Interactive Rendering and Efficient Querying for Large Multivariate Seismic Volumes on Consumer Level PCs

Liang Zhou*
SCI Institute and the School of Computing,
University of Utah

Charles Hansen†
SCI Institute and the School of Computing,
University of Utah

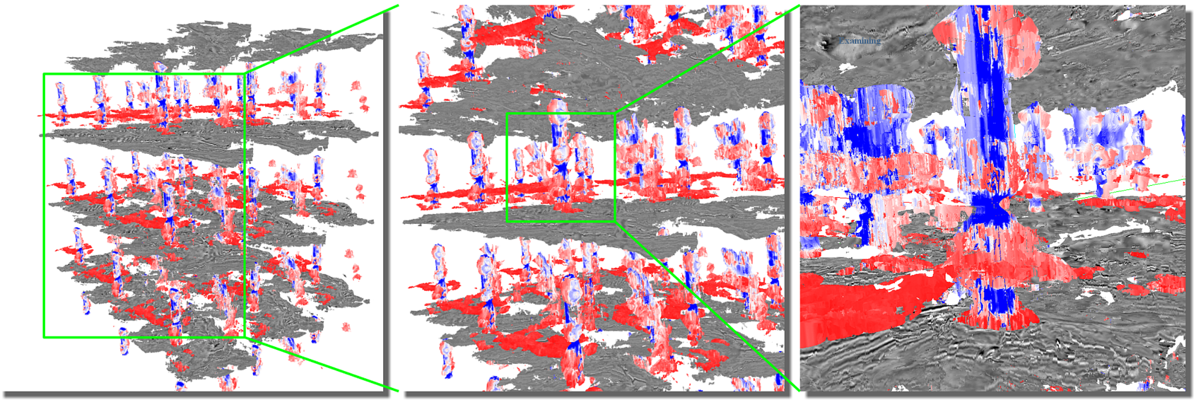

Figure 1: Our proposed method allows interactive visualization and efficient query of large multivariate seismic datasets on consumer level PCs. Shown here is a test seismic data of six attributes with size: $1278 \times 1653 \times 1704$.

Abstract

We present a volume visualization method that allows interactive rendering and efficient querying of large multivariate seismic volume data on consumer level PCs. The volume rendering pipeline utilizes a virtual memory structure that supports out-of-core multivariate multi-resolution data and a GPU-based ray caster that allows interactive multivariate transfer function design. A Gaussian mixture model representation is precomputed and nearly interactive querying is achieved by testing the Gaussian functions against user defined transfer functions on the GPU in the runtime. Finally, the method has been tested on a multivariate 3D seismic dataset which is larger than the size of the main memory of the testing machine.

Keywords: Multivariate volume, Out-of-core Methods, Transfer Functions

1 Introduction

Due to the advance in 3D seismic imaging techniques, resolution of 3D seismic datasets used in petroleum industry are usually of giga-bytes. Multiple attributes derived from the original seismic amplitude dataset have been used for aiding the interpretation of the seismic survey. With these additional attributes however, the size of the entire dataset may become far larger than the capacity of the GPU memory and even the main memory of a typical workstation.

Recently, GPU-based multi-resolution out-of-core volume rendering systems have been proposed. The Gigavoxel approach by Crassin *et al.* [1] and CERA-TVR by Engel [2] divide the dataset into multi-resolution bricks and utilize a octree structure and ray guided paging system to efficiently render large sparse volume datasets. Hadwiger *et al.* [3] propose a rendering system which uses a virtual memory system and 2D mip-mapping to support dense and noisy peta-scale microscopy scans. However, none of these methods are able to handle large multivariate volume datasets.

