Segmentation-less, automated vascular vectorization robustly extracts neurovascular network statistics from in vivo two-photon images

Samuel A. Mihelic[a], William A. Sikora[a], Ahmed M. Hassan[a], Michael R. Williamson[b], Theresa A. Jones[b], Andrew K. Dunn[a,*]

[a] *Department of Biomedical Engineering, The University of Texas at Austin, 107 W. Dean Keeton Street, Austin, TX 78712, USA*
[b] *Institute for Neuroscience, The University of Texas at Austin, 1 University Station, Stop C7000, Austin, TX 78712, USA*

Abstract

Recent advances in two-photon microscopy (2PM) have allowed large scale imaging and analysis of cortical blood vessel networks in living mice. However, extracting a network graph and vector representations for vessels remain bottlenecks in many applications. Vascular vectorization is algorithmically difficult because blood vessels have many shapes and sizes, the samples are often unevenly illuminated, and large image volumes are required to achieve good statistical power. State-of-the-art, three-dimensional, vascular vectorization approaches require a segmented/binary image, relying on manual or supervised-machine annotation. Therefore, voxel-by-voxel image segmentation is biased by the human annotator/trainer. Furthermore, segmented images oftentimes require remedial morphological filtering before skeletonization/vectorization. To address these limitations, we propose a vectorization method to extract vascular objects directly from unsegmented images. The Segmentation-Less, Automated, Vascular Vectorization (SLAVV) source code in MATLAB is openly available on GitHub. This novel method uses simple models of vascular anatomy, efficient linear filtering, and low-complexity vector extraction algorithms to remove the image segmentation requirement, replacing it with manual or automated vector classification. SLAVV is demonstrated on three in vivo 2PM image volumes of microvascu-

*Corresponding author: adunn@utexas.edu

lar networks (capillaries, arterioles and venules) in the mouse cortex. Vectorization performance is proven robust to the choice of plasma- or endothelial-labeled contrast, and processing costs are shown to scale with input image volume. Fully-automated SLAVV performance is evaluated on various, simulated 2PM images based on the large, [1.4, 0.9, 0.6] mm input image, and performance metrics show greater robustness to image quality than an intensity-based thresholding approach.

Keywords:
2000 MSC: 92C55, 42B05, 05C40
Cerebral microvascular network, two-photon microscopy, linear filtering, vectorization, classification, graph

---

1. Introduction

The neurovascular system provides oxygen and nutrients in response to local metabolic demands through the process of neurovascular coupling (Kozberg and Hillman, 2016). This process is dysregulated in pathological conditions such as hypertension and Alzheimer's disease (Girouard and Iadecola, 2006; Lecrux and Hamel, 2011). Individual capillary tracking over multiple imaging sessions would provide a useful experimental tool for measuring neurovascular plasticity in preclinical disease models. Such experiments could screen new stroke therapeutics and provide much needed relief to clinical studies (STAIR, 1999; Dirnagl, 2006; Savitz, 2007; Fisher et al., 2009). However these measurements remain intractable for large image volumes, due to difficulties in vectorization. A vectorized network consists of a graph that summarizes the connectivity and a collection of simple objects that represent individual vessel segments. High-fidelity vectorization is difficult to attain in general, but facilitates the extraction of topological and geometric statistics to summarize vascular networks (Smith et al., 2019). The state of the art approach to vascular vectorization is skeletonization, which requires a nearly-perfect image segmentation of the vascular network. From a computer vision perspective, segmentation of blood vessels from in vivo fluorescence images present three major challenges: (1) blood vessels have many sizes and bifurcations, (2) samples are unevenly illuminated, and (3) large, information-rich digital images are required to achieve both adequate statistical power and spatial resolution. It is this requirement, which is difficult to meet in general, that motivates us to revisit the vascular vectorization

workflow.

For the purposes of quantifying vascular anatomy, skeletonization techniques are used to vectorize and extract the centerlines of blood vessels (Tsai et al., 2009; Lang et al., 2012; Cao et al., 2015; Teikari et al., 2016). Skeletonization techniques perform iterative morphological filtering on binarized images, and therefore require segmented images generated from vessel enhancement filtering, thresholding, or manual or machine-learned annotation. Manual, voxel-by-voxel, image segmentation ensures a high-quality binary input image to skeletonize, but is a tedious and often heuristic task. Researchers demonstrated individual capillary tracking in living mouse cortex over several weeks, but estimates of vascular plasticity lacked statistical significance due to the small, manually segmented, sample size (Cudmore et al., 2017). Alternatively, convolutional neural networks allow computers to learn this manual task from example (Damseh et al., 2018; Haft-Javaherian et al., 2019). However, deep learners that are trained by humans in voxel-by-voxel classification have human biases and are not intrinsically robust to input image properties such as resolution and noise level.

Many filters using local curvature information show improved robustness to image quality (Sato et al., 1997; Yousefi et al., 2013, 2015). These filters require eigenvalue decomposition of the Hessian at many voxels, and are thus computationally expensive. They rely on local shape information, are difficult to extend to multiscale, and show attenuated response at vessel bifurcations. These deficiencies were addressed by (Jerman et al., 2016) using a unitless ratio of eigenvalues. Lee et al. (2014) used a Hessian-based filter to segment images using an exploratory vectorization algorithm that automatically traces vessels and terminates each trace when the filtered image drops below a predefined threshold. However, the termination criteria was arbitrary and did not produce a connected network.

Manual, machine-learned or other approaches to voxel-by-voxel, image segmentation (Araújo et al., 2019) do not guarantee any topological structure without remedial morphological filtering. As an alternative, we propose extracting vascular vectors directly from 2PM images, thereby enforcing fundamental shape and connectivity constraints upstream of vector classification. To demonstrate this workflow, we present the Segmentation-Less, Automated, Vascular Vectorization (SLAVV) method in Sections 3.1-4, (MATLAB code at https://github.com/UTFOIL/Vectorization-Public). The advantage of SLAVV is that there is no requirement to segment, interpolate, or otherwise preprocess the input 2PM image. However, the extracted vectors

are probabilistic and need to be classified in some way. A graphical user interface is used to curate the extracted vectors from in vivo, plasma- and endothelial-labeled 2PM images (Sections 4.1-2). Example vectorized statistical analysis is demonstrated for the large-volume Input Image 3 (Section 4.3). Fully-automated, global vector thresholding performance is evaluated on realistic, simulated 2PM images of varying quality (Section 4.4).

2. Experimental

2.1. 2PM Imaging

Two-photon microscopy (2PM), in vivo, three-dimensional images of murine microvasculature were acquired from two mice at two resolutions with two different sources of contrast. Table 1 summarizes the input image properties.

Table 1: Descriptions of 2PM input images used in Sections 4.1 and 4.3 to demonstrate novel vascular vectorization. The endothelium (vessel wall) was labeled in Image 1.

