Compression, Segmentation, and Modeling of Large-Scale Filamentary Volumetric Data

Bruce H. McCormick* Purna Doddapaneni David Mayerich Zeki Melek

John Keyser

Texas A&M University, Computer Science Department

ABSTRACT

We describe a method for processing large amounts of volumetric data collected from a Knife Edge Scanning Microscope (KESM). The neuronal data that we acquire consists of thin, branching structures extending over very large regions that prior volumetric representations have difficulty dealing with efficiently. Since the full volume data set can be extremely large, on-the-fly processing of the data is necessary.

CR Categories: J.3 [Computer Applications]: Life and Medical Sciences—; I.3.8 [Computer Graphics]: Applications—;

Keywords: Applications of volume graphics and volume visualization, compressed volume data

1 INTRODUCTION

We present a method for processing large amounts of volumetric data collected from a Knife Edge Scanning Microscope (KESM), making it useful for subsequent visualization and analysis. This new technique in microscopy allows data to be collected at rates far exceeding those previously possible. Our goal is to scan, reconstruct, and visualize stained brain tissue at a neuronal level of detail. We present the reconstruction pipeline, from data acquisition to the generation of threads representing neuronal processes.

2 KNIFE EDGE SCANNING MICROSCOPE (KESM)

The Knife Edge Scanning Microscope (KESM) is a unique instrument developed at Texas A&M for the collection of volumetric data from brain tissue embedded in plastic [2]. The KESM uses knife-edge scanning, where a diamond knife acts as both a cutting tool and an optical element, which allows the tissue to be imaged while it is cut. A line-scan camera is used to record a "row" of pixels seen through an objective, which is focused near the cutting edge of the knife. As successive slices of tissue are taken, a volumetric data set is formed from the "stacks" of images. Note that the high precision cutting technique produces all slices in registration - no additional processing to achieve registration is needed. The KESM currently is capable of imaging data at .25 μm resolution in the image plane, with successive slices being .5 μm to 1.5 μm in thickness. An entire mouse brain scanned in this way would yield approximately 30 terabytes of raw volumetric data, at a rate of approximately one terabyte a day. Note that the KESM is still in prototype form and operation; while we have some scanned data volumes, we have not yet attempted to capture data in the amount or at the sustained rate that should eventually be possible.

*This work was supported by Texas Higher Education Coordinating Board ATP grant 000512-0146-2001 and NSF grant CCR-0220047.

3 DATA COLLECTION

The full volume of the filamentary neuronal data set can be extremely large (in the terabytes). Due to this large size, it is not feasible to maintain an entire volumetric region in main memory at once. Therefore, our reconstruction and visualization approaches must be oriented toward representing, storing, and visualizing only the data of interest. That is, we wish to perform a rough segmentation of the data before visualization or further analysis. The neuronal data that we acquire tends to have several distinguishing features. For the samples we use (Nissl or Golgi stained neuronal tissue) the data of interest tends to be sparse, taking up only a modest portion of the overall volume. Segments of the neurons to be modeled have a very long but thin (as opposed to blobby) structure. Real-world data also tends to be noisy, with tiny, false "specks" appearing throughout the volume. Ideally, these artifacts must be identified and removed either before processing, using image-based filtering techniques, or after the data set has been established. Given a volume data set, we define an Enhanced volume data set (EVDS) as follows: in addition to the value assigned to every vertex (voxel) of the grid, selected edges between vertices of the grid are given a Boolean label. This enhancement alone can aid in topological analysis of the relevant data [1]. Edge labeling is used to provide independent information about whether two vertices sharing a common active edge belong to the same underlying object. A number of data structures can be used to describe volumetric solids [3]. We have chosen to adapt AABB-trees for our basic data structure, since they can be generated incrementally, and maintain many of the hierarchical benefits of other volumetric storage structures. An L-block (LB), derived from an AABB structure, is defined as a 3-dimensional iso-rectangular block of enhanced vertex information. Each LB includes both a header and a vertex array. The header defines both the position of its least vertex as indexed within the uniform grid of the raw data set, and its template. The vertex array contains the enhanced vertex information (voxel value(s) and edge labels) for all voxels. LBs sharing an active edge are assumed to be connected. We store connected LBs in a hierarchical data structure, similar to an AABB-tree, that we refer to as an LB covering (LBC).

4 DATA COMPRESSION

The memory needed to hold useful amounts of uncompressed neural data is exceedingly large. For example, the raw KESM data for an entire mouse brain requires approximately 30 terabytes and must be stored in real time. Volume data generated by serial sectioning and scanning of a three-dimensional specimen can be compressed in real time by incrementally generating the EVDS. We consider vertices significant if they pass a simple thresholding test, and edges are labeled active if and only if both of the adjacent vertices are significant. Voxels that do not pass the thresholding stage are considered "white space," and it is assumed that they can be ignored thereafter. The EVDS is partitioned into 2x2x2 cells. If any of the voxels in a cell is valid, that cell is stored as a LB. The compres-

