Parallel visualization of gigabyte datasets in GeoFEM

Issei Fujishiro (1) (2), Yuriko Takeshima (1) (2), Li Chen (2)(3),
Hiroko Nakamura (1), and Yasuko Suzuki (1)

(1) Ochanomizu University, Tokyo, Japan (e-mail: fuji@is.ocha.ac.jp; {yuriko, hiroko, yasuko}@imv.is.ocha.ac.jp; phone: +81-3-5978-5700, fax: +81-3-5978-5705). (2) Research Organization for Information Science & Technology, Tokyo, Japan (e-mail: chen@tokyo.rist.or.jp; phone: +81-3-3436-5271, fax: +81-3-3436-5274). (3) State Key Lab. of CAD&CG, Zhejiang University, Hangzhou, 310027, P.R. China.

Abstract

Parallel visualization of large datasets in GeoFEM is described. Our visualization subsystem supports concurrent visualization with computation, and outputs a simplified small graphic primitive set, which can be displayed by a basic viewer on clients. This subsystem provides many kinds of parallel visualization algorithms for the users to visualize their data from scalar, vector to tensor. Scalar field topology analysis, flow field topology and semi-automatic parameter design are employed to improve the quality of visualization results. We also present a simplification algorithm to reduce the number of output graphic primitives, accounting for both shape attribute and color attribute. The experimental results show the effectiveness of our subsystem.

Introduction

The numerical simulation and analysis in engineering usually consist of three main stages: generating computational grids, solving physical equations, and visualizing the resulting data. With the rapid development of computer hardware and software, the computational grid is becoming more and more complicated, and the data amount of computational result is becoming larger and larger. Therefore, it is very important to develop fast and effective visualization techniques and systems for very large data sets.

Unfortunately, most of commercial visualization software works well for relatively small data, but often fails for huge data sets due to the lack of parallel performance and the limitations of memory and storage spaces on a graphics workstation. Obviously, a better way is to do visualization with computation concurrently on the same high-performance parallel computer.

RVSLIB (Doi, 1997[1]) developed by NEC can perform visualization concurrently with calculation. It uses a computational server for visualization as well and a low-cost desktop PC as a local client for displaying visualization results. However, it finishes all mapping and rendering process in visualization on the computational server and finally outputs one or more movie files to clients. Hence a client just plays a role to show images or image sequences. Because of no geometric information in images, re-calculation must be done on the computational server even if the users just want to change viewpoints.

In this paper, we will introduce our parallel visualization subsystem developed for GeoFEM (Okuda, 2000[2]). First the framework of our parallel visualization subsystem is introduced. In a way similar to RVSLIB, we implement the concurrent mechanism for visualization and calculation on the same high-performance parallel computer, but we output geometric primitives to clients, which can avoid much re-calculation for the users to observe the visualization results. Next, some kinds of parallel visualization modules for scalar, vector, and tensor fields we have developed will be described. In order to improve the quality of visualization results, feature analysis techniques are presented. Finally, we will give our simplification algorithm taking into account both shape attribute and color attribute to reduce the number of output graphic primitives. The experimental results based on GeoFEM simulation datasets show the availability and effectiveness of our subsystem.

Parallel framework of visualization subsystem in GeoFEM

We have implemented a parallel visualization subsystem (Fujishiro, 2000[3]) to provide the users with a visual exploration environment for various types of 3D datasets arising from the analysis subsystems in GeoFEM, which is known as a large-scale finite element analysis platform for solid earth simulation (Okuda, 2000[2]). The framework of our parallel visualization subsystem is shown in Figure 1. We adopt the concurrent visualization with calculation on the same high-performance parallel computer, but we output to clients graphic primitives rather than resulting images. On each client, the users can set viewing, illumination, shading parameters, and so on, and display the graphic primitives by the GPPView viewing software, which is also developed by the GeoFEM group (Wada, 2000[4]). On the computational server, the users only specify in the batch files the visualization methods such as cross-sectioning, streamlines, and so on. Since both the visualization process and calculation process are performed on the same computer at the same time, we need not save the computational results on the disk, which can avoid the limitations of storage capacity for large-scale data. Meanwhile, we can make full use of computational server's huge memory to finish visualization. The visualization results are just a small part of the original computational

