A Multiscale Parallel Computing Architecture for Automated Segmentation of the Brain Connectome

Sylvain Jaume*, Kathleen Knobe, Ryan R. Newton, Frank Schlimbach, Melanie Blower, and R. Clay Reid

***Abstract*—Several groups in neurobiology have embarked into deciphering the brain circuitry using large-scale imaging of a mouse brain and manual tracing of the connections between neurons. Creating a graph of the brain circuitry, also called a connectome, could have a huge impact on the understanding of neurodegenerative diseases such as Alzheimer's disease. Although considerably smaller than a human brain, a mouse brain already exhibits one billion connections and manually tracing the connectome of a mouse brain can only be achieved partially. This paper proposes to scale up the tracing by using automated image segmentation and a parallel computing approach designed for domain experts. We explain the design decisions behind our parallel approach and we present our results for the segmentation of the vasculature and the cell nuclei, which have been obtained without any manual intervention.**

***Index Terms*—Brain anatomy, circuit connectome, computational architecture, data-intensive computing, electron microscopy, image segmentation, multiscale analysis, parallel computing.**

I. Introduction

MULTITERABYTE volumetric datasets of a mouse brain have been acquired in different institutions using serial-section electron microscopy [1]–[22]. Manual tracing efforts by teams of up to 70 trained biologists working for several months have led to the analysis of a portion of these datasets [23]–[30]. However, creating a complete connectome using such a labor-intensive method would require several decades and is unlikely to scale up with the size of future datasets. In order to reduce this time frame to the order of days, we propose to combine robust segmentation techniques and a flexible, parallel computing approach. We first describe our segmentation algorithm as we apply it on subvolumes of the volumetric dataset. Then we explain and justify our decision to use the Intel CnC library [31]–[36] as the framework for our parallel implementation. Next, we present some results for the segmentation of the vasculature and the cell nuclei, and finally, we discuss potential applications in other scientific domains and we summarize the article.

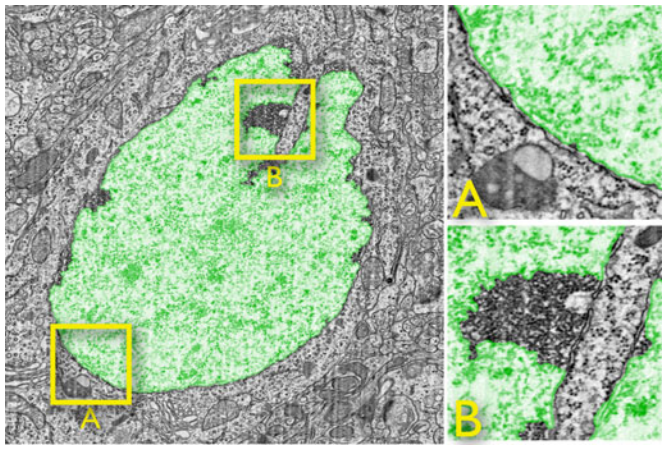

Fig. 1. Our segmentation results (green) for a cell nucleus are highly accurate. Inset A shows that the segmentation accurately follows the boundary of the cell. Inset B indicates that polyribosomes near the cell membrane can cause a segmentation error, but the error is generally very localized and does not propagate to neighboring sections.

II. Segmentation Algorithm for Subvolumes

Our segmentation algorithm is a sequence of automated image operations performed on multiscale subvolumes: extraction of subvolumes, local image filtering, and robust connected component analysis.

A. Extraction of Multiscale Subvolumes

The size of a subvolume depends on the scale of the feature of interest. For the vasculature large subvolumes are necessary, while for cell nuclei smaller subvolumes are created. Note that features split between neighboring subvolumes might be discarded because they do not exhibit a sufficient size in any subvolume. To address this issue, we create subvolumes, also called *ghost volumes*, that overlap by at least half the maximum diameter of the feature of interest. The overlap is significantly smaller along the Z axis than along the X and Y axes since the section thickness is typically larger than the in-plane pixel resolution.