Input Image	1	2	3
Mouse	1	1	2
Contrast localization	Endothelium	Plasma	Plasma
Transverse dimensions [μm]	469x469	469x469	1350x940
Pixel length [μm]	0.92	0.92	1.07
Axial depth [μm]	125	125	600
Axial voxel length [μm]	2	2	5
Bit depth	16	16	8
Numerical aperture	1.0	1.0	0.95
Vectorization preprocessing	None	None	Tiling, 3x3x3 median filter

2.1.1. Dual channel plasma- and endothelial-labeled

Mouse 1 (Young adult Tie2-GFP, FVB background, JAX stock no. 003658) was implanted with a 4mm cranial window over sensorimotor cortex (Clark et al., 2019). For imaging, the mouse was anesthetized with 1-2 % isoflurane in oxygen and head fixed. A retro-orbital injection of 0.05mL of 1 % (w/v) 70 kDa Texas Red-conjugated dextran was given to label the plasma. GFP fluorescence was localized to the endothelium. Images 1 and 2 were acquired

with a Prairie Ultima two-photon microscope with a Ti:Sapphire laser (Mai Tai, Spectra-Physics) tuned to 870 nm and a 20x 1.0 NA water immersion objective (Olympus) using Prairie View software.

2.1.2. Large volume tiling

Mouse 2 was injected intravenously with Texas Red and imaged through a cranial window using a two-photon microscope (custom microscope, Hassan et al. (2019)). The laser (1050 nm, 80 MHz repetition rate, 100 fs pulse duration (Perillo et al., 2017)) was scanned using galvo-galvo mirrors over 550 μm x 550 μm (512 x 512 pixels) and axially using a piezoelectric stage over 600 μm depth (120 slices). The images were output as 16 bit signed integers (3x10^7 voxels). To achieve the tiling in Input Image 3, a 2x3 grid of plasma-labeled images with 100 μm overlaps were median-filtered and then tiled with translational registration via cross-correlation (Preibisch et al., 2009) in Fiji (Schindelin et al., 2012).

3. Methods

3.1. Automated vessel vectorization

Novel Segmentation-Less Automated Vascular Vectorization (SLAVV) software is developed in MATLAB (version 9.6.0.1135713 (R2019a) Update 3) to extract vector sets representing vascular networks from raw gray-scale images without the need for pre-processing or specialized hardware. The SLAVV software (Sections 3.1-4, MATLAB code at https://github.com/UTFOIL/Vectorization-Public) consists of four workflows (Sections 3.1.1-4) (1) multi-scale linear filtering (2) vertex extraction (3) edge extraction and (4) network extraction. Statistics are calculated and visuals are rendered from the final step. Vectors resulting from steps 2 and 3 are probabilistic objects that require classification (Section 3.2). Example inputs and outputs (Sections 3.3-4) are shown in Sections 4.1 and 4.3. The performance of fully-automated SLAVV is measured in Section 4.4 using simulated images (Section 3.5).

3.1.1. Energy calculation

The raw (unprocessed/uninterpolated) input 3D image is matched-filtered for vessels (idealized as spherical objects) within a user-specified size range to yield a multi-scale, 4D image (Figure 2, conceptual). The matched filter is the convolution of a Laplacian of Gaussian (LoG) with standard deviation, σ,

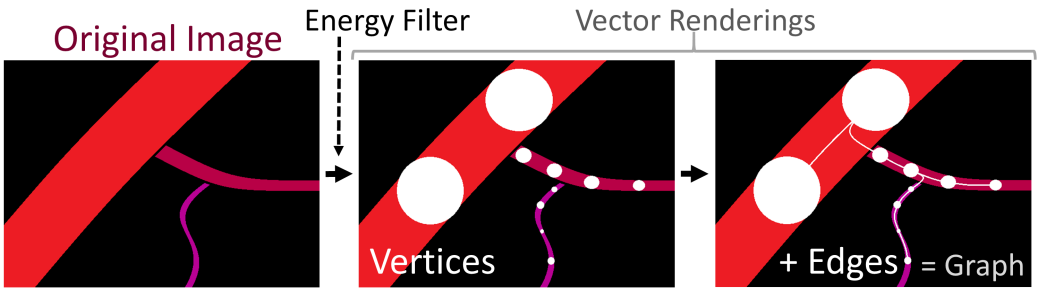

Figure 1: Overview of SLAVV approach. The purpose of SLAVV is to vectorize vascular objects from raw three dimensional images. The first step of the software is to linearly filter the input image to form "energy" and "size" images, which enhance vessel centerlines and estimate vessel sizes, respectively. Next, vertices along the blood vessels are extracted as local minima of the 3D energy image. Vertices are then connected by edges, which follow minimal energy trajectories. Finally, a graph theoretic representation of the vascular network is generated from the vertices and edges.

and an ideal (spherical or annular) kernel of radius r. Therefore, the radius, R, of the matched vessel is given by $R^2 = \sigma^2 + r^2$.

To detect a variety of fluorescent signal shapes, the ideal kernel is a combination of spherical and annular pulses. The weighting parameter f_S controls the fraction of spherical versus annular pulse in the matching kernel, thus weighting the responses from signals in the shape of the vessel wall vs lumen. Therefore, $f_S = ||K_S||/(||K_S|| + ||K_A||)$, where $||K_S||$ and $||K_A||$ are the total weights (L^1 norms) of the spherical and annular matching kernels.

The ideal kernel is convolved with a Gaussian to achieve robustness to noise and slight mismatches in size and shape. The processing parameter f_G controls the relative sizes of the Gaussian and ideal kernels, thereby trading noise robustness for accuracy of size and position. Therefore, $f_G = \sigma/(\sigma + r)$.

The scale space sampling is exponentially distributed with tunable density. The octave is defined as doubling of the vessel volume. For example, Image 1 is sampled across 16 doublings of the matched filter volume at a rate of 6 scales per octave, producing 96 discrete scale samples. To enforce a minimal amount of blurring and ensure good shape agreement between objects in the image and matched filters, a 3D (anisotropic) Gaussian model of the microscope PSF (Zipfel et al., 2003) is also convolved with each matched filter. Table 2 shows the processing parameters used to process the featured images.