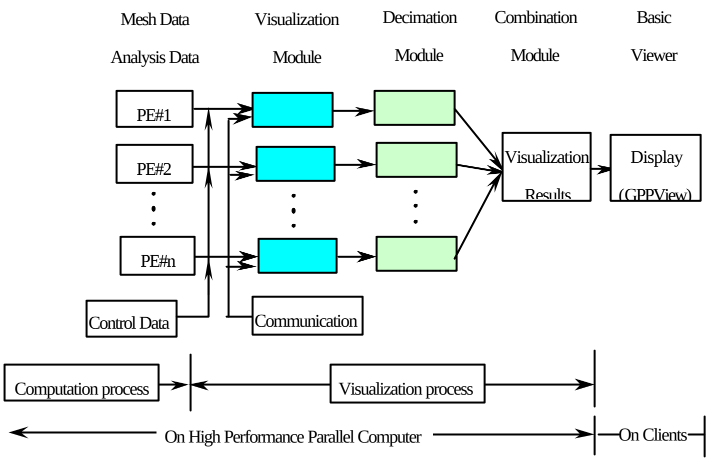


Figure 1: The parallel framework of visualization subsystem in GeoFEM

results. Moreover, according to the time requirement and the limitations of a client's memory and storage space, we use the decimation module to reduce to some degree the output graphic primitives.

Parallel visualization techniques for large-scale datasets in GeoFEM

Parallel visualization techniques for scalar datasets

Compared with other commercial software systems, our cross-sectioning module not only has a highly parallel performance, but also possesses the following advantages: (1) Cross-sections are not limited to planes. The module can generate a surface which is convenient for the users who want to observe the distribution of some physical attribute on a surface such as a global surface. (2) We totally provide ten convenient ways for the users to define a plane cross-section or a surface cross-section, and also can display multi-parallel cross-sections or multi-scalars on different cross-sections in a flexible manner. We have tested the module with a large unstructured 3D viscoelastic FEM analysis dataset for kinematic earthquake cycle in the Southwest Japan. It contains 456,365 grid points. We use the parallel computer SR2201 which is installed in the Computer Center of University of Tokyo, and has 1024PEs, 300Gflops peak speed and 224MB memory for each PE. In the case of using 32 PEs, the module required about 33 seconds to generate the image shown in Figure 2. The analysis module took about 379 seconds. The increase due to visualization is about 8.7%.

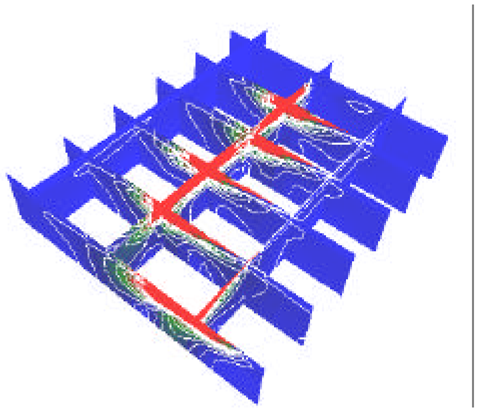
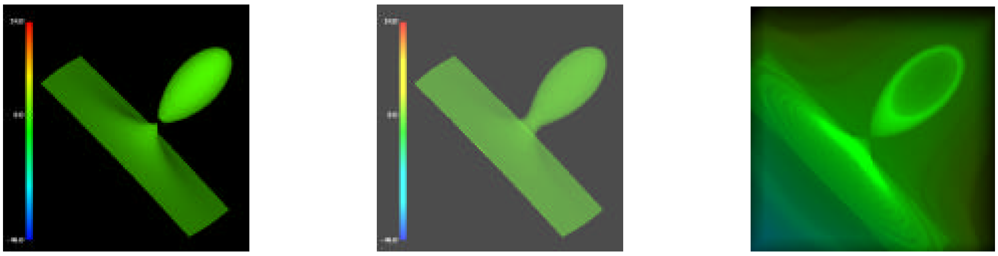


Figure 2: Cross-sectioning for large dataset with 456,365 nodes. Equivalent scalar values of stress are mapped to colors with isolines. (Data courtesy of Mikio Iizuka in GeoFEM)

