

Fusion Visualization for Fluid Dynamics in Blood Vessel

Hideo Miyachi¹, Tomoyuki Ishida², Naohisa Sakamoto³, and Koji Koyamada³

¹*CYBERNET SYSTEMS CO., LTD., Japan*

TEL: +81-3-5297-3692,

FAX: +81-3-5297-3646

miyachi@cybernet.co.jp

²*Ibaraki University, Japan*

TEL: +81-0294-38-5138

FAX:

ishida@mx.ibaraki.ac.jp

³*Kyoto University, Japan*

TEL: +81-75-753-9364

FAX: +81-75-753-9365

naohisas@viz.media.kyoto-u.ac.jp, koyamada.koji.3w@kyoto-u.ac.jp

<http://www.viz.media.kyoto-u.ac.jp/html/en/index.html>

This work was presented as a part of the 19th International Symposium on Artificial Life and Robotics, Oita, Japan, January 22-24, 2014

Abstract: The speed up of supercomputers has increased the complexity of simulations. To analyze such kind of data, new types of visualization software are needed. As one of approach for meeting this requirement, we are developing the "Fusion Visualization" in a project sponsored by the Japan Science and Technology Agency (JST). It can execute fused visualization of simulation data combining both volume and surface rendering. The overall concept was reported last year at the AROB 18th International Symposium in Korea. In this work, we are reporting the ongoing, research with an application example related to blood flow simulation.

Keywords: *Fusion visualization, Volume rendering, Scientific visualization*

1 INTRODUCTION

We are developing a new type of visualization software in which a technology named "Fusion Visualization" is included. This work has been conducted under a project sponsored by the Japan Science and Technology Agency (JST). The purpose of this project is to develop a visualization technology for large scale data generated from high performance computer

systems such as super computers. The key technology to achieve this is Particle Based Volume Rendering (PBVR). Volume rendering^{[1],[2]} usually requires less memory than those required by surface rendering. In addition to this merit, PBVR can also be easily parallelized easily. The proposed system can seamlessly fuse volume rendering results with legacy visualization results combined with surface rendering.

We have introduced a prototype at AROB 18th in Korea ^[3]. In this paper, we report the ongoing studies and achievements of the project related to last year. We will first introduce the SPT (Sorting free pre-integrated Projected Tetrahedral) technology which is another volume rendering technique. SPT has an advantage compared to SPT since it can generate better images when the objects are closely observed or zoomed. Secondly, we will present an interface to switch between PBVR and SPT. Finally, we will present a visualization example obtained by applying our approach to large scale of CFD (Computational Fluid Dynamics) simulation data with size of around 8TBytes.

2 Two types of volume rendering

2.1 PBVR

PBVR was developed by Koyamada et al. at Kyoto University ^[4]. The method assumes that light emitting particles are filled inside cells based on their density values defined by cell's transparency values. This density value can be considered as a probability in a multi-pass rendering. For example, cells with 50% transparency are expected to have their cells filled with particles 500 times out of 1,000 times image generation. The final image is obtaining by ensemble averaging all these 1,000 images. We reach to the conclusion that

the images can achieve enough quality by averaging a finite number of images.

2.2 SPT

Although the SPT ^[5] is a kind of sorting free technique proposed as a multi-pass volume rendering, it does not require particle generation process in the object space as PBVR. In the rendering process of the SPT, the tetrahedral cells are projected onto the image plane without the need for visibility ordering (Fig.1 (a)), and then their footprints are divided into three or four triangle regions by using the projected tetrahedra technique ^[6] (Fig.1 (b)). Finally, the color and opacity are assigned for each of the fragments within the footprints in the rasterization process by using the pre-integration technique ^[7] (Fig.1 (c)).

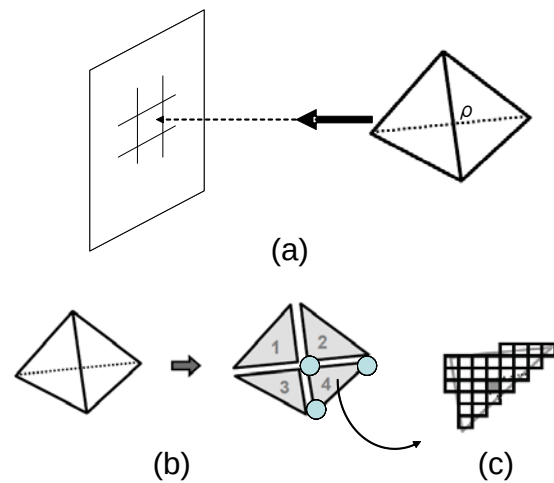


Fig.1 Concept of the SPT rendering

In the case of the SPT, the opacity can be assumed as a probability and the brightness can be calculated as an expected value of the colors. Therefore, in the final process, we can accept or discard the color by using a random

number. If the random number is smaller than the opacity, we accept the color for the fragment. Otherwise, we will discard the fragment.

