Published in final edited form as:

IEEE J Biomed Health Inform. 2019 November ; 23(6): 2551–2562. doi:10.1109/JBHI.2018.2884678.

Automatic Graph-based Modeling of Brain Microvessels Captured with Two-Photon Microscopy

Rafat Damseh[1] **[Student Member, IEEE]**, **Philippe Pouliot**[2,3], **Louis Gagnon**[4], **Sava Sakadzic**[5], **David Boas**[6], **Farida Cheriet**[7] **[Senior Member, IEEE]**, **Frederic Lesage**[1,2,3]

Rafat Damseh: rafat.damseh@polymtl.ca

[1]Institute of Biomedical Engineering, École Polytechnique de Montréal, Montreal, QC, Canada;

[2]Department of Electrical Engineering, École Polytechnique de Montréal, Montreal, QC, Canada;

[3]Research Centre, Montreal Hearth Institute, Montreal, QC, Canada;

[4]Physics Department, Université Laval, Quebec, QC, Canada;

[5]Athinoula A. Martinos Center for Biomedical Imaging, Department of Radiology, Massachusetts General Hospital, Harvard Medical School, Charlestown, MA, USA;

[6]Neurophotonics Center, Department of Biomedical Engineering, Boston University, Boston, MA, USA;

[7]Department of Computer and Software Engineering, École Polytechnique de Montréal, Montreal, QC, Canada.

Abstract

Graph models of cerebral vasculature derived from 2-photon microscopy have shown to be relevant to study brain microphysiology. Automatic graphing of these microvessels remain problematic due to the vascular network complexity and 2-photon sensitivity limitations with depth. In this work, we propose a fully automatic processing pipeline to address this issue. The modeling scheme consists of a fully-convolution neural network to segment microvessels, a 3D surface model generator and a geometry contraction algorithm to produce graphical models with a single connected component. Quantitative assessment using NetMets metrics, at a tolerance of 60 μm, false negative and false positive geometric error rates are 3.8% and 4.2%, respectively, whereas false negative and false positive topological error rates are 6.1% and 4.5%, respectively. Our qualitative evaluation confirms the efficiency of our scheme in generating useful and accurate graphical models.

Keywords

Cerebral microvasculature; deep learning; convolution neural networks; segmentation; graph; two-photon microscopy

I. Introduction

WITH the emergence of two-photon microscopy (2PM), it has become feasible to obtain microscopic measurements of the cerebral vascular geometry and, more recently, of associated oxygen distributions [1], [2]. Such investigations of cerebral microvasculature is essential to understand brain neurovascular coupling and neuro-metabolic activity [3]. It can also help in underpinning neurological pathologies associated with microvascular ischemic changes [4]. Furthermore, microscopic studies can be scaled: quantitative analysis of cerebral microvasculature has recently been used to establish a link between the microscopic vascular phenomena and the macroscopic (voxel size) observations found in the blood oxygenation level-dependent (BOLD) response [5], [6]. A key problem encountered in the efforts of [5] and [6] to model cerebral microvasculature is the construction of a sufficiently complete and detailed vascular network from noisy 2PM data. Physiological simulations were based on vascular connectivity which in turn required extensive manual annotations to reconstruct the microvascular topology of each dataset. This human interaction limits the applicability of such simulation frameworks in further neurological studies that would require scaling to large datasets.

Imaging brain tissue using 2PM is associated with high absorption and scattering of the emitted photons with depth leading to image degradation and intensity changes as laser power and detector gains are dynamically adjusted with depth. Also, scattering of excited photons by red blood cells degrades their focus and leads to shadows underneath large pial vessels. These deteriorations in the excited and emitted photons yields volumetric image intensity variations such that automatic processing of two-photon microscopic data is tedious and even problematic [1]. Modeling of angiographic information from 2PM includes segmentation of captured microvessels [7], [8], computation of the microvasculature network shape and topology [9], [10] and anatomical labeling of the extracted components (e.g. arterioles, venules and capillaries). Until recently, some work has been done to automate the processing of these microscopic datasets. However, the developed techniques were not sufficient to avoid significant manual corrections. With the emergence of sophisticated microvascular modeling, the design of a fully-automated scheme that is capable of generating topological models for vasculature in microscopy data is required to scale previous studies. This is the goal of this research work.

Numerous schemes for vascular segmentation have been proposed in the literature exploiting various image properties, such as the Hessian matrix [11], [12], moments of inertia [13], geometrical flux flow [14] and image gradient flux [15]. These schemes also vary in their segmentation strategies, e.g., vessellike prior modeling [11], [12], [15], tracing [16], evolution of deformable models [14] and deep learning [7], [17]–[19]. Applications to automatically segment microvessels captured with 2PM are further limited due to the very large number of segments, uneven intensities associated with optical imaging and shadowing effects. Previous studies have heavily relied on manual interactions to obtain satisfactory segmentations [5]. In recent works, models inspired by the success of ConvNets have been proposed to provide very good vascular segmentation [7], [17]–[19] with some applied to segmentation of microvessels in 2PM [7], [18]. In [7], the authors implemented a recursive architecture composed of 3D convolution blocks. However, their shallow model provides poor pixel-wise segmentation and is computationally demanding. In [18], the proposed scheme was not targeted to label the vascular space, but rather to provide a skeleton-like version of it. Other recent work, based on FC-ConvNets [20], performed end-to-end vessel segmentation using images from volumetric magnetic resonance angiography [19].

However, the proposed scheme is patch-based, which makes it hard to apply when there is significant variation in local features as observed in 2PM.

Automated representations of data topology has been of importance in many research disciplines: aerial remote sensing [21], [22], neuroinformatics [23]–[26], and vascular imaging [9], [27], [28]. Some techniques [24]–[26] target the reconstruction of tree-like structures, by first generating seed points based on probabilistic measurements to then apply algorithms, such as Shortest Path Tree [24], Minimum Spanning Tree (MST) [25] or kMST [24], to extract an optimal graph. On the other hand, modeling loopy curvilinear structures has been investigated in [9], [21]–[23], [28]. The problem we aim to solve in this paper resembles that investigated in the latter work since 2PM angiograms capture capillaries connecting the arteries to the veins which together also form a topology that is not tree-like. In [21], intensities in road images are clustered into superpixels. The shortest path algorithm is then used to assign connections between superpixels based on road likelihood. Optimal subsets of the connections are finally processed in a framework of conditional random fields. In [22], a thinned version of segmented road maps is simplified and then passed through a shortest path algorithm [22] to generate an undirected graph for road networks. The works in [21], [22], are designed only for two dimensional natural images that have low scale variability and semi-constant luminosity, and which have few disconnected components. In [27], automatic topological annotation of macroscopic cerebral arteries forming the circle of Willis was achieved. However, the task was performed with a predefined knowledge about the topological structure of the annotated object. In [9], [23], [28] different schemes have been developed to extract graphs from more complex three dimensional images (microscopy data) by optimizing a designed objective function. These schemes first attempt at building an overcomplete weighted connections between seed points, which are detected based on tabularity [9] or bifurcation [23], [28] measures. They then search for optimal subgraph representations by solving a linear integer programming (LIP) problem with specific constraints.

In [9], [23], [28], pre-processing steps to provide preliminary weighted graphs, forming the basis of a LIP solution, are designed for less-noisy datasets containing tubular structures of high size uniformity. High noise level and variation in tubular sizes are exhibited across microscopic angiograms acquired with two-photon fluorescence scanning. Moreover, despite the cautious problem formulation in the mentioned works designed to approach a near-optimal solution, LIP is theoretically a non-deterministic polynomial-time hard (NP-hard) problem. The hardness of the problem increases for applications on scalable datasets, which results in more decision variables. Lastly, practical solutions of LIP do not necessarily provide consistent output at different runs.

Our work below addresses the issue of extracting topological models from scalable and complex datasets, i.e. 2PM angiograms. We propose a fully-automated solution that provides a unique output for the same input data, in practical computational time. The proposed graph extraction scheme consists of 3 main stages, described in Figure 1. First, we take advantage of a recent development in deep learning semantic segmentation [20], [29] employing a fully convolution (FC) network based on the DenseNet architecture to segment potential microvessels [29]. A new well-labeled 2PM dataset is manually prepared to train our segmentation model. The volumetric output of the segmentation model is processed by 3D morphological filters to omit small isolated segments and improve the connectivity pattern of microvessels. Second, we propose a data processing flow to generate a polygonal closed-manifold geometry for the microvasculature from its volumetric mask obtained in the first stage. Finally, we exploite the 3D geometric skeletonization [30] to generate a direct representation of the microvessel network.

The paper is organized as follows. In the Methods Sections II–IV, we describe each phase of the graph extraction scheme: Section II details the proposed deep learning architecture for microvessels segmentation and explains the subsequent morphological refinements, Section III describes the techniques used to generate and post-process the polygonal mesh to represent the shape of the microvasculature, and Section IV provides the formalism of the 3D skeletonization to produce a final graph-based model of the microvascular network. The Results Section V demonstrates the validity of the proposed modeling scheme. A discussion and concluding remarks are provided in Section VI.

II. Deep Segmentation of Microvessels

In this section, various versions of the recently developed FC-DenseNets [29] are employed to address the problem of microvessel segmentation in large-scale 2PM images. Our use of the FC-DenseNets architectures is inspired by their success in achieving the state-of-the-art performance in semantic segmentation in natural images. We prove that applying this data-driven solution to segment microvessels in 2PM volumes can induce a substantially improved performance compared to that of other hand-crafted schemes. Below we discuss the architectures of the neural networks that were devised, data preparation and training procedure. Let $\mathscr{I}: P \rightarrow \mathbb{R} \mid P \subset \mathbb{R}^3$ be the 3D image representing the observed microscopic measurements. The aim is to create a binary mask $\mathscr{I}_{mask}: P \rightarrow \{1, 0\} \mid \mathscr{I}_{mask}(p) = 1 \quad \forall p \in \mathscr{O}$ and $\mathscr{I}_{mask}(p) = 0$ otherwise, where $\mathscr{O} \subset \mathbb{R}^3$ is the object that represents the vasculature.