Our visualization subsystem also provides the *Isosurface Fitting* (*IF*) method for visualizing 3D scalar fields. We take advantage of the *Marching Cube*s (MC) (Lorensen, 1987[5]) known as a highly-parallelizable isosurface construction algorithm, and existing accelerated rendering facilities for getting resulting isosurfaces. However, the IF approach visualizes only a part of the target volumetric object at a time (see Figure 3(a), for example). The other is the *direct volume rendering* (DVR) approach, which projects the entire dataset semi-transparently onto a 2D image without the aid of any intermediate geometric representations. Although DVR can produce intuitive images (see Figure 3(c)), it is intrinsically viewing-dependent, and thus requiring expensive re-computations according to changes in viewing parameter values.

To find a compromise for the accuracy versus efficiency trade-offs between these two approaches, we adopted the concept of *solid fitting* (Fujishiro, 1996[6]), which cannot be found in other commercial visualization software. Solid fitting can be viewed as a generalized IF approach, since it employs *interval volume*, which allows the users to represent as a solid, a 3D subvolume for which the associated scalar values lie within a user-specified closed interval. Such a less-constrained geometric feature extraction allows for more intuitive and informative visualization of volumetric *Region Of Interest* (ROI) than the traditional IF approach, and more computationally efficient than the DVR approach (see Figure 3(b)). Interval volume is also suitable for morphological measurement of ROIs. The quantitative properties of interval volume, such as the surface area, total volume, and field integral, are useful for the understanding of the target datasets. Figure 3 compares the three approaches for visualizing a 3D version of the Folium of Descartes.



(a) Isosurface fitting (b) Solid fitting (c) Direct volume rendering

Figure 3: Three volume visualization approaches for 3D version of Folium of Descartes.

Parallel visualization techniques for vector and tensor datasets

Since *streamline* is the most popular way for visualizing vector datasets, we have included a parallel streamline generation algorithm in our subsystem. In order to reveal much more 3D orientation information, we also implemented illuminated streamline method (Zoeckler, 1996[7]), which makes each streamline have a radius to form a streamtube (see Figure 4, for example).

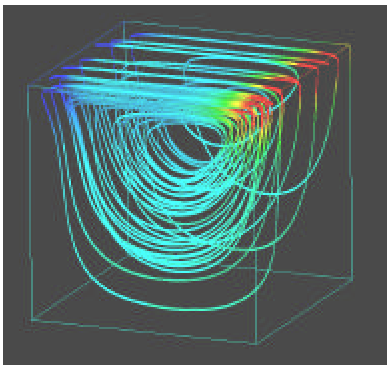


Figure 4: Illuminated streamlines for a lid-driven convection in a cubic cavity with a Reynolds number 1000. (Data courtesy of Hiroaki Matsui in GeoFEM)

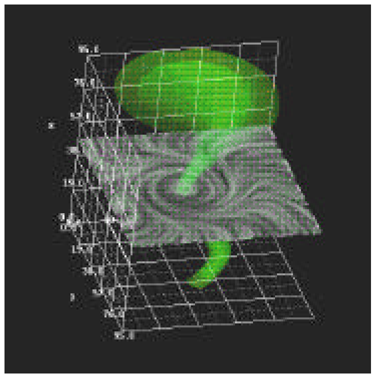


Figure 5: Near-velocity volume & LIC texture-mapped cross-section for flow volume (Data courtesy of R. Crawfis, N. Max, LLNL, UC Davis)

We also implemented the typical texture-based visualization method—LIC method (Carbral, 1993[8]) for vector data fields. LIC is a procedure that smears a given image along paths that are dictated by a vector field. It is local, one-dimensional and independent of any predefined geometry or texture, and is capable of showing the vector directions even in the area where they change quickly. It can avoid the sampling problem existed in the streamline method very well. Figure 5 shows LIC texture-mapped cross-section for a tornado flow dataset.

