Proceedings of the 2005 IEEE
Engineering in Medicine and Biology 27th Annual Conference
Shanghai, China, September 1-4, 2005

Parallel Visualization of Visible Chinese Human with Extremely Large Datasets

Liu Qian, Gong Hui, and Luo Qingming
Hubei Bioinformatics and Molecular Imaging Key Laboratory,
Huazhong University of Science and Technology, Wuhan, 430074 China
E-mail: qianliu@mail.hust.edu.cn

***Abstract*—Single sectional anatomy image of Visible Chinese Human datasets achieved recently in First Military Medical University is almost 127 MByte. The total datasets are estimated to be 1.1 TByte if 48 bits color image is stored. Personal computer and graph workstation with multiple CPUs are also impossible to process extremely large datasets. Therefore the challenge is to visualize the extremely large datasets efficiently. Visible Chinese Human datasets are so massive in size they require the use of parallel computing resources for effective visualization. At present a parallel visualization program has been developed based on parallelism visualization toolkit (pVTK) on high performance cluster to solve the problem. The visualization results and performance demonstrate that the parallel program we developed provides a promising solution of handling extremely large datasets.**

I. Introduction

With the publication of the National Library of Medicine's (NLM) Visible Human Project (VHP) [1] came opportunities to explore the human body as never before. VHP created a digital image data set of complete human male and female cadavers in magnetic resonance imaging (MRI), computed tomography (CT), and anatomical (anatomic transverse section) modes. Followed VHP, the visible Korean human (VKH) male transverse section datasets was produced [2]. Visible Chinese Human (VCH) project was start from 2001 and two datasets were obtained [3, 4]. All these datasets are massive in size from 15 Gbyte (Male), 43 Gbyte (Female) of VHP, 153.7 Gbyte (Male) of VKH, to 1.1 TByte (New male datasets in 2005). Scientists are using computer-aid visualizations and simulations to resolve anatomy models of real human body. A significant and headachy problem when the extremely large datasets is processed is how to efficiently visualize and reconstruct the large datasets. In particular, it is overload for personal computer (PC) and graph workstation even with multiple CPUs to process this kind of datasets. Visualization and simulation are usually run in parallel on high performance clusters of high-bandwidth and large memories supercomputers, such as SGI' Origin system or multiple-nodes clusters of PCs. As large scale parallel computing resources become commonplace for scientists, it is essential to develop parallel visualization software to utilize these resources.

Visualization toolkit (VTK) [5] is an open-source, object-oriented software system for computer graphics, visualization, and image processing. VTK contains many serial visualizations, graphics and imaging algorithms and is portable to a variety of hardware platforms and operating system. For parallel computing, parallelism VTK (pVTK) is scalability to use increasing numbers of computing resources to more efficiently process large datasets and offers a full range of parallel visualization algorithms. pVTK encapsulates parallel computing details in order to simplify the creation of parallel visualization programs for users or developers. There are three possible types of parallelism available in parallel visualization system: task, pipeline and data parallelism. In this paper, we report on the development of a parallel visualization program which is capable of visualizing VCH datasets base on data parallelism and message passing interface (MPI).

II. Methods and Materials

A. Parallel Visualization

Previous approaches to visualize large datasets in the field of parallel visualization have focused on the area of parallel rendering. Parallel rendering approaches include ray tracing [6], polygon rendering [7] and volume rendering [8, 9]. All these previous algorithmic work can be integrated into pVTK as modules to create full functional visualization system.

VTK is a work-flow based visualization system. There are three fundamental types of parallelism [10]. The first type of parallelism is task parallelism. It occurs when independent modules are executed in parallel. Fig. 1 shows that skin and bone of VHP are processed in independent task in a task-flow execution. Task 1 and task 2 are executed in parallel. Different slice data of skin and bone are loaded in task 1 and task 2 which can be processed in diverse algorithms. The ovals denote input and output port of each task. The master task (for example, task 1 in master node of cluster) will collect outputs of slave tasks (task 2) and finish surface rendering or volume rendering. In task parallelism, each processor can process independent

This work was supported by the National High-Tech Research and Development (863) Program of China (863 Program: 2003AA231011, 2004AA231030).

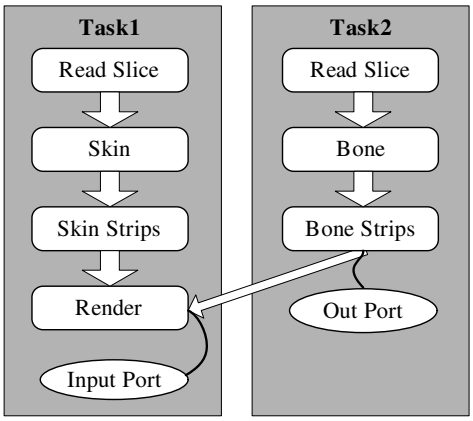


Fig.1 A sample of task parallelism pVTK

module by itself. Therefore, task parallelism is useful for running program graphs with many independent branches, such as those that implement parameter studies in parallel.

