Knife-Edge Scanning Microscopy: High-throughput Imaging and Analysis of Massive Volumes of Biological Microstructures

August 31, 2008

Chapter Number	2
Chapter Title	Knife-Edge Scanning Microscopy: High-throughput Imaging and Analysis of Massive Volumes of Biological Microstructures
Author 1	Yoonsuck Choe
Author 1 - Address	Department of Computer Science, Texas A&M Univ. 3112 TAMU, College Station, TX, 77843-3112, USA
Author 1 - Email	choe@tamu.edu
Author 1 - Tel	+1 979 845 5466
Author 2	Louise C. Abbott
Author 2 - Address	Department of Veterinary Integrative Biosciences, Texas A&M Univ. 3112 TAMU, College Station, TX, 77843-3112, USA
Author 2 - Email	labbott@cvm.tamu.edu
Author 2 - Tel	+1 979 845 5466
Author 3	Donhyeop Han
Author 3 - Address	Same as author 1.
Author 3 - Email	dhan1@neo.tamu.edu
Author 3 - Tel	+1 979 845 5466
Author 4	Pei-San Huang
Author 4 - Address	Same as author 2.
Author 4 - Email	phuang@cvm.tamu.edu
Author 4 - Tel	+1 979 845 5466
Author 5	John Keyser
Author 5 - Address	Same as author 1.
Author 5 - Email	keyser@cs.tamu.edu
Author 5 - Tel	+1 979 845 5466
Author 6	Jaerock Kwon
Author 6 - Address	Same as author 1.
Author 6 - Email	jrkwon@tamu.edu
Author 6 - Tel	+1 979 845 5466
Author 7	David Mayerich
Author 7 - Address	Same as author 1.
Author 7 - Email	mayerichd@neo.tamu.edu
Author 7 - Tel	+1 979 845 5466
Author 8	Zeki Melek
Author 8 - Address	Same as author 1.
Author 8 - Email	z0m8905@cs.tamu.edu
Author 8 - Tel	+1 979 845 5466
Author 9	Bruce H. McCormick
Author 9 - Address	Same as author 1.
Author 9 - Email	mccormic@cs.tamu.edu
Author 9 - Tel	+1 979 845 5466
Corresponding Author	Author 1

Chapter 2

Knife-Edge Scanning Microscopy: High-throughput Imaging and Analysis of Massive Volumes of Biological Microstructures

Recent advances in physical-sectioning microscopy have enabled high-throughput imaging of massive volumes of biological microstructure at a very high resolution. The Knife-Edge Scanning Microscope (KESM) we have developed is one of the few that combines serial sectioning and imaging in an integrated process. The KESM is capable of imaging biological tissue (about 1 cm^3) at 300 nm $\times$ 300 nm $\times$ 500 nm resolution within 100 hours, generating data at a rate of 180 MB/s. The resulting data per organ (e.g., a mouse brain) can easily exceed tens of terabytes. High-performance computing methods are required at every stage in the lifetime of the generated data set: (1) distributed storage and retrieval, (2) image processing to remove noise and cutting artifacts, (3) image and texture segmentation, (4) three-dimensional tracking and reconstruction of microstructures, and (5) interactive visualization. In this chapter, we will review the capabilities and latest results from the KESM (Section 2.2); and discuss the computational challenges arising from the massive amounts of data, along with a survey of our on-going efforts to address these challenges (Section 2.3–2.5). We expect the integration of high-throughput imaging and high-performance computing to lead to major break-throughs in scientific discovery in biological sciences.

2.1 Background

In this section, we will provide a brief review of high-throughput imaging and analysis methods, to provide a proper context for the work we will discuss in the remainder of the chapter.

2.1.1 High-Throughput, Physical-Sectioning Imaging

Currently, the standard approach for microscopic imaging of a volume of tissue is confocal microscopy [1]. The basic idea is to change the depth of focus (focal plane), and use a pinhole aperture to detect photons originating only from the target depth. This is called "optical sectioning" where virtual, not actual, sections are obtained. In conventional optical sectioning, the main limiting factor is not in the resolution along the x-y plane ($\sim$ 250 nm) but in that of the z direction ($\sim$ 700 nm) [2]. Although the resolution and imaging depth can be improved using more advanced schemes such as multi-photon microscopy [3], optical sectioning techniques are limited to the tissue surface and have limited z-axis resolution. Slow imaging speed is another issue, for both confocal and multi-photon, such that even in enhanced two-photon microscopy, the data rate is less than 8 MB/s (512×484 at 30 frames/s reported in [4], and about 1 frame/s in [3]).

Physical sectioning combined with microscopy is one alternative to overcome the above issues, since z-axis resolution depends only on how thin the tissue can be sectioned (it can go down to ~ 30 nm using a vibrating microtome), and there is virtually no depth limit on the tissue thickness.

The five most notable approaches in this direction are listed below:

1. All-Optical Histology [5].
2. Knife-Edge Scanning Microscopy (KESM) [6–10]
3. Array Tomography [11]
4. Serial-Block-Face Scanning Electron Microscopy (SBF-SEM) [12]
5. Automatic Tape-Collecting Lathe Ultramicrotome (ATLUM) [13]

All-Optical Histology

There are some efforts to eliminate the depth limit in optical sectioning microscopy. For example, All-Optical Histology combines multi-photon microscopy with tissue ablation to allow deeper imaging [5]. Imaging is performed using standard

optical sectioning. Next, femtosecond laser pulses are used to ablate $\sim$150 µm-thick sections on the top of the tissue block, exposing new tissue to be imaged. The main advantage of All-Optical Histology is that it overcomes the tissue thickness limit in confocal and multi-photon microscopy by ablating thick chunks of tissue. However, since multi-photon microscope is used for imaging, it suffers from the same vertical resolution limit and slow imaging speed.

Knife-Edge Scanning Microscopy

The Knife-Edge Scanning Microscope (KESM, US patent #6,744,572) has been designed at Texas A&M University (TAMU) in recent years [6–10]. The instrument, shown in Fig. 2.1, is capable of scanning a complete mouse brain ($\sim$310 mm^3) at 300 nm sampling resolution within 100 hours when scanning in full production mode. The basic idea is to simultaneously cut and image thin sections of embedded biological tissue. We will discuss KESM in more technical detail in the following section (Section 2.2).

Array Tomography

Array Tomography uses an ultramicrotome to manually section embedded tissue [11]. Adhesives on the edge of the embedded tissue block allow successive sections to stick to each other. As sequential sections are cut, this forms an array of ultrathin sections. The resulting tissue array is placed on a glass slide and can be repeatedly washed, stained, and imaged using fluorescence microscopes and scanning electron microscopes. Volume data are obtained by combining together the imaged sections. The main advantages of Array Tomography is that it enables the imaging of multiple molecular markers from the same slide, thus providing perfect registration across different data modalities. However, there are some limitations to this approach, such as limited imaging volume (order of magnitude less than KESM in linear dimension) and semi-manual operation.

Serial Block-Face Scanning Electron Microscopy

Serial Block-Face Scanning Electron Microscopy (SBF-SEM), is capable of imaging tissue volumes of approximately 500 µm in linear dimension at a resolution on the order of 10 nm $\times$ 10 nm $\times$ 50 nm (exact values range from 12.7 nm to 13.4 nm for x and y; and 55 nm to 63 nm for z) [12]. The main advantage of SBF-SEM is its extremely high resolution, where detailed cellular and subcellular

ultrastructure can be imaged. However, the volume is limited to cubes of several hundred μm, and scanning time can be prohibitive if sufficient signal-to-noise is to be achieved (a 200 μm cube at the above resolution can take up to one year). Improved staining methods to increase contrast can (theoretically) improve the speed 400-fold. SBF-SEM development is in a mature stage, where production models are now available from a commercial vendor (GATAN, Inc.).

Automatic Tape-Collecting Lathe Ultramicrotome

Automatic Tape-Collecting Lathe Ultramicrotome (ATLUM) is the latest development in serial sectioning microscopy [13]. ATLUM is capable of collecting a continuous ribbon sized 1 mm $\times$ 5 mm $\times$ 50 nm that get automatically pasted on a continuously running tape. The tape is then cut and organized into a Ultrathin Section Library. The typical volume ATLUM can handle is about 10 mm^3, which is roughly a cube with a 2.15 mm linear dimension. One distinction in ATLUM is that imaging (typically with SEM) is done only when needed, thus a digitized library is not immediately available (according to the developers of ATLUM, imaging such a volume would take hundreds of years). Thus, a more reasonable approach is to have the Ultrathin Section Library containing the chips to serve as the database itself. In the future, robotic random access SEM imaging will be used to do on-demand imaging of objects of interest.

Summary and Comparison

Even though the methods above are unified under the common theme of physical sectioning, the resolution and the typical volume they can handle all differ, putting them in a relatively complementary role with respect to each other. In other words, these methods cannot be ranked on an absolute scale since there are relative advantages and disadvantages to each method. Table 2.1 provides a summary comparison of these microscopy methods.

Finally, we wish to emphasize that no matter what physical volume these methods deal with (ranging from 100^3 μm^3 up to 1 cm^3), the resulting volume data can exceed several TBs. For example, KESM routinely generates 2 TB, and it is projected that it can generate over 20 TB with a higher resolution objective and thinner sections. Such volume of data poses serious challenges to computational analysis, and high-performance computing methods could help address these challenges.

Table 2.1: **Summary Comparison.**

Method	Resol. (x&y)	Resol. (z)	Volume	Modality	Time
All-Optical Hist.	0.5 µm	1 µm	1 cm^3	Fluorescence	∼900 hours
KESM	0.3–0.6 µm	0.5–1 µm	1 cm^3	Bright field, Fluorescence*	∼100 hours
Array Tomography	∼0.2 µm	0.05–0.2 µm	$\sim 100^3$ μm^3	Fluorescence, EM†	N/A
SBF-SEM	∼0.01 µm	∼0.03 µm	$\sim 500^3$ μm^3	EM	N/A
ATLUM	∼0.01 µm	0.05 µm	$\sim 2.15^3$ mm^3	EM	N/A

*Expected in the near future. † EM: Electron Microscopy.

