



VascuSynth: Simulating vascular trees for generating volumetric image data with ground-truth segmentation and tree analysis

Ghassan Hamarneh*, Preet Jassi

Medical Image Analysis Lab, School of Computing Science, Simon Fraser University, Canada

ARTICLE INFO

Article history:

Received 14 March 2010

Received in revised form 30 May 2010

Accepted 4 June 2010

Keywords:

Simulation

Validation

Medical image analysis

Segmentation

Tubular

Tree-like

Vasculature

Bifurcation

Ground-truth

3D images

ABSTRACT

Automated segmentation and analysis of tree-like structures from 3D medical images are important for many medical applications, such as those dealing with blood vasculature or lung airways. However, there is an absence of large databases of expert segmentations and analyses of such 3D medical images, which impedes the validation and training of proposed image analysis algorithms. In this work, we simulate volumetric images of vascular trees and generate the corresponding ground-truth segmentations, bifurcation locations, branch properties, and tree hierarchy. The tree generation is performed by iteratively growing a vascular structure based on a user-defined (possibly spatially varying) oxygen demand map. We describe the details of the algorithm and provide a variety of example results.

© 2010 Elsevier Ltd. All rights reserved.

1. Introduction

Medical images provide a wealth of data about internal anatomy essential for computer assisted surgery, computer aided diagnosis, treatment, and tracking of diseases. Images of vasculature and other tree-like structures (e.g. lung airways), in particular, provide important information about the delivery of nutrients to different organs and tissues [1]. Additionally, various pathologies may alter the vasculature. Therefore, the segmentation and analysis of images of vasculature are of utmost importance for understanding, diagnosing, and treating diseases. Examples of analyses performed on the vasculature includes discovering vascular tree topology and branching patterns, calculating radii, lengths, and tortuosity of branches using measures such as the distance metric (DM), the inflection count metric (ICM), and the sum of angles metric (SOAM) [2]. Visual inspection of the image data and manual segmentation and analysis are very time consuming, tedious, suffer from inter- and intra-operator variability and can be subjective and qualitative. This hindrance, in conjunction with the acquisition of large numbers of images of vasculatures, e.g. using phase contrast magnetic resonance angiography or computed tomography angiography, has

created a strong need for highly automated, fast, accurate and robust computerized segmentation and analysis algorithms. This strong need has been the catalyst for algorithms for segmentation and analysis of vasculature. This has been reflected by the large number of proposed algorithms for enhancement, segmentation, and analysis of vasculature and perfusion data from 3D medical images in the past few decades [3–12]. However, what has been lacking is the existence of benchmarks for the validation, evaluation, and comparison of all these approaches. Unfortunately, a database of ground-truth segmentation and analyses (branch point locations, tree-hierarchy, etc.) of vasculature does not currently exist, which makes validating and benchmarking such algorithms, as well as training machine learning techniques, very difficult. There are, however, databases or projects for validation of data other than tubular branching trees; most notably are BrainWeb [13], the Internet Brain Segmentation Repository (IBSR),¹ PET-SORTEO [14], the Non-Rigid Image Registration Evaluation Project (NIREP) [15], VALMET [16], and STAPLE [17]. This is where our work is most relevant. It is a unique validation project that generates data in a controlled manner allowing for extensive validation of existing vasculature segmentation and analysis algorithms. In particular, we

* Corresponding author. Tel.: +1 778 782 3007; fax: +1 778 782 3045.

E-mail addresses: hamarneh@cs.sfu.ca (G. Hamarneh), preetj@cs.sfu.ca (P. Jassi).

¹ Internet Brain Segmentation Repository: <http://www.cma.mgh.harvard.edu/ibsr/>.

simulate vascular trees embedded into volumetric images and generate the corresponding ground-truth segmentation (labeling), tree topology, and other measures that are essential for benchmarking and validation of the 3D vasculature segmentation and analysis algorithms.

There are two main methods for simulating vascular structures: (i) Lindenmayer systems: a set of production rules (grammar) used to iteratively generate complex shapes, often used to describe the growth of plants [18]; and (ii) iterative growth into a perfusion volume: growing a vascular structure by iteratively connecting new terminal nodes chosen from some volume [19–22]. We follow the latter approach because it allows greater control over the volume of the vasculature and is amenable to conforming to physical hemodynamics laws and blood vessel formation constraints, as we will see later.

In this work, we simulate realistic vascular trees based on user-defined (possibly spatially varying) demand for nutrients. The trees are then used to create volumetric images, which may then be customized to simulate a particular medical image acquisition modality. The result is the generation of volumetric images of vasculature, their corresponding ground-truth segmentations, bifurcation locations in 3D, and formal descriptions, using Graph eXchange Language (GXL), of the vascular tree hierarchy and branch attributes such as radii, flow, and lengths.