The 4D (multi-scale) energy image is projected across the scale coordinate

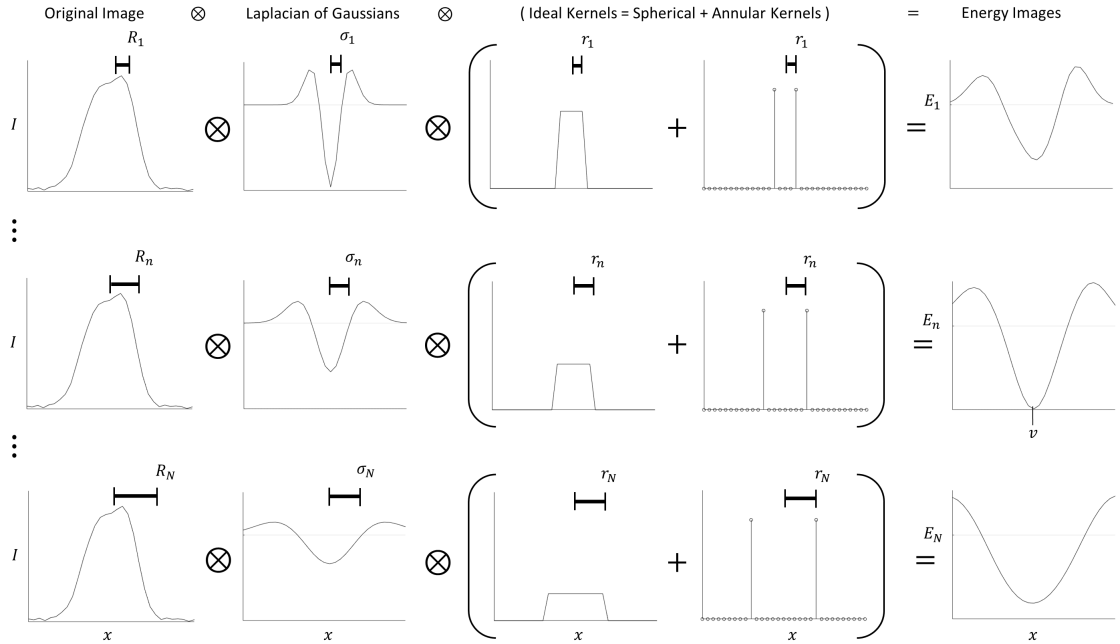

Figure 2: One-dimensional simplification of linear filtering step. To form the energy image E at scales $1...n...N$, the original image $I(x)$ is convolved with a LoG filter and an ideal kernel. The ideal kernel is a linear combination of spherical and annular pulses to match the fluorescent signal shape of vessels. σ^2 is the variance of the Gaussian, r is the radius of the ideal kernel, so $R^2 = \sigma^2 + r^2$ is the square "radius" of the LoG, matched filter. The resulting multiscale energy image is projected along the scale coordinate to form two 3D images that depict energy and size (not shown here, example in Figure 3). In this example, the kernel weighting factor was chosen so that the sums of all the spherical and annular kernels were all equal, the ratio r/σ was chosen to be 1, and a vertex was found at location v with radius R_n and energy $E_n(v)$.

to yield two 3D images (each with the same dimensions as the input) called the energy and size images (example images shown in Figure 3A). The energy image encodes "energy" which is used to estimate the likelihood of that voxel containing a vessel centerline. The size image encodes the expected radius of a vessel centered at each voxel. Because energy is the Laplacian of a real-valued image, negative values correspond to locally bright regions of the original image. Image voxels with positive energies are therefore ignored based on the assumption that the fluorescent signal is brighter than the background.

To compare Laplacian values across different scales and amounts of blur-

Table 2: Processing parameters used to vectorize the three experimental images and the set of synthetic images derived from Image 3.

Input Image	1	2	3	3-synthetic
f_G [%]	50	75	50	60, 80, 100
f_S [%]	50	50	85	100
Smallest radius [μm]	1.5	1.5	1.5	1
Largest radius [μm]	60	60	60	45
Scales per octave	6	6	3	3

ring, each second derivative is weighted by the variance of the Gaussian filter at that scale in that dimension (Lowe, 2004) to compensate for decreased derivative values due to blurring. The projection method from multi-scale, 4D image back to 3D depends on the signal type, because the annular signal is more sensitive to size mismatches and cannot be averaged across many sizes reliably. Thus, for the annular input signal (endothelial label), the estimated scale is simply the scale that yielded the least energy. For the spherical input signal (plasma label), the estimated scale is a weighted average across all scales that yielded a negative energy using energy magnitude as weights. For linear combinations of annular and spherical signals, the weighted average of the two estimates is taken. The 3D energy image is then defined by looking up the 4D energy image at the scale nearest the estimated scale at each voxel.

To improve computational efficiency, several design features are built into the energy filtering step. For larger scale objects, the image is downsampled in each dimension before filtering to have a resolution no greater than 10 voxels per object radius, and then upsampled afterwards with linear interpolation. The image volume is processed in overlapping chunks to respect memory constraints and allow for parallelization. The overlapping length is expanded for larger scales to eliminate edge effects within the image volume. To minimize computational complexity, all blurring, matched filtering, and derivative approximations are calculated in the Fourier domain with a single, combined filter for each scale sampled from analytical Fourier representations.

3.1.2. Vertex extraction

The purpose of the vertex extraction step is to identify points of interest on the vascular network to serve as seed points for the edge extraction step.

The inputs are the energy and size images, and the output is a set of vertices: non-overlapping spheres that should be concentric with vessel centerlines, share the same radius as the vessel at that central location, and densely mark the vasculature with at least one vertex for each strand in the final output network.

Vertex center points are located by searching the energy image for local minima in all three-dimensions. The sizes of the vertices are assigned by referencing the size image at their center points. Vertex volumes are painted onto a blank canvas from lowest to highest energy to obtain the most probable non-overlapping subset (Algorithm 1, conceptual).

Algorithm 1: Extracting vertices from the energy and radius images.

```
Function vertices = find_vertices(energy, radius)
  Initialize vertices as an empty list
  Let mask be the vector of ones with length of energy
  While there is a negative entry in mask·energy
    Let location =argmin(mask·energy)
    Let mask(location) = 0
    If location is a local minimum of energy
      Initialize vertex with fields:
        vertex.location = location
        vertex.radius = radius(location)
        vertex.energy = energy(location)
      Append vertex to vertices:
      Let interval = [-vertex.radius,vertex.radius]
      Let mask(location+interval) = 0
```

3.1.3. Edge extraction

The purpose of the edge extraction step is to trace the vessel segments between pairs of vertices with a simple automated algorithm. The inputs are the set of vertices and the energy and size images. The output is a set of edges: vessel traces connecting vertex pairs, summarized by ordered lists of centerpoint 3-space coordinates and associated radii. Edges are ordered from most to least likely by the maximum energies along their traces. For edges with the same energy, the shorter one is considered more likely. This decision selects for the child edge whenever a child and parent have the same maximum energy.

Edges are traced by deterministically exploring the space around each vertex by repeatedly visiting the lowest energy neighbor, in a watershed manner, until a terminal vertex is found. When a terminal vertex is found, its spherical volume is blocked off to other edges and the newfound edge is traced back to the original vertex following the shortest path with lowest maximum

energy. To guarantee that this algorithm (Algorithm 2) uses finite time and memory, maxima were placed on the number of edges that each seed vertex could find (4 edges per vertex) and on the length of trace (20-100 times seed vertex radius).

Algorithm 2: Extracting edges associated to a vertex set from an energy image.

```
Function edges = find_edges(vertices, energy, max_number_edges)
  Initialize edges as an empty list
  (Parallel) For each vertex_A in vertices
    Initialize explored_locations and vertices_B as an empty list
    Initialize number_of_edges_found to 0
    Initialize energy_map and pointer_map as empty sparse arrays
    Initialize location as vertex_A.location
    Let pointer_map(location) = −∞
    While true
      If location = vertex_B.location for some vertex_B distinct from vertex_A
        Initialize traced_locations as a list with first entry location
        While pointer_map(location) ≥ 0
          Let location = explored_locations(pointer_map(location))
          Append location to traced_locations
        Initialize edge with fields:
          edge.locations = traced_locations
          edge.vertex_A = vertex_A
          edge.vertex_B = vertex_B
        Append edge to edges and vertex_B to vertices_B
        Increment number_of_edges_found
        pointer_map(traced_locations) = - number_of_edges_found
        If number_of_edges_found = max_number_edges, break
      Append location to explored_locations
      Let neighbors list the locations in the 3x3x3 cube around location
      Delete any neighbor in neighbors where pointer_map(neighbor) = 0
      For vertex_B in vertices_B
        Let plane be the plane that faces vertex_A at vertex_B
        Delete any neighbor in neighbors beyond plane from vertex_A
      Let pointer_map(neighbors) = cardinality(explored_locations)
      Let energy_map(neighbors) = energy(neighbors)
      Let energy_map(location) = 0
      If min(energy_map) ≥ 0, break
      Let location = argmin(energy_map)
```

Algorithm 2 extracts edges from each origin vertex in the order of increasing maximum energies. To avoid double-counting stretches of vessel segments, vertices are not allowed to make multiple connections in the same direction. Once a vertex discovers a neighbor vertex by means of an edge, it is not allowed to search farther in the direction of that neighbor.