2.1.2 Volumetric Data Analysis Methods

Once the volume data are obtained from physical sectioning microscopy, the next task is to extract objects of interest from the data set. This is a non-trivial task due to distortions, noise, and artifacts resulting from the cutting process (often due to vibrations known as *chatter*). Furthermore, the difficulty in automation for this kind of data is dramatically increased by the density and the huge number of objects in the microscopy data. This is unlike volume data from medical imaging (magnetic resonance [MR] imaging or computer-aided tomography [CT]) where object count and density are both low, for which various automated methods exist (e.g., National Library of Medicine's Insight Toolkit, `http://www.itk.org/`). Note that medical imaging instumentation is also makeing great advances, thus rendering existing algorithms insufficient (micro-CT, diffusion tensor imaging, etc.).

In this section, we will review three main approaches taken in 3D reconstruction.

Image registration, segmentation, and reconstruction

A traditional approach for 3D reconstruction is to consider the volume data as a stack of images, while processing one image at a time. This is often necessary because the z-axis resolution is so much lower than the lateral resolution for most imaging modalities. First, depending on the imaging modality, alignment (registration) of successive images in the image stack constituting the volume data may be necessary. Next, the foreground objects and the background need to be segmented. Finally, the segmented objects in individual images need to be stitched together (reconstruction) to form a geometric representation of objects such as neurons and vasculatures.

The process is often times manual, and computer algorithms only play a supportive role. For example, see the RECONSTRUCT package [14], and Neuron_Morpho [15]. There are some efforts to automate the entire process [16]. One disadvantage of the segment-then-reconstruct approach is that it is hard to incorporate the 3D context in the process when there is ambiguity (foreground or background? connect or not?) since the approach operates primarily in 2D. Finally, manual algorithms are far too time-consuming for the dense, high-frequency structures found in high-throughput microscopy.

Convolutional neural networks

Jain et al. recently developed an alternative approach for 3D reconstruction, using convolutional networks [17]. Convolutional networks is a special type of artificial neural network that can be trained using supervised learning algorithms such as backpropagation [18]. The input is a cubic volume of data, and the output is the segmentation of the input volume. With an $n \times n \times n$ voxel volume, connecting to another $n \times n \times n$ voxel volume in the hidden layers, the number of connection weights (tunable parameters) can become prohibitive (for full connectivity, we need n^6). Convolutional networks avoid such an issue by *sharing* the connection weights. So, for one connection bundle connecting two $n \times n \times n$ voxel volumes only n^3 connection weights would be needed, instead of n^6. In some sense, these n^3 connections work as a filter, performing a convolution operation in 3D. Thus, the hierarchy in the neural network works as a series of 3D convolutions and combinations of the results.

For the actual training, small cubic samples from the raw data volume are plugged into the input, while manually labeled target values are plugged into the output. A gradient-based backpropagation algorithm is the used to adjust the connection weights. Thus, the learning process is basically converging to the most effective 3D convolution kernels. One disadvantage of this approach is that, even though the convolution strategy drastically reduces the number of tunable parameters, the training can be very slow, often lasting for weeks on modern parallel computers. The slowness is partly due to the computationally demanding 3D convolution operation, and partly to the gradient-descent learning algorithm. The advantage of the convolutional network approach is that fairly high levels of accuracy can be reached based on a small amount of manually labeled ground truth.

Vector tracking

The reconstruction methods discussed above (also see [19] for a review of traditional methods such as such as voxel-based skeletonization) have their own advantages, but in many cases they are not efficient enough computationally to be applied to large volume data sets, especially those with fiber-like structures (see [20] for a review). Part of the problem is that each and every voxel has to be visited (multiple times) during processing. One way to overcome this problem is to visit only a subset of relevant voxels.

One of the latest approaches addressing the above issues is the vector tracking algorithm [20, 21], where only local information around the object of interest is examined, thus avoiding visiting every voxel in the volume data set. The basic idea is to begin with a seed point, estimate the trajectory of a fiber by means of template matching. Subsequently, small steps are taken along the fiber, continually adjusting the current position based on the estimated trajectory. Our group at Texas A&M generalized the approach for more robust tracking [22], which we will describe in detail in Sec. 2.3–2.4 (also see [23], which uses level sets).

2.2 Knife-Edge Scanning Microscopy

The instrument comprises four major subsystems: (1) precision positioning stage (Aerotech), (2) microscope/knife assembly (Micro Star Technology), (3) image capture system (Dalsa), and (4) cluster computer (Dell). The specimen, a whole mouse brain, is embedded in a plastic block and mounted atop a three-axis precision stage. A custom diamond knife, rigidly mounted to a massive granite bridge overhanging the three-axis stage, cuts consecutive thin serial sections from the block. Unlike block face scanning, the KESM concurrently cuts and images (under water) the tissue ribbon as it advances over the leading edge of the diamond knife. A white light source illuminates the rear of the diamond knife, providing illumination at the leading edge of the diamond knife with a strip of intense illumination reflected from the beveled knife-edge, as illustrated in Fig. 2.2. Thus, the diamond knife performs two distinct functions: as an optical prism in the collimation system, and as the tool for physically cutting thin serial sections. The microscope objective, aligned perpendicular to the top facet of the knife, images the transmitted light. A high-sensitivity line-scan camera repeatedly samples the newly cut thin section at the knife-edge, prior to subsequent major deformation of the tissue ribbon after imaging. The imaged stripe is a 20 µm-wide band locate

at the bevel at the very tip of the diamond knife, spanning the entire width of the knife. Finally, the digital video signal is passed through image acquisition boards and stored for subsequent analysis in a small dedicated computing server. The current server is a dual processor PC (3.2 GHz/2MB Cache, Xeon) with 6 GB of memory, built-in 1 TB storage, connected to an archival RAID attachment. The process of sectioning and imaging is fully automated with minimal human intervention. Fig. 2.3 shows a screen-shot of the stage controller/imaging application developed and maintained in our lab.

A quick calculation puts us in context, regarding the massiveness of the data that KESM can produce. Consider the acquisition of volume data representing a plastic-embedded mouse brain (15 mm Anterior-Posterior, 12 mm Medial-Lateral, 6 mm Dorsal-Ventral). A 40X objective has a field of view (knife width) of 0.625 mm. Sixteen strips (each 0.625 mm wide by 15 mm long) are cut for each z-axis section (like plowing a field). For a (z-axis) block height of 6 mm, 12,000 sections must be cut, each 0.5 μm thick. The integrated tissue ribbon length (15 mm/strip $\times$ 16 strips/section $\times$ 12,000 sections/mouse brain) is 2.9 km. The tissue ribbon is line-sampled at 300 nm resolution, near the Nyquist rate for an ideal optical resolution of (0.77)(532 nm)=(0.80 NA) = 512 nm. Based on this, the total data size (assuming one byte per voxel) comes out to 20 terabytes (TB) (at half the resolution in each dimension, it would be $\sim$2.5 TB). The tissue ribbon can be sampled at 11 mm/s by line sampling at 44 kHz (180 MB/s), the camera maximum (Dalsa CT-F3-4096 pixels). Sampling the 2.9 km tissue ribbon requires 265,000 s = 73 hr. Because mice brains are not cubical, stage return takes time, etc., we add 50% overhead, resulting in $\sim$100 hr.

Figs. 2.4 and 2.5 show typical data that can be obtained using the KESM [9]. Nissl staining dyes the RNA in the cytoplasm of all neurons and the DNA in cell bodies in all cells. However, the dendritic arbors and axons remain unstained. Thus, Nissl staining allows us to reconstruct the distribution of all cell bodies in the mouse brain, and in particular their distribution within the six layers of the cerebral cortex. Golgi staining, in contrast, reveals the entire structure of neurons, as it stains just 1% of the neurons in the tissue. Individual neurons can be seen clearly, permitting reconstruction. India ink enables high-contrast staining of the entire vascular network. Fig. 2.6 shows a rendering of the data volume using Amira [24].

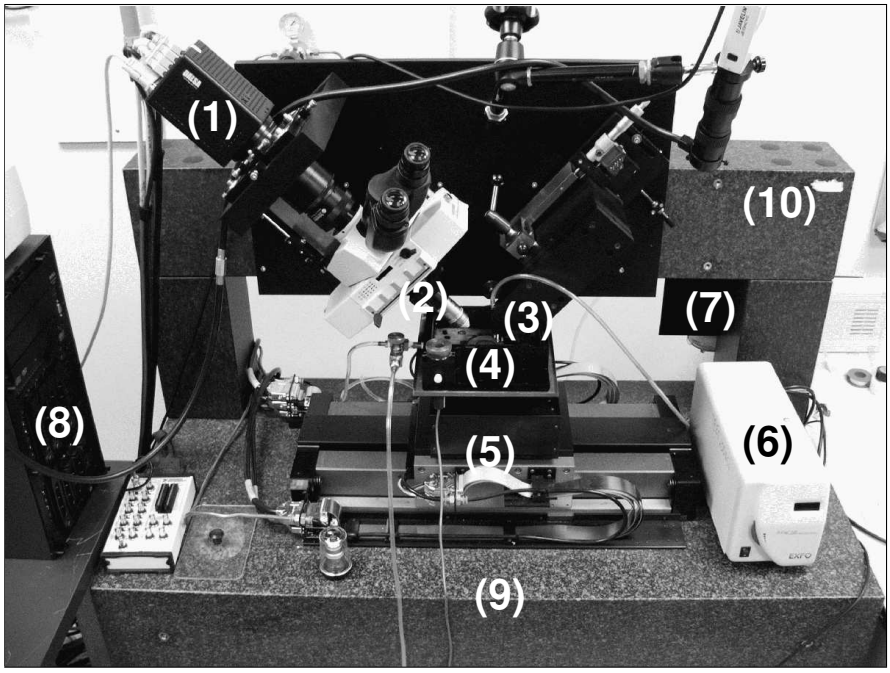
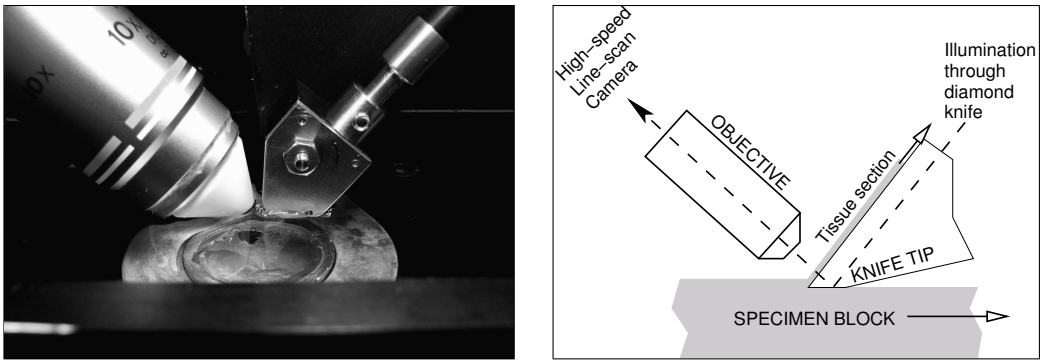


Figure 2.1: **The Knife-Edge Scanning Microscope (KESM).** A photo of the KESM is shown with its major components marked: (1) high-speed line-scan camera, (2) microscope objective, (3) diamond knife assembly and light collimator, (4) specimen tank (for water immersion imaging), (5) three-axis precision air-bearing stage, (6) white-light microscope illuminator, (7) water pump (in the back) for the removal of sectioned tissue, (8) PC server for stage control and image acquisition, (9) granite base, and (10) granite bridge.