2.3 Switching between PBVR and SPT

These aforementioned two kinds of volume renderings have the following characteristics.

PBVR:

- a) The number of particles can be controlled. Rather than the number of cells, it is related to the number of pixels in the final image. It means that we can reduce the complexity to any level even though the original data size is over billion cells.
- b) Once reducing the data (after generating particles), we can visualize it interactively on a commodity personal computer.
- c) Zoom operation invokes the particle re-generation. The calculation can take much time, however image quality degradation can occur without the particle re-generation.

SPT:

- d) The entire cells are loaded for the zoom operations it does not invoke any special load. It allows us an interactive visualization.
- e) Large computer resource proportional to the data size is usually required. Therefore the maximum data size which is able to be handled is limited by the available resource, mainly graphics memory size of the computer.

From the above, we have a plan to implement a switching interface between these two methods. On the stage of bird's-eye view of the entire object, the PBVR method is the most recommended. On the stage of the detail analysis, and after detecting an interesting region, in a size which can be loaded entirely in the graphics memory, the PST becomes the recommended method. This switching should be executed automatically by the system. However, in current implementation this switching can only be operated manually.

3 An application example

3.1 System overview

The outward appearance of the 2nd prototype system is shown in Fig.2. It is implemented on AVS/Express^[8] as same as the 1st prototype. AVS/Express is a commercial software package for general purpose visualization which was originally developed by AVS Inc. (Advanced Visual Systems Inc., MA., USA) and is currently maintained by both AVS Inc. and CYBERNET (CYBERNET SYTEMS CO., LTD. Tokyo, Japan). The system has object-oriented architecture and user programs can be embedded easily into the system. Our project is not only a research project, but it is also business oriented. To develop a system with friendly man-machine interface and with a compatibility with existing system are part of the missions of our project.

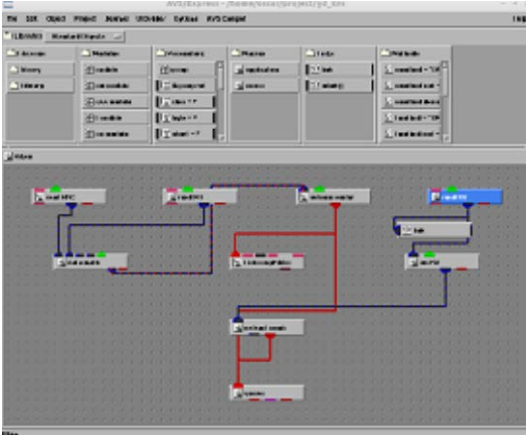


Fig.2 Appearance of the proposed system

3.2 Blood flow simulation

We have applied our prototype to visualize blood flow simulation data^[9]. This simulation uses a full Eulerian method that employs a uniform grid system for all materials, such as vessel wall and blood including red blood cells and platelets which are represented by hyperelastic materials. As initial condition, red blood cells and platelets are arranged randomly. Then pressure gradient causes the flow of the blood. Both ends of the blood vessel will give a circular boundary condition. The number of nodes is 1,270 million in total (3074x642x642). The output data is divided into 4700 region (10x10x47 files) and each file includes 66x66x66 nodes with overlapping nodes between neighbor regions. The overview of the data is shown in Fig.3. The transparency tube represents the wall of blood vessel. The red objects represent the red blood cells. The colored particles represent the platelets. In this figure, the fluid is flowing from the left to right hand side.

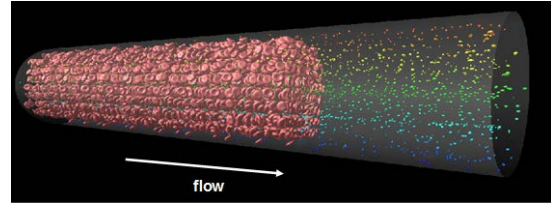


Fig.3 Overview of the blood flow simulation data.

3.3 Visualization by using PBVR and SPT

Some visualization images obtained by using PBVR and SPT are shown in Fig.4 and Fig.5 respectively^[10].