A. Networks Architecture

Generally, in a neural network with layers l_n, $n = 0,1,2,\ldots$, N, each layer l_n performs a non-linear transformation $H_n(\cdot)$ that might include several operations, e.g. batch normalization, rectified linear units (ReLUs), convolutions or pooling. Let x_n denote the output of the layer l_n. In DenseNets [31], consisting of building units called dense blocks, the output x_n of the layer l_n is connected to its previous layers contained in the same dense block [31] by cascading operations. The layer transition is therefore described by the following formula:

$$x_n = H_n\left(\left[x_{n-m-1}, \ldots, x_{n-2}, x_{n-1}\right]\right) \tag{1}$$

where $[\cdot]$ operation denotes the concatenation process and m denotes the number subsequent layers contained in a dense block.

DenseNets are extended to a fully convolution scheme [29], analogous to that proposed in [20], comprising down- and up-sampling paths. The down-sampling pattern is composed of dense blocks and transition-down (T-down) layers, where T-down layers consist of three operations: batch normalization, convolution and max-pooling. Alongside the down-sampling path, the up-sampling path is inserted to recover the input spatial resolution. It is composed of dense blocks, transition-up (T-up) layers and skip connections where T-up layers are built of transposed convolution operations. To avoid the increase in the number of feature-maps in the up-sampling path, the input of dense blocks in this path is not concatenated with its output. Henceforth, we refer to dense blocks in the down-sampling path as *type A*, and that of the up-sampling path as *type B*. As in typical fully convolution networks [20], skip connections – from the down-sampling path to the up-sampling one – have been introduced to allow the network to compensate for the loss of information due to the pooling operations in the T-down layers.

In this work, to perform the segmentation task, we investigate three different architectures of FC-DenseNets varying in their deepness, namely, Net71, Net97 and Net127. Our architectures consist of 71, 97 and 127 convolution layers, respectively, with an input size of 256×256×1 for each. The detailed specifications for implemented convolution and pooling layers, with their number of feature-maps, in each architecture are listed in Table I. The down-sampling and up-sampling paths have 36 convolution layers each, whereas the networks bottlenecks have 11, 13 and 15, respectively. The down-sampling path is composed of dense blocks *type A*, that have input-output concatenation. This concatenation is omitted in the dense blocks *type B* forming the up-sampling path. Each T-down block is composed of one convolution layer followed by dropout [32] with $p = 0.2$ and non-overlapping max-pooling. Each T-up block is a transposed-convolution layer with $stride = 2$. The growth rate of feature maps in each dense block is set to $k = 18$ and a dropout of $p = 0.2$ is applied in all contained layers. In each architecture, one convolution layer is applied on the input and another is placed before the last (softmax) layer that provides the bi-class predictions.

B. Data Collection

To train our segmentation networks, manual annotations were performed to create a ground truth labeling of mice cerebral microvessels captured using our custom-built two-photon laser scanning microscope. To image microvessels, 200 μL 2MDa dextran-FITC (50 mg/ml in saline, Sigma) was injected through the tail vein of mice. Due to the injected fluorescent dye, the plasma appeared bright in the images while red blood cells (RBCs) appeared as dark shadows. Acquisition was performed using 820 nm, 80 MHz, 150 fs pulses from MaiTai-BB laser oscillator (Newport corporation, USA) through an electro-optic modulator (ConOptics, USA) to adjust the gain. The optical beam was scanned in the x-y plane by galvanometric mirrors (Thorlabs, USA). Reflected light was collected by a 20× objective (Olympus XLUMPLFLNW, NA=1). Fluorescent photons were separated by dichroic mirrors, passed through a filter centered at 520 nm and relayed to a photomultiplier tube (PMT, R3896, Hamamatsu Photonics, Japan) for detection of the dextran-FITC. Manual annotations from the 3D microscopic measurements were done slice-by-slice. The annotation process was carried out with the assistance of MayaVi visualization tool to consider the final 3D structure of the microvessel network. Eight angiograms were labeled to produce a training dataset of images $\mathbf{T} = \left\{\left(\mathbf{I}^{(q)}, \mathbf{I}_{mask}^{(q)}\right) | q \in Q = [1, 2, 3, \ldots, 396]\right\}$, each of size 256 × 256. Next we omit the superscript q for notational simplicity.

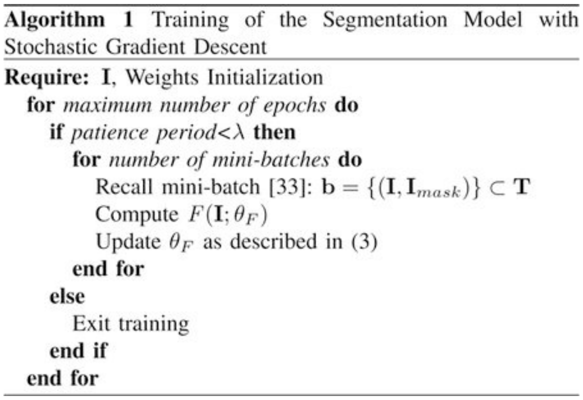
C. Networks Objective & Training

Let $p_x, p_y, p_z \in \mathbb{R}$ define the dimensions of one of our networks output $F(\mathbf{I};\ \theta_F)$, where θ_F is vector containing the model parameters. The model was trained to minimize the following cross-entropy loss function [33]:

$$\Psi(F) = -\Big[\Sigma_{p_x, p_y} \mathbf{I}_{mask} \log\big(F(\mathbf{I};\theta_F)\big|p_z = 0\big) + \Sigma_{p_x, p_y}\big(1 - \mathbf{I}_{mask}\big)\log\big(F(\mathbf{I};\theta_F)\big|p_z = 1\big)\Big]. \qquad (2)$$

The stochastic RMSprop gradient descend algorithm was employed to solve our model parameters based on backpropagation [33]. Given an initial learning rate ρ and a decay parameter γ, model parameters were updated as follows:

$$\theta_{F_{(t+1)}} = \theta_{F_{(t)}} - \frac{\rho}{\sqrt{\zeta_{(t)}}}\nabla_{\theta_F}\Psi\big(F(\mathbf{I};\theta_F)\big),\ \text{where} \quad \zeta(t) = (1-\gamma)\Big[\nabla_{\theta_F}\Psi\big(F(\mathbf{I};\theta_F)\big)\Big]^2 + \gamma\zeta_{(t-1)}. \qquad (3)$$



Algorithm 1 Training of the Segmentation Model with Stochastic Gradient Descent

```
Require: I, Weights Initialization
  for maximum number of epochs do
    if patience period<$\lambda$ then
      for number of mini-batches do
        Recall mini-batch [33]: $\mathbf{b} = \{(\mathbf{I}, \mathbf{I}_{mask})\} \subset \mathbf{T}$
        Compute $F(\mathbf{I};\theta_F)$
        Update $\theta_F$ as described in (3)
      end for
    else
      Exit training
    end if
  end for
```

The training procedure is described in Algorithm 1. In each training iteration, a set of samples from our dataset ("mini-batch") was processed by the network models with their current parameters. Then, the gradient of the loss function Ψ was computed with respect to θ_F and parameters were updated by stepping in the descending direction of the gradient as stated in (3). In the algorithm, the training process was early-stopped if no minimization of the loss function Ψ was achieved after a certain number of consecutive training epochs, denoted as the patience period λ.

D. Morphological Post-Processing

The trained segmentation models can accept a 2D input **I** of an arbitrary size and produce an output $F(\mathbf{I})$ of the same size. To segment microvessels in a 3D two-photon angiogram and obtain I_{mask}, we first stacked the outputs of the trained neural networks applied to each slice in the z-direction. Morphological 3D closing and opening filtering was then applied on the resultant image stack with a spherical shape filter of radius=8 to refine the stacked segmentation outputs. It has been found that by applying this 3D morphological processing to obtain I_{mask}, a reduction in false positive structures emerging after the 2D segmentation is achieved.

III. Surface Modeling

This section describes the process used to construct a triangulated mesh $\mathscr{M} = (\mathbf{V}, \mathbf{C})$ representing the shape of microvasculature, where $\mathbf{V} = \left[\mathbf{v}_1^T, \mathbf{v}_2^T, \ldots, \mathbf{v}_d^T\right]^T$ are the vertices positions, $\mathbf{v} \in \mathbb{R}^3$, and $\mathbf{C} = \left\{\mathbf{c}_i\right\}_{i = 1, \ldots, l}$, $\mathbf{c} \in \mathbf{v} \times \mathbf{v}$, are the edges connecting the vertices of the mesh. The extracted mesh should form a 2D manifold to allow successful geometrical contraction and convergence towards a curve skeleton formulation. Many computational paradigms have been proposed to provide isosurface representations of the binary output $\mathscr{I}_{mask}$ obtained in the previous section. Examples are those based on the concepts of Marching Cubes (MC) [34]–[36], dual contouring [37] and advancing-front construction [38]. The MC algorithm [34] is arguably the most widely used due to its robustness and speed, however it has an ambiguity problem in its classical formulation due to lookup table redundancy. MC with asymptotic decider [35] was proposed to give better, topologically consistent, surface models. In [36], an interior ambiguity test was added to the scheme of [35] to achieve even higher topological correctness. A modified version of [36], introduced by Liener et. al.[39], was utilized to generate an initial triangulated mesh model for the microvessels surface.

The output from the MC algorithm is a polygonal mesh with a high level of triangulation redundancy. The algorithm produces a large number of small coplanar triangles resulting in a big size model. Direct processing of this raw 3D model to generate a curve skeleton abstraction would be computationally expensive. Also, the MC process introduces excessive roughness with stair-step effects that need to be removed. We applied a polygonal simplification procedure to circumvent these concerns. Many mesh simplification approaches have been developed in the literature [40]–[44] with varied characteristics: topological invariance, view-dependency – based on objects location, illumination and motion in the scene – and the polygonal removal mechanism. The vertex clustering technique in [40] works by superimposing a grid with a predetermined resolution to the cells resulting after clustering the original meshes vertices. The resampling technique [41] is based on performing low-pass filtering of the original volumetric data. These approaches are topology insensitive and provide a drastic reduction in the number of elements of the original geometry. Other simplification approaches [42], [43] have been developed to ensure appearance/rendering quality in complex scenes that hinder the rendering process by assuming different levels of detail for the discovered objects. The vertex merging technique proposed by Garland et al. [44] gives an accurate geometry for the simplified model and works on non-manifold inputs that could result from the MC process [39]. We used these features of the vertex merging technique [45] to simplify our surface model.

The process undertaken to create a surface model is not guaranteed to be a manifold and even less a closed manifold [46]. Further processing was used to eliminate possible non-manifold local defects such as self-intersections based on the work done in [47]. Lastly, the hole filling algorithm illustrated in [48] was used to ensure a watertight surface model that can be further used in the next stage of our modeling pipeline.