(a) (b)

Figure 2.2: **Tissue Sectioning and Imaging in KESM.** (a) A close-up of the parts 2, 3, and 4 in Figure 2.1 is shown. To the left is the microscope objective, and to the right the diamond knife and light collimator. Submerged under water in the center is the plastic-embedded brain tissue held in a specimen ring. (b) The principal of operation of KESM is illustrated. The objective and the knife is held in place, while the specimen affixed on the positioning stage moves (arrow with solid line) at the resolution of 20 nm and travel speed of 1–5, and gets scraped against the diamond knife (5 mm wide for 10X objective), generating a thin section flowing over the knife (arrow with solid line). Line-scan imaging is done near the very tip of the knife where the distortion is minimal (maximum 106 μm from the tip of the knife). Illumination if provided through the diamond knife (arrow with dashed line indicates the light path). Note that the size of the knife is exaggerated. Adapted from [9].

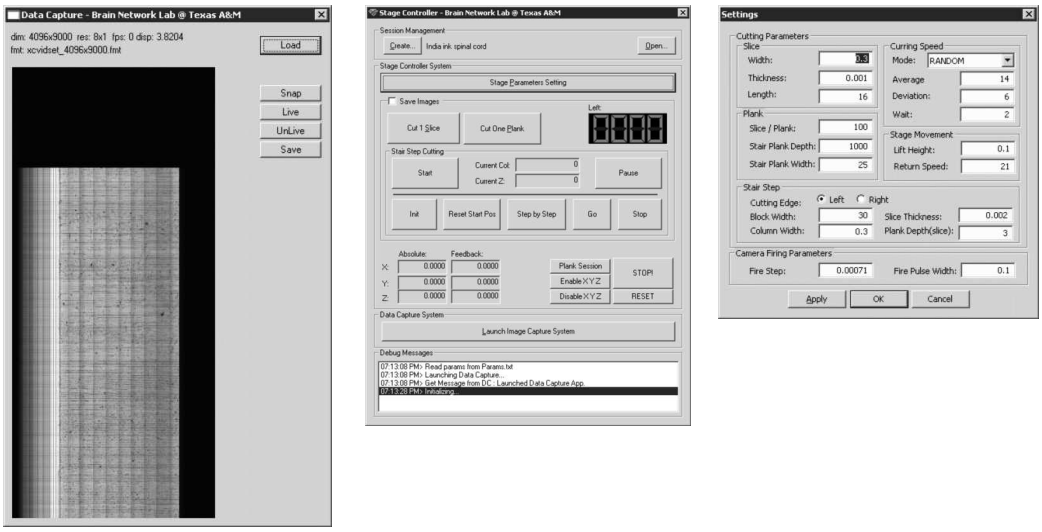


Figure 2.3: **KESM Automation Software.** A screenshot of the automated stage control and image capture application is shown.

2.3 Tracing in 2D

As we have seen in the previous section, the amount of data generated by high-throughput microscopy instruments is staggering. Simply iterating through the entire data set can take a long time. Thus, traditional image processing and 3D reconstruction algorithms are not suitable for this kind of data, since they require intense computation on every pixel/voxel in the data volume. Can et al. [21] and Haris et al. [27] developed template-based methods to overcome this issue.

In this section, we will present an extended template-based 2D tracking algorithm that can track fiber-like structures in biological volume data, such as neuronal processes or vascular networks. The basic idea is shown in Fig. 2.7 and 2.8. A moving window template is generated around a seed point (the size of the window is proportional to the fiber width), and the pixels along the circumference of the moving window are sampled to construct an intensity profile. The intensity profile is convolved with a Gaussian filter, and based on this *fiber cross-sections* (FCSs) are identified, where the border of the window overlaps the underlying object (Fig. 2.7). The current position is then moved to the center of the FCS, and a Cubic Tangential Trace Spline (CTTS) is used to interpolate between the current and the next seed point (or center point) (Fig. 2.8, left). CTTS is an interpolation method we developed by combining Lagrange interpolation [28] and

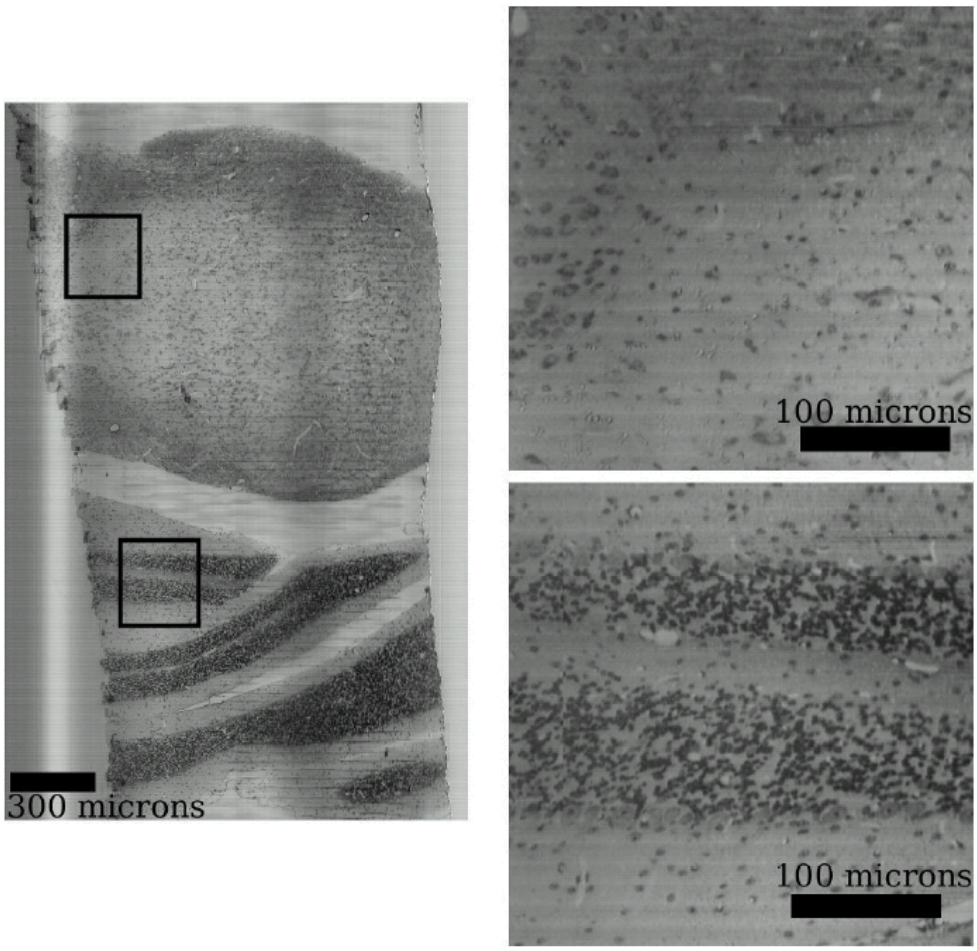


Figure 2.4: **Nissl Data from KESM.** Coronal section of mouse brain stem is shown, with part of the cerebellum visible to the bottom half. Close-up of the inserts are shown to the right. The pixel resolution of the images is 0.6 μm/pixel, with a section thickness of 1 μm. Adapted from [9].