For tensor datasets, we have implemented a parallel *hyperstreamline* algorithm (Delmarcelle, 1993[9]), which can visualize 3D second-order tensor fields along continuous paths, and can display nine components (three eigenvectors) of a tensor field simultaneously. According to the direction of major display eigenvector, the module first generates a trajectory from a seed point, then attaches ellipses at each point on the trajectory, which forms streamtubes. The direction and magnitude of the long axis and short axis of each ellipse are decided by the direction and magnitude of the other two eigenvectors at this point respectively. The colors on the tube surface can display the magnitude of major display eigenvector at each point on the trajectory. An example for stress tensor data is shown in Figure 6.

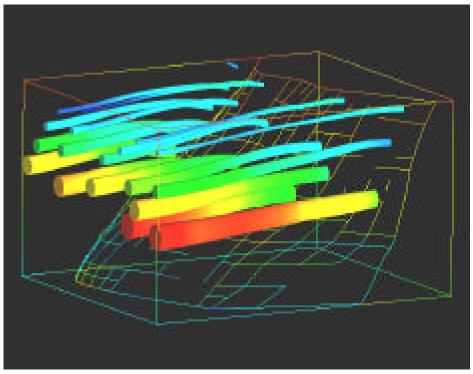


Figure 6: Hyperstreamlines for a fault analysis data around Japanese Islands. (Data courtesy of Mikio Iizuka in GeoFEM)

Feature analysis techniques

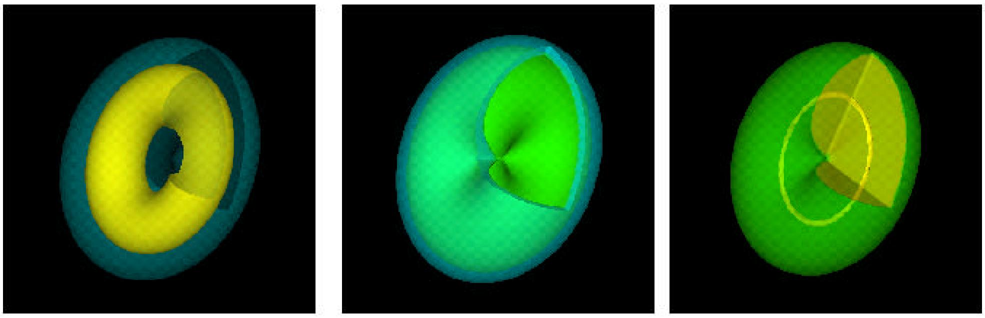
Adaptive extraction of isosurfaces/interval volumes based on Hyper Reeb graph

The concept of the Reeb graph was originally imported into CG fields by Shinagawa, et al. in order to reconstruct a topologically-correct surface from cross-sectional contours extracted from CT images (Shinagawa, 1991[10]). Hyper Reeb graph (HRG) is an extension of the Reeb graph concept to 3D volume fields (Fujishiro, 2000[11]). Theoretically, a volume can be decomposed into an infinite number of isosurfaces with different target values. The topological features of each isosurface can be captured by using the Reeb graph with a common direction for the height function. Therefore, by examining the sequence of isosurfaces in terms of the structure of Reeb graphs, we can find a particular field value, termed critical field value (CFV), for which the topological equivalence of consecutive isosurfaces is not maintained. Based on the HyperReeb graph, we can improve the quality of indirect volume visualization results. We consider the following two options:

- Method 1: Simultaneous display of *m+1* semi-transparent isosurfaces, each of which is extracted with a field value $(f_i + f_{i+1})/2$ at the midpoint of the topologically-equivalent field interval $[f_i, f_{i+1}](i = 1,\dots,m)$. We can determine a plausible value for the opacity of each isosurface so as to reflect the mutual relationships among $l_{i,i+1}(i = 1,\dots, m)$ in order to allow us to understand the relative thickness of topologically-equivalent field intervals.

- Method 2: Decomposition of a given volume V into a sequence of m+1 non-overlapping interval volumes $IV(f_i, f_{i+1})(i = 0,..., m)$;i.e. $V = \bigcup_{i=0}^{m} IV(f_i, f_{i+1})$. Topological equivalence gives the rigid basis for the volume decomposition. In addition, the boundaries of each interval volume convey informative shapes of isosurfaces with CFVs, at exactly the location where the topology of level surfaces changes.