Algorithm 2 extracts non-overlapping centerline traces for the edges from each origin vertex. When two edge traces with the same origin vertex would

share a common segment of centerline voxels, the algorithm only counts the redundant trace in the first-extracted, lower-energy (more-probable) edge. That edge is called the parent, and the higher-energy one the child edge. To promote vertex placement near vessel bifurcations, vertices are added (after Algorithm 2) wherever a child edge meets its parent.

In cases where two vertices discover each other, only the most probable edge that connects them is kept. Small cycles of three fully-connected vertices are eliminated using graph theoretic methods. Two vertices are adjacent by a small cycle if they can be mutually reached using one and two edges. The connected components of the adjacency matrix by these small cycles is calculated, and the least probable edge is removed from each component. The adjacency matrix is then recalculated and the removal repeated until there are no adjacencies by small cycles.

3.1.4. Network extraction

The purpose of the network extraction step is to organize the vectors to facilitate statistical calculations and visualization. The inputs are the sets of vertex and edge objects, and the outputs are sets of strands, bifurcations and endpoints. Bifurcations and endpoints are special vertices. Bifurcations connect three or more edges, and endpoints connect one. A strand is a set of one or more consecutive edges that connects exactly two special vertices.

Bifurcations are found by calculating the adjacency matrix for the graph of the vertices and edges. Bifurcations are vertices associated with 3 or more edges. These vertices and their associated edges are then removed from the graph, leaving only vertices with 1 or 2 edges. The connected components of this graph identify the majority of the strand objects. The remaining strands and strand fragments are the single edges that connect to bifurcation vertices.

Once all the edges are assigned to strands and a subset of the vertices to bifurcations, a smoothing operation is applied that keeps the special vertices fixed and smooths the positions and sizes along the 1-dimensional strands. Gaussian smoothing kernels are defined for discrete locations along each strand with standard deviations equal to their radii. Smoothing kernels are further weighted by the energy value at each strand location. Algorithm 2 guarantees negative energy at every location along an edge, so the weighting is well defined and will favor the more likely, lower-energy locations. The variable smoothing kernel is then applied to each strand location to spatially average along the strand trace to increase the statistical significance of the five local quantities: 3-position, size, and energy.

3.2. Vector Classification

Due to the property of the SLAVV software extracting probabilistic vectors directly from grayscale images (without the need for image segmentation), a vector classification step after the edge extraction step is used to provide a deterministic input to the network statistic calculations. Vector classification is performed manually or automatically.

3.2.1. Interactive curation software

Two built-in, graphical curator interfaces are used to manually classify probabilistic vertices and edges into true and false categories in Sections 4.1-3. The automatically generated vectors are rendered transparently over the raw image. The user sets the intensity limits of the underlying raw image to accommodate variable brightness and contrast. The user may also view the probabilistic vectors with their brightnesses dependent on their energies to assess the accuracy of the automated filtering and vector extraction steps. The interface enables navigation within the input image volume to view any rectangular sub-volume as a maximum intensity projection in z. Manual edits to vectors can then be made within the sub-volume.

The interface gives two vector classification tools: point-and-click true/false toggling and local energy thresholding (within the navigation volume). The interface also enables the addition of vertices to a volume in a manual way that is assisted by the energy and size images. The user specifies the x-y location with a point-and-click and the software automatically selects the z position that minimizes the energy and the size from the corresponding location in the size image. The user may also add an edge by selecting any two vertices. The locations and associated sizes of the added edge are simply linear interpolations of the vertex sizes and locations.

3.2.2. Global energy thresholding

During the characterization of fully-automated SLAVV performance in Section 4.4, the vertex and edge objects are both segmented using global thresholds on the energy feature. Thresholds are chosen with the knowledge of the ground truth image to maximize voxel-by-voxel classification accuracy. For the vertex objects, global thresholding applies a user-defined upper limit on the acceptable energy associated with any vertex. Vertices with energies above the threshold are classified as false and eliminated. Since each vertex serves as a seed point (and possible termination point) in the edge extraction

algorithm, vertices are segmented prior to edge extraction to improve performance. For the edge objects, global thresholding applies a user-defined upper limit on the acceptable maximum energy associated with any edge. Edges with maximum energy above the threshold are eliminated.

3.3. Outputs

3.3.1. Three-dimensional scalar fields: vector renderings

Outputs of the energy, vertex, edge, and network steps (Sections 3.1-4) are automatically output as TIFF files of 3D scalar fields at input image resolution. These can be opened for viewing, merging/overlaying, etc. for example, in ImageJ (Schindelin et al., 2012). Vertices and edges are rendered as energy-weighted, centerline- or volume-filled objects. Centerline filling creates single voxel wide, continuous traces. Volume filling places spherical structuring elements of estimated radius, concentric with the centerline traces.

3.3.2. Visual projections: Strands, depth, and 3D perspective

Strand-resolved or depth-resolved z-projections are created in MATLAB by rendering strand centerline voxels with colors based on the unique strand identity or the z-component of each centerline location, respectively. Three-dimensional visualization is done in MATLAB with the ISOSURFACE and ISONORMALS functions (Copyright 1984-2017 The MathWorks, Inc.) by upsampling the strand sample points, and rendering each as a volume-filled object. The colors are again mapped from the strand unique identifiers.

3.4. Statistical analysis of vectors

Statistical calculations are performed within the SLAVV software to extract features of the network output such as volume, surface area, length, and number of bifurcations (Section 4.3). These calculations operate on the network output strands idealized as consecutive circular, right cylinders (Figure 4B). For analyzing distributions (histograms) of vessel quantities, the lateral surface area is used as the weighting quantity, because the lateral surface area is directly proportional to the chemical flux across the vessel wall.

3.5. Simulating 2PM images

Realistic 2PM images of varying contrast and noise are simulated from a ground truth vectorization for the purposes of measuring automated vectorization performance (Section 4.4). First the ground truth vectorization is

rendered in a binary volume-fill at high resolution, blurred with an overestimated Gaussian model for the PSF (Zipfel et al., 2003), and downsampled to the resolution of the original raw image. The intensities are then linearly transformed to have a positive background value I_B and a larger foreground value I_F.

To simulate Poisson distributed noise, a normally distributed random variable is added to each voxel with variance equal to the voxel intensity. The foreground and background values are varied to achieve different contrast and noise levels. Contrast is defined as the difference between the foreground and background intensities, $I_F - I_B$, and noise as the uncertainty in that difference, $\sqrt{I_F + I_B}$. For comparison purposes, simulated image quality is summarized by the contrast to noise ratio.