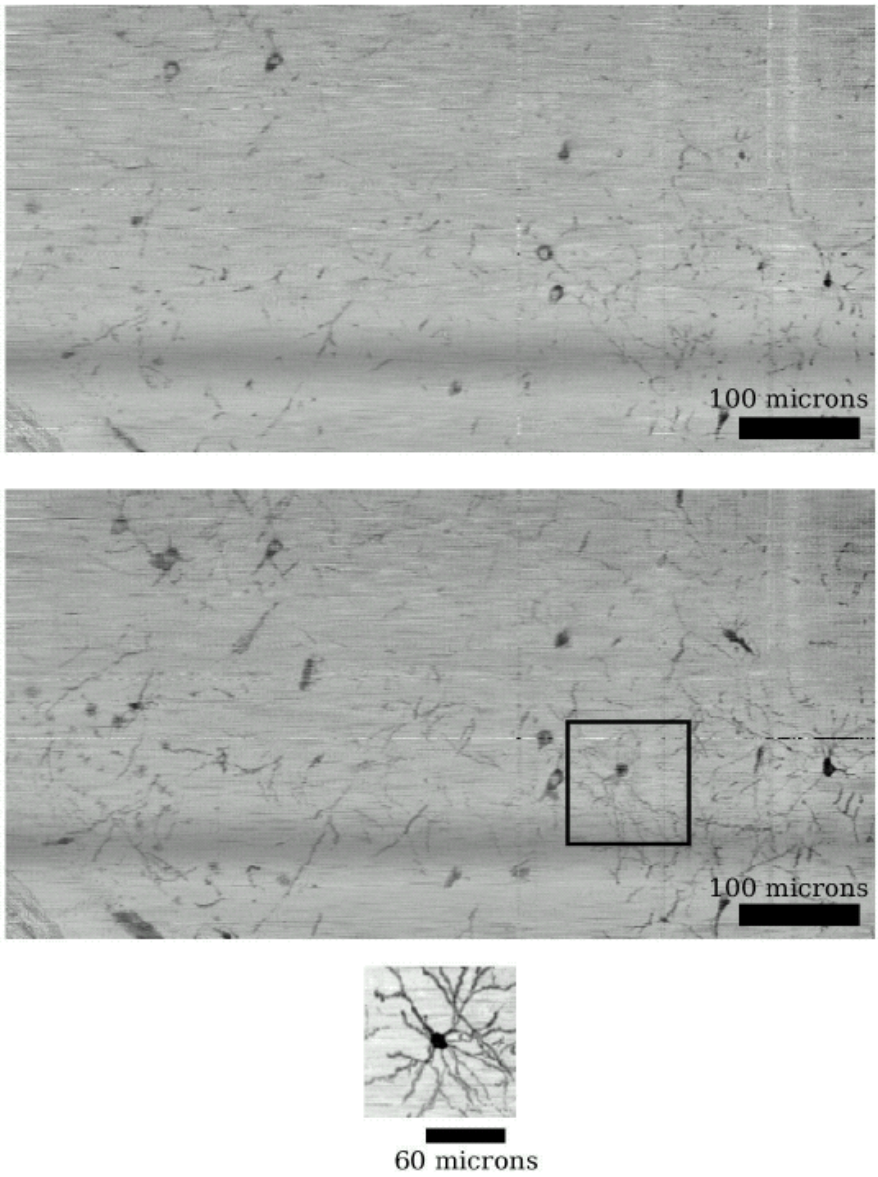
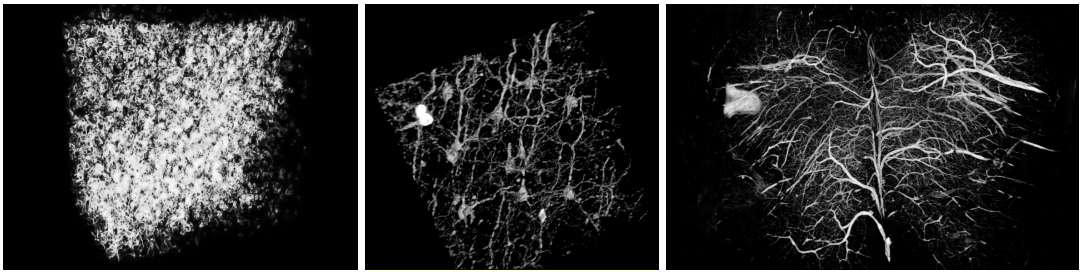


Figure 2.5: **Golgi Data from KESM.** The stack of images generated by KESM can be viewed from the side of the stack (resectioning). A single resectioned plane is shown at the top. Golgi stain results in sparse data, so often times it is easier to see familiar structures by overlaying multiple planes (middle). A single neuron can be observed when approximately 300 planes are overlayed (bottom). Adapted from [9].



(a) Nissl (b) Golgi (c) India Ink

Figure 2.6: **Volume Visualization of KESM Data.** Volume visualizations of KESM data using Amira [24] are shown. (a) Nissl-stained mouse cerebral cortex ($\sim 300^3$ μm^3 cube). (b) Golgi-stained mouse cerebral cortex ($\sim 300^3$ μm^3 cube). (c) India-ink-stained vasculature in mouse spinal cord (1.8 mm $\times$ 1.8 mm $\times$ 1.2 mm, adapted from [25, 26]). Voxel size $\sim$ 0.6 μm $\times$ 0.7 μm $\times$ 1 μm.

B-splines [29], for fast and accurate interpolation. When there is a branch, linear interpolation is used instead of the CTTS (Fig. 2.8, right). To make sure that the interpolation line is over actual data, the pixels directly below the spline are checked. All the other pixels in the moving window are safely ignored, greatly reducing the amount of data to be inspected.

Our algorithm (MW-CTTS, for moving window with CTTS) improves upon the previous approaches [21, 27] by adding robustness to noise, improved branch handling, and longer tracked distance from a single seed point. We quantitatively measured the noise-robustness of our algorithm using synthetic data with added noise. Two types of fibers were generated (line-type and curve-type), with varying fiber widths (20 to 50 pixels). We randomly generated 2,400 such synthetic inputs and ran our algorithm against Can et al.'s and Harris et al.'s. Fig. 2.9 shows the results. Our method was found to be more robust compared to these two methods. We also ran our algorithm against the two algorithms above on a real vascular data set (Fig. 2.10(a)). Can et al.'s method cannot handle branches (Fig. 2.10(b)), and Haris et al's can show limited coverage due to incorrect branch handling (Fig. 2.10(c)). On the other hand, our algorithm shows the longest tracked distance, coupled with accurate traces (Fig. 2.10(d)). Our algorithm is also fast compared to the two other methods. Tab. 2.2 summarizes the total distance tracked and the speed of tracking.

The algorithm described above is efficient, requiring $O(k)$ complexity (in the number of pixels examined), where k is the length of the tracked fiber. The number of pixels needed to be inspected depends on (1) the moving window's side length,

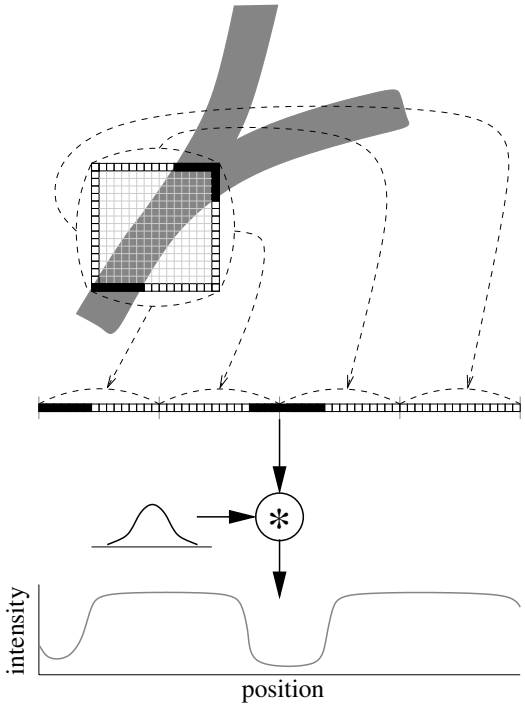


Figure 2.7: **Identifying Cross-Section of Fiber Object and the Moving Window.** An illustration of the method used to identify fiber segments (the "Y"-shaped object in the background) that overlap with the circumference of the moving-window template is shown (the blacksolid pixels). First, all the pixels are taken from the moving window's boundary, and convolution is done with a Gaussian filter to obtain an intensity profile (bottom). The valleys in the intensity profile are identified as fiber cross sections. This process helps deal with inherent noise in biomedical images. Note that the actual algorithm extracts a "band" of pixels from the window border, and apply a small 2D Gaussian filter to extract the intensity profile.

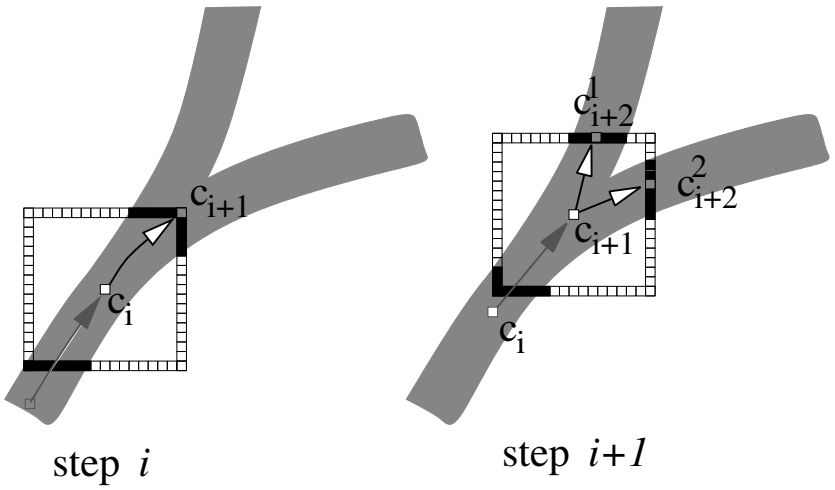
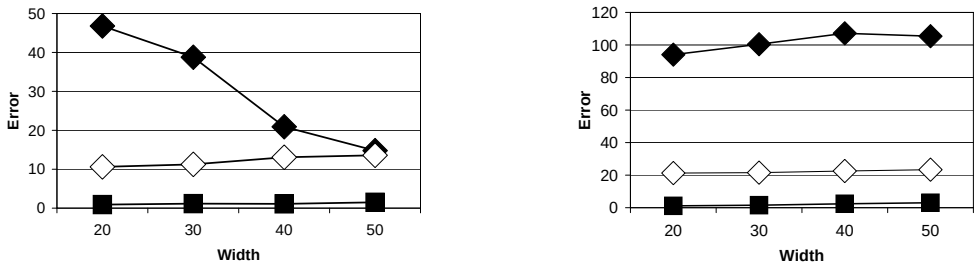


Figure 2.8: **Fiber Tracing with Moving Window Templates.** An illustration of the fiber tracing algorithm is shown. (Left) Given an initial seed point (or moving window center from the previous step) c_i, first we construct a moving window of an appropriate size and identify fiber cross-sections (FCS) in the advancing direction. By taking the center of the identified FCS, we can obtain the next center point c_{i+1} (gray pixel). In this example, there is no branch within the moving window. Once the target is found, a cubic tangential trace spline (CTTS) is used to connect the two center points c_i and c_{i+1}, and pixels underneath the interpolation line are checked to certify that the interpolation is over an existing fiber object. (Right) Starting from the moving window center estimated from the previous step (c_{i+1}), another moving window is constructed and the candidate FCSs identified. In this case we have two candidates, thus we generate two center points for the continued trace (c^1_{i+2} and c^2_{i+2}, marked as gray pixels). For cases like this including branches, we used linear inerpolation. Note that the full grid is not shown, to avoid clutter.