B. Local Image Operations

We perform local operations on the images based on the intensity distribution of the feature of interests, in a similar fashion as typically performed on a region of interest. First, we manually segment the feature of interests in three randomly selected

Manuscript received April 3, 2011; revised July 14, 2011 and August 16, 2011; accepted August 24, 2011. Date of publication September 15, 2011; date of current version December 21, 2011 *Asterisk indicates corresponding author.*

*S. Jaume is with the Department of Neurobiology, Harvard Medical School, Boston, MA 02115 USA (e-mail: sylvain@csail.mit.edu).

K. Knobe, R. R. Newton, F. Schlimbach, and M. Blower are with Intel Corporation, SSG/DPD/TPI, 77 Reed Road, Hudson, MA 01749, USA (e-mail: kath.knobe@intel.com; ryan.r.newton@intel.com; frank.schlimbach@intel.com; melanie.blower@intel.com).

R. C. Reid is with the Department of Neurobiology, Harvard Medical School, Boston, MA 02115 USA (e-mail: clay_reid@hms.harvard.edu).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TBME.2011.2168396

subvolumes and perform an histogram analysis for the different features. Then we select the intensity threshold that results in the minimum misclassification. Finally, since small clusters of pixels are typically noise, we remove them and fill the holes within large clusters of pixels. This sequence of intensity thresholding and noise removal results in large areas that characterize the features of interest that we try to segment.

C. Data Recovery and Connected Components Analysis

It is typical that part of a tissue sample gets damaged during the sectioning or the staining. This artifact results in partially imaged sections. Fortunately, performing the sectioning process at a thickness below the size of the features of interest allows for some data recovery by interpolation across the stack of imaged sections. First we sum up the intensity of pixels across Z stacks of $\frac{N}{10}$ sections, where N is the total number of sections within a subvolume. Since the previous operations produced binary sections, the pixel intensity within each section is either 0 or 1. Then, if the total intensity of a pixel after the summation is larger than $\frac{N}{20}$, the pixel is classified as belonging to a feature of interest. Finally, we perform a 3-D connected components analysis, compute their respective number of pixels, and threshold out the smaller components.

III. Parallel Computing Approach

The development of parallel computing code can be challenging. In parallel computing solutions using libraries such as OpenMP [37], specific keywords need to be inserted inside the computation methods. These modifications require that the programmer is not only a domain expert, but also a parallel computing expert. Our approach is a departure from this programming methodology in the sense that only the signatures of the computation methods are modified and only a small header file describing the relations between data items and computation methods, also called computation steps, is added.

A. Subvolume Items and Segmentation Steps

In our application, the data items are the image subvolumes, and the computation steps are the segmentation methods for the different features of interest. Every image subvolume is indexed by a triplet $\langle i, j, k \rangle$, which we call a subvolume control tag. The segmentation method will consume a pair of subvolume control tag and subvolume data item. The method is then triggered and upon its execution it will produce a set of segment data items, where the entire set is indexed by a segment set control tag $\langle x, y, z \rangle$, where $x = i$, $y = j$, $z = k$.

B. Segment Sets and Merge Steps

To connect segments across neighboring subvolumes, we write a merge method that compares the segments across subvolumes, and connects the segments that overlap. The merge method thus consumes a segment set control tag and produces a connectivity tag $\langle u, v, w, a, id \rangle$, where $u = x$, $v = y$, $w = z$, $a \in 1, 2, 3$ defines the direction of the neighboring subvolume