The second type of parallelism is pipeline parallelism. This type of parallelism occurs when a series of connected modules execute in parallel on independent data elements. These elements are usually elements of a time series or independent subsets of a single dataset. Pipeline parallelism is shown in Fig. 2. Between consecutive nodes, output port of last node is input port of next node. Node 1 reads slice data; Node 2 processes former subsets data from Node 1, the same as following nodes. Pipeline parallelism is useful for processing time-varying datasets with many independent resources, for example, simultaneously reading from disk, computing results and rendering with graphics hardware.

The third type of parallelism is data parallelism. The program executed in each node (task) is same program with independent subsets data. Data parallelism is described in Fig. 3. It shows data are loaded and processed in parallel. The results of the last data parallel module are usually merged to create a single process result, such as "display". Data parallelism is useful for processing large datasets. With data parallelism, a large dataset is partitioned into many independent subsets that are processed in parallel. System models for achieving parallelism include shared-memory and distributed-memory processes.

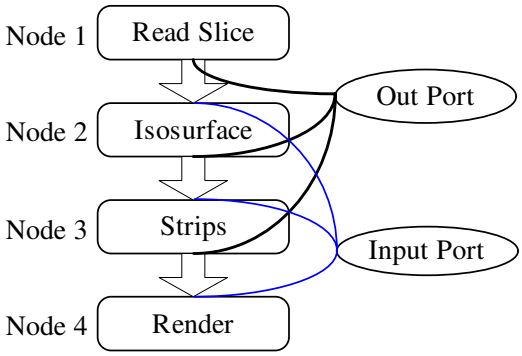


Fig.2 A sample of pipeline parallelism pVTK

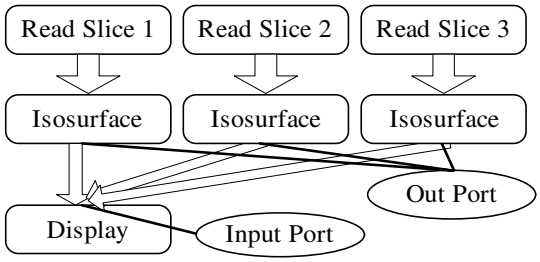


Fig.3 A sample of data parallelism pVTK

B. *Datasets of VCH*

VCH datasets was achieved by First Military Medical University. The data are transverse cross-sectional photographic images of a frozen-make cadaver with a resolution of 0.2 mm and a slice interval of 0.2 mm. The cadaver specimen was fixed and frozen in standing posture instead of lying. To distinguish arteries from veins, arteries were perfused by red perfusate from right common carotid artery. When specimen was embedded into gelatin, four register marks were mounted in four corner of whole specimen. Datasets were well preprocessed. First, four register marks were located of each slice and coordinates matrix were determined. Slices registering were performed by coordinate matrixing. Then, registered slices were segmented by computer-aided manual work. Segmentation data were processed by data parallel programs.

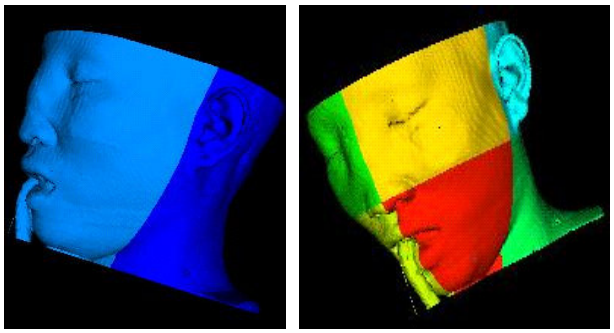
C. *Computing Resources*

Imaging preprocessing was implemented in IBM graphic workstation with dual Xoen CPUs, 4 Gbyte memories, and 3DLabs WildCats special graphic card. Parallel computing was implemented in LangChao (Beijing, China) high performance computing cluster with 34 Xoen CPUs, 34 Gbyte memories, InfiniBand high-bandwidth data switch, and 7 TByte storages. The maximum peak value in high-performance Linpack benchmark test is 102.6 GFlops. The operation system of each node is Linux with MPI libraries installed. High performance computing cluster locates in Huazhong University of Science and Technology, China.