4. Results

4.1. Demonstration of SLAVV

To demonstrate the SLAVV approach, Images 1, 2, and 3 were vectorized (Section 3.1), manually curated (Section 3.2), and visualized (Section 3.3). Example inputs (plasma- or endothelial-labeled fluorescent images), intermediates (energy, size, vertex, and edge images), and outputs (3D rendering, 2D projections, and bulk network statistics) were produced for a small volume of vasculature in Figure 3. Figure 3A shows example inputs and intermediates. Input images were linearly filtered at many scales (Section 3.1.1) to produce the energy and size images which estimate the centerline positions and radius of vessels, respectively. Vertices were extracted (Section 3.1.2) as the minima of the energy image, with associated radii looked up from the size image. Edges were traced (Section 3.1.3) from one vertex to another along the path of minimum energy, and the size of the vessel at each location was again read from the size image. Once vectorized, the vessels were classified/curated (Section 3.2) and then rendered in 3D or 2D projections (Section 3.3). The strand objects (collections of non-bifurcating edges) were each labeled with a unique color in the 3D rendering in Figure 3B. The 2D projections in Figure 3C show vector vessel depth or direction information as a volume filling (at one quarter of the estimated vessel radius) overlaying the input image. The runtimes of the automated vectorization steps are shown in Table 3 for each of the three input images. Table 4 shows total network statistics of interest.

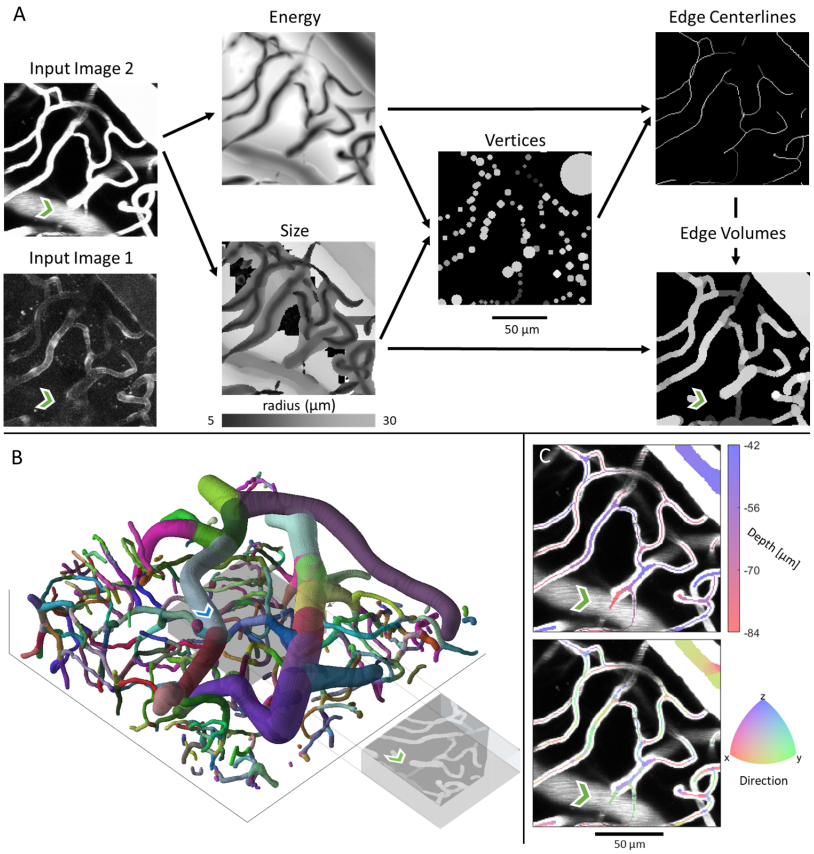

Figure 3: Example projections of original two-photon images, intermediate outputs, and vector renderings of manually assisted SLAVV applied to Input Image 2. The green chevron is directly underneath a medium sized vessel which bleeds into the original projection volume for the plasma-labeled Image 2 but not the endothelial-labeled Image 1. A. Either Image 1 or 2 could be the input (Image 2 outputs shown). The original image is subject to multiscale, LoG, matched filtering to obtain 3D energy and size images. The energy image is used to estimate vertex centers and the size image to estimate their radii. Vertices are used as genesis and terminus points for energy image exploration in the centerline extraction algorithm. Finally, indexed edge volumes are recalled from the size image to form the volume-filled vector rendering. Gray-scale coloring in the vector renderings corresponds to the energy values and thus vector probabilities. B. 3D visual output of SLAVV. Colors represent strands, which are defined as non-bifurcating vessel segments. Each strand is assigned a random color. The image is 125 μm in z and 460 in x and y. The projection volume used in the other panels is shown as a gray box in the center of the larger volume. The blue chevron marks the vessel that bleeds into the Input Image 2 at the green chevron. C. Depth and direction outputs from SLAVV. Vector volumes are rendered over the original projection at a quarter of their original radius. Direction is calculated by spatial difference quotient with respect to edge trajectory. The centerline for the vessel above the green chevron lies above the projection volume.

Table 3: Runtimes of automated processing steps in seconds. Image 3-Synthetic runtimes are similar to those of Image 3.

	Image 1	Image 2	Image 3
Energy filter runtime [s]	986	1,037	7,853
Vertex extraction runtime [s]	22	35	96
Edge extraction runtime [s]	73	204	141
Network analysis runtime [s]	19	0.5	34
Number of Voxels	1.67×10^7	1.67×10^7	1.64×10^8

Table 4: Bulk network statistics. Image 3-synthetic statistics are exactly those of Image 3.

	Image 1	Image 2	Image 3
Num. bifurcations	130	194	3,383
Length [mm]	16.2	17.0	443
Area [mm^2]	0.495	0.534	13.0
Volume [mm^3]	2.17×10^{-3}	2.46×10^{-3}	4.40×10^{-2}
Image Volume [mm^3]	2.75×10^{-2}	2.75×10^{-2}	0.761

4.2. Interactive curation software

For the vectorization of raw input images, an interactive vector curation interface (Section 3.2) was used to segment the automatically-generated vertex and edge objects into true and false categories and to patch false negatives. Vertex and edge curation costs for the three curated image volumes are shown in Table 5. Image 3 required the most time because it was the largest volume. Image 1 required more time than Image 2 although it was the same field of view, because the endothelial label was not as strong as the plasma label. The large number of vertex selections for Image 3 were due to the imperfect tiling process (Section 2.1.2) causing image discontinuities away from the image boundaries. Many of these selections were done simultaneously using a click-and-drag box selection. Local thresholds were placed on the energy feature of the vertices of the max energy feature of the edges in order to coarsely classify vectors over large regions of the image at once. The Final Total rows refer to the number of vector objects (vertices or edges) remaining in the final curated datasets.

Table 5: Summary of human effort toward manual vector classification on a graphical curator interface. Selections were point-and-click classifications of objects either individually or over a rectangular volume.

		Image 1	Image 2	Image 3
Vertices	Duration [min]	101	24	206
	Local Thresholds	2	5	110
	Selections	4724	230	19,845
	Additions	62	18	28
	Final Total	1,380	1,397	30,586
Edges	Duration [min]	207	140	410
	Local Thresholds	9	0	1
	Selections	145	66	96
	Additions	152	63	557
	Final Total	1,193	1,323	31,523

4.3. Example anatomical statistics

To demonstrate its statistical power, the SLAVV software was used to automatically vectorize and manually classify a large volume of capillaries, venules, and arterioles. Figure 4A shows a 3D rendering of the vectorized vessels (color-coded by strand) beneath a maximum intensity projection of the superficial layers of the tiled input Image 3. The vectorized vessels were idealized as collections of cylinders attached at the centers of their faces, and connected face to face (Figure 4B). Figure 4C shows lateral-area weighted histograms of vessel statistics (depth, radius, inclination (i.e. angle from x-y plane)). Figure 4D shows bulk statistics (length, area, volume, radius, inclination) binned by depth (z coordinate). The total summary statistics (Section 3.4) and a binary mask (Section 3.3.1) derived from this vector set serve as ground truth statistics and image segmentation for the automated SLAVV performance evaluation in Section 4.4.