In this work, we extend the approach by Hadwiger *et al.* to support interactive rendering of large multivariate seismic datasets on a consumer level PC. On the other hand, data value querying raises another challenging issue for multivariate datasets especially when the data is very dense as hierarchical acceleration techniques, e.g. octrees may not be beneficial. As such, we propose an efficient data querying technique based on precomputed per-block Gaussian mixture models and run-time ellipse-polygon intersection detection. An interactive exploration system has been built to allow the user to visualize the multivariate volumes as well as to edit multivariate transfer functions with the query feedback on parallel coordinate plots.

2 Multivariate Out-of-core Volume Rendering

Multivariate multi-resolution data blocks are stored in our virtual memory structure. The associated ray caster is able to support multivariate transfer functions (TFs) which are interactively defined by the user.

2.1 Virtual Memory Structure for Multivariate Volumes

We share the same virtual memory hierarchy as in the work of Hadwiger *et al.*, namely, in a top-down manner: page table directory, page table and block caches. The difference is that instead of storing a single scalar volume in the block cache, we store data of all attributes at a given block location contiguously in the block cache. The page table entries are set to point to the beginning of the first attribute of each block. When the volume renderer makes paging requests, the virtual memory system updates all attribute blocks of the requested block location. Also, we store our multi-resolution

*e-mail: zhoul@cs.utah.edu
†e-mail: hansen@cs.utah.edu

IEEE Symposium on Large Data Analysis and Visualization 2013
October 13 - 14, Atlanta, Georgia, USA
978-1-4799-1658-0/13/$31.00 ©2013 IEEE

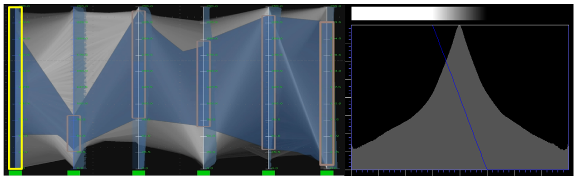

Figure 2: The TFs classifying the channel system in Figure 1. The *visibilisty* TF is shown to the left where the blue PCP indicates the query result with the user defined TF widget. While the *appearance* TF to the right sets a gray-level color map for the amplitude attribute signifies with yellow TF widget on the PCP.

blocks in a block file for each attribute to avoid the block building process from 2D tiles that intersect with the viewport. During initialization, our system fills the block cache by simply fetching blocks from the block cache files and sets flags in the page table and page table directory accordingly.

2.2 Multivariate Transfer Functions

Multivariate TFs are supported in our method, and to reduce the computational complexity, we separate the n-dimensional value space formed by n attributes into $n-1$ 2D space. The user designs the transfer function interactively on a parallel coordinate plot (PCP) based editor as shown in Figure 2. We define a so called *visibility* TF, comprised by the $n-1$ 2D space, which determines the visibility of voxels and also define an *appearance* TF of 1D which controls the visual appearance of the visible voxels. The user defines the multivariate visibility TF by manipulating TF widgets on the parallel coordinate axes and designs the appearance TF by clicking on a desired axis and editing in a 1D TF editor to set color and opacity. Alternatively, the visibility TF can be modified in a 2D TF editor for a chosen pair of attributes for more refined result.

In the TF sampling function of the GPU ray caster, we first determine the visibility of a voxel based on current visibility TF using an ID map which stores the coverage of all user defined visibility TFs by bitwise `OR`. If any attribute value of current voxel falls outside the coverage of current visibility TF, the voxel is skipped. Otherwise, the visible voxel is rendered with the user designed appearance TF.

3 Efficient Multivariate Query

To allow efficient data query on the noisy seismic datasets, we propose a two-stage approach that utilizes Gaussian Mixture Model (GMM) to compactly approximate the multidimensional distribution of data. The method first computes GMM for each block in a pre-computation stage and then tests ellipse-polygon intersection in runtime to query data values for voxels selected by user defined TFs. The per-block GMM is required to compute only once for a dataset and the runtime querying achieves near interactive performance.