In the remainder of the paper, we describe the details of the underlying physics-based vascular model in Section 2. In Section 3, we describe the vasculature generation algorithm that we adopted, which is initialized with a user-defined oxygen (nutrient) demand map and other tunable parameters (Sections 3.2–3.3). We then describe the iterative tree-construction algorithm for selecting terminal nodes, creating bifurcating branches, calculating branch radii and flow, and updating the supply of nutrients (Sections 3.4–3.10). We describe the graph representation in Section 3.11 and the creation of volumetric images in Section 3.12. We follow with comprehensive results and examples in Section 4, and concluding remarks in Section 5.

2. Vasculature flow model

We begin, in this section, by describing the underlying physics-based vascular model that governs the way that the vascular structure is generated. Specifically, we discuss the flow and radii constraints of a branch in the vascular structure. Additionally, we present equations to calculate the reduced resistance as well as the pressure drop along a branch in the vascular structure. The flow and radii constraints, as well as the reduced resistance equations, are used to determine the radii of the branches in the generated vascular structure. The equations and constraints presented here will be used in later sections when developing the iterative vascular generation.

Two flow constraints in a vascular network are of particular importance for the simulation: (i) Conservation of flow, under which the flow entering a bifurcation point through the parent Q_{parent} must equal the total flow leaving a bifurcation point via the ‘left’ and ‘right’ child branches, Q_{left} and Q_{right} , i.e.

$$Q_{parent} = Q_{left} + Q_{right}. \quad (1)$$

(ii) The flow into each terminal node Q_{term} must be the same² and equal to the perfusion flow Q_{perf} at the root of the tree divided

by the total number of terminal nodes N , i.e.

$$Q_{term} = \frac{Q_{perf}}{N}. \quad (2)$$

The radius of the parent branch r_{parent} is related to the radii of its two children, r_{left} and r_{right} , as follows

$$r_{parent}^\gamma = r_{left}^\gamma + r_{right}^\gamma \quad (3)$$

with γ set between 2.55 and 3 [20,23]. The flow resistance R_i induced along a branch i with radius r_i and length L_i is

$$R_i = \frac{8\eta L_i}{\pi r_i^4} \quad (4)$$

where η is the viscosity of the fluid, which is assumed constant in this work for simplicity. The reduced resistance R^* for a terminal branch³ $term$ is obtained by substituting $r=1$ in (4),

$$R_{term}^* = \frac{8\eta L_t}{\pi} \quad (5)$$

R^* for a bifurcation sub-tree with (non-terminal) branch i as its root is given by [20]

$$R_i^* = \frac{8\eta L_i}{\pi} + \left[\frac{(r_{left}/r_i)^4}{R_{left}^*} + \frac{(r_{right}/r_i)^4}{R_{right}^*} \right]^{-1} \quad (6)$$

As we will see later, the reduced resistance is used for calculating the radii of branches. The pressure drop along a branch i is calculated as

$$\Delta P_i = Q_i R_i \quad (7)$$

Each terminal node is assumed to have the same terminal pressure, which means the pressure drop from the perfusion point to any terminal node must be the same. Since the flow at each terminal node is the same and likewise the terminal pressure, then the resistance from the perfusion point to any terminal node must also be the same.

3. Vascular tree generation

3.1. Overview

The tree generation is performed by iteratively growing a vascular structure based on a user-defined (possibly spatially varying) oxygen demand map (Section 3.2) while enforcing the physical constraints (Section 2). At a high level (Fig. 1), during each iteration, a candidate terminal node is first chosen according to the oxygen demand map. Then an existing branch is bifurcated to supply this terminal node. The choice of the branch to bifurcate and the location of the bifurcation point along the branch is chosen in an optimal way as we explain later. The oxygen supply map is then updated accordingly, and a new terminal node is selected, and so on. After the tree has completed its growth, the final radii of the tree branches are calculated according to the aforementioned physical flow and radii constraints.

3.2. Oxygen demand map

An oxygen demand map $ODM(x, y, z) : \Omega \subset \mathbb{R}^3 \rightarrow [0, 1]$ is a 3D scalar volume that covers the entire perfusion volume Ω , with values ranging from 0 for ‘no demand’ to 1 for maximum demand for oxygen. ODM is provided as an input to the algorithm. A simple

² This assumption avoids having varying terminal flows and subsequently avoids the knapsack combinatorial optimization problem of choosing candidates.

³ We assume a cut-off in the resolution of our model and hence ignore the capillary loops bringing blood back to the heart. With this assumption we assign a resistance even to the terminal branches of our model.