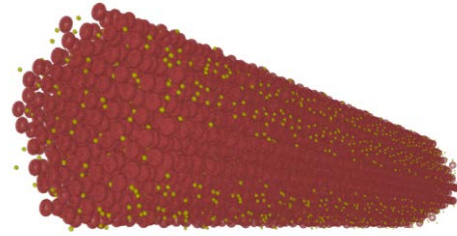


Fig.4 Visualization result by using PBVR. The red objects represent red blood cells and the yellow objects represent platelets.

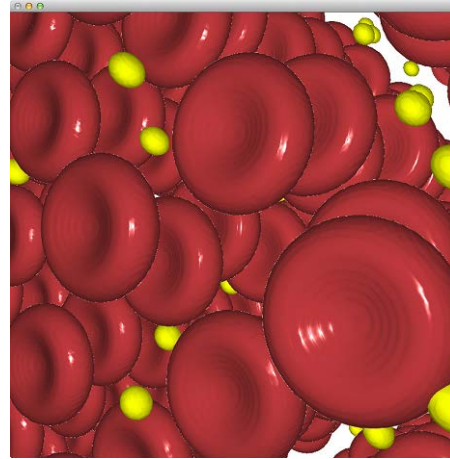


Fig.5 Visualization result by using SPT. The red objects represent red blood cells and the yellow objects represent platelets.

The image shown in Fig.4 is a result of ensemble averaging of 81 images produced by rendering of 86 million particles. After generating the particles from VOF (Volume Of

Fluid) function values for red blood cells and platelets defined over 1.2 billion nodes, PBVR showed a performance of 334 msec/frame on a personal computer with the following specification: Intel Core2 Quad 2.83 GHz CPU, 8GB of main memory and NVIDIA GeForce GTX580 graphics board (with graphics memory of 1.5GB). The size of image is 800x800 pixels.

The image shown in Fig.5 is an ensemble average image of 4,500 images generated by rendering the one-tenth of the entire data, i.e. about 120 million cube cells process the VOF functions data for red blood cells and platelets on each node. The image size is also 800x800 pixels. It achieved a performed of 40 msec/frame. It is easy to note that the image in Fig.5 is clearer than Fig.4, because SPT shows the full data.

4 Detailed analyses

We are able to understand the overview of simulated blood flow by visualizing the images by using both PBVR and SPT. The velocity of blood flow near the center of blood vessel is higher than when near to the vessel wall. The velocity gradient makes the rotation movement of red blood cells. Volume rendering works with scalar values. Therefore, CFD researchers who need vector value visualization do not use volume rendering so much. Fusion visualization technique that we are proposing gives the users a seamless visualization switching from the overview to the detailed

visualization, and from volume rendering to surface rendering. The final stage is the analysis in detail. After extracting an interesting block from the large data, we can visualize it in the traditional way.

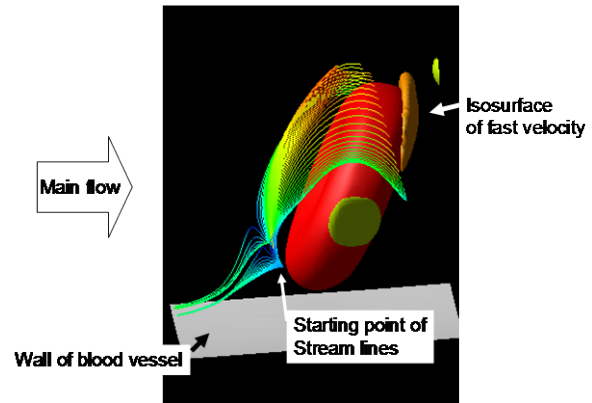


Fig.6 Visualization result in the traditional way.

A visualization result using traditional approach is shown in Fig.6. It includes streamlines and isosurfaces. Velocity, blood vessel wall, red blood cells and platelets are shown by isosurfaces. The latter two by using values of VOF functions. In this case, the main flow (the velocity in X plus direction) is dominant. Hence, all streamlines are drawn by horizontal lines if we use the original velocity values (Fig.7).

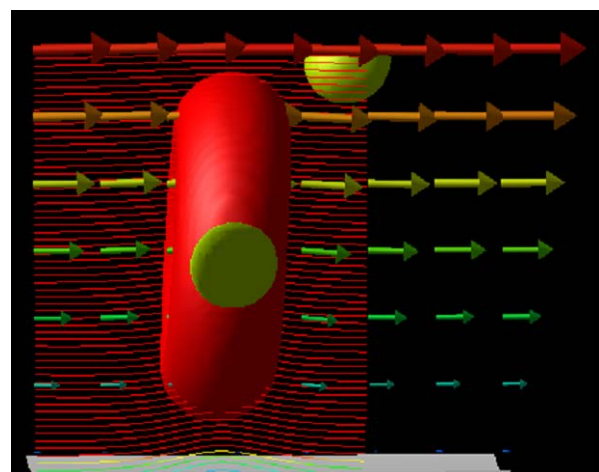


Fig.7 Streamlines drawn with original velocity values

To emphasize V and W velocities, U velocity in Fig.6 is converted by using the following formula (1).

