developing an in-situ visualization tool called VISMO (<u>VIS</u>ualization <u>MO</u>dule) [4] which is primarily for PIC (Particle In Cell) simulations [5]. This tool is provided as a Fortran module. VISMO employs the MPI/OpenMP hybrid programming model. The image composition is done by the binary swap method [6] when the number of MPI processes is a power of two. VISMO's visualization methods are:

- sphere glyph for charged particles,
- isosurface, color slice, and volume rendering for scalar data, and
- arrow glyph and tubed streamline for vector data.

We coupled VISMO with PASMO [7], a parallelized PIC simulation code, and conducted a test run of VISMO on a supercomputer (Hitachi SR16000 Model M1, POWER7). The numbers of CPU cores used in this test run reached 2,048.

3 FUNCTION FOR GENERATING POINT CLOUDS

In order to give some interactivity to VISMO, we have devised a new function to save opaque visualization objects, such as isosurfaces and arrow glyphs, as point clouds. A point cloud is a data set of points with some attributions. Point clouds are used in various fields such as architecture, civil engineering, and digital archives of historic objects. Point clouds are rendered directly, or processed into polygons before being rendered. There is even a convenient library, Point Cloud Library [8], for processing point clouds.

3.1 Visualization Methods as Point Cloud Generators

We devised the visualization methods of VISMO and made them into point cloud generators. The size of point clouds is controlled by the image size.

VISMO uses ray casting methods for all the scalar visualization methods. The ray casting method can be used as a point cloud generator for isosurfaces. A similar method is described in Ref [3]. The ray is extended to collect points from not only the first surface that the ray runs against but also from the surface hidden behind it. The ray never stops unless it exits from the data. Furthermore, our method emits special rays to some pixels after the first raycast. The special ray can detect the isosurface closest to it and collect points on the surface. The pixels are chosen with the condition that the first ray passing it ran against the isosurface or passed near the surface. For color slice, points are generated on the plane. The density of points is adjusted considering the pixel size and the distance between the viewing point and the subdomain. For arrow glyphs and tubed streamlines, collecting points from their surfaces is not efficient. 3D reconstruction of an arrow glyph requires only position, direction, and the ID number on it. 3D reconstruction of streamlines requires position, the magnitude of the vector, and the ID number. For arrows, all the points are collected. For streamlines, not all the calculated points by Runge-Kutta integration are collected. Some points are skipped depending on the size of the pixel and the distance between the viewing points and the subdomain, such as color slice.

3.2 Development of a Special Viewer

We developed a special viewer that reads the point clouds output by VISMO and displays visualized objects based on the Kyoto Visualization System library [9]. The point clouds of isosurfaces and slices are displayed as points directly. The point clouds of arrow glyphs and streamlines are not displayed directly, but the polygons of 3D objects, namely arrow glyphs and tubed streamlines, are created. This viewer is supposed to be used on a PC or GWS with graphics hardware. Users can change the viewing point interactively.

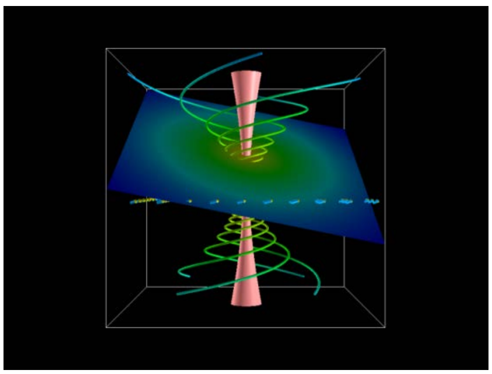

Figure 1: Visualization image of test data.

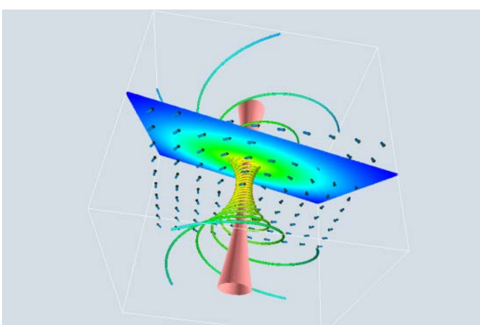

Figure 2: Point clouds are displayed by the special viewer. The point size is four .