4.4. Objective performance evaluation of fully-automated SLAVV

To remove the dependence on (subjective) manual annotation/segmentation in determining ground truth images (and therefore image segmentation accuracy), two-photon images were simulated (Section 3.5) from an assumed ground truth anatomy with various levels of contrast and noise. To enable the tracking of bulk network statistic accuracies in addition to image segmentation accuracy, a ground truth vector set (with known network statistics)

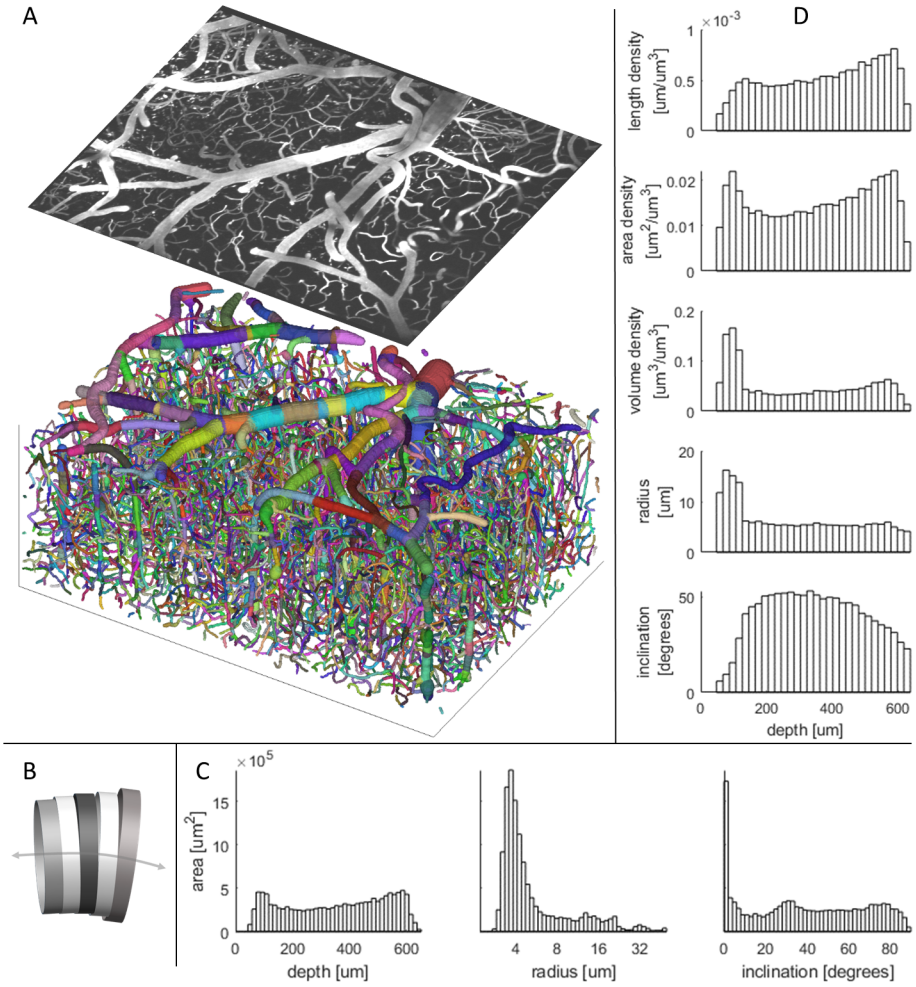

Figure 4: Example statistics of the microvasculature calculated from manually assisted SLAVV applied to Input Image 3. A. 3D rendering of strand objects similar to Figure 3B. Size of the image is 600 μm in z and 1350x940 in x-y. Projection of the first 70 μm of the original image is shown with the same perspective. B. Cartoon depiction of consecutive cylinder representation of a vessel segment used in calculations. C. Histograms of depth, radius, and inclination (angle from the x-y plane). The large peak in the inclination histogram at horizontal alignment is due to low axial resolution (5 μm). Contributions of cylinders to the bins are weighted by their lateral areas. D. Depth-resolved anatomical statistics output from SLAVV. Cylinders are binned by depth. Their heights, lateral areas, and volumes are summed and divided by the image volume apportioned to each bin to yield length, area, and volume densities of vasculature. Average radius and inclination in each depth bin are lateral area-weighted averages.

was used to generate the ground truth segmented image. Three simulated images of varying quality were generated from the vector set from Image 3 and Section 4.3. The noise and contrast settings for the simulated images are shown in Figure 5A, and maximum intensity projections are shown in the top row of Figure 5B.

The simulated images were vectorized to characterize the performance of the fully-automated (Section 3.2.2) SLAVV software. Three values (60, 80, and 100 %) were used for the f_G processing parameter in the energy calculation step. Vertices were extracted for each energy image and classified using global thresholding with a dense sweep of thresholds. The curated vertex sets were rendered at the resolution of the input image and compared to the ground truth image voxel-by-voxel to calculate the confusion table enumerating the false positives FP, true positives TP, false negatives FN, and true negatives TN. The vertex set yielding the highest classification accuracy $(TP+TN)/(TP+TN+FP+FN)$ was passed to the edge extraction step.

Edges were extracted for each vertex set and similarly classified using many global thresholds. Edge sets were rendered and voxel-by-voxel compared to the ground truth to find the sensitivity and specificity for each threshold. An ROC curve was made to observe the voxel-by-voxel classification strength of global thresholding on the edge objects. The edge set from each threshold sweep with maximal voxel-by-voxel classification accuracy was passed on to the network calculation step to compute statistics and compare to those of the ground truth.

The simulated image quality was benchmarked using a commonly-used, voxel-by-voxel, intensity thresholding classifier (Figure 5, black series) that has perfect accuracy in the limit of perfect image quality (CNR $\rightarrow \infty$). To demonstrate the sensitivity of the intensity thresholding classifier to image quality, the volume was computed from the intensity thresholded image with the best voxel-by-voxel classification accuracy. For the higher quality images, the volume accuracy was comparable to that of the vectorized approaches, however, the topology was unconstrained, leading to salt and pepper segmentation errors. These topological errors made surface area calculation highly sensitive to noise for the intensity classifier approach (result not shown). For the lowest quality image, the intensity classifier yielded a maximal segmentation accuracy of around 95 %, corresponding to the operating point that assigns all voxels to background (0 % volume accuracy).

As a vessel segmentation tool on input images of poor quality ($1 \leq$ CNR

≤ 4), three fully-automated implementations of SLAVV outperformed the intensity classifier. The peak, voxel-by-voxel classification accuracy of fully-automated SLAVV was above 97 % for all image qualities tested. However, the number of bifurcations detected was very sensitive to noise. A larger fraction of Gaussian in the filtering kernel ($f_G \rightarrow 100\%$) provided greater robustness to noise for bifurcation detection, but resulted in an underestimate of volume even for the higher quality input images.

5. Conclusion

We present the SLAVV method to vectorize microvascular networks directly from unprocessed, in vivo 2PM images of the mouse cortex. This vectorization method removes any preprocessing requirements (image segmentation, interpolation, denoising) by utilizing linear filtering and low-complexity vector extraction algorithms. The SLAVV method is shown to perform similarly on plasma- and endothelial-labeled images, enabling statistical calculations for the bulk network (length, area, volume, bifurcation frequency) and for individual capillaries, arterioles, or venules. Automated and manual processing costs are shown to scale proportionally with input image volume. Fully-automated SLAVV performance is proven robust to image quality compared to a common intensity-based thresholding approach. The SLAVV method is expected to enable longitudinal capillary tracking through accurate and efficient vectorization of time-lapse, in vivo images of mouse cerebral microvasculature.