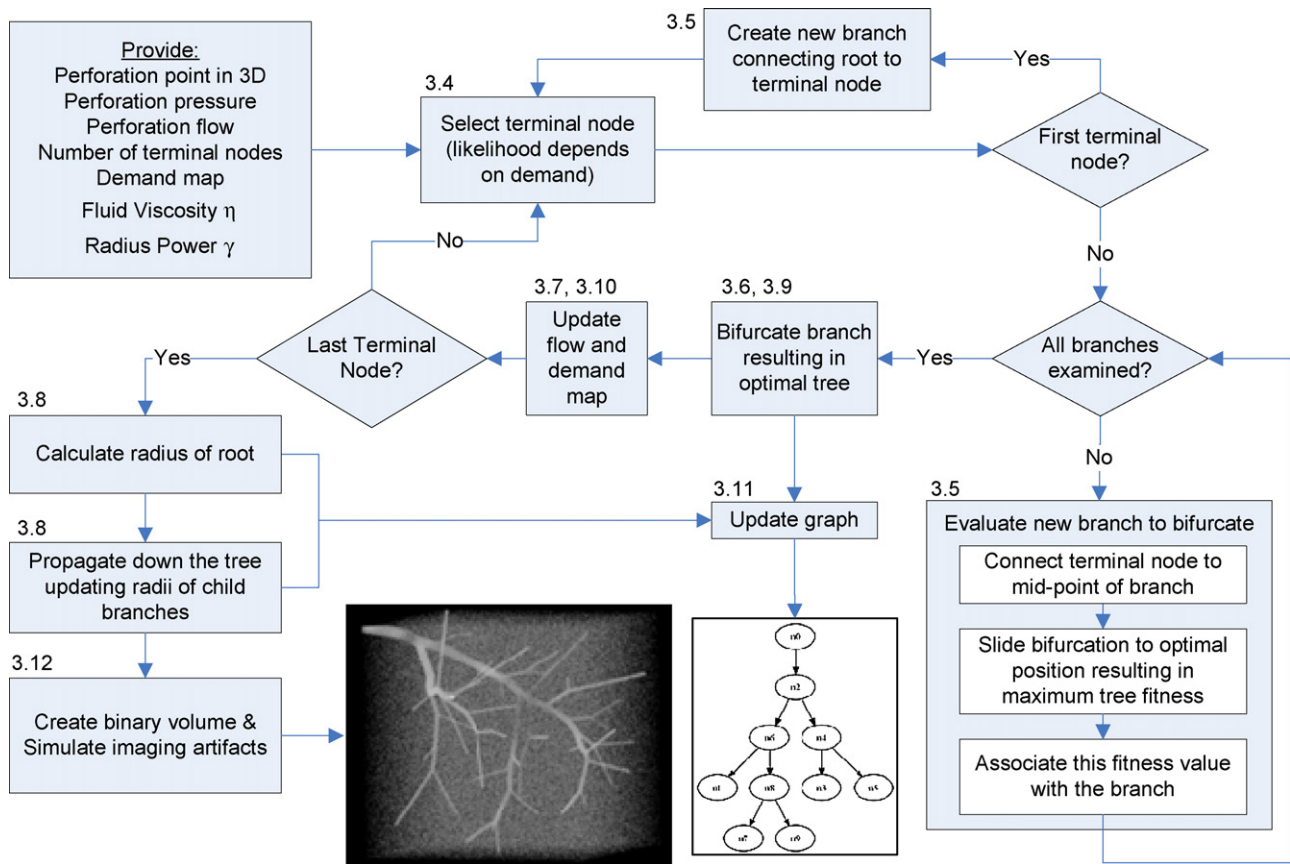


Fig. 1. Flow chart depicting the overall progress of the algorithm (corresponding section number is shown at the top left of a block).

case is to have a constant (homogeneous) oxygen demand map. The steps in Sections 3.4–3.10 are then repeated to iteratively construct tree branches that connect to terminal nodes, until a user-defined number N of accepted terminal nodes is reached.

3.3. Perforation and terminal pressure

A tree trunk node position, or perforation point, $p = (p_x, p_y, p_z)$ must be provided to the algorithm. Altering this value will change the initial location of the vascular structure. The perforation pressure is the pressure of the vascular structure at the perforation point, i.e. $P_{perf} = P(p)$. Since the flow is conserved (Section 2), any increase in pressure requires a decrease in volume, which in turn means that as the perforation pressure increases, the radii of the branches decrease, and vice versa. The terminal pressure P_{term} is the value of the pressure at any terminal node $t_i = (t_x, t_y, t_z)$, i.e. $P_{term} = P(t_i)$, for any $i = 1, 2, \dots, N$. Note that $P_{term} \leq P_{perf}$ must be ensured. P_{perf} and P_{term} are used for calculating the radius of the root branch, which is then used to calculate the radii of all other branches (Section 3.8).

3.4. Selecting terminal nodes

A terminal node position is randomly selected from the perfusion volume such that the likelihood (probability) of selecting this node is proportional to the value of the ODM at the location of that node. The Mersenne Twister algorithm is used for the selection of a candidate terminal node [24]. The selected node is then used as a candidate for being attached to the tree. The candidate node is rejected, if all paths from the tree to the node pass through regions

of zero demand, or if the node is within a certain user-specified distance, D_B , to any existing branch.

