Geometric Analysis of 3D Electron Microscopy Data

Ullrich Köthe, Björn Andres, Thorben Kröger, Fred Hamprecht

Multidimensional Image Processing Group, University of Heidelberg

Abstract We present a complete pipeline for the segmentation and analysis of 3-dimensional electron microscopy data. Considerable algorithmic optimizations and parallelization have been applied to make the system applicable to data as large as 8 gigavoxels. Discrete geometry plays a prominent role at several processing stages (initial watershed segmentation, cell complex representation, reduction of oversegmentation by a graphical model, topological and geometric feature computation). We will demonstrate our algorithms and visualization tools in an on-site software demo.

1 Introduction

Understanding the human brain is one of the most challenging problems in science. High-resolution 3-dimensional electron microscopy of brain tissue is an important tool in this area. Special staining techniques are used to mark the cell membranes of all neurons, and a segmentation of these images will eventually provide a complete map of the neurons, their adjacency and their network of synaptic connections. This information can be represented as a graph, the so called *connectome* [13], which is an invaluable input for subsequent brain function analysis.

Isotropic resolution of at least 25 nm is necessary for reliable segmentation and interpretation of these images. Since the smallest known functional units of the mammalian brain beyond single neurons (the *cortical columns*) comprise about 1 mm^3 of neural tissue, the data for a single cortical column will eventually consist of about 40000^3 voxels. Currently available data sets contain 2000^3 to 6000^3 voxels (8...216 GBytes). Figure 1 left shows a small sub-region of a data set we are working on, which has been acquired by *serial block-face scanning electron microscopy* (SBFSEM [6]). Our analysis proceeds in the following steps:

1. Compute feature vectors describing the local neighborhood of every voxel.
2. Compute each voxel's membrane probability.
3. Compute an initial over-segmentation by means of the seeded watershed algorithm.
4. Compute a cell complex representation of the segmentation.
5. Compute features for all surface segments.
6. Reduce oversegmentation by a probabilistic graphical model on surface segments.
7. Characterize and visualize the resulting neural regions.

Digital geometry and mathematical morphology play a prominent role in this approach: watershed segmentation, creation of a cell complex representation, extraction of topological and geometric features for the different cells, and visualization of intermediate and final results. Space only permits a brief description of steps 1 to 3: Features comprise gradient magnitudes, eigenvalues of the structure tensor and Hessian matrix at multiple scales, as well as statistics of these measurements in isotropic neighborhoods. Feature vectors are transformed into membrane probabilities with a random forest classifier [4] that is trained from labels provided by a human expert. Watersheds are determined with a seeded version of the Vincent-Soille algorithm [14] where seeds are defined as connected regions of voxels whose membrane probability is very low ($< 0.5\%$).

2 Computing Cell Complexes on Large Datasets

Kovalevsky [9] proved that a topologically consistent representation of a N-dimensional segmentation requires explicit representation of all cell types up to dimension N. Therefore, we need to represent surfaces (2-cells), surface intersections (1-cells), and junctions (0-cells) in addition to the 3-dimensional regions (3-cells). Generalized combinatorial maps [5, 11] are the most powerful representations in this context because they not only store cells and their adjacencies, but also encode the topology of their embedding into 3D space. Unfortunately, these maps require a massive number of auxiliary *darts*, so that they are not feasible for our data which typically contain about 3 million regions and 80 million cells in total. The slightly weaker *cell complex representation* [8] which needs significantly less storage, is sufficient in our context, because the embedding can be easily reconstructed on demand from the labeled watershed image. The cell complex is constructed by a 3-dimensional generalization of the *crack insertion algorithm* [10]:

1. Create a topological grid with twice the resolution of the original grid. This is necessary in order to store explicit labels for the cells of dimension < 3.
2. Map region labels from the watershed segmentation onto topological grid points with three even coordinates. Each component of like-labeled points becomes a 3-cell.
3. Mark topological grid points with a single odd and two even coordinates as *active* when they are located between two differently labeled region points. Connected components of those active points become 2-cells (surfaces) of the cell complex.
4. Likewise, create 1-cells as connected components of active points with two odd and one even coordinates which are located between two or more differently labeled surface points.
5. Label topological grid points with three odd coordinates when they are located between two or more differently labeled 1-cells to get 0-cells.
6. For each k-cell, create a list of the points (coordinates) belonging to this cell.
7. Create adjacency lists for the cells' bounding relation.