6. Discussion

6.1. Advantages of the direct vectorization approach

The SLAVV method is advantageous because it is robust to input signal shape, quality, and resolution. Through efficient linear filtering, vessel centerlines and sizes are estimated along with an "energy" or goodness of fit metric. Vessel objects are extracted using simple algorithms to utilize the energy information along with size and topological (connectivity) constraints. Unlike image segmentation approaches, which classify voxels as true or false from grayscale images, the SLAVV approach directly extracts elementary vectors from the grayscale image. In doing so, it removes the needs for pre- or post-image processing (interpolation, morphological filtering, skeletonization), which improves computational efficiency and removes sensitivities to

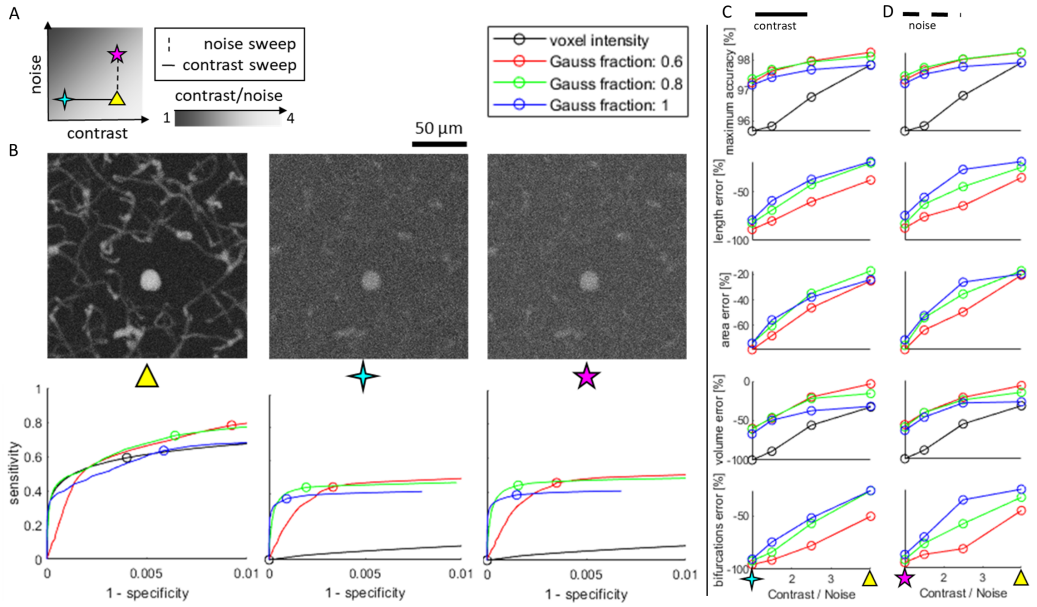

Figure 5: Objective performance of fully-automated SLAVV software. A. Simulated images of varying quality are generated from the vector set in Figure 4. Image quality is swept along contrast and noise axes, independently. Example maximum intensity projections are shown for three extremes of image quality (triangle: best quality, 4-point star: high noise, 5-point star: lowest contrast). The legend shows the labels for the four segmentation methods used in B-D. Images are vectorized using SLAVV with different amounts of Gaussian filter, f_G (60, 80, or 100 % of matched filter length). B. Vasculature is segmented from three simulated images in one of four automated approaches: by thresholding either voxel intensity or maximum energy feature on edge objects produced by three automated vectorizations. Voxel-by-voxel classification strengths of thresholded vectorized objects or voxel intensities are shown as ROC curves for each of the three input images. Operating points with maximal classification accuracy are indicated by circles in the bottom row of B and plotted in the top row of C&D across all input images. C&D. Bulk network statistics (length, area, volume, and number of bifurcations) were extracted from vectors or binary images resulting from maximal accuracy operating points. Performance metrics were plotted against CNR (image quality) for a (C) contrast or (D) noise sweep. Thresholding vectorized objects to segment vasculature demonstrated a greater robustness to image quality than thresholding voxel intensities. Surface area, length, and number of bifurcations were not extracted from binary images, because these images were topologically very inaccurate.

image quality, resolution, and signal shape (spherical vs annular). Although the SLAVV software does not segment the image before extracting vectors, it can nonetheless be used as a segmentation tool (by rendering the extracted vectors at the resolution of the original). Unlike voxel-by-voxel image segmentation methods, the output of SLAVV has guaranteed shape and connectivity constraints that are realistic for vascular networks.

6.2. Performance of SLAVV on endothelial-labeled inputs

An endothelial labeled image, such as Input Image 1 (Figure 3) is very difficult to segment using voxel-by-voxel classification, because the signal only outlines the object instead of filling its volume. The SLAVV approach is robust to this input signal shape (Tables 3-5), because it can be tuned to detect a combination of spherical and annular shapes (Table 2 and Figure 2). However, manual vector classification is less efficient on the endothelial-labeled input Image 1 than the plasma-labeled input Image 2 (Table 5), because the automated vector extraction is less robust to noise for the endothelial signal which was weaker and spanned fewer voxels. Despite these obstacles, the SLAVV method enables the calculation of morphological statistics from endothelial contrast (Table 4) that would otherwise be very difficult. Additionally, because the initial filtering is linear, the superposition principle guarantees good performance on any combination of plasma- and endothelial-labeled inputs.

6.3. Anatomical statistic accuracy

Previous anatomical statistics describing cerebral microvasculature in mice have largely come from two-photon images of post-mortem brain tissue. In a comprehensive review by Tsai et al. (2009), the vascular volume density was reported to be between 0.5 and 1 % and the length density between 0.7 and 1.1 m/mm^3. These results are comparable to the bulk statistics obtained in the cursory anatomical study shown in Figure 4 and Table 4. For example, the vectors extracted from Image 3 had a length density of 0.6 m/mm^3. Interestingly, the corresponding volume density was 6 %. This larger volume estimate could be attributed to a higher vascular perfusion under anesthesia and in vivo compared to post-mortem imaging.

6.4. Fully-automated SLAVV performance

The vectorization software utilized size, shape, and topographical information to achieve a greater robustness to image quality. Therefore, it is unsurprising that there was an image quality threshold below which the vector

energy classifier outperformed the voxel intensity classifier. This objective performance evaluation demonstrates a rigorous framework for testing the accuracy of vectorization techniques in estimating bulk network statistics such as surface area, length, and number of bifurcations.

6.5. Future work

The automated vectorization software will be expanded to accept time-lapse images, register vascular objects between imaging sessions, and detect any changes. Time-lapse images will be analyzed to estimate vascular plasticity statistics such as angiogenesis or angionecrosis. Time-resolved neurovascular statistics in murine cortex will be estimated with high accuracy and precision and in large volume. Good estimates of these statistics will inform fundamental neurovascular research. Measured statistics will be compared between treatment groups in a relevant medical experiment, for example to study the effect of a drug on capillary plasticity in adult murine cortex.