3.5. Selecting a branch to supply a terminal node

If no branches exist yet in the tree then a branch is created from the tree trunk node position p to the first terminal node t_1 . Otherwise, each existing branch in the tree is temporarily connected to the candidate terminal node and each such solution is evaluated. Preference is given to trees that minimize the objective function

$$\sum_{\forall j} L_j^\mu r_j^\lambda \quad (8)$$

where j is an index over all branches in the tree. Increasing λ will favor the configurations where the radii are smallest (and vice versa), while increasing μ will favor configurations with shorter branches. If $\mu = 1$ and $\lambda = 2$, then the objective function is proportional to the volume $\sum_{\forall j} L_j \pi r_j^2$, i.e. union of cylindrical branches.

According to [20], the branches closest to the candidate terminal node are most likely to be chosen as the minimizers of (8), with the 1st closest being chosen 60% of the time, dropping down to $\leq 1\%$ for the 10th closest. In our work, we rely on this conclusion and examine only the k closest branches (k is user-specified). The optimal branch which results in the minimal value of the objective function is chosen and that connection is made permanent. If every branch examined was rejected (i.e. all branches pass through a 0 demand region), then the terminal node is rejected and a new one is chosen (Section 3.4).

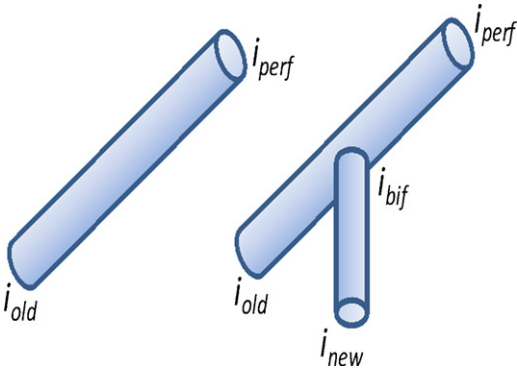


Fig. 2. Introducing a bifurcating branch.

3.6. Branch bifurcation points

With an accepted terminal node i_{new} , the chosen branch to bifurcate, originally connecting nodes $i_{perf} \Rightarrow i_{old}$ and denoted $i_{perf} \Rightarrow i_{old}$, is now split into two parts via a new point i_{bif} (Fig. 2). The location of i_{bif} is chosen arbitrarily midway at this stage and will be updated later (Section 3.9). This results in two branches $i_{perf} \Rightarrow i_{bif}$ and $i_{bif} \Rightarrow i_{old}$, and a newly created branch $i_{bif} \Rightarrow i_{new}$, i.e. a Y junction is formed.

3.7. Updating the branches' reduced resistance and flow

After a terminal node i_{new} has been connected to the vascular structure at i_{bif} , the reduced resistance and flow are updated. The new branch $i_{bif} \Rightarrow i_{new}$ is assigned a flow value of Q_{term} . To satisfy the constraint in (1), Q_{term} is added to all the ancestor branches of $i_{bif} \Rightarrow i_{new}$, stopping at the root, up to which the flow accumulates to $Q_{perf} = NQ_{term}$ (agreeing with (2)). The reduced resistance R^* of $i_{bif} \Rightarrow i_{new}$ is calculated using (5) as it is initially connected to a terminal node. Then, recursively, R_i^* of the parent branch is updated using (6) until the reduced resistance of the root R_{perf}^* has been calculated.

3.8. Updating the branch radii

The radius of a branch is calculated by first determining the fraction of the branch's radius to its parent's radius, and then multiplying the parent radius by the calculated fraction. In order to calculate the radii of all branches of the tree, the radius of the root-most (perforation) branch is first calculated, then the radii of its two children are calculated, then the radii of the children's children, and so on, until reaching the terminal branches. The perforation radius is calculated by dividing (4) by (5) to obtain

$$r_{perf} = \left(\frac{R_{perf}^*}{R_{perf}} \right)^{1/4} \quad (9)$$

Using (7) to calculate R_{perf} , we re-write (9) as

$$r_{perf} = \left(\frac{Q_{perf} R_{perf}^*}{P_{perf} - P_{term}} \right)^{1/4} \quad (10)$$

and calculate R_{perf}^* from (5). Once the radius of the root branch is calculated, the radii of the left and right child branches are calculated as follows. Dividing (3) by r_{left}^γ and re-arranging the terms, we calculate the fraction f_{left} (and similarly for f_{right}) as

$$f_{left} = \frac{r_{left}}{r_{perf}} = \left(1 + \left(\frac{r_{right}}{r_{left}} \right)^\gamma \right)^{-1/\gamma} \quad (11)$$

r_{left} is then obtained by multiplying f_{left} by r_{perf} in (10). Note that the ratio r_{right}/r_{left} must be known in order to calculate f_{left} (and similarly for f_{right}). This ratio is obtained by equating ΔP_{left} to ΔP_{right} (both calculated from (7)) and substituting R_{left} by R_{left}^*/r_{left}^4 to obtain

$$\frac{r_{left}}{r_{right}} = \left(\frac{Q_{left} R_{left}^*}{Q_{right} R_{right}^*} \right)^{1/4} \quad (12)$$

With this r_{right}/r_{left} ratio in hand, f_{left} and f_{right} are calculated and multiplied by r_{perf} to obtain r_{left} and r_{right} , respectively. The procedure is repeated to update the flow and radii for subsequent branch generations.