To speed up computations, a large volume can be split into blocks that can be processed in parallel. To integrate the resulting pieces into a consistent whole, block must start and end at odd topological coordinates, and neighboring blocks must have one voxel overlap. Since the cell complex does not fit into memory at once, a sophisticated file format is required which supports fast access to subsets of the data and fast insertion of newly processed pieces. We found the Hierarchical Data Format (HDF5 [1]) to be ideally suited for this purpose. On our 2000^3 data set, the entire processing chain from region label image to cell complex takes about a day and results in a data structure of about 229 GB for the topological grid, 2 GB for the adjacency information, and 23 GB for the lists of coordinates constituting each cell. A detailed description of the algorithm can be found in [2], figure 1 right shows surface segments overlaid over the raw data.

3 Topological and Geometric Features

Since the watershed algorithm produces an oversegmentation, a correct segmentation can only be obtained by deleting surface segments in order to merge falsely split regions. We perform this task by means of a probabilistic graphical model [3] whose parameters are learned from training data. In our model, a random variable is assigned to each surface segment which takes a value of 1 when the system is certain that the corresponding surface should be kept, and 0 otherwise. A global energy function measures the probability of each configuration of kept/deleted surfaces (i.e. of each 0/1 assignment), and a (locally) optimal solution is computed by means of the belief propagation algorithm [15]. The definition of the energy function relies heavily on methods of

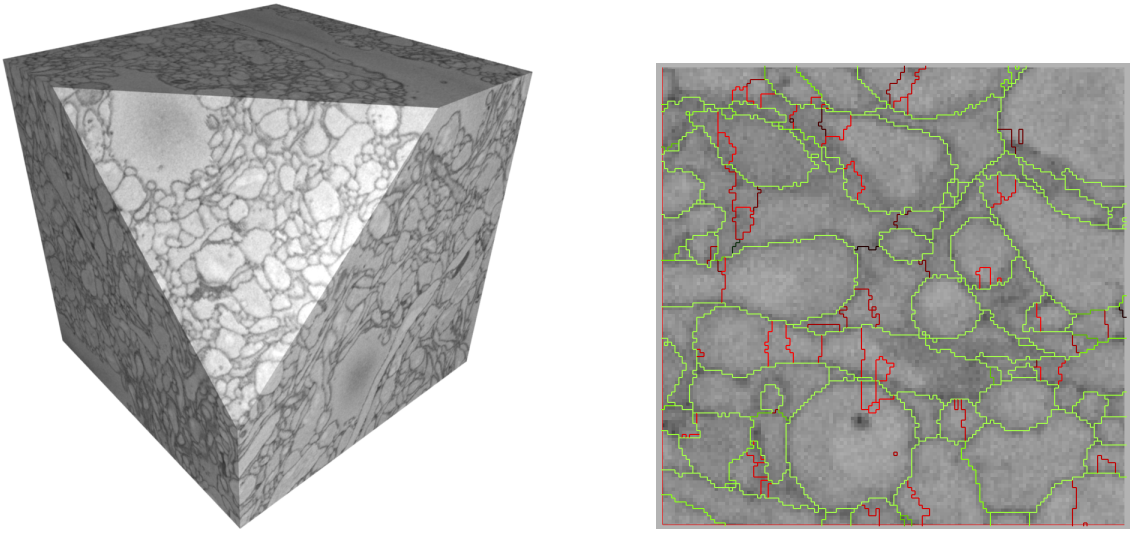

Figure 1: Left: 250^3 subset of the raw data. Right: Raw data with color-coded surface segment overlay (yellow: "keep", red: "delete", black: "uncertain", according to the unary potential p_1).