IV. Graph Extraction

Various definitions have been suggested as a basis to compute formal skeletons (also called medial axes) [49]. These definitions emerged from conceptualizing the skeleton based on maximally-inscribed balls, Grassfire Analogy, Maxwell Set or Symmetry Set [50]. Let $\mathscr{S} = d\mathscr{O}$ be the 2D-manifold surface of the 3D-manifold volume $\mathscr{O}$ of interest. The idea of maximally-inscribed balls defines the skeleton of $\mathscr{O}$ to be the locus of the centers of maximally inscribed balls in $\mathscr{O}$. Extension to the Medial Axis transform is done by associating the radii of these balls to their center locations. The Grassfire analogy defines the skeleton as the locations inside $\mathscr{O}$ where quenching happens if we let the surface $\mathscr{S}$ propagate isotropically toward the interior of $\mathscr{O}$. The Maxwell Set definition captures the curve skeleton as an extension of the Voronoi diagram by encoding the loci that are equidistant from at least two points on the surface $\mathscr{S}$. In the Symmetry Set definition, a curve skeleton is regarded as the infinitesimal symmetry axes generated by first linking each pair-point on the surface to calculate their symmetry centers and then combine these centers to form a curvilinear infinitesimal axis. In all definitions, skeletonization depends only on the shape of $\mathscr{O}$ rather than its position or size in its embedding space.

In this study, the target is to generate a graphed form of 1D curve skeletons that capture the topology of the microvasculature. In general, the medial axes for 3D objects are 2D [49]. Therefore, two main solutions have been proposed, following the procedures to either simplify the 2D medial axes [51], [52] or to calculate the curve skeletons directly from 3D objects [30], [53], [54]. The techniques in [51], [52] are computationally expensive and very dependent on the quality of the associated surface skeletonization. From a geometric perspective, the techniques that are based on shape contraction [30], [54] are proven to be the most successful in extracting accurate and yet smooth curve skeletons from a variety of different shapes. Moreover, such techniques provide skeletonization embedded in the geometrical space, and thus it coincides with our goal in generating a graph-based representation of microvessel networks. Therefore, we adapt the work proposed in [30] to perform graph extraction in our last stage of the designed pipeline.

A. Geometric Contraction

Based on the triangulated mesh $\mathscr{M} = (\mathbf{V}, \mathbf{C})$ generated in the previous section, our goal is to extract a geometric graph $\mathscr{G}$ with node positions $\mathbf{N} = \left[\mathbf{n}_1^T, \mathbf{n}_2^T, \ldots, \mathbf{n}_m^T\right]^T, \mathbf{n} \in \mathbb{R}^3$, and edges (connections) $\mathbf{E} = \{\mathbf{e}_i\}_{i=1,\ldots,h}$, $\mathbf{e} \in \mathbf{n} \times \mathbf{n}$. We follow the geometrical contraction technique in [30] to achieve our goal by iteratively moving the vertices of $\mathscr{M}$ along the corresponding curvature normal directions. At iteration t, we minimize the constrained energy function:

$$\begin{aligned}\mathscr{E} = & \left\| \mathbf{W}_S \mathbf{L} \mathbf{V}^{t+1} \right\|^2 + \\ & \sum_{i=1}^{d} \mathbf{W}_{V,i}^2 \left\| \mathbf{v}_i^{t+1} - \mathbf{v}_i^t \right\|^2 + \\ & \sum_{i=1}^{d} \mathbf{W}_{M,i}^2 \left\| \mathbf{v}_i^{t+1} - \xi(\mathbf{v}_i) \right\|^2 \end{aligned} \quad (4)$$

where $\mathbf{W}_S$, $\mathbf{W}_V$ and $\mathbf{W}_M$ are diagonal matrices of size $d \times d$ that impose constraints on the geometrical contraction process, such that for the ith diagonal element: $\mathbf{W}_{S,i} = w_S$, $\mathbf{W}_{V,i} = w_V$ and $\mathbf{W}_{M,i} = w_M \cdot w_S$, w_V and w_M serve as tuning parameters to control the smoothness, velocity and mediality of the contraction process at each iteration. In (4), the function $\xi(\cdot)$ maps the vertex $\mathbf{v}_i$ to the corresponding Voronoi pole calculated before the iteration process [30], and $\mathbf{L}$ is the $d \times d$ curvature-flow Laplace operator with its elements obtained as:

$$\mathbf{L}_{ij} = \begin{cases} w_{ij} = \cot(\alpha_{ij}) + \cot(\beta_{ij}) & i, j \in \mathbf{C} \\ -\sum_{(i,k) \in \mathbf{C}}^{k} w_{i,k} & i = j \\ 0 & otherwise, \end{cases} \quad (5)$$

where α_{ij} and β_{ij} are the angles opposite to the edge (i, j) in the two triangles which have this edge in common [55].

B. Geometric Decimation

At each iteration step in the geometric contraction process, higher levels of local anisotropy is introduced in the processed mesh. We follow the procedure described in [54] to reduce the mesh at each contraction step maintaining a water-tight manifold structure. After completing the iteration process, the degenerated surface mesh is reduced to form a graph model by applying a series of shortest edge-collapses [54] until all mesh faces have been removed. Lastly, to ensure that the output of our modeling framework is valid for running physiological simulations, we extract the largest single connected component (LSCC) of the resulting graph to be our final graphical model.

V. Validation Experiments

In this section, we carry out various experimentations to evaluate the designed pipeline in producing graph-based representation for microvessels captured with 2PM. First, we describe the parametric settings used to train the segmentation model, extract 3D surface models and generate final geometric graphs. We illustrate the manual procedure followed to prepare a graph-based ground truth baseline. Next, we discuss the validation mechanism and the metrics used to study the performance of the designed pipeline. Lastly, based on these metrics, we present a comprehensive evaluation of our proposed modeling scheme.

A. Baseline, Parameters and Implementation

Here we discuss the issue of finding a baseline performance for the purpose of validation and the parametric setup of the model used to generate results. Also we describe implementation details.

1) Baseline: To study the segmentation performance of the our deep segmentation models before and after applying the 3D morphological processing (3DM). Segmentation results were compared with that of manual segmentation (gold-standard), optimally oriented flux method [15] (OOF) and the hessian-based method in [12], which are state-of-the techniques used in 3D vascular segmentation.

To our knowledge, this work is the first fully automatic scheme designed to extract graph-based representation from sizable 3D 2PM datasets. Other baseline methods in the literature applicable to similar datasets rely on human interaction. Hence, in order to validate the correctness of our graphing scheme, we compare our results with manually prepared graphs recently utilized in [5], [56]. To create the manually-graphed datasets in [5], [56], manual annotations were performed on contrast-enhanced structural images of the cortical vasculature which were then skeletonized using erosion. Ensuing graphs were hand-corrected and verified by visual inspection to generate a single connected component. Datasets of six graphs (from six different mice) were prepared from angiograms with 1.2 × 1.2 × 2.0 μm voxel sizes acquired with a $20X$ Olympus objective (NA= 0.95). To illustrate our choice of the surface modeling and mesh contraction processes in the proposed pipeline, we compare our graph outputs with that resulting from applying the widely used 3D Thinning method [57] on the same binary masks.

2) Training the Segmentation Model: Implementation of the deep-learning portion of the pipeline was done in Python under the Theano framework. We ran the experiments on two 12GB NVIDIA TITAN X GPUs. The model was initialized using the HeUniform method as in [29]. The training was performed by setting $\rho = 10^{-3}$ and $\gamma = 0.995$ in (3) for all epochs of training. To monitor the training process, we randomly selected 25% of the training data as a validation set. The selection of the validation set was performed by randomly picking several 2D slices after grouping all the slices forming the 2PM stacks in one pile. The validation set was used to early-stop the training based on the acquired accuracy, with a patience of 25 epochs. The model was regularized with a weight decay of 10^{-5}.

3) 3D Modeling and Contraction Process: We have followed a brute-force selection of parameters used in the processes of creating 3D surface and carrying out geometric contraction, based on the quality of the generated graphs. To reduce the 3D model using the vertex merging technique in [44], we set the target number of faces in the reduced model to be half of that of the original one. Our code for generating and processing the 3D surface model was based on the VTK Python library. The parameters w_S, w_V and w_M were set to 1, 20 and 35, with $\epsilon_{vol} = 10^{-6}$. We built Python bindings using SWIG to call the C++ API, CGAL, containing the implementation of the geometric contraction algorithm.

B. Metrics

To quantify the performance of the various segmentation methods, the metrics of sensitivity = $TN/(TN + FP)$, specificity= $TP/(TP+FN)$, accuracy = $(TP+TN)/(FP+ FN)$ and Intersection Over Union (IoU) = $TP/(TP + FP + FN)$, were used. The terms TP and FP denote respectively the positive predictions (vessel) with true and false ground truth, whereas TN and FN denote respectively the negative predictions (not vessel) with true and false ground truth. It is to be noted that to generate binary masks using the methods in [15] and [12], empirical thresholding is performed to achieve the highest IOU value.

For the quantification and visualization of errors incurred by our graphing scheme, we used the NetMets metrics proposed in [58]: four measures were computed to compare both the geometry and connectivity between two interconnected graphs. Namely, the Graph False Negative/Positive Rate metrics GFNR and GFPR reflected the false negative rate and the false positive rate, respectively, in the geometry of the extracted graph while the metrics CFNR and CFPR provided measures about the false negative rate and the false positive rate in the topology of the graph.

Let us define $\mathscr{G}_r$ and $\mathscr{G}_e$ as the ground truth and experimentally generated graphs. In the validation process using NetMets, we first performed a two-way matching between the junction nodes in both graphs. These nodes, denoted as $\mathbf{J}_r$ and $\mathbf{J}_e$ for $\mathscr{G}_r$ and $\mathscr{G}_e$, respectively, were those representing bifurcations in the vascular network or terminals of vessel pathways. Graph paths between junction nodes were referred to as graph branches $\mathscr{B}$. Junction node mappings $\mathbf{J}_r$ to $\mathbf{J}_e$ were denoted as $\mathbf{J}_r \rightarrow \mathbf{J}_e$, whereas $\mathbf{J}_e \rightarrow \mathbf{J}_r$ denoted the opposite matching processes. In the matching process, we assigned each junction node in the first graph with one in the second graph based on the shortest Euclidean distance measured. GFNR was computed as $1/n_{\mathbf{J}_r} \sum_{i \in \mathbf{J}_r} 1 - e^{D\left(i, \mathscr{G}_e\right)^2/2\delta^2}$ and that of GFPR as $1/n_{\mathbf{J}_e} \sum_{i \in \mathbf{J}_r} 1 - e^{D\left(i, \mathscr{G}_r\right)^2/2\delta^2}$, where $n_{\mathbf{J}_r}$ and $n_{\mathbf{J}e}$ are the number of junction nodes in $\mathscr{G}_r$ and $\mathscr{G}_e$, respectively, $D(i, \mathscr{G})$ is the Euclidean distance between a node i and its matched node in $\mathscr{G}$, and δ is a sensitivity parameter.