3.9. Optimizing the bifurcation location

Once the radii and flow values have been updated, the location of the bifurcation point, i_{bif} , is translated along the branch to minimize the aforementioned objective function in (8). Once i_{bif} is moved, the entire tree must be updated again. This process is repeated until convergence. Any location of i_{bif} is rejected if either the branch $i_{bif} \Rightarrow i_{new}$, $i_{perf} \Rightarrow i_{bif}$, or $i_{bif} \Rightarrow i_{old}$ pass through a region of zero oxygen demand. If all possible i_{bif} positions are rejected then the branch is rejected.

3.10. Updating the demand map to reflect existing supply

Since branches created in the volume supply nutrients to regions in the vicinity of terminal nodes perfused by these branches, the demand for nutrients in nearby regions must be lowered. The positions in the volume close to the position of a perfused terminal node are assumed to draw from this supply through the arteriolar networks, and hence the value of the oxygen demand map ODM at these nearby voxel locations is updated as

$$ODM(v) \leftarrow \begin{cases} ODM(v) & d(v, t) > D_T \\ ODM(v)w(d(v, t)) & d(v, t) \leq D_T \end{cases} \quad (13)$$

where $d(v, t)$ is the Euclidean distance between the voxel position v and the terminal node position t and $w(d(v, t)) \in [0, 1]$ is a monotonically increasing function (e.g. polynomial or exponential). The equation essentially states that the demand at locations far away from the perfused nodes ($d(v, t) > \text{threshold } D_T$) is not affected, whereas the demand at nearby locations ($d(v, t) \leq D_T$) is decreased via the weighting by w , and the farther these locations are to the perfused terminal node the smaller the drop in demand.

3.11. Graph-based modeling of the vasculature

During the iterations of the algorithm, an attributed graph data structure $\mathcal{G}(\mathcal{V}, \mathcal{E})$ is constructed to describe the hierarchy and connectivity of the branches (parent–child relationships). \mathcal{V} is the set of vertices v_i representing the bifurcation points of the vascular tree and \mathcal{E} is the set of edges e_i representing the branches. Vertex and edge attributes are also encoded in the graph to further capture the properties of the vasculature. Associated with each $v_i \in \mathcal{V}$ are the 3D coordinates of the spatial location of the branching point. Associated with each $e_i \in \mathcal{E}$ are the length, radius, and flow in the branch represented by e_i , and indices to the two bifurcation or terminal nodes it connects. The attributed graph is formalized using the Graph eXchange Language,⁴ an open standard exchange format for graphs [25], adhering to the eXtensible Markup Language (XML) and constrained by a Document Type Definition (DTD). Our implementation is also capable of exporting the tree in the DOT

⁴ GXL: <http://www.gupro.de/GXL/>.