III. RESULT

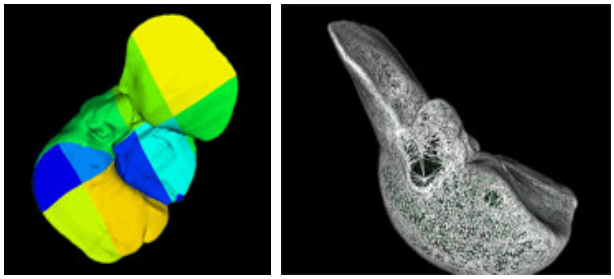
A. *Parallel Rendering*

Parallel visualization programs firstly were tested by using small scale CT datasets in VTK example package. The datasets is 128*128*93 pixels in size with 16 bits color. Data parallel programs were implemented in 1 node, 2 nodes and 8 nodes of cluster. Visualization computing in 2 nodes and 8 nodes are faster than that in single node which was considered as serial program. The results of surface rendering in 2 nodes and 8 nodes were illustrated in Fig. 4. Before parallel computing, datasets were divided into equal size data portions. Data portioning were distributed to each process executing in cluster node. In Fig. 4, one color denotes one process contains a data



2 Nodes 8 Nodes

Fig.4 Isosurface rendering using 2 nodes and 8 nodes by data parallel computing. Each color denotes one data portioning.



Isosurface Polygonal Grid

Fig. 5 Isosurface rendering and polygonal grid of liver of VCP.

TABLE I
NUMBERS OF POINTS, TRIANGLE CELL AND TRIANGLE TRIP

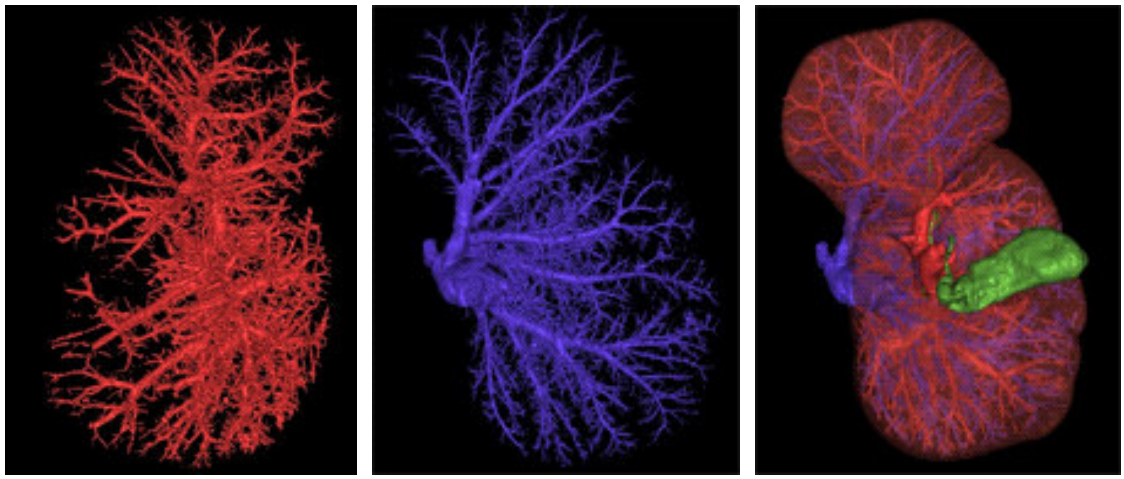
	Point	Triangle Cell	Triangle Strip
Contour	44862	143360	18944
Hepatica Artery	1089709	3218747	472482
Hepatica Vein	1438833	4037845	612418
Cholecyst	49840	162729	21299

portioning.

For large scale datasets, segmentation data of liver of VCH were visualized in parallel. The segmentation data size is 1712*1768*1000 with 24 bit color. There are four segmentation data: surface contour, hepatica artery, hepatica vein, and cholecyst. Fig. 5 shows surface contour was parallel rendered with 10 nodes and polygonal grid of reconstructed surface. Because rendering results is too large to display in normal PC, adjacent triangle cells in polygonal grid were merged to reduce triangle cells number. The numbers of points, triangle cells, and triangle strip in the rendering of surface contour, hepatica artery, hepatica vein, and cholecyst were listed respectively in Table I. Because there are a large amounts of small blood vessels in liver, the reconstructed numbers of triangle cells and strips from blood vessels are much more than contour and cholecyst. In order to reconstruct detail structure of arteries and venules, surface smoothing was carefully treated. Consequently, the reconstructed time of liver is mostly spent on blood vessels. The reconstructed efforts of liver were shown in Fig. 6. Blood vessels which diameter is 1 mm were clearly reproduced. Surface contour, hepatica artery, hepatica vein, and cholecyst were denoted in different colour. Surface contour was set to semitransparent.