Now, to compute CFNR and CFPR as described in [58], core graphs $\breve{\mathscr{G}}_r$ and $\breve{\mathscr{G}}_e$ were extracted from $\mathscr{G}_r$ and $\mathscr{G}_e$. Core graphs were obtained by first reducing both graphs, $\mathscr{G}_r$ and $\mathscr{G}_e$, by eliminating each junction node and its immediate branches if the distance D from the corresponding nodes was greater than δ. Then, in the reduced graphs of $\mathscr{G}_r$ and $\mathscr{G}_e$, we further eliminated junction nodes, and their corresponding branches, that did not form the same matching pair in the two-way matching process. Lastly, we compared the reduced $\breve{\mathscr{G}}_r$ and $\breve{\mathscr{G}}_e$ to calculate the following: $\mathscr{B}_{TP}$ = number of graph branches in $\breve{\mathscr{G}}_r$ or $\breve{\mathscr{G}}_e$; $\mathscr{B}_{FN}$ = number of graph branches in $\mathscr{G}_r$ but not in $\breve{\mathscr{G}}_e$; $\mathscr{B}_{FN}$ = number of graph branches in $\mathscr{G}_e$ but not in $\breve{\mathscr{G}}_e$. CFNR and CFPR were calculated as $\mathscr{B}_{FN}/\left(\mathscr{B}_{FN}+\mathscr{B}_{TP}\right)$, and $\mathscr{B}_{FP}/\left(\mathscr{B}_{FP}+\mathscr{B}_{TP}\right)$, respectively.

C. Results

1) Microvessels Segmentation: In this section, we study the performances of the various segmentation schemes used to generate microvessels binary masks. Table II provides the averaged quantitative results obtained after applying the various segmentation schemes on the 2PM slices in our validation set. From the table, despite the slight degradation in sensitivity scores, one can notice that carrying out the 3DM post-processing on the FC-DenseNets outputs substantially improves the measures of accuracy, specificity and IOU. The table obviously shows that the method in [15] is not a suitable choice for extracting microvessel maps. The highest sensitivity value of 98.9% ± 0.01 was achieved by the Net71 architecture but at the expense of very poor specificity. The method in [12] achieved the highest specificity value of 57.33% ± 2.20. However, it produces vessel masks with poor IOU scores. The table proves that the Net97+3DM architecture achieves the best accuracy and IOU values, 92.3% ± 0.1 and 41.1% ± 0.4, respectively, with comparable values of sensitivity and specificity corresponding to the schemes of Net97 and that in [12], respectively.

To qualitatively assess the segmentation performances, Fig. 2 depicts two raw 2PM slices, obtained from our validation set, with their binary masks counterparts generated by applying the various segmentation schemes. It is clearly seen that the scheme in [15] and those of Net71, Net71+3DM, Net127 and Net127+3DM, produce microvessel mappings that suffer from over-segmentation. On the other hand, the scheme in [12] is highly specific in a way that generates outputs with large portion of false negatives, thus missing important vesselness structures. This observation coincides the low IOU measure of this scheme reported in Table II. From the figure, one can see that the schemes of Net97, Net97+3DM exhibit better segmentation results with superiority of the Net97+3DM scheme in providing less over-segmentation. Hence, based on the previous experimental demonstration, in our study, we choose the scheme Net97+3DM to generate binary maps needed for extracting graph-based models of microvessels.

2) Graph Modeling: The six ground-truth graph datasets were carefully annotated, ensuring a fully interconnected topology for the captured micro vessel networks. Each graph provides a model for descending arterioles, capillary bed, and then ascending venules. These graphical models have been employed in [5] to simulate oxygen-dependent quantities across the network, and then generate synthetic MRI signals based on these measurements. Fig. 3 shows the six maximum intensity projections of the raw cerebral micro vascular spaces captured with 2PM, with their corresponding ground truth and experimental graphical modelings depicted in Fig. 4. The reader is also referred to Supplemental Fig. 1 in the supplementary material for assessing the proposed graph models based on visualizing only thin slices (10–30 μm thick) of the stack in Fig. 1 (a,1) at various depth levels. Empirical visual inspection shows that the outputs of our graph modeling scheme are significantly superior to that of the 3D Thinning method and are globally comparable to the ground-truth annotations. However, some defects are present for some boundary parts of the vascular networks, especially the capillary bed where one can observe that microvessel terminals are present for the experimentally generated models but not in the manually prepared ones. It is to be noted that the experimental models in Fig. 4 are fully interconnected graphs obtained after attaining the LSCC from the raw graph outputs. The ratio of these LSCCs to the original models generated from the first-sixth 2PM datasets, respectively, are listed in Table III. The table demonstrate that the proposed scheme generates less disconnected components compared to that produced by the 3D Thinning method.

As previously mentioned, in order to quantify the performance of the modeling schemes, we performed a two-way matching, based on the shortest Euclidean distance, between junction nodes in the ground truth graphs and those in the generated ones. Fig. 5 plots the estimated probability distribution, using a Gaussian kernel, of the mapping distances D obtained after performing the matching process. It is seen that, in the case of the proposed graph modeling, the $\mathbf{J}_e \rightarrow \mathbf{J}_r$ matching process produced D mappings similar to that resulting from the $\mathbf{J}_e \rightarrow \mathbf{J}_r$ process. This implies that the localization of junction nodes coincides in both manual and experimental modelings, thus resulting a negligible difference between the statistics obtained from the two matching processes. On the other hand, the figure shows a big statistical difference between $\mathbf{J}_e \rightarrow \mathbf{J}_r$ and $\mathbf{J}_e \rightarrow \mathbf{J}_r$ calculated based on the graphs extracted by the 3D Thinning method.

Table III shows the number of nodes in the graphs generated manually, by the 3D Thinning method and by our automatic processing. It is seen from the table that, in all cases, the 3D Thinning method produces sparser models having dramatically fewer nodes than that in the ground truth ones. Our modeling scheme produces fewer nodes in some cases (datasets 1, 5 and 6), whereas the opposite was observed in the other cases. To interpret this, Fig. 6 depicts magnified versions of the dataset 1 graph. It is clearly seen that manually processed graphs are composed of nodes with almost uniform Euclidean distances between all nodes and their neighboring ones. This is due to regraphing procedures [56]. In contrast, our automatic graphing results in non-regularized placement of nodes due to the dependency of the geometric contraction process on the complexity and tortuosity of the surface modeling for microvessels. However, the figure explains the superiority of the proposed modeling scheme compared to the 3D Thinning method in extracting more accurate graphs. Also from Table III, one can note a variation in the number of junction nodes observed in the ground-truth and the experimental models generated by the proposed scheme. Hence, our modeling produces either denser or sparser interconnections than that of the manually processed graphs when applied to different raw 2PM angiographies.

Fig. 7 plots the measurements of the GFNR, GFPR, CFNR, CFPR metrics obtained for the various datasets at different tolerance values δ. It is noted that these measurements are bounded between zero and one; the lower the value the better the performance. The figure illustrates that in all cases, the proposed modeling achieves lower geometrical and topological error rates than that of the 3D Thinning method. At the highest tolerance, $\delta = 60$, our scheme achieves an average of 3.8 ± 2.1% and 4.2 ± 2.6% of false negative and false positive error rates, respectively, in modeling the geometry of the vascular networks. These error rates decrease to 3.5±2.6% and 3.0 ± 1.9% respectively, when excluding boundary microvessels (≈ 25 % of the angiograms) from our assessment. At the same value of δ, our scheme is able to capture the topology of the vascular networks with an average false negative and false positive error rates of 6.1 ± 2.6% and 4.5 ± 2.9%, respectively. When excluding the boundary microvessels, the average error rates decrease to 4.4 ± 2.8% and 1.5 ± 0.9%, respectively. It is to be noted the 3D Thinning method achieves even higher false negative rates when excluding the boundary microvessles, thus proving that this method is poor in capturing the geometrical and topological details from the segmented angiograms. We provide a visual illustration in Fig. 8 to interpret the improved performance when excluding the boundary vascular structures in the experimental graphs generated by the proposed scheme. The figure shows the false positive and false negative edges obtained at $\delta = 60$ to compute the corresponding CFNR, CFPR values. One can inspect that a large portion of the mismodeled edges are those located at the boundary of the angiograms. As a result, a better performance of our modeling is achieved at the center of the angiograms, due to 2-photon sensitivity limitations. One goal of our proposed scheme is to generate topological models of microvessel networks that can be used for calculating physiological quantities. Towards that goal, we study the propagation of microvessels in the modeled networks. Fig. 9 depicts examples of ground-truth and experimental sub-graphs generated by propagating through the microvessel networks, beginning with nodes at the start pf a penetrating arteriole and ending with bifurcation nodes after 5 levels of consecutive branching. Despite some few mismodeled network branches, propagation through our experimental networks is very similar to that achieved in ground-truth ones.

3) Computational Complexity: To provide a comprehensive performance evaluation of the proposed graphing scheme, analysis of its complexity in terms of computational time is necessary. We have run the experiments on a 3.0 GHz Rayzen AMD processor (8 cores, 16 threads in each) with 64 GB of RAM. Segmentation of microvessels has been carried out on a 12GB NVIDIA TITAN X GPU. The times required to perform the computations in each stage of our modeling pipeline are reported in Table IV. It is noted that most of the computation times (≈ 91.8%) is used for segmenting microvessels and contracting the generated mesh-based models. From the table, the averaged calculation time for processing one 2PM angiogram, begining with the segmentation of microvessels and ending with the extraction of their graphed networks, reaches 53 minutes. Hence, our fully automatic scheme is proved to provide a very reliable solution when applied to study cerebral microvasculature in large cohorts and can save weeks of manual labor.