The curation software will be improved by giving the user more control over the displayed objects and more access to object features and properties. Additionally, the interface will be adapted to manually register objects between multiple images. Such an interface will be used to curate an automatic vector registration algorithm for tracking vascular dynamics.

7. Declaration of competing interest

The authors have no conflicts of interest to declare.

8. Acknowledgments

National Institutes of Health (NIH) (EB011556, NS082518, NS108484, T32EB007507)

Appendix A. Algorithm language

The pseudocode language used in Algorithms 1 and 2 resembles MATLAB or Python (Van Rossum and Drake Jr, 1995) syntax. For easier interpretation, <u>functions</u> are underlined, **variables** are bolded, and *keywords* are italicized.

References

Araújo, R.J., Cardoso, J.S., Oliveira, H.P., 2019. A deep learning design for improving topology coherence in blood vessel segmentation, in: International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer. pp. 93–101.

Cao, Y., Wu, T., Li, D., Ni, S., Hu, J., Lu, H., et al., 2015. Three-dimensional imaging of microvasculature in the rat spinal cord following injury. Scientific reports 5, 12643.

Clark, T.A., Sullender, C., Kazmi, S.M., Speetles, B.L., Williamson, M.R., Palmberg, D.M., Dunn, A.K., Jones, T.A., 2019. Artery targeted photothrombosis widens the vascular penumbra, instigates peri-infarct neovascularization and models forelimb impairments. Scientific reports 9, 2323.

Cudmore, R.H., Dougherty, S.E., Linden, D.J., 2017. Cerebral vascular structure in the motor cortex of adult mice is stable and is not altered by voluntary exercise. Journal of Cerebral Blood Flow & Metabolism 37, 3725–3743.

Damseh, R., Pouliot, P., Gagnon, L., Sakadzic, S., Boas, D., Cheriet, F., Lesage, F., 2018. Automatic graph-based modeling of brain microvessels captured with two-photon microscopy. IEEE journal of biomedical and health informatics .

Dirnagl, U., 2006. Bench to bedside: the quest for quality in experimental stroke research. Journal of Cerebral Blood Flow & Metabolism 26, 1465–1478.

Fisher, M., Feuerstein, G., Howells, D.W., Hurn, P.D., Kent, T.A., Savitz, S.I., Lo, E.H., 2009. Update of the stroke therapy academic industry roundtable preclinical recommendations. Stroke 40, 2244–2250.

Girouard, H., Iadecola, C., 2006. Neurovascular coupling in the normal brain and in hypertension, stroke, and alzheimer disease. Journal of applied physiology 100, 328–335.

Haft-Javaherian, M., Fang, L., Muse, V., Schaffer, C.B., Nishimura, N., Sabuncu, M.R., 2019. Deep convolutional neural networks for segmenting

3d in vivo multiphoton images of vasculature in alzheimer disease mouse models. PloS one 14, e0213539.

Hassan, A.M., Wu, X., Jarrett, J.W., Xu, S., Yu, J., Miller, D.R., Perillo, E.P., Liu, Y.L., Chiu, D.T., Yeh, H.C., et al., 2019. Polymer dots enable deep in vivo multiphoton fluorescence imaging of microvasculature. Biomedical optics express 10, 584–599.

Jerman, T., Pernuš, F., Likar, B., Špiclin, Ž., 2016. Enhancement of vascular structures in 3d and 2d angiographic images. IEEE transactions on medical imaging 35, 2107–2118.

Kozberg, M., Hillman, E., 2016. Neurovascular coupling and energy metabolism in the developing brain, in: Progress in brain research. Elsevier. volume 225, pp. 213–242.

Lang, S., Müller, B., Dominietto, M.D., Cattin, P.C., Zanette, I., Weitkamp, T., Hieber, S.E., 2012. Three-dimensional quantification of capillary networks in healthy and cancerous tissues of two mice. Microvascular research 84, 314–322.

Lecrux, C., Hamel, E., 2011. The neurovascular unit in brain function and disease. Acta physiologica 203, 47–59.

Lee, J., Jiang, J.Y., Wu, W., Lesage, F., Boas, D.A., 2014. Statistical intensity variation analysis for rapid volumetric imaging of capillary network flux. Biomedical optics express 5, 1160–1172.

Lowe, D.G., 2004. Distinctive image features from scale-invariant keypoints. International journal of computer vision 60, 91–110.

Perillo, E.P., Jarrett, J.W., Liu, Y.L., Hassan, A., Fernée, D.C., Goldak, J.R., Bonteanu, A., Spence, D.J., Yeh, H.C., Dunn, A.K., 2017. Two-color multiphoton in vivo imaging with a femtosecond diamond raman laser. Light: Science & Applications 6, e17095–e17095.

Preibisch, S., Saalfeld, S., Tomancak, P., 2009. Globally optimal stitching of tiled 3d microscopic image acquisitions. Bioinformatics 25, 1463–1465.

Sato, Y., Nakajima, S., Atsumi, H., Koller, T., Gerig, G., Yoshida, S., Kikinis, R., 1997. 3d multi-scale line filter for segmentation and visualization of curvilinear structures in medical images, in: CVRMed-MRCAS'97, Springer. pp. 213–222.

Savitz, S.I., 2007. A critical appraisal of the nxy-059 neuroprotection studies for acute stroke: a need for more rigorous testing of neuroprotective agents in animal models of stroke. Experimental neurology 205, 20–25.

Schindelin, J., Arganda-Carreras, I., Frise, E., Kaynig, V., Longair, M., Pietzsch, T., Preibisch, S., Rueden, C., Saalfeld, S., Schmid, B., et al., 2012. Fiji: an open-source platform for biological-image analysis. Nature methods 9, 676–682.

Smith, A.F., Doyeux, V., Berg, M., Peyrounette, M., Haft-Javaherian, M., Larue, A.E., Slater, J.H., Lauwers, F., Blinder, P., Tsai, P., et al., 2019. Brain capillary networks across species: a few simple organizational requirements are sufficient to reproduce both structure and function. Frontiers in physiology 10, 233.

STAIR, S.T.A.I.R., 1999. Recommendations for standards regarding preclinical neuroprotective and restorative drug development. Stroke 30, 2752.

Teikari, P., Santos, M., Poon, C., Hynynen, K., 2016. Deep learning convolutional networks for multiphoton microscopy vasculature segmentation. arXiv preprint arXiv:1606.02382 .

Tsai, P.S., Kaufhold, J.P., Blinder, P., Friedman, B., Drew, P.J., Karten, H.J., Lyden, P.D., Kleinfeld, D., 2009. Correlations of neuronal and microvascular densities in murine cortex revealed by direct counting and colocalization of nuclei and vessels. Journal of Neuroscience 29, 14553–14570.

Van Rossum, G., Drake Jr, F.L., 1995. Python reference manual. Centrum voor Wiskunde en Informatica Amsterdam.

Yousefi, S., Liu, T., Wang, R.K., 2015. Segmentation and quantification of blood vessels for oct-based micro-angiograms using hybrid shape/intensity compounding. Microvascular research 97, 37–46.

Yousefi, S., Qin, J., Zhi, Z., Wang, R.K., 2013. Label-free optical lymphangiography: development of an automatic segmentation method applied to

optical coherence tomography to visualize lymphatic vessels using hessian filters. Journal of biomedical optics 18, 086004.

Zipfel, W.R., Williams, R.M., Webb, W.W., 2003. Nonlinear magic: multiphoton microscopy in the biosciences. Nature biotechnology 21, 1369.