(a) Line-type input (b) Curve-type input

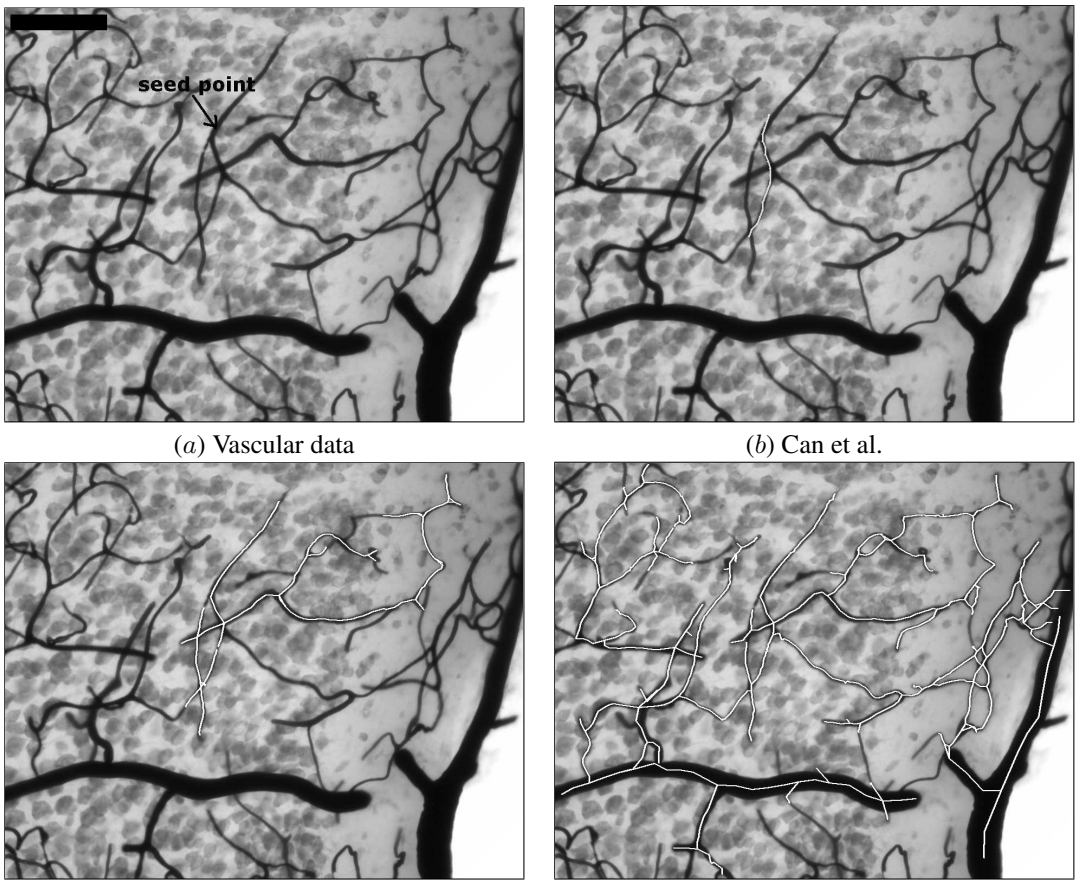
Figure 2.9: **Performance of MW-CTTS Tracing Algorithm on Synthetic Data.** The average performance of our MW-CTTS tracing algorithm on synthetic data (linear or curvy objects) are shown (squares), compared with that of Can et al. [21] (solid diamonds) and Haris et al. [27] (open diamonds). In both cases, our algorithm showed superior performance (near zero error).

(2) the fiber width, (3) the size of the Gaussian filter, and (4) total number of moving windows. With the fiber width n, moving window side length of $2n\epsilon$ (ϵ is a small parameter between 1.1 and 1.5), a Gaussian filter size of 5×5, and the total number of moving windows of $\frac{k}{2n\epsilon}$, the number of pixels come out to $(2n\epsilon \times 4) \times (5 \times 5) \times \frac{k}{2n\epsilon} = 100k$, i.e., $O(k)$. This calculation shows that our algorithm can scale up quite well to large volumes of data. It also helps that vascular and neural data are very sparse ($< 6\%$ for vascular and $<$ 1–2% for neural data).

In the next section, we will show how a similar approach can be extended into 3D.

2.4 Tracing in 3D

In this section, we discuss the extension of vector tracking techniques to three-dimensional data sets like those found in high-throughput microscopy. In general, 2D techniques have limited use since a single section contains a very limited amount of data. One method of dealing with this would be to combine several sections together, creating images similar to those in Fig. 2.10. Two-dimensional tracking methods can then be used to trace filament data. However, this method



(a) Vascular data (b) Can et al.

(c) Haris et al. (d) MW-CTTS

Figure 2.10: **Tracing Results Using MW-CTTS Compared to Other Algorithms.** Tracing results on a vascular image is shown for three different algorithms (scale bar = 20 µm, mouse cortex with vasculature stained with India ink and cell nuclei stained with Nissl, imaged using conventional light microscope). All three algorithms were initiated with the same single seed point in (a). Our MW-CTTS algorithm exhibits the most extensive trace.

	Can et al.		Haris et al.		MW-CTTS	
	total dist.	μsec/dist.	total dist.	μsec/dist.	total dist.	μsec/dist.
#1	68 px	50.21177	831 px	8.791092	936 px	6.705957
#2	25 px	100.6608	1826 px	16.84223	1912 px	12.57990
#3	76 px	85.74671	1473 px	22.86413	7845 px	9.974812
#4	272 px	14.07918	2190 px	12.16080	3712 px	11.60712
#5	196 px	16.77046	1461 px	14.55146	4045 px	10.91478
#6	216 px	20.83727	1449 px	16.10389	7731 px	11.80340

Table 2.2: Performance comparison using one seed point on 6 mouse cerebellum data. The μm/pixel is 1.1. We performed the trace on a PC with Intel Pentium 4 (2.4 GHz) processor, 512MB of memory under Windows XP operating system. C/C++ programming language and OpenGL library were used. Note that MW-CTTS always has the largest number of traced distance within real-time.

causes a loss in resolution along the projected axis.

Instead, we employ tracking techniques that work completely in three dimensions. One of the most basic techniques for segmentation in 2D and 3D data sets is template matching [30, 31]. In this case, a template is created that has a structure similar to the objects to be segmented. We then run the template across the image (or volume), centering the template at each pixel (or voxel). We then compare the local region with the template. Template matching has many advantages, the most important of which is that it is robust in the presence of noise. In addition, the operation is highly parallel and different regions can be tested for correlation to the template simultaneously.

Although this works well for structures that have a fixed size and are rotationally invariant, segmentation of varying structures, such as blood vessels, can be time consuming. This is because the template must also be sampled at different sizes and orientations, turning what was a 3D spatial problem into a seven-dimensional problem; three spatial, three rotational, and one in scale-space. This is particularly problematic when dealing with large data sets. Additional problems arise when trying to discretize these other dimensions. While the spatial dimensions of the image are already discrete, only a finite number of template sizes and orientations can be tested.

We solve this problem by using the vector tracking methods described in Sec. 2.3 along with a heuristic-based approach to reduce the number of samples re-

quired to identify a vascular or neuronal filament. By starting from an initial seed point, we predict the path of the filament by taking samples of the cross section along several directions (Fig. 2.11). These samples are then compared with a template. The sample that provides the best response is selected as the trajectory of the filament. We then take a small step along this direction, at which point we take another series of samples (Fig. 2.12). Every step on the way, we need to make small adjustments such as axis correction and estimation of the fiber radius. These techniques can be used to limit sampling to the filament surface [32] and cross section [33].

This kind of tracking allows us to reduce the required number of samples in several ways:

- Similar to vector tracking, samples are only taken local to the filament.
- By understanding the curvature of the filament, we can sample a limited number of orientations based on the orientation used in the previous step.
- Since filaments gradually change size, we can limit the number of samples in the scale dimension to sizes local to the previous step.

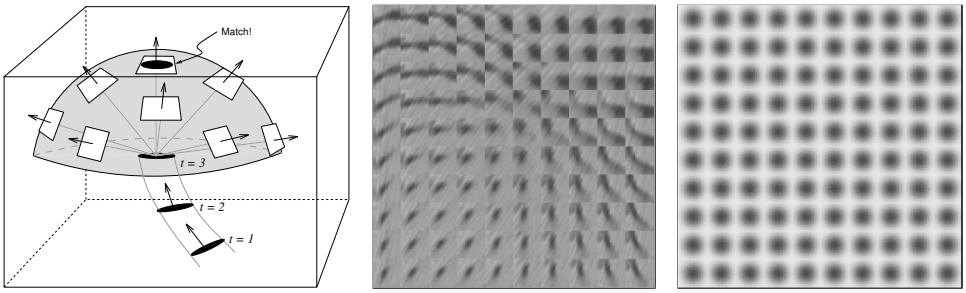
Finally, we note that evaluating each sample is a highly parallel operation that maps well to modern graphics processing units (GPUs). By implementing these tracking methods completely on commodity GPUs, we are able to achieve even greater speedup over standard template matching schemes (Fig. 2.14). See [33] for details.

2.5 Interactive Visualization

Both the size and density of structures in high-throughput data sets poses several problems in modeling and visualization. In this section, we discuss interactive visualization techniques that we use to explore dense bundles of filaments in volumetric networks such as neuronal trees and microvascular networks.

Direct volume visualization and isosurface construction are standard methods used to visualize scalar volume data sets like those created using KESM. Unfortunately, these methods require that a large data set or surface be stored in memory during visualization, which is often impossible on the limited memory of current graphics cards.

Streamline and stream tube methods [34] provide a means of limiting the amount of active data by storing only the simplest representation of the filament



(a) Template matching (b) Tangential slices (c) Templates

Figure 2.11: **Fast Fiber Tracing.** The basic idea behind our fast fiber tracing algorithm is shown. (a) Starting from a seed point ($t = 1$), the direction of the fiber is estimated ($t = 2$ to $t = 3$). The next point in the fiber trajectory is estimated using a template-based approach. Images are formed as interpolated slices (computed via graphics hardware) through the data volume—sampling in several directions around the current point. (Note that the distance between trace points is exaggerated.) (b) A typical array of slices from the tangential planes are shown. Only a small number of these resemble a circular cross-section of a fiber (upper-right corner). The best matching one is selected as the next point in the fiber trajectory. (c) An array of templates are shown. Using the graphics processing unit (GPU), comparison of b and c can be done extremely fast, through the texture rendering logic. Adapted from [26].