Figure 7 visualizes the metatorus volume with the above two methods in a comprehensible manner. The selected isosurfaces in Figure 7(a) can also be utilized as an effective set of basic frames for the *flipbook* approach to volume rendering. On the other hand, the set of interval volumes in Figure 7(b) is expected to provide a good initial guess for more sophisticated volume segmentation.



(a) (b)

Figure 7: Geometric object extraction from a metatorus volume based on the HRG. (a) Simultaneous display of two isosurfaces with 0.21 (ellipsoid) and 0.64 (torus); (b) Decomposition into two interval volumes $IV[0.15,0.271]$ and $IV[0.271,1.0]$.

Adaptive LIC image generation for vector fields based on significance map

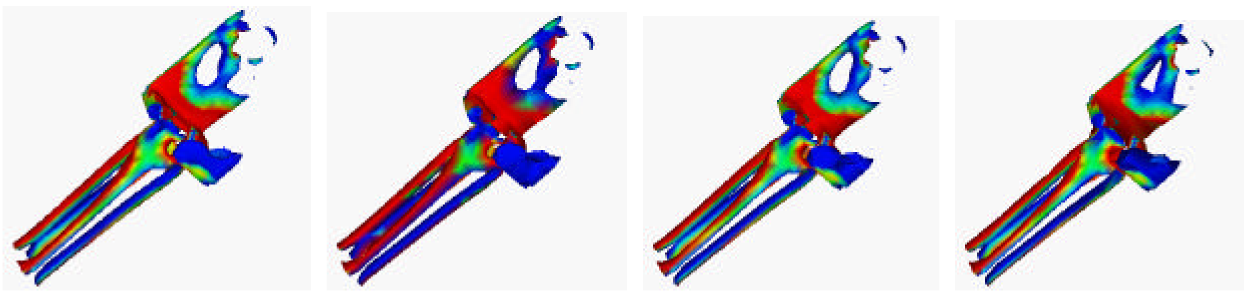
Texture-based methods provide a very promising way to visualizing vector fields. However, most of the existing methods treat every pixel equally, thus leading to fixed detail over the entire texture space without any designated highlights. In fact, it is quite common for a flow field to have extremely non-uniform distribution of detail. It is very time-consuming to generate a finer image so as to ensure a sufficient precision everywhere for significant features.

We present a method to ameliorate the problem (Chen, 2000[12]). We introduce the "significance map", which is derived from both intrinsic properties of a given vector field and user-guided highlights. We adopt the flow topology analysis technique (Chong, 1990[13]) to determine the significance value at each point. In a case that a user is interested in certain areas, his/her specification may be used to adjust the topology-derived value. Based on the significance map, we propose techniques to accelerate LIC texture image generation, to highlight important structures in a vector field, and to generate an LIC texture image with different granularities. Figure 8 shows an example of our method. Figure 8(a) is generated by the original LIC method. Figure 8(b) is generated by adjusting texture opacities to highlight the vortices. And the convolution length at each pixel is shortened greatly in the regions far away vortices, by which it is 5 times faster than the original one. In Figure 8(c), a coarser granularity is selected in lower significant areas, which can involve less cells to generate the texture image. It is about 4 times faster than before.

(a) (b) (c)

Figure 8: Adaptive LIC image generation based on significance map. (a) A cross-section texture image generated by the original LIC method; (b) A texture image generated by our significance-driven LIC method, which not only accelerates the image generation, but also highlights the vortices; (c) A texture image generated with different texture granularities. (Data courtesy of Anlu Ren, Zhejiang University, P.R. China)

Parallel decimation



(a) (b) (c) (d)

Figure 9. Simplifying interval volume of mechanical part. (a) Original (6,706 patches); (b) 50% patches reduced by accounting for geometry only (r = 0.0); (c) 50% patches reduced by optimized combination of color/geometry deterioration (r =0.63); (d) 50% patches reduced by accounting for color only (r = 1.0)

In order to make efficient the transmission and rendition of geometric primitive datasets for the display on clients, a simplification scheme to decimate triangle patches (Schroeder, 1992[14]) is extended. The extended algorithm accounts for the color attributes as well as geometric features to select best edges to be collapsed (Nakamura, 2000[15]). Although analogous concepts can be found in the literature (Hoppe, 1999[16]), what distinguishes our algorithm from the others lies in its auxiliary mechanism to determine the combination ratio r of the color/geometry components in an error metric automatically by reflecting the coherence structure of a given two-vector volumetric dataset. This is expected to be extremely useful for retaining meaningful details obtained by large-scale computations even within overall data reduction effects. Figure 9 uses a mechanical part volume to illustrate the feasibility of our optimized simplification scheme.