B. Computing Performance

Visualization computing time was recorded during isosurface rendering. The parallel program of hepatica artery rendering took over 20 minutes on a single processor but only about 30 seconds on 32 processors. Fig. 7 shows the performance results of running datasets of liver parallel program using MPI for communication on a cluster using 1 to 32 nodes. The results show a significant speedup from 1 to 32 processors. The data parallel results validate the scalability of parallel program for large datasets processing. Additionally, large scale datasets of liver can not be processed in single PC or graphic workstation because of the memories can not exceed 4 Gbyte for that machine, while shared-memory system (SGI) and distributed-memory system (cluster) can handle this problem.



(a) (b) (c)

Fig. 6 3D visualization of hepatica artery (a, red), hepatica vein (b, blue), cholecyst (c, green) and liver (c).

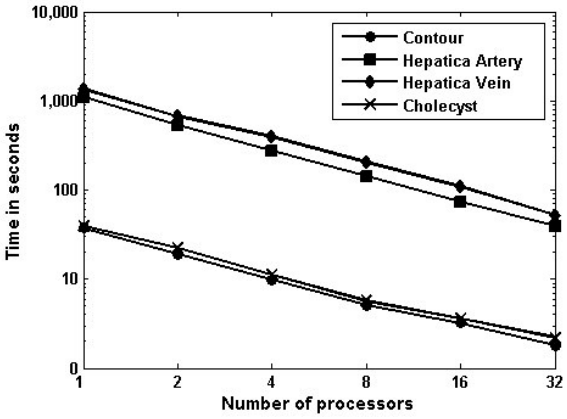


Fig. 7 Performance comparison of data parallel program using MPI and pVTK on cluster using 1 to 32 nodes

IV. CONCLUSION

In this paper, a parallel visualization program has been developed based on parallelism visualization toolkit (pVTK) on high performance cluster to solve the problem of extremely large scale datasets. The visualization results and computing performance demonstrate that the parallel program we developed provides a promising solution of handling extremely large VCH datasets.

ACKNOWLEDGMENT

We would like to thank Drs. Shizheng Zhong, Lin Yuan, Lei Tang in First Military Medical University for kindly offering the VCH datasets. We would like to acknowledge the students, Ting Li, Xueling Bai, Hua Chen, Li Wu, Chuanhua Cao, Yinping Liao, Zheng Fang, Li Yu, Zhe Feng, Weihua Luo, Anan Li, Xiaohua Lv, and Shan Ju, who work hard on datasets segmentation.

REFERENCES

[1] M. J. Ackerman, "The Visible Human Project: a resource for anatomical visualization," Medinfo., 1998, Vol. 9, pp. 1030-1032.

[2] J. Y. Kim, M. S. Chung, W. S. Hwang, J. S. Park, and H. S. Park, "Visible Korean Human: another trial for making serially-sectioned images," Stud. Health Technol. Inform., 2002, Vol. 85, pp. 228-233.

[3] S. X. Zhang, P. A. Heng, Z. J. Liu, L. W. Tan, M. G. Qiu, and Q. Y. Li, et al., "The Chinese Visible Human (CVH) datasets incorporate technical and imaging advances on earlier digital humans" J. Anat., 2004, Vol. 204, pp. 165-173.

[4] L. Tand, L. Yang, W. H. Huang, X. H. Wang, H. W. Hong, J. H. Fang, et al., "Data collecting technology on Virtual Chinese Human" Chinese J. Clin. Anat., 2002, Vol. 20, pp 324-329.

[5] W.J. Schroeder, K.M. Martin, and W.E. Lorensen, *The Visualization Toolkit An Object Oriented Approach to 3D Graphics*, Prentice Hall: Englewood cliffs, 1996.

[6] E. Reinhard, A. G. Chalmers, and F.W. Jansen. "Overview of parallel photo-realistic graphics," In Proceedings of Eurographics 98, 1998.

[7] T.W. Crockett. "An introduction to parallel rendering," Parallel Computing, 1997, Vol. 23, pp. 819–843.

[8] C. Wittenbrink, "Survey of parallel volume rendering algorithms," In Proceedings of Parallel and Distributed Processing Techniques and Applications, 1998, pp. 1329–1336

[9] R. Yagel, "Towards real time volume rendering," In Proceedings of GRAPHICON'96, 1996, Vol. 1, pp. 230–241.

[10] J. Ahrens, C Law, W. Schroeder, K. Martin, M. Papka, "A Parallel Approach for Efficiently Visualizing Extremely Large, Time-Varying Datasets," *Technical Report*, Los Alamos National Laboratory