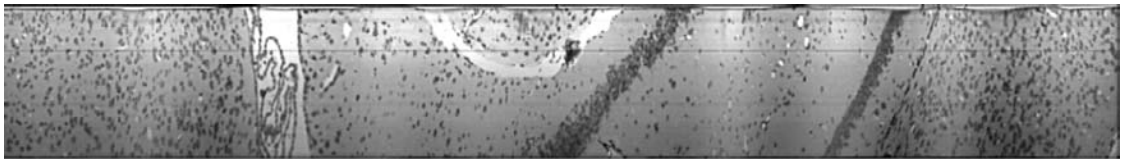

Figure 1: A portion of a slice from the BTS. A Nissl stain was used, so only cell bodies are stained. The section shown is only 500 pixels by 5500 pixels.

sion achieved will depend on the stain, the threshold used, and the density of the data. To test, we have considered two specific sets of the data, one consisting of mostly dendrites and the other mostly of cell bodies. Because of the different types of data prevalent, these should have different analysis statistics. For the first case we have considered 90 sections, each of 250x230 resolution. And for the second we have considered 100 sections, each of 500x500 resolution. Each voxel of the data set represents a volume of 0.37 μm by 0.37 μm by 0.5 μm. The initial data requires approximately 5 MB and 25 MB of storage space respectively. With realistic threshold levels, we form 65,611 LBs and 50,218 LBs, requiring about 1.1 MB and 0.8 MB to store, yielding a compression factor of approximately 4 and 31 respectively(i.e. compressing data to 20% and 3.2% respectively). Additional data compression is done by merging LBs where appropriate, which minimizes the header overhead. Because LBs store all data in an iso-rectangular volume, expanding an LB might require storing additional "white" space along with relevant data. We use a cost function based strictly on the relative storage requirements for the merged and unmerged LBs. For our test data sets, our merging strategy reduces the total number of LBs twofold (28,938) and fourfold (10,684), requiring less than 0.4 MB and 0.18 MB of storage respectively. Note that these strategies are well suited for processing 3D microscopic data where data arrives one "section" at a time and each section must be processed in real time. Due to the amount of data, it is not practical to store many sectional images in memory at once. Merging LBs requires only storage of the LBs that cover portions of the immediately preceding section.

5 Noise Reduction, Data Segmentation and Thread Generation

Some image based filtering of the data is done before edge labeling to get rid of some of the scanning artifacts. After initial data compression, the LBs are processed for noise reduction to improve the quality of reconstruction. First, we remove large smear noise that exists on single layers. Second, tiny disconnected LBs with no active edges are removed, since they are unlikely to be a part of a larger neuronal structure.

Taking advantage of the fact that neural data is both sparse and clustered, our data is further combined into clusters, each expressed as an LBC, and defined as groups of interconnecting LBs. If two LBs border on each other and at least one of the voxels has an active link to a voxel in the other LB, they are considered to be in the same cluster. Since the voxels themselves are used to determine cluster boundaries, this scheme effectively segments the data, i.e. it does not group two pieces of data that should have been separate. If LBs are clustered before merging, the space of LBs to be examined for potential merging is reduced, thus speeding up the algorithm.

An expanded connectivity graph is formed by linking the overlapping dilated LBs. To identify the major threads, the graph needs to be simplified using graph algorithms. The expanded connectivity graph is simplified into a tree format by temporarily removing "fine-scale" detail. The LB structure allows us to work with packages, rather than the data contained within. A "thread axis" is constructed, around which a medial axis approximation can be obtained. The medial axis representation can be iteratively refined to match the LB representation using radius approximation by using actual LB contents.

6 Visualization

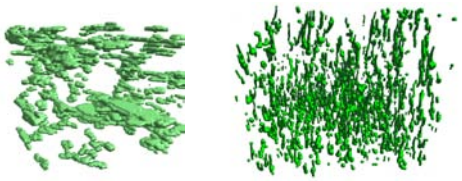

Figure 2: Two visualizations from different sets, a) isosurfaces , b) generated threads

Packing the data inside LBs allows us to display as bounding boxes interactively. The data content is hidden for faster frame rates, enabling the user to walk through the data easily. The internal data is displayed only for the very nearby LBs, or when the viewpoint is not changing. The bounding box information is also valuable in as a means of culling for faster raytracing of the volume data. The edge labeling and LBC structure is useful in grouping connected data together, and this connectivity information is used in fast isosurface generation. LBs can be isosurfaced in parallel, only exchanging information along boundary values when necessary. Knowing connectivity information ensures that the independently generated isosurfaces for each LB will match across LB boundaries.

7 Conclusion

We presented a method for processing large amounts of volumetric data collected from a Knife Edge Scanning Microscope (KESM) for subsequent visualization and analysis. We have described our current implementation of the pipeline from data acquisition until partial thread generation. We have demonstrated that we can process the data on-the-fly and do noise detection and compression using partially available data. LBC storage enables parallel execution of thread generation.

References

[1] T. Y. Kong and A. Rosenfeld. Digital topology: introduction and survey. *Computer Vision, Graphics, and Image Processing*, 48:357–393, 1989.

[2] B. H. McCormick. Development of the brain tissue scanner. Technical Report 18, Department of Computer Science, Texas A&M University, College Station, TX, March 2002.

[3] A. S. Winter. *Volume Graphics, Field Based Modelling and Rendering*. PhD thesis, Department of Computer Science, University of Wales, Swansea, December 2002.