Information Sciences 370–371 (2016) 463–475

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

Efficient volume rendering methods for out-of-Core datasets by semi-adaptive partitioning

Jian Xue[a,*], Jun Yao[a], Ke Lu[a], Ling Shao[b], Mohammad Muntasir Rahman[a]

[a] School of Engineering Science, University of Chinese Academy of Sciences, No.19A Yuquan Road, Beijing 100049, China
[b] Department of Computer Science and Digital Technologies, Northumbria University, Newcastle upon Tyne, NE1 8ST, United Kingdom

ARTICLE INFO

Article history:
Received 18 April 2016
Revised 8 July 2016
Accepted 6 August 2016
Available online 11 August 2016

Keywords:
3D image processing
Scientific visualization
Volume rendering
Out-of-core algorithm

ABSTRACT

Volume rendering methods are widely used for the high-quality visualization of various 3D datasets, especially scalar field datasets (e.g., 3D images). However, when rendering datasets with ultra-high spatial resolutions, which occupy massive (out-of-core) storage space, some traditional in-core volume rendering algorithms cannot function because the large input data can hardly be handled in the main memory. Simple modifications based on disk cache I/O do not perform well because of the overheads associated with external memory access. To solve this problem, this paper describes a semi-adaptive partitioning strategy and an efficient out-of-core visualization framework with improved volume rendering algorithms. Under this new partitioning strategy, an out-of-core dataset is spatially divided into small sub-blocks of different sizes, which are organized by a binary space partitioning (BSP) tree. Each sub-block can be loaded into the fast texture memory of the graphics hardware to be rendered by our improved volume rendering algorithms. The final result is obtained by composing the projection images of all sub-blocks after traveling the BSP tree according to the viewpoint. Experimental results indicate that the new methods are effective and efficient for visualizing out-of-core 3D scalar field datasets.

1. Introduction

For the visualization of 3D scalar field datasets (e.g., 3D images), volume rendering is widely used because of its ability to generate high-quality 2D projection images with sufficient detail. Its most common application is in medical imaging field, which is useful for doctors to view the regions of interest in the datasets obtained from the image acquisition devices such as CT (Computed Tomography), MRI (Magnetic Resonance Imaging) and Ultrasonography machines. Besides, it can also be used for nondestructive flaw detection in the industrial field, because the resulting image presents the whole scalar field distribution, thus the inside structure of the scanned object can be seen clearly through appropriate transfer functions.

Most conventional volume rendering algorithms were designed and implemented under the assumption that the main computer memory would be sufficiently large to handle the entire dataset, that is, the dataset to be processed could be loaded into the main system memory before the algorithm started running. However, with the rapid progress of data acquisition devices and related techniques, the resolution of the data being acquired continues to increase, leading to very large (out-of-core) datasets. These large datasets usually occupy tens or hundreds of gigabytes of storage space. This presents a

* Corresponding author.
E-mail addresses: xuejian@ucas.ac.cn (J. Xue), luk@ucas.ac.cn (K. Lu).

http://dx.doi.org/10.1016/j.ins.2016.08.017
0020-0255/© 2016 Published by Elsevier Inc.

challenge to conventional volume rendering methods, as it becomes impossible to load the entire input data into the main memory. To process such large datasets, algorithms must deal with small chunks of data while the rest of the dataset is stored out-of-core or in external memory (e.g., hard disk). However, most traditional volume rendering algorithms do not perform well in the transition form in-core to out-of-core processing conditions. The major bottleneck tends to be the I/O between the fast internal memory and the slower external memory.

In this paper, we attempt to solve this problem using a uniform out-of-core visualization framework and new volume rendering algorithms for out-of-core datasets. The new framework offers transparent and efficient access to the out-of-core volume dataset cached in the hard disk, which makes it easy to transfer common data-processing algorithms from in-core to out-of-core. Furthermore, we develop new out-of-core volume rendering algorithms based on a kind of semi-adaptive partitioning strategy, and achieve good performance by minimizing the time required to reload volume data from the external memory to the internal memory.