$$U' = (U - U_{\text{main}}) * a \quad (1)$$

Where U': Converted velocity to generate Fig.6

U: Original U velocity

U_{main} : Velocity of the main flow

a: Constant value for adjustment

We cannot specify U_{main} only from the data in this block. We have specified the speed of red blood cell as U_{main}. It assumes that it is equal to the velocity in almost all center of the red blood cell in the picture. It means that the camera of computer graphics is moving together with the red blood cell. The constant is adjusted manually in order to avoid streamlines' penetration to the red blood cells. It is a visualization technique to prevent that the streamlines hidden. Finally, an example of time series visualization is shown in Fig.8.

5 CONCLUSION

We presented the 2nd version prototype of Fusion Visualization system which enables us to visualize seamlessly both volume and surface rendering results. By using this system, we could visualize and analyze a simulated blood flow data with size of 8 TBytes. We have successfully confirmed that our developing system can be utilized for large-scale data visualization.

Acknowledgment: This work was partially supported by the project A-STEP<AS2415031H>: "The research and development of Fusion Visualization technology", funded by the Japan Science and Technology Agency (JST). We would like to thank Dr. Kazuyasu Sugiyama from RIKEN for providing us the simulated blood flow data.

REFERENCES

- [1] Krüger, J., and R. Westermann. (2003), Acceleration Techniques for GPU-Based Volume Rendering., In Proceedings of IEEE Visualization,;287–292.
- [2] Engel, K., M. Kraus, and T. Ertl.(2001), High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading., In Proceedings of the SIGGRAPH/Eurographics Workshop on Graphics Hardware 2001: 9–16.
- [3] Hideo Miyachi, Koji Koyamada and Naohisa Sakamoto(2013), Fusion visualization of surface and volume on AVS/Express, In Proceeding of the 17th AROB.
- [4] Naohisa Sakamoto, Takuma Kawamura, Koji Koyamada(2010), Improvement of particle-based volume rendering for visualizing irregular volume data sets, Computers & Graphics 34(1):34-42
- [5] Naohisa Sakamoto, Koji Koyamada(2012), Stochastic Approach for Integrated Rendering of Volumes And Semitransparent surface, In Proceedings of the 2012 Workshop on Ultrascale Visualization.
- [6] Kazuyasu Sugiyama, Yasuhiro Kawashima, Shigeo Noda, Satoshi Ii, Hiroshi Koyama, Shu Takagi, Yoichiro Matsumoto and Ryutaro Himeno (2013), Massively parallel computing of novel fluid-structure interaction solver on the K Computer, In Proceesing of the Symposium on High Performance Computing and Computational Science, IPSJ-HPCS2013005
- [7] P. Shirley and A. Tuchman(1990), A Polygonal Approximation to Direct Scalar Volume Rendering, In Proceedings of San Diego Workshop on Volume Visualization:63-70

- [8] K. Engel, M. Kraus, and T. Ertl(2001), High-Quality Pre-Integrated Volume Rendering Using Hardware-Accelerated Pixel Shading, In Proceedings of Eurographics/SIGGRAPH Workshop on Graphics Hardware:9-16
- [9] Craig Upson, Thomas Faulhaber , Jr., David Kamins, David H. Laidlaw, David Schlegel, Jefrey Vroom, Robert Gurwitz, Andries van Dam(1989),The Application Visualization System: A Computational Environment for Scientific Visualization, IEEE Computer Graphics and Applications, 9(4):30-42
- [10] Hideo Miyachi, Koji Koyamada, Naohisa Sakamoto, Kazuyasu Sugiyama(2013), Applying Fusion visualization to blood hemorheological information including red blood cells (in Japanese), In Proceedings of the 41st Annual Symposium on Visualization, The Visualization Society of Japan