VI. Conclusion and Future Work

In this work, we have proposed a novel fully-automatic processing pipeline to produce graphical models for cerebral microvasculature captured with 2-photon microscopy. Our scheme is composed of three main processing blocks. First, a 3D binary mapping of microvasculature is obtained using a fully-convolution deep learning model. Then, a surface mesh is computed using a variant of the marching cube algorithm. Lastly, a reduced version of the generated surface model is contracted toward the 1D medial axis of the enclosed vasculature. The contracted mesh is post-processed to generate a final graphical model with a single connected component. We used a set of manually processed graphing of 6 angiograms to validate our graphs. From the quantitative validation based on NetMet measures, our model was able to produce accurate graphs with low geometrical and topological error rates, especially at a tolerance > $30\mu m$. Further qualitative assessment has shown that automatic processing generates realistic models of the underlying microvascular networks having accurate propagation through the modeled vessels. One important issue that could be addressed in a future work is related to the difficulty in generating watertight surface models. The employed contraction algorithm is not applicable to surfaces lacking such characteristics. Introducing a geometric contraction not restricted to such conditions on the obtained surface model could be an area of further investigation.

Supplementary Material

Refer to Web version on PubMed Central for supplementary material.

Acknowledgments

Canadian Institutes of Health Research (CIHR, 299166) operating grant and a Natural Sciences and Engineering Research Council of Canada (NSERC, 239876–2011) discovery grant to F. Lesage and NSERC Discovery Grant, RGPIN-2014–06089 to P. Pouliot.

References

[1]. Helmchen F and Denk W, “Deep tissue two-photon microscopy,” Nature methods, vol. 2, no. 12, pp. 932–940, 2005. [PubMed: 16299478]

[2]. Parpaleix A, Houssen YG, and Charpak S, “Imaging local neuronal activity by monitoring po2 transients in capillaries,” Nature medicine, vol. 19, no. 2, pp. 241–246, 2013.

[3]. Gagnon L, Smith AF, Boas DA, et al., “Modeling of cerebral oxygen transport based on in vivo microscopic imaging of microvascular network structure, blood flow, and oxygenation,” Frontiers in computational neuroscience, vol. 10, 2016.

[4]. Weller RO, Boche D, and Nicoll JAR, “Microvasculature changes and cerebral amyloid angiopathy in alzheimer’s disease and their potential impact on therapy,” Acta Neuropathologica, vol. 118, no. 1, p. 87, 2009.

[5]. Gagnon L, Sakadzic S, Lesage F, et al., “Quantifying the microvascular origin of bold-fmri from first principles with two-photon microscopy and an oxygen-sensitive nanoprobe,” Journal of Neuroscience, vol. 35, no. 8, pp. 3663–3675, 2015. [PubMed: 25716864]

[6]. Baez-Yanez MG, Ehses P, Mirkes C, et al., “The impact of vessel size, orientation and intravascular contribution on the neurovascular fingerprint of bold bssfp fmri,” Neuroimage, vol. 163, pp. 13–23, 2017. [PubMed: 28890417]

[7]. Teikari P, Santos M, Poon C, et al., “Deep learning convolutional networks for multiphoton microscopy vasculature segmentation,” arXiv preprint arXiv:1606.02382, 2016.

[8]. Bates R, Irving B, Markelc B, et al., “Segmentation of vasculature from fluorescently labeled endothelial cells in multi-photon microscopy images,” IEEE transactions on medical imaging, 2017.

[9]. Turetken E, Benmansour F, Andres B, et al., “Reconstructing curvilinear networks using path classifiers and integer programming,” IEEE transactions on pattern analysis and machine intelligence, vol. 38, no. 12, pp. 2515–2530, 2016. [PubMed: 26891482]

[10]. Bates R, Irving B, Markelc B, et al., “Extracting 3d vascular structures from microscopy images using convolutional recurrent networks,” arXiv preprint arXiv:1705.09597, 2017.

[11]. Frangi AF, Niessen WJ, Vincken KL, et al., “Multiscale vessel enhancement filtering,” in International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 1998, pp. 130–137.

[12]. Jerman T, Pernus F, Likar B, et al., "Enhancement of vascular structures in 3d and 2d angiographic images," IEEE transactions on medical imaging, vol. 35, no. 9, pp. 2107–2118, 2016. [PubMed: 27076353]

[13]. Hoyos MH, Orlowski P, Piatkowska-Janko E, et al., "Vascular centerline extraction in 3d mr angiograms for phase contrast mri blood flow measurement," International Journal of Computer Assisted Radiology and Surgery, vol. 1, no. 1, pp. 51–61, 2006.

[14]. Vasilevskiy A and Siddiqi K, "Flux maximizing geometric flows," IEEE transactions on pattern analysis and machine intelligence, vol. 24, no. 12, pp. 1565–1578, 2002.

[15]. Law MW and Chung AC, "Three dimensional curvilinear structure detection using optimally oriented flux," in European conference on computer vision, Springer, 2008, pp. 368–382.

[16]. Poon K, Hamarneh G, and Abugharbieh R, "Live-vessel: Extending livewire for simultaneous extraction of optimal medial and boundary paths in vascular images," Medical Image Computing and Computer-Assisted Intervention-MICCAI 2007, pp. 444–451, 2007.

[17]. Maninis K-K, Pont-Tuset J, Arbelaez P, et al., "Deep retinal image understanding," in International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2016, pp. 140–148.

[18]. Bates R, Irving B, Markelc B, et al., "Extracting 3d vascular structures from microscopy images using convolutional recurrent networks," arXiv preprint arXiv:1705.09597, 2017.

[19]. Merkow J, Marsden A, Kriegman D, et al., "Dense volume-to-volume vascular boundary detection," in International Conference on Medical Image Computing and Computer-Assisted Intervention, Springer, 2016, pp. 371–379.

[20]. Long J, Shelhamer E, and Darrell T, "Fully convolutional networks for semantic segmentation," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.

[21]. Wegner JD, Montoya-Zegarra JA, and Schindler K, "Road networks as collections of minimum cost paths," ISPRS Journal of Photogrammetry and Remote Sensing, vol. 108, pp. 128–137, 2015.

[22]. Mattyus G, Luo W, and Urtasun R, "Deeproadmapper: Extracting road topology from aerial images," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 3438–3446.

[23]. Turetken E, Benmansour F, Andres B, et al., "Reconstructing loopy curvilinear structures using integer programming," in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 1822–1829.

[24]. Peng H, Long F, and Myers G, "Automatic 3d neuron tracing using all-path pruning," Bioinformatics, vol. 27, no. 13, pp. i239–i247, 2011. [PubMed: 21685076]

[25]. Wang Y, Narayanaswamy A, and Roysam B, "Novel 4-d open-curve active contour and curve completion approach for automated tree structure extraction," in Computer Vision and Pattern Recognition (CVPR), 2011 IEEE Conference on, IEEE, 2011, pp. 1105–1112.

[26]. Turetken E, Benmansour F, and Fua P, "Automated reconstruction of tree structures using path classifiers and mixed integer programming," in Computer Vision and Pattern Recognition (CvPR), 2012 IEEE Conference on, IEEE, 2012, pp. 566–573.

[27]. Bogunovic H, Pozo JM, Cardenes R, et al., "Anatomical labeling of the circle of willis using maximum a posteriori probability estimation," IEEE transactions on medical imaging, vol. 32, no. 9, pp. 1587–1599, 2013. [PubMed: 23674438]

[28]. Almasi S, Xu X, Ben-Zvi A, et al., "A novel method for identifying a graph-based representation of 3-d microvascular networks from fluorescence microscopy image stacks," Medical image analysis, vol. 20, no. 1, pp. 208–223, 2015. [PubMed: 25515433]

[29]. Jégou S, Drozdzal M, Vazquez D, et al., "The one hundred layers tiramisu: Fully convolutional densenets for semantic segmentation," arXiv preprint arXiv:1611.09326, 2016.

[30]. Au OK-C, Tai C-L, Chu H-K, et al., "Skeleton extraction by mesh contraction," ACM Transactions on Graphics (TOG), vol. 27, no. 3, p. 44, 2008.

[31]. Huang G, Liu Z, Weinberger KQ, et al., "Densely connected convolutional networks," arXiv preprint arXiv:1608.06993, 2016.

[32]. Srivastava N, Hinton GE, Krizhevsky A, et al., “Dropout: A simple way to prevent neural networks from overfitting.,” Journal of machine learning research, vol. 15, no. 1, pp. 1929–1958, 2014.

[33]. Goodfellow I, Bengio Y, Courville A, et al., Deep learning. MIT press Cambridge, 2016, vol. 1.

[34]. Lorensen WE and Cline HE, “Marching cubes: A high resolution 3d surface construction algorithm,” in ACM siggraph computer graphics, ACM, vol. 21, 1987, pp. 163–169.

[35]. Nielson GM and Hamann B, “The asymptotic decider: Resolving the ambiguity in marching cubes,” in Proceedings of the 2nd conference on Visualization’91, IEEE Computer Society Press, 1991, pp. 83–91.

[36]. Chernyaev E, “Marching cubes 33: Construction of topologically correct isosurfaces,” Tech. Rep, 1995.

[37]. Ju T, Losasso F, Schaefer S, et al., “Dual contouring of hermite data,” in ACM transactions on graphics (TOG), ACM, vol. 21, 2002, pp. 339–346.

[38]. Tristano JR, Owen SJ, and Canann SA, “Advancing front surface mesh generation in parametric space using a riemannian surface definition.,” in IMR, 1998, pp. 429–445.

[39]. Lewiner T, Lopes H, Vieira AW, et al., “Efficient implementation of marching cubes’ cases with topological guarantees,” Journal of graphics tools, vol. 8, no. 2, pp. 1–15, 2003.

[40]. Low K-L and Tan T-S, “Model simplification using vertex-clustering,” in Proceedings of the 1997 symposium on Interactive 3D graphics, ACM, 1997, 75–ff.

[41]. Nooruddin FS and Turk G, “Simplification and repair of polygonal models using volumetric techniques,” IEEE Transactions on Visualization and Computer Graphics, vol. 9, no. 2, pp. 191–205, 2003.

[42]. Cohen J, Varshney A, Manocha D, et al., “Simplification envelopes,” in Proceedings of the 23rd annual conference on Computer graphics and interactive techniques, ACM, 1996, pp. 119–128.

[43]. Luebke D and Erikson C, “View-dependent simplification of arbitrary polygonal environments,” in Proceedings of the 24th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., 1997, pp. 199–208.