discrete geometry. First, in order to assign a random variable to each surface segment, these segments and their constituting points must be identified by the labels derived during the creation of the cell complex representation. Second, geometric and topological features for these segments are used in the definition of the probabilities that comprise the global energy function. The energy to be maximized is defined as

$$E(x) = \log\, p(x) \quad \propto \quad \alpha \sum_{i=1}^{S} \log\, p_1(x_i) + (1-\alpha)\left[\sum_{k=1}^{I_3} \log\, p_3\left(x_{k_1}, x_{k_2}, x_{k_3}\right) + \sum_{k=1}^{I_4} \log\, p_4\left(x_{k_1}, x_{k_2}, x_{k_3}, x_{k_4}\right)\right]$$

where S, I_3, and I_4 are the number of surface segments, ternary and quaternary intersections respectively, and x_i denotes the state ("keep" vs. "delete") of surface segment i. The unary potentials $\log p_1$ of the surface segments describe the log probabilities for each segment to be correct, based on features pertaining to one segment alone. These probabilities summarize various measures of membrane strength, as well as geometric features such as size and curvature. In contrast to voxel-based features, we are now able to compute features on data-dependent neighborhoods defined by the cells' shapes, similar to the superpixel approach of [12]. Feature measurements are transformed into probabilities by a second random forest classifier. The color-coding in figure 1 right illustrates the values of the potential p_1 for the surfaces depicted.

The ternary and quaternary potentials $\log p_3$ and $\log p_4$ assess properties of configurations of three or four adjacent surface segments, i.e. of surfaces that share a common intersection (a common 1-cell). The cell complex representation is obviously required to identify these configurations, but discrete geometry is also necessary to evaluate their probabilities. These probabilities, also learned by a random forest from expert labels, have two effects: On the one hand, they prevent dangling or isolated surfaces that could occur when surface segments are deleted without regard to neighboring segments. On the other hand, they favor configurations that lead to *good continuation* of the resulting surfaces. That is, adjacent segments are likely to be kept when they enclose an angle around 180°, whereas a segment is likely to be deleted when is meets the other segments at an angle of about 90°. These angles are measured by means of standard estimators of tangent planes known from discrete geometry [7].

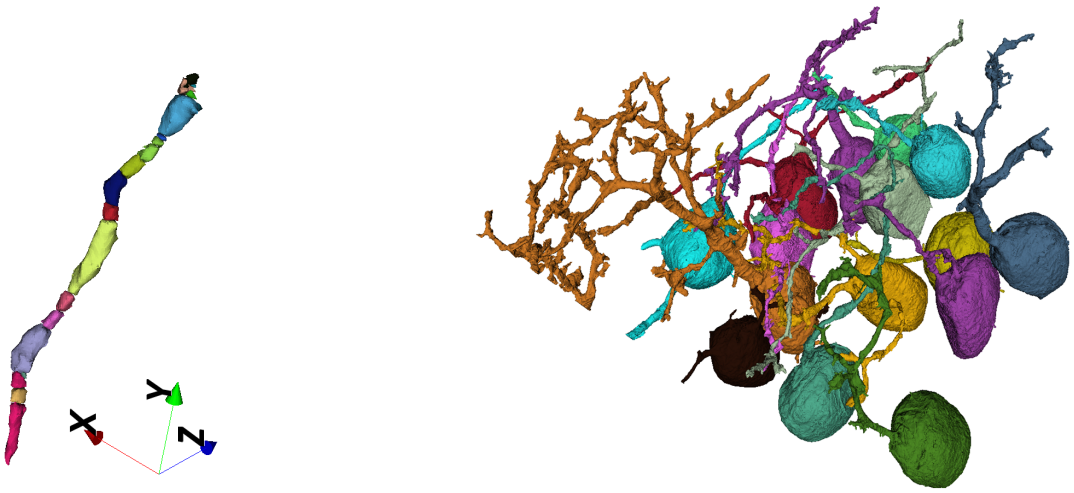

Figure 2: Left: A region that has been correctly merged by the graphical model after severe oversegmentation. Right: Some neurons of the final segmentation of the entire data set.

This objective function provides a well-defined probabilistic model for the reduction of oversegmentation. Since every surface segment is part of several intersections, global optimization of the objective leads to an implicit non-local propagation of local information. A locally optimal solution is found by belief propagation [15], and results are very satisfactory both empirically and w.r.t. ground truth, see fig. 2 left. The entire workflow (from initial computation of voxel features to convergence of the graphical model) takes about 1 week on 16 parallel machines.