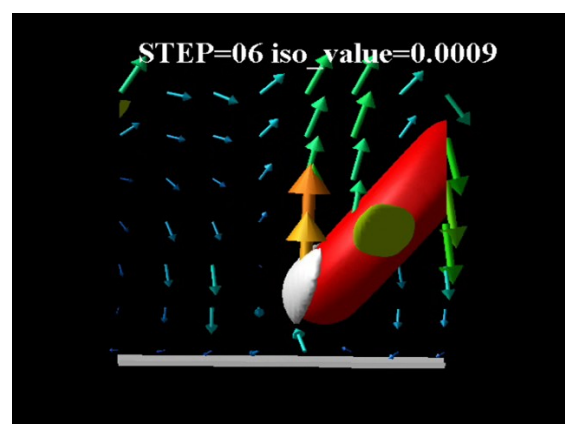
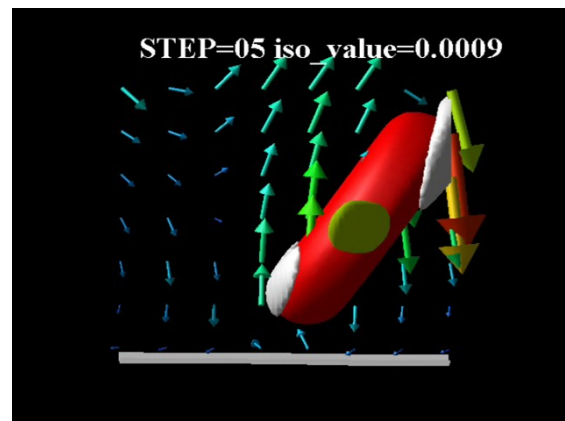
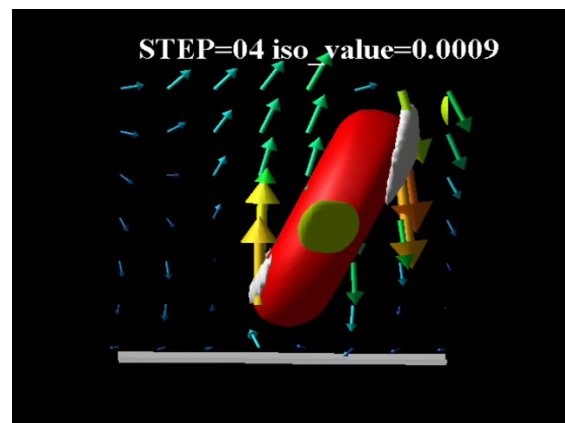
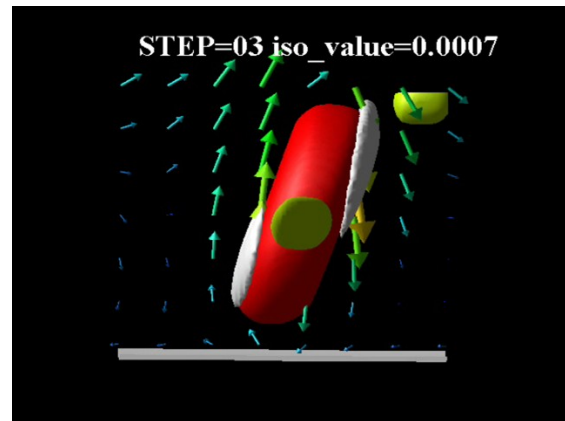
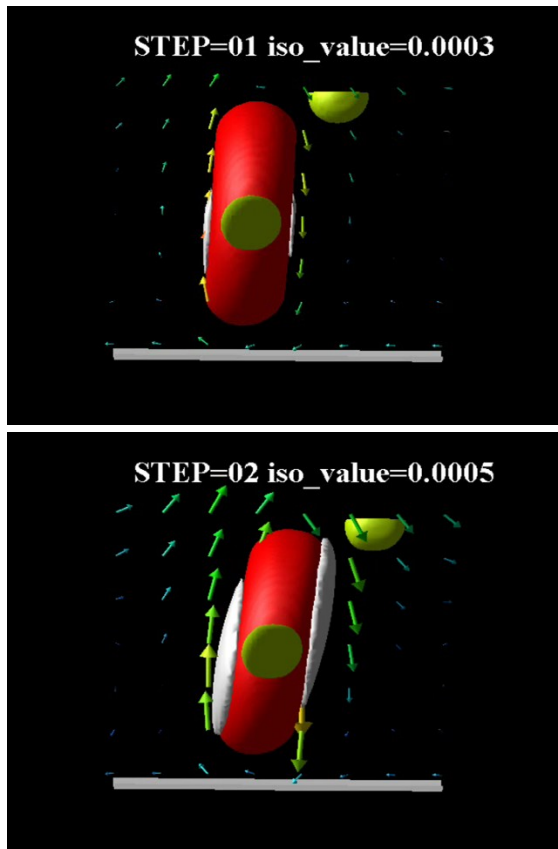


Fig.8 Time series visualization of the simulated blood flow. Arrows represent the velocity orientation. The gray isosurfaces represents the fastest regions in the block