3rd Asia-Pacific Conference on Plasma Physics, 4-8,11.2019, Hefei, China

4-D Street View: Movie-based visualization method for HPC

Akira Kageyama
Department of Computational Science, Kobe University
kage@port.kobe-u.ac.jp:

In these years, the in-situ visualization has been featured as a promising approach to visualizations in High Performance Computing (HPC). Major visualization applications today, i.e., post-hoc, visualization tools, are now armed with in-situ libraries or tools, such as Catalyst and libsim.

Unless we adopt special methods for the Computer Graphics (CG) rendering, the in-situ visualization deprives users of interactive control of visualization parameters, including the applied visualization algorithms, their parameters, and the camera settings including the view point and view angle. In the in-situ visualizations, we have to re-submit the same simulation job to a supercomputer system for different visualization or camera settings. It is not reasonable to consume computer resources when we, for example, just want to observe a phenomenon from a different view position.

We proposed an interactive in-situ visualization method that can be applied with the standard CG rendering methods [1]. The key idea is to place multiple cameras and to perform in-situ visualizations with every camera at once. In the paper [1], the cameras are supposed to focus on specified locations in the simulation region, such as the center of the region. We have recently relaxed this constraint. The cameras are now visualize the whole view solid angle of four pi steradian (See Fig. 1). The images are stored as omni-directional movie files. The output of this in-situ visualization method is a set of multiple (same number as the cameras) omni-directional movie files, rather than numerical data files. Each movie file is tied with the camera's location.

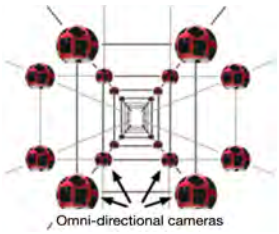

Fig. 1: Multiple omni-directional visualization cameras scattered in the simulation region.

In our visualization method, we interactively analyze the movie data base. A specially designed application program extracts a sequence of still images from the movie files and show the sequence as an animation on a window of user's PC monitor screen (See Fig. 2).

The application program enables its user to interactively change the camera position by means of the PC's mouse and the keyboard and it extracts a sequence of images on the screen, as shown in Fig. 2. We can, interactively, go back and forth not only in space but also in time. This is the 4-D Street View method that we propose for the HPC visualization in the future.

To realize this method, we need two kinds of application programs; one is an omni-directional, multi-viewpoints, and in-situ visualization library for supercomputer simulation codes; another is a special movie player that extracts a sequence of images from the movie data base produced with the library.

We have developed "Membrane" as the former library [2-4]. Membrane is a parallel program in the MPMD (Multi Program, Multiple Data) style. It separates the simulation and visualization parts.

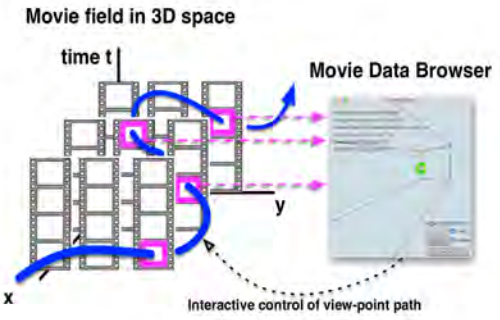

Fig. 2: Concept of the 4-D Street View method.

We have also made Movie Data Browser as the special-purpose movie player, which is developed in the framework of KVS. KVS is an open-source API for CG programs developed by N. Sakamoto of Kobe University.

We will report the development of these tools for 4-D Street View, as well as its applications to fluid and MHD simulations.

References

[1] A. Kageyama and T. Yamada, An Approach to Exascale Visualization: Interactive Viewing of In-Situ Visualization, Comput. Phys. Comm., vol.185, pp.79-85 (2014)

[2] K. Yamamoto and A. Kageyama, MHD Relaxation with Flow in a Sphere, Procedia Computer Science, vol.80, pp.1374-1381 (2016)

[3] A. Kageyama, et al., Membrane Layer Method to Separate Simulation and Visualization for Large-scale In-situ Visualizations, Proc. SIMULTECH 2018, pp.106-111 (2018)

[4] K. Yamamoto and A. Kageyama, In-Situ Visualization with Membrane Layer for Movie-Based Visualization, Lecture Notes in Computer Science (LNCS) vol.11540, pp.588-594 (2019)