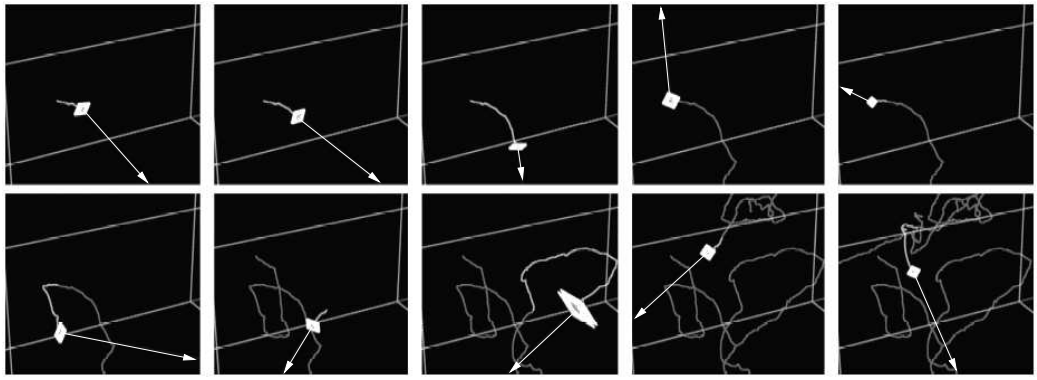


Figure 2.12: **Tracing Process.** An example of using the tracing algorithm to track a sequence of fibers is shown. The small squares show the current trace center point, and the white arrows indicate the prediction direction. Gray trailing traces show the fibers that have been traced so far. Adapted from [26].

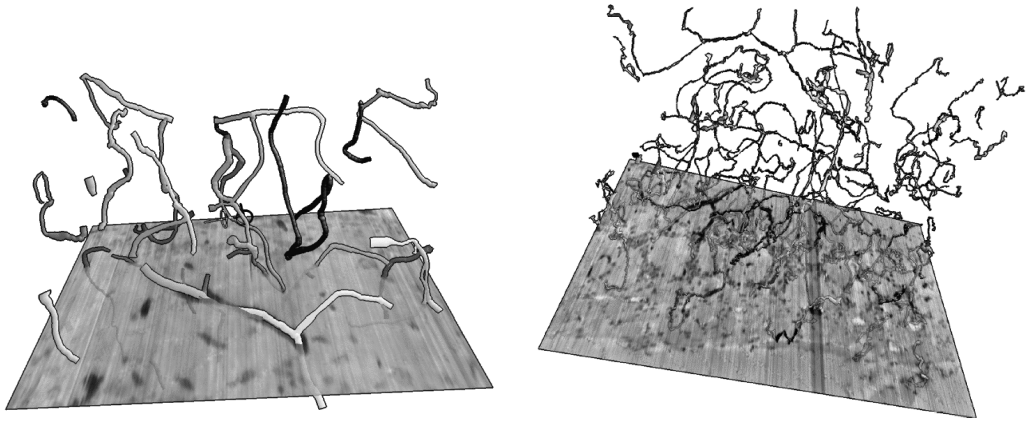
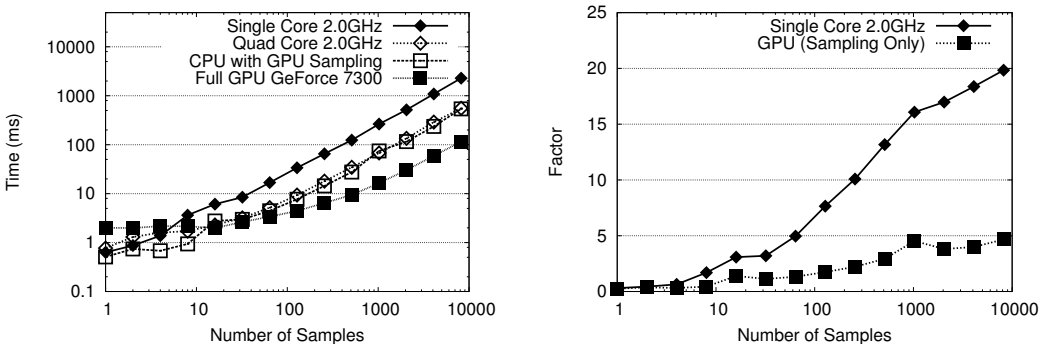


Figure 2.13: **Tracing Results.** The tracing results on a vascular data set are shown, with a slice of the actual data shown on the bottom. The fibers are colored with different shades of gray to represent independent traces. Adapted from [26].

data. These methods are often used for visualization in Diffusion Tensor MRI to display bundles fibers in white-matter regions of the brain. At the resolution of high-throughput microscopy, however, filaments rarely travel in coherent groups. Rendering high-throughput microscopy data using these techniques creates images that are cluttered and difficult to understand.

The techniques that we use are based on ideas demonstrated by streamlines. We first segment a filament network, as described in Sec. 2.4. We then render the filaments as a series of billboards that are always oriented towards the viewer (Fig. 2.15). This eliminates the need to store large volumes, in the case of volume rendering, or polygon meshes, in the case of surface representations. Instead, only the series of line segments and radii are used to display the network (Fig. 2.16).

This allows a user to interactively explore the filament data but does little to reduce the clutter produced by the dense network of filaments. Since we have the positional information for each filament, we can selectively visualize parts of the network based on the connectivity and trajectory of filaments [35]. By selecting an orientation, we can highlight filaments that travel in the selected direction, looking for trends in the data set. We call this *orientation filtering* and use it to understand the flow of axons in a complex network as well as general trends in vascular flow through different types of tissue.



(a) Run time (b) Speedup

Figure 2.14: **Fast Fiber Tracing Using the Graphics Processing Unit (GPU).** Performance figures demonstrate the speedup obtained by using GPU computation. The GPU version results (a) Computation time taken to trace the same amount of data on different combinations of CPUs (single- or multi-core) and GPUs are shown. The use of GPU gives an order-of-magnitude reduction in computation time.(b) The speedup achieved by using the full capacity of GPUs as compared to that of single-core CPU (diamond) or GPU sampling only (square) is shown, as a speed-up ratio (Factor = comparison time/full GPU time). The results show an almost 20-fold speedup compared to single-core CPU-based runs. Adapted from [26].

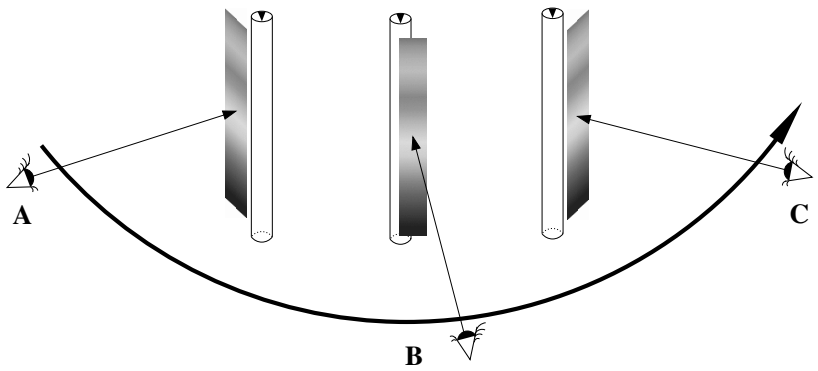


Figure 2.15: **Self-Orienting Surfaces.** The geometric data is replaced by a single geometric object (such as a surface). This object (surface) is oriented to always face the viewer. The object itself is also modified so that the appearance is correct from any angle. Note that the orientation of the data (cylinder) is not changing (small arrow head on top of the cylinders).

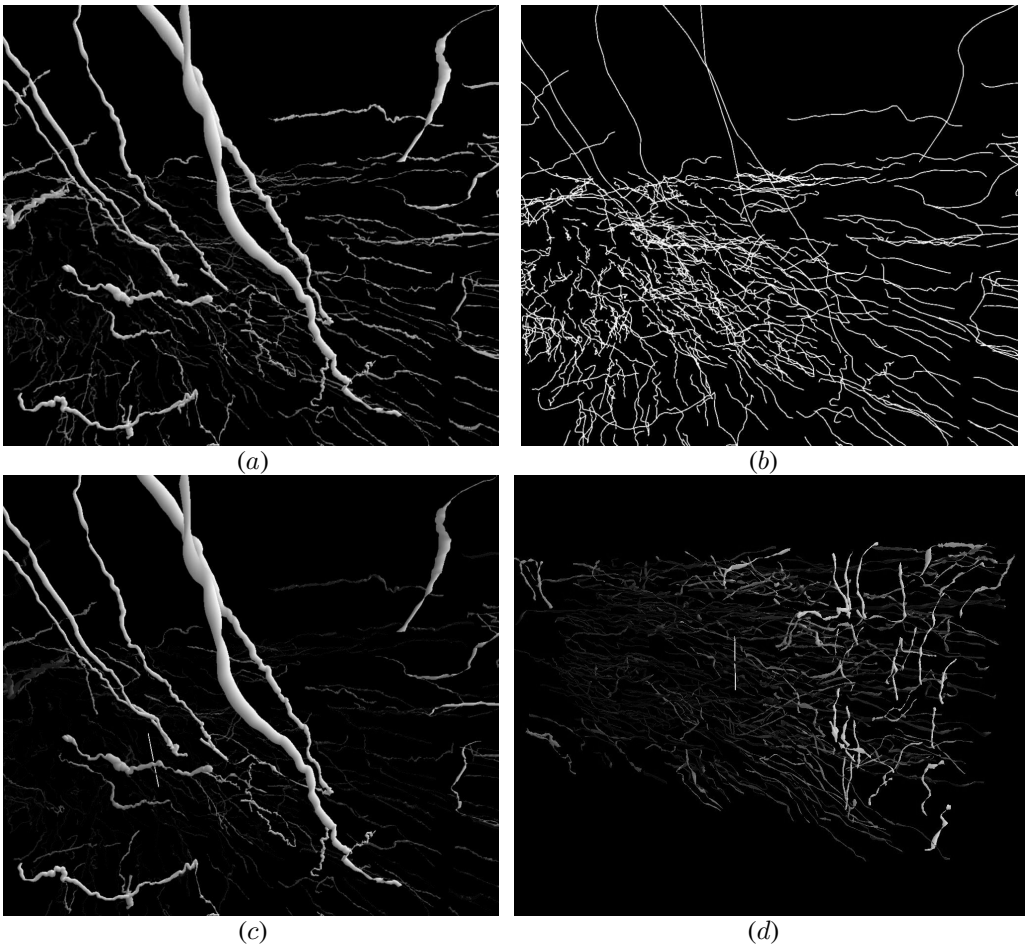


Figure 2.16: **Interactive Visualization of Fibrous and Thread-like Data.** (a) Self-orienting surfaces implemented on the GPU give high quality renderings of neuron data viewed close up, and (b) allow large volumes to be viewed at interactive rates. (c) Orientation filtering allows us to interactively explore orientation of fibers, allowing us (d) to discover fiber bundle directions across larger data sets. Adapted from [35].