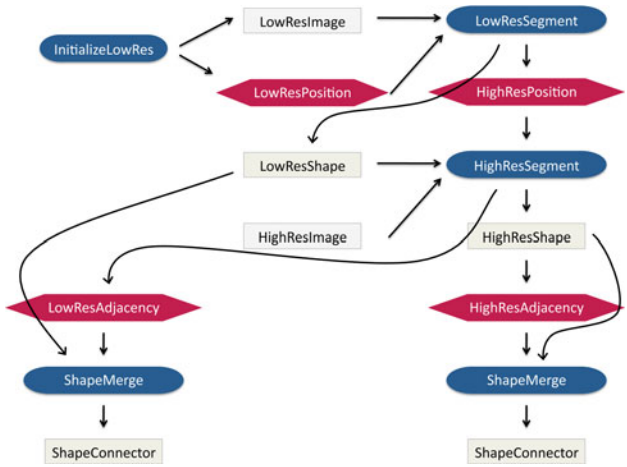

Fig. 2. Our parallel computing implementation can be represented in a graph connecting the computation steps *(blue)*, the data items *(grey)*, and the control tags *(red)*. We use the Intel CnC library in order to separate the development of the computation steps, and the development of the header file that describes the relations in the graph above. This separation allows for an implementation that can be easily optimized to new hardware configurations.

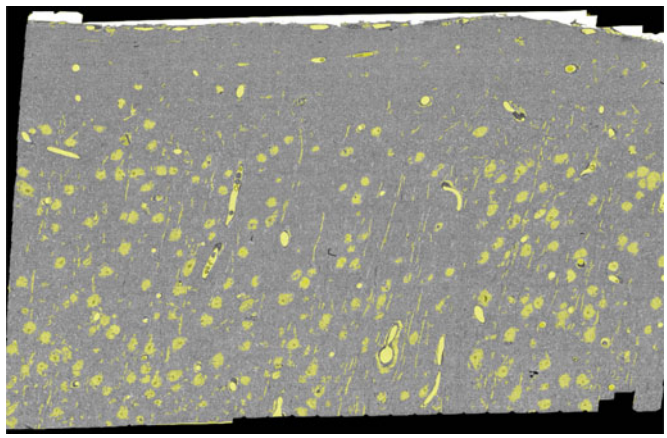

Fig. 3. Overlay of our segmentation results for nuclei and blood vessels on one section of the electron microscopy datasets allows for a visual inspection of the segmentation accuracy. This presentation can help the expert to visualize the features of interest when reviewing the stack of images.

(+X axis, +Y axis or +Z axis), id is the id of the segment in the second subvolume.

C. Reconstruction Step

Upon completion of all the merge steps, a reconstruction step is executed to create the segmentation results for the entire dataset. This light-weight step simply follows the list of connectivity tags and puts together the segments that overlap partially. Fig. 2 shows a graphical presentation of the elements in the Intel CnC header file.

IV. Results

We applied our parallel implementation to segment the vasculature and the cell nuclei in the entire dataset. The overlay of our segmentation results on one section of the dataset is shown in Fig. 3. A surface rendering in 3-D can be seen in Fig. 4.

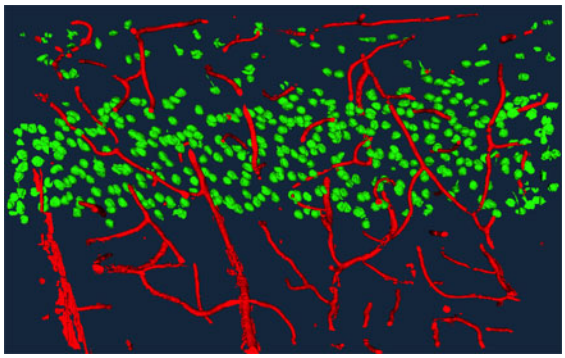

Fig. 4. We created a surface rendering of the segmentation results for cell nuclei *(*green) and the vasculature *(red)* using triangulation of binary masks and OpenGL visualization. This representation helps the expert to understand how the anatomy of the mouse brain is organized in space.