2. Related work

Volume ray casting is the best-known image-based volume rendering technique. Since the earliest ray casting algorithms were presented in the 1980s [7,17], a number of improved methods have been proposed. However, most of these are not suitable for out-of-core datasets, because they require random access to the voxels when casting rays through the volume data field, which would result in substantial disk I/O overheads.

The rapid development of graphics hardware has introduced the potential for fast volume rendering methods [1]. Carbral et al. [3] proposed the use of 3D texture mapping hardware to accelerate volume rendering. Since then, some improved algorithms have been presented [11,27,28]. However, because of the random access to volume data and the limitation of the 3D texture size, these methods cannot be directly used for out-of-core datasets. To address these problems, Laine et al. [16] presented an efficient GPU-based voxel representation for visualizing large and complex geometry. Fogal et al. [10] analyzed the critical factors and bottleneck in the GPU-based volume ray casting algorithms for large-scale datasets, and implemented a ray-guided volume renderer which achieved better performance than previous work. Okuyan et al. [21,22] presented a view-dependent refinement scheme for unstructured tetrahedral volumetric meshes and a GPU-based volume rendering algorithm that uses a cell projection-based ray-casting algorithm designed for CPU implementations. Using these cell projection and GPU acceleration, their method enables the direct volume rendering of large, unstructured volumetric grids.

For the visualization of out-of-core datasets, Silva et al. [25] examined various out-of-core algorithms for scientific visualization and computer graphics. Their paper discussed many out-of-core techniques for visualization, such as the work of Farias and Silva [9], who introduced a set of techniques for the direct volume rendering of arbitrarily large unstructured grids on machines with limited memory, including an extension of the ZSWEEP algorithm [8] to an out-of-core setting. Chiang et al. [5] presented a unified infrastructure for parallel out-of-core volume rendering of large unstructured grids on distributed-memory parallel machines. Recently, Nuber et al. [20] presented a method for visualizing higher-resolution datasets based on an out-of-core point-based rendering approach. Gobbetti et al. [12] used a coarse volume hierarchy in which binary space partitioning (BSP) is applied to the out-of-core triangle mesh. Their technique allows the interactive view-dependent rendering of huge complex 3D models. Crassin et al. [6] implemented an N^3-tree with brick structure and an associated ray-casting rendering method to achieve real-time level of detail (LoD) rendering performance for billions of voxels.

However, the methods described above were primarily concerned with interactive performance for their specific applications. Thus, they usually focused on eliminating unnecessary data processing to achieve acceptable visual quality using some well-designed data structures. When high-resolution and precise results are required, these well-designed but complicated LoD or view-dependent data structures and processing procedures often cannot obtain the expected benefits. Faced with these problems, some researchers have begun to develop new methods for out-of-core data on large-scale parallel computers or in distributed environments. Peterka et al. [24] ran a parallel volume rendering algorithm at massive concurrency, rendering large datasets with 4096 cores on an IBM BG/P system. Stuart et al. [26] proposed a programming model based on MapReduce to realize parallel volume rendering on multiple GPUs. Howison et al. [14] presented a hybrid parallelism approach, using distributed-memory parallelism across a large number of nodes to increase the available FLOPs and memory. This method ensures that each node performs its part of the larger computation as efficiently as possible. Hassan et al. [13] adopted distributed ray casting to do real-time interactive volume rending. Cao et al. [4] presented a hybrid parallel rendering scheme which reduced the data transfer delays during remote rendering. Pan et al. [23] also proposed a hybrid architecture based on various hardware environments and technologies to achieve interactive operations on very large volumetric datasets. However, there has been a lack of research on out-of-core volume rendering algorithms that run efficiently on normal PCs or workstations.

3. Out-of-core visualization framework

In order to provide an efficient and flexible underlying processing mechanism for the out-of-core datasets and make the implementation of visualization algorithms simpler, we design and implement an out-of-core volume data visualization framework, which is described in the following subsections.