2.6 Discussion

The main contribution of our work is three-fold: (1) high-throughput serial-sectioning imaging of brain microstructures, (2) fast and accurate algorithms for 2D and 3D reconstruction of neural and vascular morphology, and (3) interactive real-time visualization of the reconstructed structures. These techniques are expected to provide critical neuroanatomical data at a submicron-level for whole small animal brains. There are several open issues that need to be addressed. In the following sections, we will discuss validation and editing, as well as how to exploit parallelism.

2.6.1 Validation and editing

Quality control becomes a critical issue for any automated data processing procedure, and our parallel 3D reconstruction algorithm is no exception. In medical imaging, this process is called "validation" [36–38]. The challenge here is that obtaining full ground truth for validation could require performing a reconstruction of the entire data set. Typically, two approaches are taken for validation: (1) use of manually labeled data as ground truth, and (2) use of digital phantoms, i.e., synthetically generated data mimicking structures observed in a known ground truth (see [38] for a review). The two approaches have their own limitations, but they are complementary. We are currently looking into optimally combining these two approaches for accurate, efficient, high-throughput validation.

For the first approach requiring ground truth, manual editing is critical. Manual editing of small volumes, for example, a 300µm $\times$ 300µm $\times$ 300µm cube of mouse brain would not take too long (about a week). Consider Fig. 2.17 which illustrates the image acquisition and reconstruction process. Given the data volume (image stack I) obtained from KESM, domain experts can label the data to provide the ground truth $\hat{M}_{\mathrm{e}}$ (note that the domain expert's labeling is in itself an estimation of the true ground truth M). Concurrently with manual labeling, the data volume I can be put through our segmentation and reconstruction algorithms, producing the estimated microstructure $\hat{M}$. Validation can then be done by comparing $\hat{M}_{\mathrm{e}}$ (by the domain expert) and $\hat{M}$ (by the reconstruction algorithm). Measures like mutual information can be used to quantitatively measure the difference. Furthermore, we can train validation functions using machine learning techniques, and use the resulting validator for large-scale validation (see Chapter 7 by Yom-Tov on machine learning for large-scale datasets).

For the second approach, digital phantoms can be used. Digital phantoms

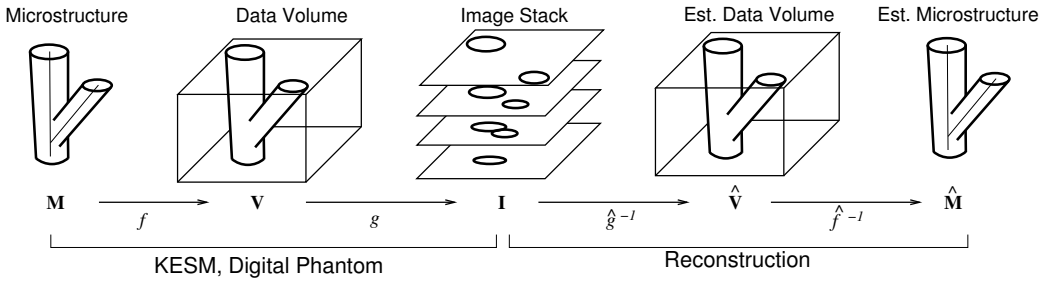


Figure 2.17: **Microstructure-to-image Mapping and Reconstruction.** The process by which a microstructure (real or synthetically generated) is turned into a stack of images in KESM and how they are reconstructed is shown. Modeling this process ($g \circ f$, composition of g and f) enables the generation of realistic synthetic image data (digital phantoms). On the other hand, the task of recovering the structural descriptions from the image data is basically the inverse: $\hat{f}^{-1} \circ \hat{g}^{-1}$, a composition of the segmentation ($\hat{g}^{-1}$) and the 3D reconstruction process ($\hat{f}^{-1}$). (The "$\hat{}$" symbol indicates that these functions are estimates.) Validation can take three approaches: (1) given I from KESM and $\hat{M}_{\mathrm{e}}$ from human labeling, generate reconstruction $\hat{M}$ from I and compare it with $\hat{M}_{\mathrm{e}}$, (2) given a digital phantom I with ground truth M, reconstruct $\hat{M}$ from I and compare it with M, or (3) Given I and its reconstruction $\hat{M}$, use the mapping $g \circ f$ to create $\hat{I}$ and show the difference image $\hat{I} - I$ that can be accepted or rejected.

are synthetically generated, realistic data, produced from a known ground truth model (see e.g., [39]). Given a known ground truth M, the 3D volume V is generated in a straight-forward manner, and then image formation, distortion, and noise models are used to produce the final product, the image stack I (Fig. 2.17). The KESM imaging process can be modeled (as in [40]), and based on this, noisy digital phantoms (I) can be generated from a synthetic ground truth (M). The reconstruction algorithm can then be executed on this digital phantom, and the estimated microstructure $\hat{M}$ compared to the ground truth (M).

These two approaches will help perform large-scale validation of our automated algorithms.

2.6.2 Exploiting parallelism

With the advent of high-throughput microscopy instruments like KESM, data acquisition is no longer a bottle neck: the reconstruction and computational analysis

becomes a major bottle neck [16]. Fortunately, many tasks involved in reconstruction and analysis can be conducted locally, thus allowing straight-forward parallelization (see, e.g., [41]). For example, the large data volume can be partitioned into small unit cubes that can fit in several GB of memory, and the reconstruction can be done within each cube on multiple computing nodes in parallel. The latest techniques developed for parallel feature extraction and 3D reconstruction using high-performance computing can be adapted easily for these tasks (see Chapter 6 by Rao and Cecchi and Chapter 8 by Cooper et al.).

Besides parallelizing by dividing our data set into multiple regions, each to be handled by a single computing node, we can exploit the avaiable GPUs and multi-cores on modern CPUs to run multiple processes simultaneously tracking different fibers in the same unit cube. This will allow for larger memory blocks, and thus fewer artificial boundaries across which fiber tracking is inherently more difficult.

Even though these computations can be carried out locally, in the end, the results of these computations need to be merged, since processes from neurons (especially long axons) can project across the full span of the brain. After initial unit cubes are processed, we must merge data from adjacent unit cubes. It may also be necessary to modify our reconstruction in one unit cube based on the data obtained from the boundary of an adjacent unit cube. Some ideas on how a similar merger can be achieved is discussed in [42].

Another important observation is that the result from parallel reconstruction and merging need to be stored. As morphological information from adjacent unit cubes are combined, the fibers within each area must be merged. This can create potential problems, as the merged fiber-like data loses the locality it previously enjoyed. Furthermore, the data structure could become extremely complex (e.g., the entire vascular system could be one connected component); methods for representing such complex structures *hierarchically* will be needed.

Once all the reconstructions are done and stored so that a network of connected components emerges (e.g., neuronal or vascular networks), large-scale network analysis needs to be conducted to extract principles that connect network structure to function [43–47]. For such a task, parallel processing techniques developed for computational biology applications could be utilized (see Chapter 4 by Wagner).

Finally, various visualization techniques may also require parallel and high-performance computing capabilities. Machine learning techniques to visualize relevant features in the input (Chapter 10 by Xiao et al.), and the use of new high-performance computing paradigms (Chapter 13 by Singh et al.) show that visualization also needs to tap into high-performance computing.

2.7 Conclusion

Emerging high-throughput microscopy techniques such as the Knife-Edge Scanning Microscopy are starting to generate immense volumes of high-quality, high-resolution data from whole animal organs such as the mouse brain. Once these data are made available, the real bottle neck becomes computational analysis. Computational demand is very high at every stage of analysis, from image processing, 3D reconstruction, up to validation and network analysis (e.g., for neuronal and vascular networks). This challenges can be met effectively only through a combination of (1) efficient and accurate algorithms and (2) high-performance computing. In this chapter, we surveyed high-throughput microscopy techniques with a focus on our own KESM, and efficient algorithms (MW-CTTS and template-matching vector tracking) for morphological reconstruction. An extension of these methods, coupled with high-performance computing paradigms, can open the door to exciting new discoveries in biological sciences.

Acknowledgments

The results reported in this chapter have been supported in part by the National Science Foundation (MRI award #0079874 and ITR award #CCR-0220047), Texas Higher Education Coordinating Board (ATP award #000512-0146-2001), National Institute of Neurological Disorders and Stroke (Award #1R01-NS54252); and the Department of Computer Science, and the Office of the Vice President for Research at Texas A&M University. We would like to thank Bernard Mesa at Micro Star Technologies for helping with the design and implementation of the KESM, and the anonymous reviewer for constructive feedback. We dedicate this chapter to Bruce H. McCormick, our dear colleague and the main architect of the KESM, who passed away while this chapter was being written.

Bibliography

[1] K. Carlsson, P. Danielsson, R. Lenz, A. Liljeborg, and N. Åslund, "Three-dimensional microscopy using a confocal laser scanning microscope," *Optics Letter*, vol. 10, pp. 53–55, 1985.