A. Data Acquisition

We extracted a volumetric sample of tissue within the visual cortex of a mouse, and performed the tissue fixation within resin. Then we sectioned the sample at a 45 nanometer-thickness, we stained each section for electron microscopy imaging. We acquired 1153 serial sections under a transmission electron microscope where the electron beam is projected onto a phosporus plate 1.5 m below the sample and a CCD camera captures a gray-level picture. The section is incrementally displaced along the X axis and the Y axis in order to cover the entire length and width of the section.

B. Data Size and Computation Time

The digitized section composed after mosaicking and alignment of the camera images has a length of 135 200 pixels and a width of 119 600 pixels. The total size for the uncompressed dataset is approximately 19 terabyte, which makes it the largest imaging dataset for a single volume. An array of 5-RAID disks was used to store the data and was connected through a SATA port onto a workstation with two 3.2 GHz quad-core processors with 32 GB memory. Each subvolume was 3250 pixel by 3250 pixel in-plane and 50 sections high. The overlap margin was 650 pixels in the $-$X, $+$X, $-$Y, and $+$Y directions, and the overlap in $-$Z and $+$Z was five sections. The total computation time for the entire 19 terabyte dataset was two days and 10 h.

C. Computation Speedup

To assess the speedup due to the parallel implementation, we performed an experiment on a subset of the dataset. We divided the length and the width of the dataset by 16, and extracted a subvolume with this reduced dimensions, i.e., 1153 sections of 8450 pixel by 7475 pixel images, around the center of the original dataset. The execution of the serial implementation took 1 h and 26 min, while the execution of the parallel implementation took 14 min, resulting in a 6.14 speedup factor. The segmentation results were identical. Given that our hardware had two quad-core processors, the maximum speedup factor would be 8. Since the imaging operations on the individual subvolumes are the most time-consuming, we speculate that our application could be considered as *embarrasingly parallel*. In this context, a speedup close to the total number of subvolumes (i.e., $46 \times 52 = 2392$) could be attained on a cluster with an equal number of processors.

V. Validation

We perform the following validation experiment to compare the accuracy of our segmentation results versus the tracing by an expert. First, we randomly select one cell nucleus within the volumetric dataset and extract a 3250 pixel $\times$ 3250 pixel subvolume S_d centered on this cell nucleus. Second, we extract a 3250 pixel $\times$ 3250 pixel subvolume S_s centered at the same position within the segmentation results, and we locate the largest connected component C_s. Then an expert manually contours the cell nucleus in every section of S_d, so that a volumetric component C_m can be reconstructed from the stack of contours. We denote $\bar{C}_s$, the pixels, within S_s that do not belong to C_s. Similarly, $\bar{C}_m$ is the set of pixels within S_m that do not belong to C_m. In this experiment, 49% of N, the total number of pixels within the subvolume, belong to C_m, thus $\#C_m \approx \#\bar{C}_m$. Finally, we compute normalized values for the true positives TP, true negatives TN, false positives FP, and false negatives FN as formulated in (1), (2), (3), and (4). An exact segmentation would lead to $TP = 100\%$, $TN = 100\%$, $FP = 0\%$, and $FN = 0\%$. Our results are $TP = 95\%$, $TN = 98\%$, $FP = 2\%$, and $FN = 5\%$.

$$TP = \#\{C_s \cap C_m\}/\#C_m \tag{1}$$

$$TN = \#\{\bar{C}_s \cap \bar{C}_m\}/\#\bar{C}_m \tag{2}$$

$$FP = \#\{C_s \cap \bar{C}_m\}/\#\bar{C}_m \tag{3}$$

$$FN = \#\{\bar{C}_s \cap C_m\}/\#C_m. \tag{4}$$

The high values for TP and TN demonstrate the accuracy of our segmentation results. The relative larger value for FN compared with FP indicate that portions of the nucleus were not included in the segmentation results. The overlay of the segmentation results with the original sections shows that dark areas close to the membrane of the nucleus are often mis-segmented. These dark areas are typically clusters of polyribosomes, as shown in Fig. 1.