[44]. Garland M and Heckbert PS, “Surface simplification using quadric error metrics,” in Proceedings of the 24th annual conference on Computer graphics and interactive techniques, ACM Press/Addison-Wesley Publishing Co., 1997, pp. 209–216.

[45]. Luebke DP, “A developer’s survey of polygonal simplification algorithms,” IEEE Computer Graphics and Applications, vol. 21, no. 3, pp. 24–35, 2001.

[46]. Ovreiu E, “Accurate 3d mesh simplification,” PhD thesis, INSA de Lyon, 2012.

[47]. Guéziec A, Taubin G, Lazarus F, et al., “Cutting and stitching: Converting sets of polygons to manifold surfaces,” IEEE Transactions on Visualization and Computer Graphics, vol. 7, no. 2, pp. 136–151, 2001.

[48]. Liepa P, “Filling holes in meshes,” in Proceedings of the 2003 Eurographics/ACM SIGGRAPH symposium on Geometry processing, Eurographics Association, 2003, pp. 200–205.

[49]. Tagliasacchi A, Delame T, Spagnuolo M, et al., “3d skeletons: A state-of-the-art report,” in Computer Graphics Forum, Wiley Online Library, vol. 35, 2016, pp. 573–597.

[50]. Siddiqi K and Pizer S, Medial representations: mathematics, algorithms and applications. Springer Science & Business Media, 2008, vol. 37.

[51]. Dey TK and Sun J, “Defining and computing curve-skeletons with medial geodesic function,” in Symposium on geometry processing, vol. 6, 2006, pp. 143–152.

[52]. Jalba AC, Kustra J, and Telea AC, “Surface and curve skeletonization of large 3d models on the gpu,” IEEE transactions on pattern analysis and machine intelligence, vol. 35, no. 6, pp. 1495–1508, 2013. [PubMed: 23599061]

[53]. Manzanera A, Bernard TM, Preteux F, et al., “Medial faces from a concise 3d thinning algorithm,” in Computer Vision, 1999. The Proceedings of the Seventh IEEE International Conference on, IEEE, vol. 1, 1999, pp. 337–343.

[54]. Tagliasacchi A, Alhashim I, Olson M, et al., “Mean curvature skeletons,” in Computer Graphics Forum, Wiley Online Library, vol. 31, 2012, pp. 1735–1744.

[55]. Desbrun M, Meyer M, Schroder P, et al., “Implicit fairing of irregular meshes using diffusion and curvature flow,” in Proceedings of the 26th annual conference on Computer graphics and interactive techniques.

[56]. Sakadzic S, Mandeville ET, Gagnon L, et al., “Large arteriolar component of oxygen delivery implies a safe margin of oxygen supply to cerebral tissue,” Nature communications, vol. 5, p. 5734, 2014.

[57]. Homann H, “Implementation of a 3d thinning algorithm,” 2007.

[58]. Mayerich D, Bjornsson C, Taylor J, et al., “Netmets: Software for quantifying and visualizing errors in biological network segmentation,” BMC bioinformatics, vol. 13, no. 8, S7, 2012.

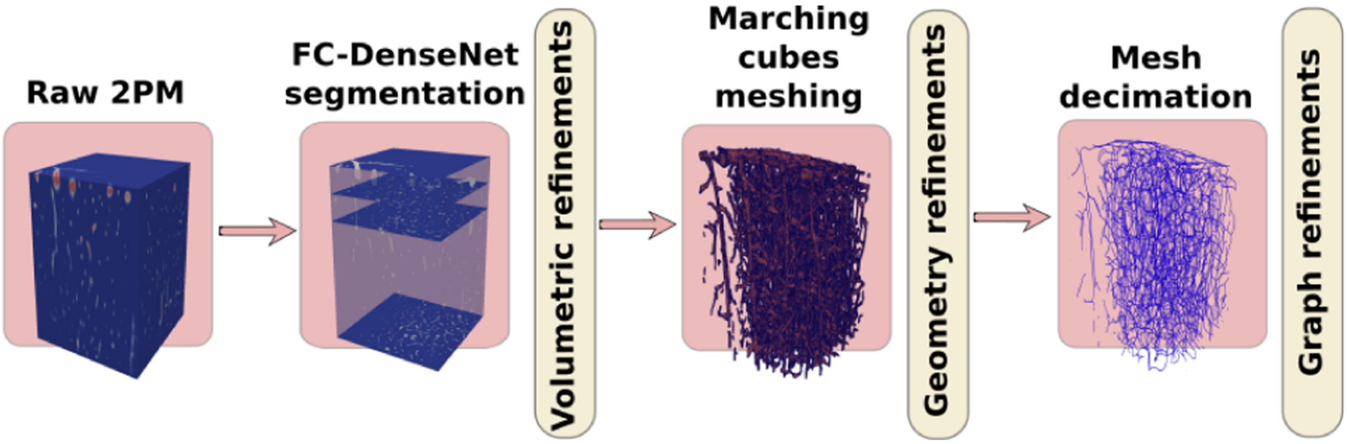
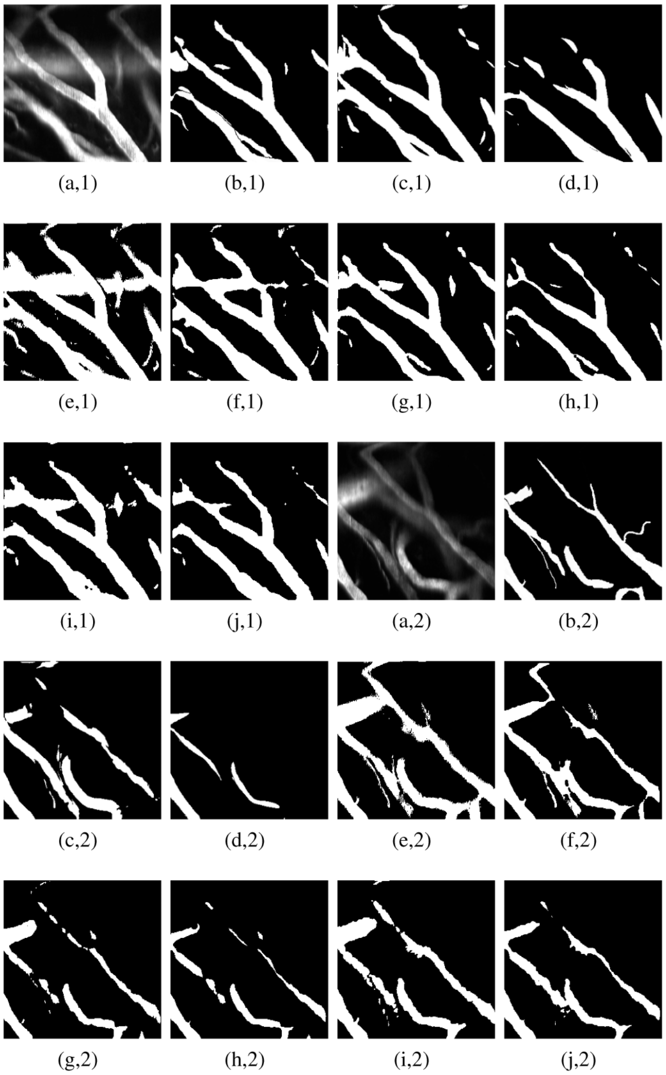


Fig. 1:

A schematic diagram describing the proposed multistage graph extraction scheme.



(a,1) (b,1) (c,1) (d,1)

(e,1) (f,1) (g,1) (h,1)

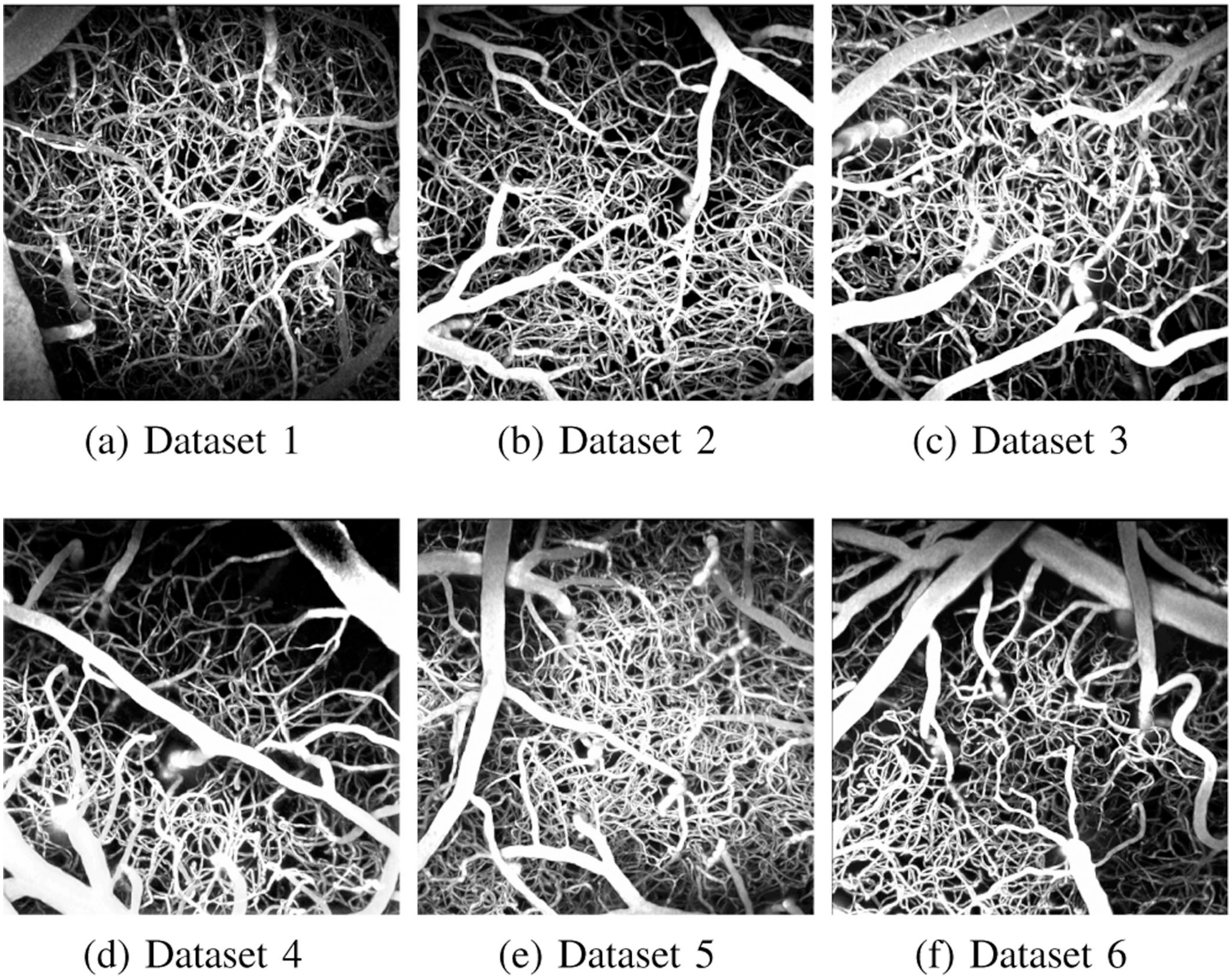
(i,1) (j,1) (a,2) (b,2)

(c,2) (d,2) (e,2) (f,2)

(g,2) (h,2) (i,2) (j,2)

Fig. 2:

Examples of microvessel masks obtained after applying the various segmentation schemes: (a,1–2) Raw 2PM, (b,1–2) true label, (c,1–2) OOF method [15], (d,1–2) Hessian-based method [12], (e,1–2) Net71, (f,1–2) net71+3DM , (g,1–2) Net97, (h,1–2) Net97+3DM, (i,1–2) Net127 and (j,1–2) Net127+3DM.