3.3 Experiment

We developed a program that creates a scalar data and a vector data with mathematical functions and coupled it with VISMO. We carried out the coupled program on a supercomputer (Fujitsu PRIMEHPC FX100). The number of MPI processes used in this texperiment is 32. That of OpenMP threads is 8. The grid size of the data is 1024^3. The file sizes of the generated point clouds are about 2.8KB for the arrow glyphs, 865KB for the isosurface, 5.5MB for the slice, and 166KB for the streamlines with the same visualization parameters which outputs figure 1. Figure 2 is a snapshot from the interactive visualization of the point clouds with a different viewing point. The objects hidden behind the other objects are clearly seen. The visualization was carried out based on only one viewing point, but the objects can be viewed from all viewing points. It took about 10.4 second to output Figure 1 and 7.7 second to output the point clouds for figure 2 respectively. Though the size of point cloud for figure 2 is much larger than that of figure 1 (80KB for png format), the required computational resources were almost the same. This is clearly a big advantage for this method.

4 Conclusions

We devised our in-situ visualization tool VISMO to save point clouds of isosurfaces, slices, arrow glyphs, and streamlines. For isosurface visualization, we enabled the rays to travel and collect point clouds in the data even after running against the first surfaces. Furthermore, special rays to collect surface points for improving the quality of point clouds are emitted from some pixels after the first raycast. For vector data visualization, we don't save any point cloud of the surfaces of arrows and streamlines, but their position points and directions for 3D reconstruction. By this development, users can view the visualization objects such as isosurfaces and arrow glyphs from any viewing point.

Acknowledgements

This work was supported by The Okawa Foundation. This work was also supported and under the auspices of the National Institute for Fusion Science (NIFS) Collaborative Research Program (NIFS15KNTT030) and the NIFS/NINS under the project Formation of International Scientific Base and Network. This work is partially supported by "Joint Usage/Research Center for Interdisciplinary Large-scale Information Infrastructures" in Japan and JSPS KAKENHI Grant Number 23340182.

References

[1] A. Kageyama and T. Yamada. An Approach to Exascale Visualization: Interactive Viewing of In-Situ Visualization. *Computer Physics Communications*, volume 185, pages 79-85, 2014.

[2] J. Ahrens, S. Jourdain, P. O'Leary, J. Patchett, D. H. Rogers and M. Petersen. An Image-based Approach to Extreme Scale in Situ Visualization and Analysis. In *Proceedings of the 2014 International Conference for High Performance Computing, Networking, Storage and Analysis (SC)*. Piscataway, NJ, USA, pages 424–434, 2014.

[3] Y. Ye, R. Miller, and K.-L. Ma. In Situ Pathtube Visualization with Explorable Images. In *Proceedings of EGPGV 2013*, pages 9-16 2013.

[4] N. Ohno and H.Ohtani. Development of In-Situ Visualization Tool for PIC Simulation. *Plasma and Fusion Research*, volume 9 3401071, 2014.

[5] C. K. Birdsall and A. B. Langdon. *Plasma Physics Via Computer Simulation*, McGraw-Hill, New York, 1985.

[6] K. -L. Ma, J. S. Painter, C. D. Hansen and M. F. Krogh. Parallel Volume Rendering Using Binary-Swap Compositing. *IEEE Computer Graphics and Applications*, volume 14, pages 59–67, 1994.

[7] H. Ohtani and R. Horiuchi. Open Boundary Condition for Particle Simulation in Magnetic Reconnection Research. *Plasma and Fusion Research*, volume 4, 024, 2009.

[8] R. B. Rusu and S. Cousins. 3D is here: Point Cloud Library (PCL). In *IEEE International Conference on Robotics and Automation (ICRA)*, Shanghai, China, 2011.

[9] N. Sakamoto. Kyoto Visualization System. available at https://github.com/naohisas/KVS (accessed 14 September 2015).