We believe that the extension of our multiscale segmentation algorithm to smaller features, such as mylenated axons and dendrites, is possible. The segmentation step for the nuclei needs to produce a triplet $\langle i, j, k\rangle$, which will then be consumed by a segmentation step for the mylenated axons and dendrites. It is possible to refine the segmentation by using different image operations for mylenated axons and dendrites, and for basal dendrites and apical dendrites. The rationale is the same as explained in this paper: in every subvolume the segmentation step at the large scale is performed first and generates the control tag that will trigger further segmentation steps at finer scales.

VI. Conclusion

We have presented a solution for scaling-up the tracing of the connectome using automated segmentation and parallel computing. The idea is to apply robust image segmentation techniques in carefully-sized subvolumes and to merge the segmentation results as soon as overlapping subvolumes have been computed.

Our decision to use Intel CnC library has been critical to rapidly port our serial segmentation code into a parallel implementation. We have demonstrated that our implementation can segment the vasculature and the cell nuclei with high accuracy and without manual intervention. We believe that our approach could be extended beyond the automated tracing of a connectome and could solve other data-intensive applications where the analysis of large volumetric datasets requires an automated and accurate solution.

ACKNOWLEDGMENT

The authors thank W. C. Lee Allen, D. G. C. Hildebrand, H. S. Kim, S. Butterfield, and W. Tobin for valuable discussions regarding electron microscopy, and V. Bonin, M. L. Andermann, A. Kerlin, L. Glickfield, and S. Chatterjee for sharing their insights about the anatomy of the brain.

REFERENCES

[1] K. L. Briggman and W. Denk, "Towards neural circuit reconstruction with volume electron microscopy techniques," *Curr. Opin. Neurobiol.*, vol. 16, pp. 562–570, 2006.

[2] J. W. Lichtman and J. R. Sanes, "Ome sweet ome: What can the genome tell us about the connectome?" *Curr. Opin. Neurobiol.*, vol. 18, no. 3, pp. 346–53, Jun. 2008.

[3] J. Lu, J. C. Tapia, O. L. White, and J. W. Lichtman, "The interscutularis muscle connectome," *PLoS Biol.*, vol. 7, no. 2, p. e32, Feb. 10, 2009. [Erratum in: PLoS Biol. 2009 Apr;7(4):e1000108.]

[4] Y. Mishchenko, T. Hu, J. Spacek, J. Mendenhall, K. M. Harris, and D. B. Chklovskii, "Ultrastructural analysis of hippocampal neuropil from the connectomics perspective," *Neuron*, 2010.

[5] E. Jurrus, R. Whitaker, B. W. Jones, R. Marc, and T. Tasdizen, "An optimal-path approach for neural circuit reconstruction," in *Proc. IEEE ISBI*, 2008, pp. 1609–1612.

[6] J. Lu, J. C. Fiala, and J. W. Lichtman, "Semi-automated reconstruction of neural processes from large numbers of fluorescence images," *PLoS One*, vol. 4, no. 5, p. e5655, May 2009.

[7] Y. Mishchenko, "Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs," *J. Neurosci. Methods*, vol. 176, pp. 276–289, 2009.

[8] T. Hu and D. Chklovskii, "Reconstruction of sparse circuits using multi-neuronal excitation (RESCUME)," in *Proc. 23rd Annual Conference on Neural Information Processing Systems (NIPS) 2009*, Vancouver, B.C., Canada, Dec. 7–10, vol. 2, pp. 790–798.

[9] J. R. Anderson, B. W. Jones, J.-H. Yang, M. V. Shaw, C. B. Watt, P. Koshevoy, J. Spaltenstein, E. Jurrus, U. V. Kanan, R. T. Whitaker, D. Mastronarde, T. Tasdizen, and R. E. Marc1, "A computational framework for ultrastructural mapping of neural circuitry," *PLoS Biol.*, vol. 7, p. e1000074. 10.1371/journal.pbio.1000074, 2009.