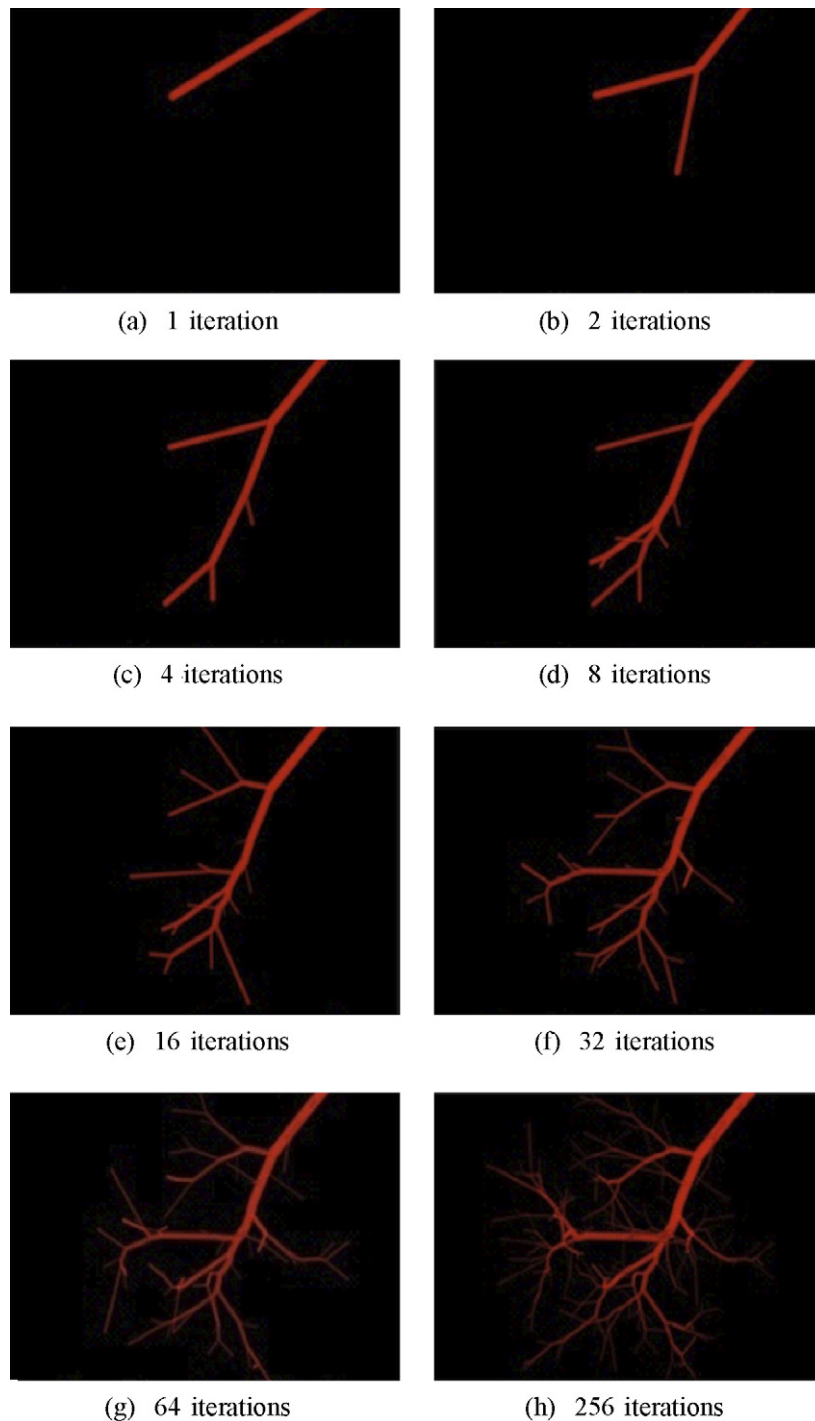


Fig. 3. Progress of vasculature generation for a constant oxygen demand map. The iteration number (also current number of terminal nodes) is shown below each sub-figure.

language,⁵ a plain text graph description language, which can be imported into different software. We use the Graph Visualization software, Graphviz,⁶ to display the resulting graphs.

3.12. Volumetric data generation

The previous sections described the construction of the vascular tree contained within the spatial limits of the non-zero

demand map. The resulting vascular tree produced is a set of connected branches with known starting and ending points and with known radii. A volumetric image is then produced using a rasterization procedure based on the branching vasculature as follows.

First, voxel dimensions in mm are specified, which reflect the desired resolution of the output 3D image. The field of view of the medical image data is set to be the bounding box of the non-zero demand map. Next, a 3D array *Vol* with enough rows, columns and slices (i.e. with enough voxels) to tile the entire field of view is created and is initialized with zeros. To model partial volume effect, an iterator traverses all the array voxels (i, j, k) examining whether

⁵ DOT: http://en.wikipedia.org/wiki/DOT_language.

⁶ GraphViz: <http://www.graphviz.org/>.

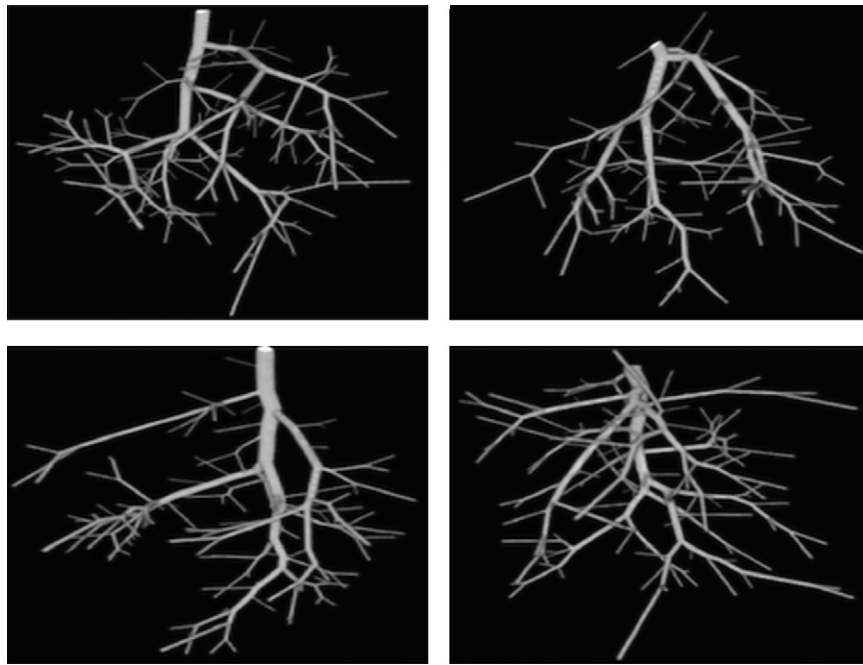


Fig. 4. Examples of generated vasculature volumes rendered with maximum intensity projections (MIP).