[2] J. B. Pawley, *Handbook of Biological Confocal Microscopy*. New York: Plenum Press, 1995.

[3] W. Denk, J. H. Strickler, and W. W. Webb, "Two-photon laser scanning fluorescence microscopy," *Science*, vol. 248, pp. 73–76, 1990.

[4] G. Y. Fan, H. Fujisaki, A. Miyawaki, R.-K. Tsay, R. Y. Tsien, and M. H. Elisman, "Video-rate scanning two-photon excitation fluorescence microscopy and ratio imaging with cameleons," *Biophysical Journal*, vol. 76, pp. 2412–2420, 1999.

[5] P. S. Tsai, B. Friedman, A. I. Ifarraguerri, B. D. Thompson, V. Lev-Ram, C. B. Schaffer, Q. Xiong, R. Y. Tsien, J. A. Squier, and D. Kleinfeld, "All-optical histology using ultrashort laser pulses," *Neuron*, vol. 39, pp. 27–41, 2003.

[6] B. H. McCormick, "System and method for imaging an object," 2004. USPTO patent #US 6,744,572 (for Knife-Edge Scanning; 13 claims).

[7] B. H. McCormick, "The knife-edge scanning microscope," tech. rep., Department of Computer Science, Texas A&M University, 2003. http://research.cs.tamu.edu/bnl/.

[8] B. H. McCormick, L. C. Abbott, D. M. Mayerich, , J. Keyser, J. Kwon, Z. Melek, and Y. Choe, "Full-scale submicron neuroanatomy of the mouse brain," in *Society for Neuroscience Abstracts*, Washington, DC: Society for Neuroscience, 2006. Program No. 694.5. Online.

[9] D. Mayerich, L. C. Abbott, and B. H. McCormick, "Knife-edge scanning microscopy for imaging and reconstruction of three-dimensional anatomical structures of the mouse brain," *Journal of Microscopy*, vol. 231, pp. 134–143, 2008.

[10] B. H. McCormick and D. M. Mayerich, "Three-dimensional imaging using Knife-Edge Scanning Microscope," *Microscopy and Microanalysis*, vol. 10 (Suppl. 2), pp. 1466–1467, 2004.

[11] K. Micheva and S. J. Smith, "Array tomography: A new tool for imaging the molecular architecture and ultrastructure of neural circuits," *Neuron*, vol. 55, pp. 25–36, 2007.

[12] W. Denk and H. Horstmann, "Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure," *PLoS Biology*, vol. 19, p. e329, 2004.

[13] K. Hayworth and J. W. Lichtman, 2007. Automatic Tape-Collecting Lathe Ultramicrotome (ATLUM), `http://www.mcb.harvard.edu/lichtman/ATLUM/ATLUM_web.htm`.

[14] J. C. Fiala and K. M. Harris, "Extending unbiased stereology of brain ultrastructure to three-dimensional volumes," *J. Am. Med. Inform. Assoc.*, vol. 8, pp. 1–16, 2001.

[15] K. M. Brown, D. E. Donohue, G. D'Alessandro, and G. A. Ascoli, "A cross-platform freeware tool for digital reconstruction of neuronal arborizations from image stacks," *Neuroinformatics*, vol. 3, pp. 343–359, 2007.

[16] D. Chklovskii, "From neuronal circuit reconstructions to principles of brain design," in *Proceedings of the 5th Computational and Systems Neuroscience Meeting (COSYNE 2008 Abstracts)*, p. 331, 2008.

[17] V. Jain, J. F. Murray, F. Roth, H. S. Seung, S. Turaga, K. Briggman, W. Denk, and M. Helmstaedter, "Using machine learning to automate volume reconstruction of neuronal shapes from nanoscale images," in *Society for Neuroscience Abstracts*, Washington, DC: Society for Neuroscience, 2007. Program No. 534.7. Online.

[18] Y. LeCun, B. Boser, J. S. Denker, D. Henderson, R. E. Howard, W. Hubbard, and L. D. Jackel, "Backpropagation applied to handwritten zip code recognition," *Neural Computation*, vol. 1, pp. 541–551, 1989.

[19] J. J. Capowski, ed., *Computer Techniques in Neuroanatomy*. Plenum, 1989.

[20] K. A. Al-Kofahi, S. Lasek, D. H. Szarowski, C. J. Pace, G. Nagy, J. N. Turner, and B. Roysam, "Rapid automated three-dimensional tracing of neurons from confocal image stacks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 6, pp. 171–187, 2002.

[21] A. Can, H. Shen, J. N. Turner, H. L. Tanenbaum, and B. Roysam, "Rapid automated tracing and feature extraction from retinal fundus images using direct exploratory algorithms," *IEEE Transactions on Information Technology in Biomedicine*, vol. 3, pp. 125–138, 1999.

[22] D. M. Mayerich, L. C. Abbott, and B. H. McCormick, "Imaging and reconstruction of mouse brain vasculature and neighboring cells using knife-edge scanning microscopy," in *Society for Neuroscience Abstracts*, Washington, DC: Society for Neuroscience, 2006. Program No. 694.4. Online.

[23] B. Busse, S. J. Smith, C. A. Taylor, and R. Y. Arakaki, "Development of a semi-automated method for three-dimensional neural structure segmentation," in *Society for Neuroscience Abstracts*, Washington, DC: Society for Neuroscience, 2006. Program No. 834.13. Online.

[24] Zuse Institute Berlin (ZIB) and Mercury Computer Systems, Berlin, "Amira: Advanced 3D visualization and volume modeling," 2006. http://www.amiravis.com.

[25] D. M. Mayerich and J. Keyser, "Filament tracking and encoding for complex biological networks," in *Proceedings of Solid Modeling*, 2008. To appear.

[26] D. M. Mayerich, Z. Melek, and J. Keyser, "Fast filament tracking using graphics hardware," Tech. Rep. TAMU-CS-TR-2007-11-3, Department of Computer Science, Texas A&M University, 2007.

[27] K. Haris, S. Efstratiadis, N. Maglaveras, C. Pappas, J. Gourassas, and G. Louridas, "Model-based morphological segmentation and labeling of coronary angiograms," *IEEE Trans. Med. Imag.*, vol. 18, pp. 1003–1015, October 1999.

[28] R. W. Hamming, *Numerical Methods for Scientists and Engineers*. New York: NY: McGraw-Hill, 1962.

[29] T. Pavlidis, *Algorithms for graphics and image processing*. Computer Science Press, 1982.

[30] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Prentice Hall, 2nd ed., 2002.

[31] Y. Sato, S. Nakajima, N. Shiraga, H. Atsumi, S. Yoshida, T. Koller, G. Gerig, and R. Kikinis, "Three-dimensional multi-scale line filter for segmentation and visualization of curvilinear structures in medical images," *Medical Image Analysis*, vol. 2, pp. 143–168, 1998.

[32] K. Al-Kofahi, S. Lasek, D. Szarowski, C. Pace, G. Nagy, J. Turner, and B. Roysam, "Rapid automated three-dimensional tracing of neurons from confocal image stacks," *IEEE Transactions on Information Technology in Biomedicine*, vol. 6, pp. 171–186, 2002.

[33] D. M. Mayerich, Z. Melek, and J. Keyser, "Fast filament tracking using graphics hardware," *Technical Report*, vol. TAMU-CS-TR-2007-11-3, 2007.

[34] C. Stoll, S. Gumhold, and H.-P. Seidel, "Visualization with stylized line primitives.," p. 88, 2005.

[35] Z. Melek, D. Mayerich, C. Yuksel, and J. Keyser, "Visualization of fibrous and thread-like data," *IEEE Transactions on Visualization and Computer Graphics*, vol. 12, no. 5, pp. 1165–1172, 2006.

[36] T. Yoo, N. J. Ackerman, and M. Vannier, "Toward a common validation methodology for segmentation and registration algorithms," in *Lecture Notes In Computer Science, Vol. 1935; Proceedings of the Third International Conference on Medical Image Computing and Computer-Assisted Intervention*, (London), pp. 422–431, Springer, 2000.

[37] S. J. Warfield, K. H. Zou, and W. M. Wells, "Validation of image segmentation and expert quality with expectation-minimization algorithm," in *Lecture Notes In Computer Science, Vol. 2488; Proceedings of the 5th International Conference on Medical Image Computing and Computer-Assisted Intervention*, pp. 298–306, 2002.

[38] D. L. Pham, C. Xu, and J. L. Prince, "Current methods in medical image segmentation," *Annual Review of Biomedical Engineering*, vol. 2, pp. 315–337, 2000.

[39] R. A. Koene, "Large scale high resolution network generation: Producing known validation sets for serial reconstruction methods that use histological images of neural tissue," in *International Conference on Complex Systems*, 2007. [Presentation].

[40] J. S. Guntupalli, "Physical sectioning in 3D biological microscopy," Master's thesis, Department of Computer Science, Texas A&M University, 2007. To be released in December 2007.

[41] A. R. Rao, G. A. Cecchi, and M. Magnasco, "High performance computing environment for multidimensional image analysis," *BMC Cell Biology*, vol. 8(Suppl 1), p. S9, 2007.

[42] J. Kwon, D. Mayerich, Y. Choe, and B. H. McCormick, "Lateral sectioning for knife-edge scanning microscopy," in *Proceedings of the IEEE International Symposium on Biomedical Imaging*, 2008. In press.

[43] O. Sporns and G. Tononi, "Classes of network connectivity and dynamics," *Complexity*, vol. 7, pp. 28–38, 2002.

[44] M. Kaiser and C. C. Hilgetag, "Nonoptimal component placement, but short processing paths, due to long-distance projections in neural systems," *PLoS Computational Biology*, vol. 2, pp. 805–815, 2006.

[45] D. Chklovskii, T. Schikorski, and C. Stevens, "Wiring optimization in cortical circuits," *Neuron*, vol. 34, pp. 341–347, 2002.

[46] R. Milo, S. Shen-Orr, S. Itzkovitz, N. Kashtan, D. Chklovskii, and U. Alon, "Network motifs: Simple building blocks of complex networks," *Science*, vol. 298, pp. 824–827, 2002.

[47] A.-L. Barabási, *Linked*. Cambridge, MA: Perseus Publishing, 2002.