[10] Y. Kubota, S. Hatada, and Y. Kawaguchi, "Important factors for the three-dimensional reconstruction of neuronal structures from serial ultrathin sections," *Front. Neural Circuits*, vol. 3, Article 4, Epub, May 29, 2009.

[11] W. Jeong, J. Schneider, S. G. Turney, B. E. Faulkner-Jones, D. Meyer, R. Westermann, R. C. Reid, J. Lichtman, and H. Pfister, "Interactive histology of large-scale biomedical image stacks," *IEEE Trans. Vis. Comput. Graph.*, vol. 16, no. 6, pp. 1386–1395, 2010.

[12] W. Jeong, J. Beyer, M. Hadwiger, R. Blue, C. Law, A. Vazquez, C. Reid, J. Lichtman, and H. Pfister, "SSECRETT and NeuroTrace: interactive visualization and analysis tools for large-scale neuroscience datasets," *IEEE Comput. Graph. Appl.*, vol. 30, pp. 58–70, 2010.

[13] S. C. Turaga, J. F. Murray, V. Jain, F. Roth, M. Helmstaedter, K. Briggman, W. Denk, and H. S. Seung, "Convolutional networks can learn to generate affinity graphs for image segmentation," *Neural Comput.*, vol. 22, pp. 511–538, 2010.

[14] R. Kumar, A. V. Reina, and H. Pfister, "Radon-like features and their application to connectomics," in *Proc.IEEE Comput. Soc. Workshop Math. Methods in Biomed. Image Anal. (MMBIA) 2010*.

[15] L. Xiao, X. Yuan, Z. Galbreath, and B. Roysam, "Automatic and reliable extraction of dendrite backbone from optical microscopy images," in *Proc. 2010 Int. Conf. Life Syst. Model Simul. Intell. Computing, and Proc. 2010 Int. Conf. Int. Comput. Sust. Energy and Enviro.: Part III (LSMS/ICSEE)*, Wuxi, China, Sep. 17–20, pp. 100–112.

[16] E. Bas and D. Erdogmus, "Piecewise linear cylinder models for 3-dimensional axon segmentation in brainbow imagery," in *Proc. 2010 IEEE Int. Symp. Biomed. Imag.: From Nano to Macro*, Apr. 14–17, Rotterdam, The Netherlands, pp. 1297–1300.

[17] G. Gonzalez, E. Turetken, F. Fleuret, and P. Fua, "Delineating trees in noisy 2D images and 3D image-stacks," in *Proc. IEEE Conf. Comput. Vision and Pattern Recog. (CVPR), 2010*, San Francisco, CA, Jun. 13–18, pp. 2799–2806.

[18] A. Stepanyants, L. M. Martinez, A. S. Ferecsko, and Z. F. Kisvrday, "The fractions of short- and long-range connections in the visual cortex," *PNAS*, vol. 106, no. 9, pp. 3555–3560, 2009.

[19] S. Jaensch, M. Decker, A. A. Hyman, and E. W. Myers, "Automated tracking and analysis of centrosomes in early caenorhabditis elegans embryos," *Bioinformatics*, vol. 26, no. 12, pp. i13–i20, 2010.

[20] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers, "V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets," *Nature Biotechnol.*, vol. 28, pp. 348–353, 2010.

[21] S. Lang, P. Drouvelis, E. Tafaj, P. Bastian, and B. Sakmann, "Fast extraction of neuron morphologies from large-scale SBFSEM image stacks," *J. Comput. Neurosci.*, DOI: 10.1007/s10827-011-0316-1.