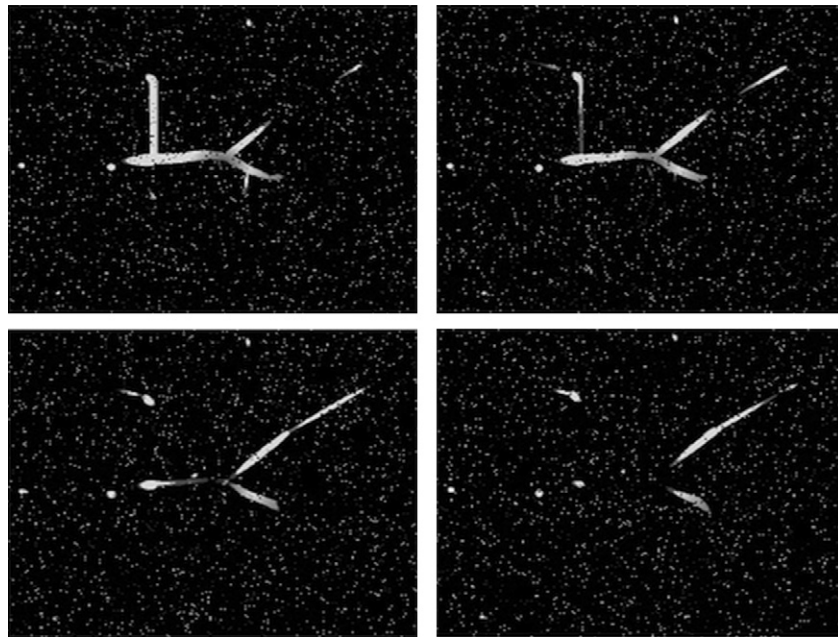


Fig. 5. Sample adjacent multi-planar reconstruction (MPR) (slices) from a single vasculature volume.

the voxel lies completely or partially within any of the cylinder-like branches. If the voxel intersects with a branch then the intensity of that voxel is calculated by dividing the voxel into eight equal subvoxels and adding the contributions $c_n(i, j, k)$ made by the subvoxels, i.e. $Vol(i, j, k) = \sum_{n=1}^8 c_n(i, j, k)$. If the subvoxel is contained within the branch, then the subvoxel contributes to the resulting intensity of the voxel by an eighth of the maximum intensity value, i.e. $c_n(i, j, k) = (1/8)256$. If the subvoxel is not in the branch, then the subvoxel does not contribute to the final intensity of the voxel, i.e. $c_n(i, j, k) = 0$. To create a more realistic 3D medical image, we simulate specific artifacts (e.g. noise and non-homogeneity) depending on the modality of interest and the intended use of the simulated volumetric data.

Once the noise-free volume is generated, it may be used as an input to any of the variety of modality-specific simulators. There exists numerous works on the simulation of different medical image modalities (not the simulation of the underlying geometry, which we do). We review a few but consider the choice of which simulator to be used in the subsequent stages beyond the scope of this paper. For example, the Physics-Oriented Simulated Scanner for Understanding MRI (POSSUM)⁷ [26] can be used for simulating structural or functional MR images using different pulse sequence and inhomogeneities. For simulating positron emission tomogra-

⁷ POSSUM: <http://www.fmrib.ox.ac.uk/fsl/possum/>.

Table 1

Graph eXchange Language of Simulated Vasculature. XML-based GXL shows example node (bifurcation) and edge (branch) attributes such as flow and radius.

```
<?xml version="1.0" encoding="UTF-8" ?>
<gxl xmlns:xlink="http://www.w3.org/1999/xlink">
<graph id="tree_structure.xml" edgeids="true" edgemode="directed" hypergraph="false">
<node id="n0">
<attr name="nodeType"><string>root node</string></attr>
<attr name="position"><tup>
<float>0</float>
<float>50</float>
<float>50</float></tup>
</attr>
</node>
<node id="n8">
<attr name="nodeType"><string>bifurcation</string></attr>
<attr name="position"><tup>
<float>0.527111</float>
<float>49.6762</float>
<float>50.2838</float></tup>
</attr>
</node>
<edge id="e8" to="n8" from="n0">
<attr name="flow"><float>8.33</float></attr>
<attr name="radius"><float>0.0940703</float></attr>
<attr name="length"><float>0.6806</float></attr>
</edge>
```

phy (PET) or single photon computed tomography (SPECT), the Simulation System for Emission Tomography (SimSET)⁸ [27] can be used. For simulating ultrasound image the reader is referred, for example, to [28].