3.1 Per-block Gaussian Mixture Model Computation

Assuming the datasets follow Gaussian distribution, we are able to describe the distribution using GMM which is very compact in terms of storage. GMMs are computed using the well-known expectation maximization algorithm, and we pre-compute GMMs for each block at its finest resolution for only once. In the same fashion of our TF space as described in Section 2.2, we compute GMMs in the $n-1$ 2D space. The computation is performed using CUDA thrust library and the result is written to a file which records the mean value and covariance matrix for each Gaussian distribution for each block. We empirically choose the number of Gaussians to be three as it strikes a balance between the closeness of GMM approximation of the original distribution and the compactness of storage.

3.2 Runtime Ellipse-Polygon Intersection Test

During visualization, the system queries data values for user defined TFs on the GPU with the GMM information stored as a texture. For any given pair of attributes, each Gaussian distribution is a 2D ellipse while each user defined TF is a 2D polygon. As such, we are able to conduct the query using ellipse-polygon intersection detection: i.e. if any part of the ellipse intersects with the TF polygon in any 2D sub-space of the $n-1$ 2D TF space, all values in the distribution are selected. Ellipse-polygon intersection is hard in the original space, and as such we compute a circle-triangle intersection in a transformed space. It is known that the ellipse can be transformed from a circle using matrix $\Sigma^{1/2}$ which is the square root matrix of matrix Σ which holds the eigen vectors of the ellipse. Therefore, the ellipse can be transformed back to a circle using the inverse matrix $\Sigma^{-1/2}$. The 2D polygon can be triangulated, and the triangles that form the polygon can also be transformed using $\Sigma^{1/2}$ into the circle's space, and then a much easier circle-triangle intersection test can be performed. Consequently, the query result is rendered using PCP by transforming the data values inside the Gaussian blobs to lines in the PCP.

4 Result

The proposed approach has been tested on a machine with Nvidia GTX480 with 1.5GB memory and a single Intel Core i5 processor with 16GB memory. Due to the restriction of usage of the datasets provided by our collaborators, we created a test dataset by repeating a small 100MB public domain seismic dataset with its five derived attributes three times in x and y axes and four times in z axis. The total size of the dataset is then 21.6GB, and we achieved frame rates from 2 FPS to 25 FPS with different settings of transfer functions on a frame buffer of 800×800. The querying time varies from 30 ms to 4 s which is positively correlated with the number of voxels that passed runtime testing. As shown in Figure 1, a channel system and a salt dome structure have been classified using the multivariate TFs. The channel system is colored using an appearance TF with a gray-level color map on the seismic amplitude attribute, while the salt dome structure is colored with a red-to-blue color map on the thickness attribute.

5 Future Work

In the future, we would like to further optimize the proposed method using multi-threading for data paging and exploit empty space skipping for multivariate TFs in the ray caster. We would also like to integrate the method into a full fledged interactive multivariate exploration environment designed for domain experts as proposed in [4].

References

[1] C. Crassin, F. Neyret, S. Lefebvre, and E. Eisemann. Gigavoxels: ray-guided streaming for efficient and detailed voxel rendering. In *Proceedings of the 2009 symposium on Interactive 3D graphics and games*, I3D '09, pages 15–22, New York, NY, USA, 2009. ACM.

[2] K. Engel. CERA-TVR: A framework for interactive high-quality teravoxel volume visualization on standard pcs. In *Posters at Large-Data Analysis and Visualization 2011 LDAV 2011*, 2011.

[3] M. Hadwiger, J. Beyer, W.-K. Jeong, and H. Pfister. Interactive volume exploration of petascale microscopy data streams using a visualization-driven virtual memory approach. *Visualization and Computer Graphics, IEEE Transactions on*, 18(12):2285–2294, 2012.

[4] L. Zhou and C. Hansen. Transfer function design based on user selected samples for intuitive multivariate volume exploration. In *Proceedings of the 2013 IEEE Pacific Visualization Symposium (PacificVis)*, pages 73–80, 2013.