[22] A. Akselrod-Ballin, D. Bock, R. C. Reid, and S. K. Warfield, "Accelerating feature based registration using the Johnson-Lindenstrauss Lemma," in *Proc. 12th Int. Conf. Med. Image Comput. Comput.-Assisted Intervention: Part I (MICCAI)*, London, U.K., Sep. 20–24, pp. 632–639, 2009.

[23] Concurrent Collections for C/C++ Application Programming Interface, [Online]. Available: http://software.intel.com/en-us/sites/whatif/runtime_api/index.html

[24] Intel Concurrent Collections for C/C++, [Online]. Available: http://softwarecommunity.intel.com/articles/eng/3862.htm

[25] Habanero Multicore Software Project, [Online]. Available: http://habanero.rice.edu

[26] (2010). Concurrent Collections in Habanero-Java (HJ), [Online]. Available: http://habanero.rice.edu/cnc-download

[27] M. Grossman, A. Simion, Z. Budimlic, and V. Sarkar, "CnC-CUDA: Declar-ative programming for GPUs," in *Proc. 23rd Int Workshop Languages and Compilers for Parallel Comput. (LCPC)*, Houston, TX, Oct. 7–9, 2010, pp. 230–245.

[28] R. Newton, F. Schlimbach, M. Hampton, and K. Knobe, "Capturing and composing parallel patterns with Intel CnC," in *Proc. 2nd USENIX Workshop Hot Topics Parallelism (HotPar'10)*, Berkeley, CA, Jun.2010.

[29] M. G. Burke, K. Knobe, R. Newton, and V. Sarkar, "The concurrent collections programming model," in *Encyclopedia of Parallel Computing*, D. Padua, Ed. New York: Springer-Verlag, 2011.

[30] Z. Budimlic, M. Burke, V. Cave, K. Knobe, G. P. Lowney, R. Newton, J. Palsberg, D. Peixotto, V. Sarkar, F. Schlimbach, and S. Tasirlar, "The CnC programming model," *J. Scientific Programming*, 2010.

[31] D.B. Chklovskii, S. Vitaladevuni, and L. K. Scheffer, "Semi-automated reconstruction of neural circuits using electron microscopy," *Curr. Opin. Neurobiol.*, vol. 20, no. 5, pp. 667–675, 2010.

[32] V. Jain, B. Bollmann, M. Richardson, D. R. Berger, M. N. Helmstaedter, K. L. Briggman, W. Denk, J. B. Bowden, J. M. Mendenhall, W. C. Abraham, K. M. Harris, N. Kasthuri, K. J. Hayworth, R. Schalek, J. C. Tapia, J. W. Lichtman, and H. S. Seung, "Boundary learning by optimization with topological constraints," in *Proc. IEEE Comput. Vision and Pattern Recog. (CVPR)*, San Francisco, CA, 2010.

[33] V. Jain, H. S. Seung, and S. C. Turaga, "Machines that learn to segment images: A crucial technology for connectomics," *Curr. Opin. Neurobiol.*, vol. 20, pp. 1–14, 2010.

[34] D. D. Bock, W.-C. A. Lee, A. M. Kerlin, M. L. Andermann, G. Hood, A. W. Wetzel, S. Yurgenson, E. R. Soucy, H. S. Kim, and R. C. Reid, "Network anatomy and in vivo physiology of visual cortical neurons," *Nature*, vol. 471, pp. 177–182, 10 Mar. 2011.

[35] K. L. Briggman, M. Helmstaedter, and W. Denk, "Wiring specificity in the direction-selectivity circuit of the retina," *Nature*, vol. 471, p. 183188, 10 Mar. 2011.

[36] M. Helmstaedter, K. L. Briggman, and W. Denk, "High-accuracy neurite reconstruction for high-throughput neuroanatomy," *Nature Neurosci.*, vol. 14, pp. 1081–1088, 2011.

[37] R. Chandra, L. Dagum, D. Kohr, D. Maydan, J. McDonald, and R. Menon, *Programming in OpenMP*. New York: Academic, 2001.