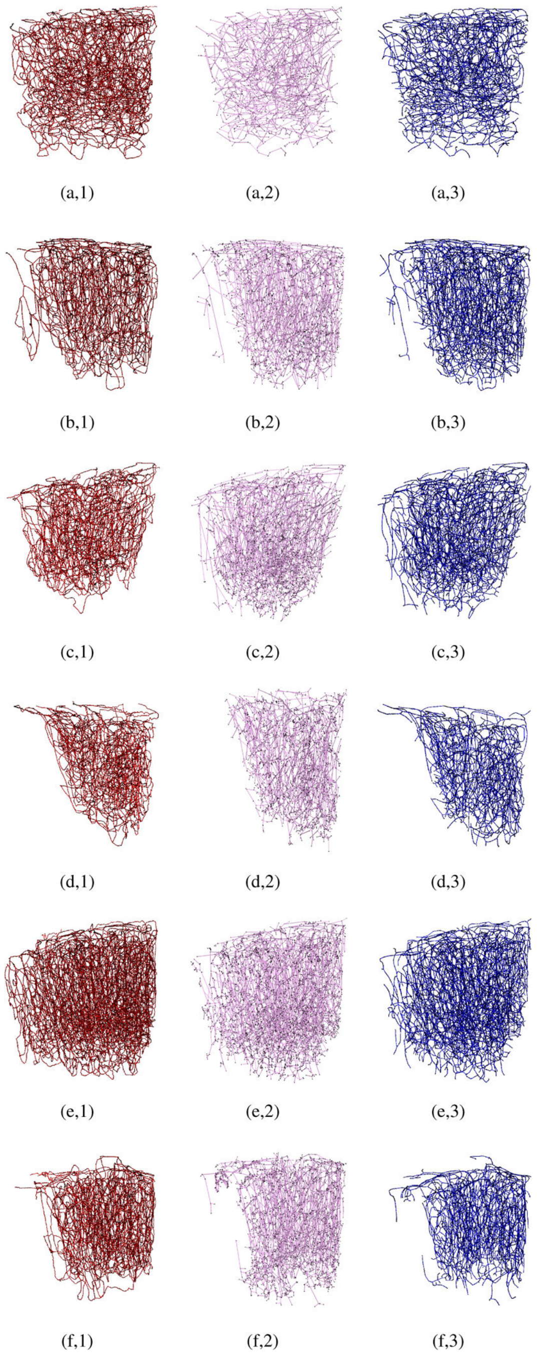


(a) Dataset 1 (b) Dataset 2 (c) Dataset 3

(d) Dataset 4 (e) Dataset 5 (f) Dataset 6

Fig. 3:

Maximum intensity projections of raw angiograms used to validate the proposed graph extraction scheme.



(a,1) (a,2) (a,3)

(b,1) (b,2) (b,3)

(c,1) (c,2) (c,3)

(d,1) (d,2) (d,3)

(e,1) (e,2) (e,3)

(f,1) (f,2) (f,3)

Fig. 4:

Visual assessment of our graph modeling scheme applied on the six raw 2PM datasets. For all datasets (1–6), manually processed graphs (ground-truth) are depicted in the left column (a-f,1); the graphs generated based on 3D Thinning are depicted in the middle column (a-f, 2); the graphs generated using our scheme are depicted in the right column.

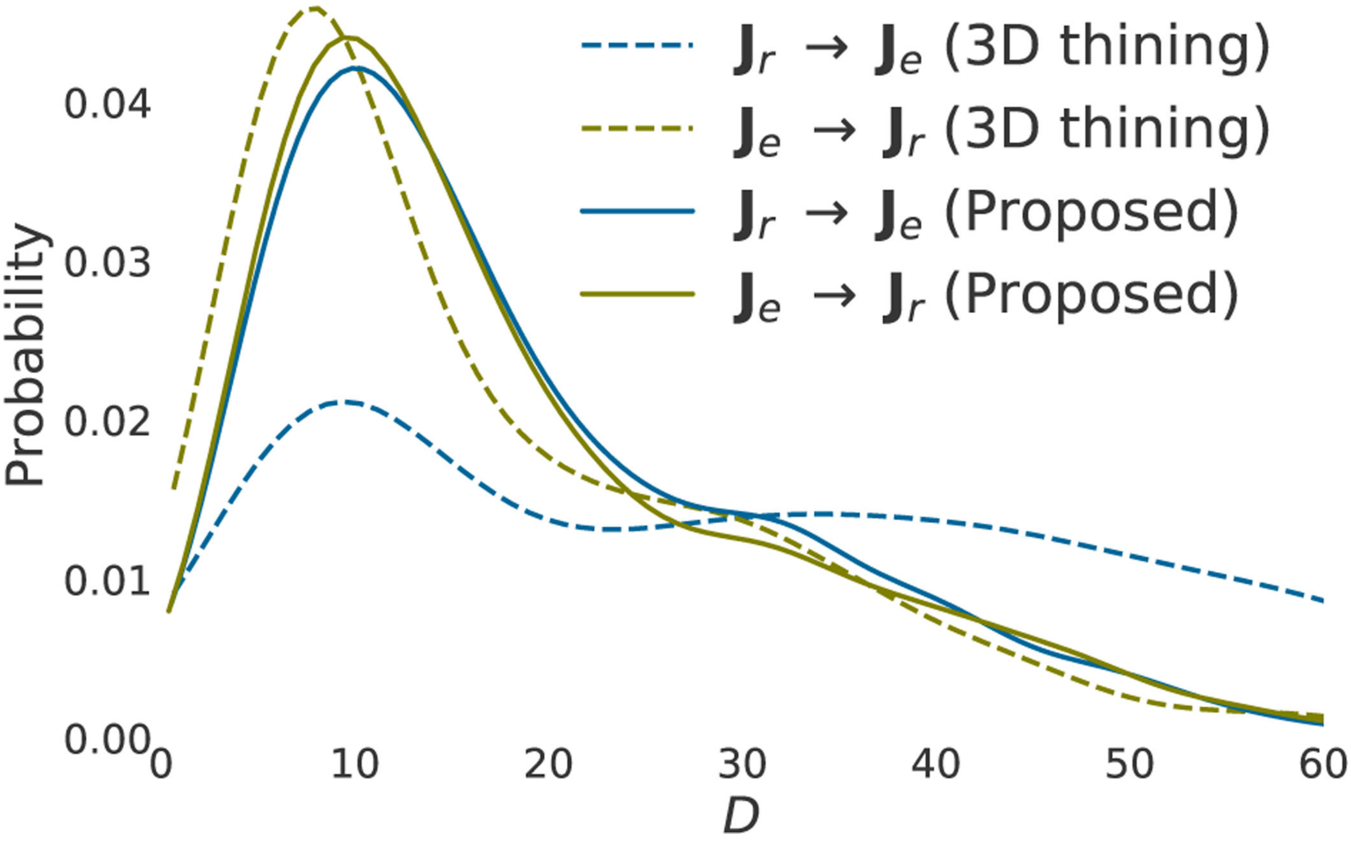


Fig. 5:

Experimental probability distributions of mapping distances D for the two-way matchings between the ground-truth and experimental graphs.

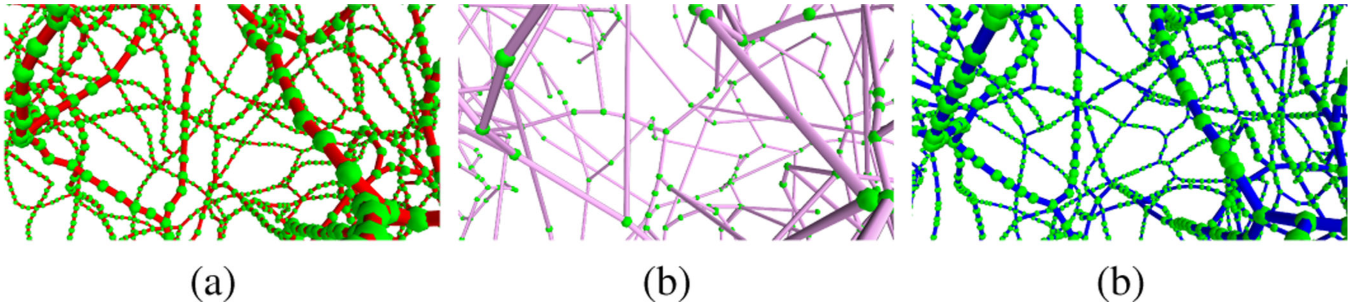
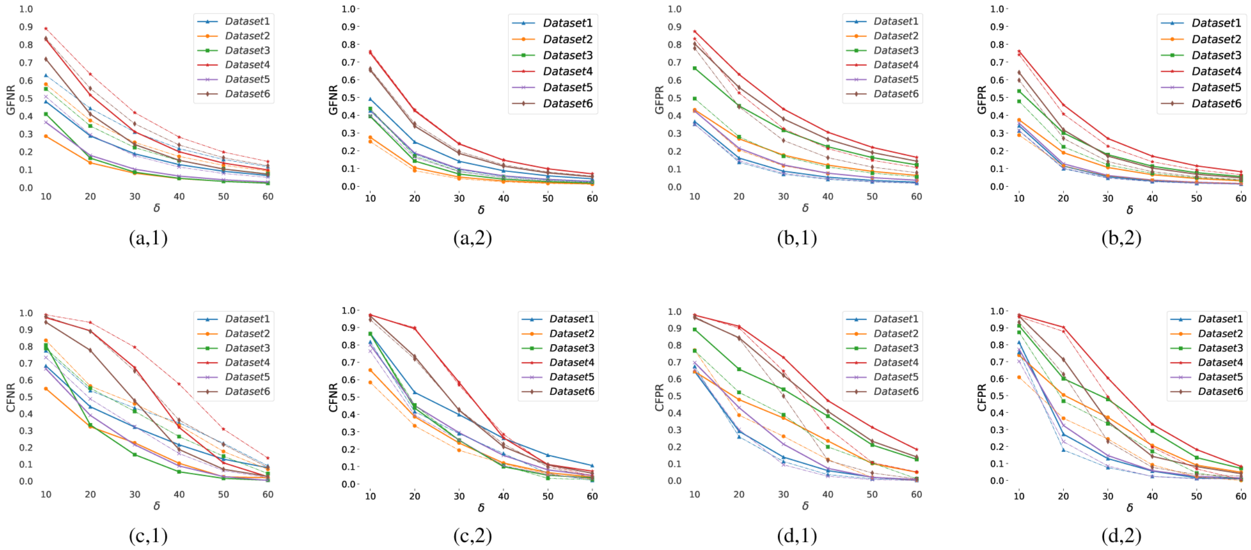