4 Visualization

Visualization of the results is another important part of the project. On the one hand, the visualization of individual regions (i.e. neurons) and their relations helps biologist understanding the detailed anatomy of the brain. On the other hand, it is a indispensable tool for image analysis in order to improve the segmentation method: When the segmentation does not conform to ground-truth provided by the biologist (for small subsets of the data), it is possible to find out exactly where the algorithm went wrong, and why it arrived at incorrect surface probabilities. Visualization of 3-dimensional data, especially of the size encountered in this project, is a challenging problem, and methods of discrete geometry are once again central to its solution.

In particular, our software supports several visualization modes:

- In the standard view, the original data are displayed on three orthogonal, axis-aligned slices which can be placed arbitrarily in the data set by simple interactions. On top of these slices, any segment of the cell complex representation can be displayed as an overlay. Overlays can be switched on and off interactively and via programming. This is possible because the set of points constituting each segment is explicitly known.

- Overlays may also be color coded in order to visualize features and probabilities. Thus, undesirable feature assignments (that would lead to false removal or false preservation of surfaces) can be quickly spotted.

- Regions and sets of regions can be surface rendered and arbitrary rotated on a mouse click. To this end, the interpixel boundary of each region is triangulated (by splitting each surface square into a pair of triangles and subsequent standard mesh simplification).

An example of the resulting segmentation can be seen in figure 2.

5 Conclusions

We presented a hierarchical segmentation algorithm for the detection of neurons in SBFSEM data. At the first level, supervoxels are determined by a seeded watershed algorithm. Since supervoxels partition the domain in a data-driven manner, more informative features can be computed for the graphical model that forms the second level of our algorithm. Thanks to the balancing between probabilities of individual surface patches (unary potentials) and surface configurations (higher-order potentials), oversegmentation can be successfully reduced without introducing significant undersegmentation. However, segmentation accuracy must still be improved about threefold in order to be usable for connectome determination.

Parallelization reduced the computation time on 2000^3 voxels to about a week. Feature computation, classification, and cell complex construction are relatively easy to parallelize, whereas parallelization of more complex parts (watersheds, graphical model optimization) was not necessary as they consume only a small part of the total time.

References

[1] HDF5 data storage technologies, 2010. http://www.hdfgroup.org/HDF5/.

[2] B. Andres, U. Koethe, T. Kroeger, and F. A. Hamprecht. Representing the geometry and topology of large volume segmentations. submitted, 2010.

[3] C. Bishop. *Pattern Recognition and Machine Learning*. Springer, 2006.

[4] L. Breiman. Random forests. *Machine Learning*, 45(1):5–32, Oct. 2001.

[5] G. Damiand. Topological model for 3d image representation: Definition and incremental extraction algorithm. *Comput. Vis. Image Underst.*, 109(3):260–289, 2008.

[6] W. Denk and H. Horstmann. Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. *PLoS Biol*, 2(11):e329, 10 2004.

[7] R. Klette and A. Rosenfeld. *Digital geometry*. Morgan Kaufmann, 2004.

[8] V. Kovalevsky. Algorithms in digital geometry based on cellular topology. In: R. Klette and J. Zunic (eds.): *Combinatorial Image Analysis, Proc. IWCIA '04*, Springer LNCS 3322, pp. 366–393, 2004.

[9] V. A. Kovalevsky. Finite topology as applied to image analysis. *Comput. Vision Graph. Image Process.*, 46(2):141–161, 1989.

[10] U. Köthe. Deriving topological representations from edge images. In: T. Asano, R. Klette, and C. Ronse (eds.): *Theoretical Foundations of Computer Vision*, Springer LNCS 2616, pp. 320–334, 2002.

[11] P. Lienhardt. Topological models for boundary representation: a comparison with n-dimensional generalized maps. *Computer-Aided Design*, 23(1):59–82, 1991.

[12] X. Ren and J. Malik. Learning a classification model for segmentation. In: *Proc. ICCV '03*, pp. 10–17, 2003.

[13] O. Sporns, G. Tononi, and R. Kötter. The human connectome: A structural description of the human brain. *PLoS Comput Biol*, 1(4):e42, 09 2005.

[14] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Patt. Anal. Mach. Intell.*, 13(6):583–598, 1991.

[15] J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. In: G. Lakemeyer and B. Nebel (eds.): *Exploring artificial intelligence in the new millennium*. Morgan Kaufmann, 2003.