Conclusions

The parallel visualization subsystem in GeoFEM has been described. It can perform visualization concurrently with computation on a high-performance parallel computer, and output simplified geometric primitives to clients. It provides many kinds of parallel visualization techniques, covering from scalar data, vector data to tensor data. Some feature analysis techniques are presented to improve the quality of visualization results. Future work includes developing more sophisticated visualization techniques, and improving the quality and efficiency of visualizations.

Acknowledgments

The authors have benefited from continuous discussions with GeoFEM stuff. This research was funded by Special Promoting Funds of Science & Technology.

References

[1] Doi, S., et al., 1997, *RVSLIB: A library for concurrent network visualization of large-scale unsteady simulation.* SPEEDUP Journal, 11, 59-65.

[2] Okuda, H., Nakamura, H. and Yagawa, G., 2000, *GeoFEM: Parallel FE solid earth simulation platform computational performance and solid earth applications*. CD-ROM Proceedings of the 4th International Conference on Supercomputing in Nuclear Applications, Tokyo, ES-G3.

[3] Fujishiro, I., Takeshima, Y., Nakamura, H., and Suzuki, Y., 2000, *Parallel visualization utilities for solid earth simulation platform GeoFEM.* In CD-ROM Proceedings of the 4th International Conference on Supercomputing in Nuclear Applications,Tokyo, HPCC-I-3-G1.

[4] Wada, Y. http://www.tokyo.rist.or.jp

[5] Lorensen, W.E. and Cline, H.E., 1987, *Marching Cubes: A high resolution 3D surface construction algorithm.* ACM Comp. Graph, **21**(4), 163-169.

[6] Fujishiro, I., Maeda, Y., Sato, H. and Takeshima, Y., 1996, *Volumetric data Exploration using interval volume.* IEEE TVCG, **2**(2), 144-155.

[7] Zoeckler, M., Stalling, D. and Hege, H.-C.,1996, *Interactive visualization of 3D vector fields using illuminated stream lines.* In Proceedings of IEEE Visualization, 107-113.

[8] Cabral, B., Leedom, C., 1993, *Image vector field using line integral convolution.* In Com-

puter Graphics Proceedings, Annual Conference Series. ACM SIGGRAPH, 263-272.

[9] Delmarcelle, T. and Hesselink, L., 1993, *Visualizing second-order tensor fields with hyper-streamlines*. IEEE CG&A, **13**(4), 25-33.

[10] Shinagawa, Y. and Kunii, T.L., 1991, *Constructing a Reeb graph automatically from cross sections*. IEEE CG&A, **11**(6), 41-51.

[11] Fujishiro, I., Azuma, T., Takeshima, Y. and Takahashi, S., 2000, *Volume data mining using 3D field topology analysis*. IEEE CG&A, **20**(5), 46-51.

[12] Chen, L., Fujishiro, I. and Peng, Q.S., 2000, *Fast LIC image generation based on significance map*. In Proceedings of the Third International Symposium on High Performance Computing, 537-546.

[13] Chong, M.S., Perry, A.E., Cantwell, B.J., 1990, *A general classification of 3D flow fields*. Physics of Fluids Ann., **2**(5), 765-777.

[14] Schroeder,W.J., Zarge, J. A. and Lorensen, W.E.,1992, *Decimation of triangle meshes*. ACM Comp. Graph., **26**(2), 65-70.

[15] Nakamura, H., Fujishiro, I., and Takeshima, Y., 2000, *Towards optimizing local feature metric for simplifying colored interval volumes*, In Proceedings of Work in Progress, IEEE Visualization 2000, Salt Lake City.

[16] Hoppe, H., 1999, *New quadric metric for simplifying meshes with appearance attributes*. In Proceedings of IEEE Visualization '99, 59-66.