Fig. 6:

Magnified perspective view of Fig. 4 (a,1): (a), Fig. 4 (a,2): (b) and Fig. 4 (a,3): (c) with enlarged and recolored graph nodes.

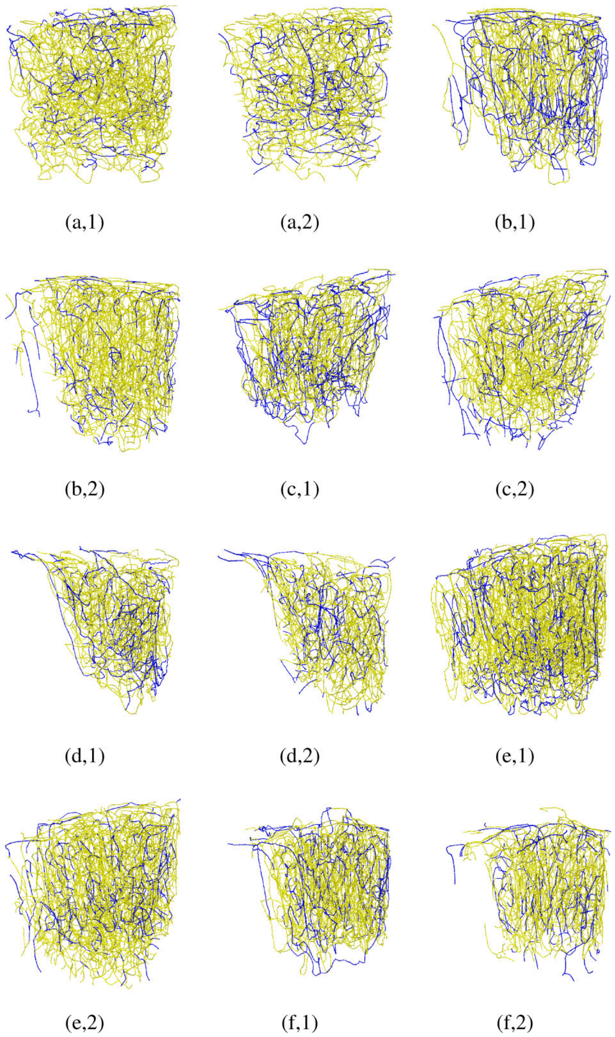


(a,1) (a,2) (b,1) (b,2)

(c,1) (c,2) (d,1) (d,2)

Fig. 7:

Quantitative assessment of the 3D Thinning method (a-d, 1) and the proposed graphing scheme (a-d,2), as a function of the δ parameter based on (a,1–2) GFNR, (b,1–2) GFPR, (c, 1–2) CFNR and (d,1–2) CFPR metrics. Dashed lines quantify the performance excluding the graphs boundary parts from the assessment process.



(a,1) (a,2) (b,1)

(b,2) (c,1) (c,2)

(d,1) (d,2) (e,1)

(e,2) (f,1) (f,2)

Fig. 8:

Visual illustration of mismodeled edges (blue-colored) for all datasets: false negative edges in (a-f,1); false positive edges in (a-f,2).

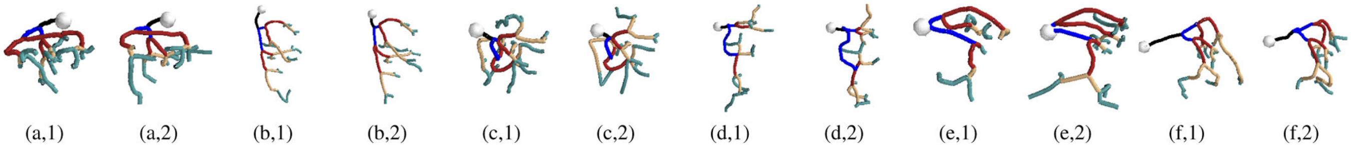


Fig. 9:

Propagating through the various microvessel networks for five consecutive branching levels beginning at the start of a penetrating arteriole (represented by white spheres), ground-truth (a-f,1) and automatically modeled (a-f,2). Each branching level is assigned a different colour.

TABLE I:

The various FC-DenseNet architectures tested to perform the segmentation task. At each component, we report the number of convolution layers followed by the number of feature maps.

	Architecture		
Component	**Net71**	**Net97**	**Netl27**
Input	-; 1	-; 1	-; 1
3×3 Convolution	1; 48	1; 48	1; 48
Dense block *type A*	4; 120	4; 120	4; 120
T-down layer	1; 120	1; 120	1; 120
Dense block *type A*	5; 210	5; 210	5; 210
T-down layer	1; 210	1; 210	1; 210
Dense block *type A*	7; 336	7; 336	7; 336
T-down layer	1; 336	1; 336	1; 336
Dense block *type A*	9; 498	9; 498	9; 498
T-down layer	1; 498	1; 498	1; 498
Dense block *type A*	-	11; 696	11; 696
T-down layer	-	1; 696	1; 696
Dense block *type A*	-	-	13; 930
T-down layer	-	-	1; 930
Bottleneck	11; 198	13; 234	15; 270
T-up layer + concatenation	-	-	1; 1200
Dense block *type B*	-	-	13; 234
T-up layer + concatenation	-	1; 930	1; 930
Dense block *type B*	-	11; 198	11; 198
T-up layer + concatenation	1; 696	1; 696	1; 696
Dense block *type B*	9; 162	9; 162	9; 162
T-up layer + concatenation	1; 498	1; 498	1; 498
Dense block *type B*	7; 126	7; 126	7; 126
T-up layer + concatenation	1; 336	1; 336	1; 336
Dense block *type B*	5; 90	5; 90	5; 90
T-up layer + concatenation	1; 210	1; 210	1; 210
Dense block *type B*	4; 72	4; 72	4; 72
1×1 Convolution	1; 2	l; 2	1; 2
Softmax	-; 2	-; 2	-; 2
Convolution layer	**Transition-down layer (T-down)**	**Transition-up layer (T-up)**	
Batch Normalization	Batch Normalization	3×3 Transposed convolution, stride = 2	
ReLu	ReLu		
3×3 Convolution Dropout $p = 0.2$	1×1 Convolution Dropout $p = 0.2$ 2×2 Max-pooling		

TABLE II:

Quantitative performance evaluation of the various segmentation schemes.

	Method in [15]	**Method in [12]**	**Net71**	**Net71+3DM**	**Net97**	**Net97+3DM**	**Net127**	**Net127+3DM**
Accuracy(%)	85.3 ± 0.6	92.1 ± 0.1	82.7 ± 0.3	88.3 ± 0.2	90.3 ± 0.1	**92.3** ± **0.1**	88.8 ± 0.3	91.1 ± 0.1
Sensitivity(%)	95.1 ± 0.1	94.64 ± 0.1	**98.9** ± **0.01**	97.1 ± 0.06	97.7 ± 0.1	96.74 ± 0.1	98.7 ± 0.1	97.17 ± 0.2
Specificily(%)	31.0 ± 2.3	**57.33** ± **2.20**	31.2 ± 0.9	36.2 ± 1.3	51.3 ± 1.9	56.5 ± 2.1	46.8 ± 1.6	50.8 ± 1.8
IOU(%)	28.9 ± 0.4	34.4 ± 0.4	30.9 ± 0.4	34.2 ± 0.3	40.2 ± 0.4	**41.1** ± **0.4**	38.9 ± 0.4	40.2 ± 0.4

TABLE III:

Number of nodes in the ground-truth graphs and in those generated by the 3D Thinning method and the proposed scheme. The generated graphs are obtained after extracting the LSCC from their original version.

	Ratio of LSCC to origional models		Number of all nodes			Number of junction nodes		
	$\mathscr{G}_e$ **(3D thining)**	$\mathscr{G}_e$ **(Proposed)**	$\mathscr{G}_r$	$\mathscr{G}_e$ **(3D thining)**	$\mathscr{G}_e$ **(Proposed)**	$\mathscr{G}_r$	$\mathscr{G}_e$ **(3D thining)**	$\mathscr{G}_e$ **(Proposed)**
Dataset 1	95.5%	99.5%	18624	1670	15902	1496	1371	1289
Dataset 2	86.6%	99.0%	16122	3292	17524	827	1890	906
Dataset 3	91.6%	99.0%	12301	4989	16843	782	2638	876
Dataset 4	87.4%	98.5%	9678	3239	10127	640	1668	718
Dataset 5	89.3%	99.0%	27745	6619	23307	1843	3615	1783
Dataset 6	87.2%	97.5%	13723	4527	11871	841	2377	894

TABLE IV:

Computational times (in seconds) required by each processing stage in our modeling pipeline.

	Dataset					
	1	**2**	**3**	**4**	**5**	**6**
Segmentation/ morphology	1053/37	1183/43	1059/42	1111/44	1262/43	1113/40
Surface Modeling	177	206	204	250	291	195
Graph Extraction	1981	2463	2210	1524	2930	1860