Finally, to provide more challenging datasets for the evaluation and machine learning of image processing and analysis algorithms, we follow an approach similar to [29]. In [29], curved branches are obtained via spatially warping the resulting volume by using B-spline interpolation of 3D control points' displacement throughout the volume (see [30] for details). The volumes may be further deteriorated by Gaussian blurring, by intensity dropouts (to introduce gaps in the vessel tree) and by Gaussian, uniform, or impulse noise with user-specified parameters. The resulting volumes and the corresponding ground-truth analyses are now ready to be used for benchmarking existing or newly proposed 3D image segmentation and analysis algorithms, such as those surveyed in [31].

4. Results

Fig. 3 depicts an example of the progress of the iterative vasculature generation algorithm given a uniform oxygen demand map. Fig. 4 shows several automatically generated simulated vasculature volumes using various parameters. It is important to note that these parameters are not low-level, ad-hoc parameters that have been tweaked to produce a desired result, but are meaningful and understandable physical quantities or relate to physical phenomenon such as flow, number of branches, etc. Therefore, the examples presented in Fig. 4 and subsequent figures, where we vary the simulation parameters, are intended to validate that changes in the parameters produce expected and intuitive results. These results in turn demonstrate the flexibility of generating different types of realistic datasets that present varying degrees of challenges and complexities for automated segmentation and analysis algorithms, a desirable trait for our benchmarking and validation data. Maximum intensity projections (MIP) are used in Fig. 4 to show the

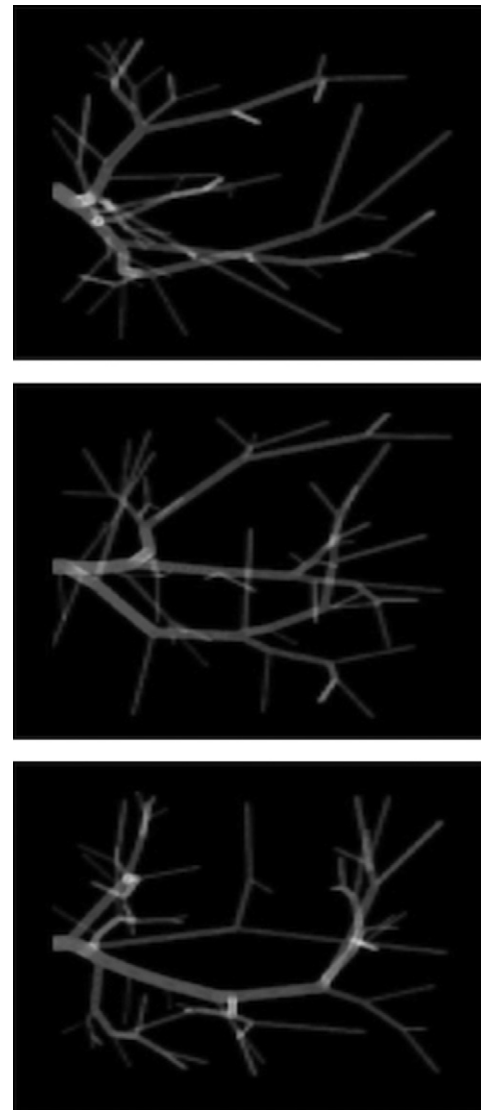


Fig. 6. Examples of generated volumes using exactly the same parameters. The change in vasculature is due to the randomness in selecting terminal nodes. The parameters used are: Perforation point (0,50,50); $P_{perf} = 133$ mmHg; $P_{term} = 83$ mmHg; $Q_{flow} = 138.83$ cm³/min/g (8.33 m³/h/kg); $\eta = 36$ mPa s; $\gamma = 3$; $\lambda = 2$; $\mu = 1$; $D_T = 1$ mm; voxel size = 0.043 mm³; $N = 50$; $k = 5$.

entire vasculature compared to multiplanar reconstruction (MPR), shown in Fig. 5 for comparison.

Table 1 contains a snippet of the automatically generated GXL summarizing bifurcation point positions, branch attributes (flow, length, radii), and tree topology (or connectivity) of vasculature. Fig. 6 shows several examples generated using the exact same parameters. Note that the different results are due to the randomness in the terminal node selection (Section 3.4). This randomness may be avoided, if desired, by setting the initial seed values of the random number generator to a fixed value. Fig. 7 depicts the generation of curved vasculature from linear (straight) branches via a deformable, B-spline spatial warp.

In the following set of results, we focus on demonstrating the effect on the resulting vasculature as we vary the different parameters required by the algorithm. Here, the random number generator seed value is fixed in order to focus on comparing the effect of changing the simulation parameters themselves and not changes due to the random node sampling. Fig. 8(a) shows the resulting vasculature from a spatially invariant oxygen demand compared to how the simulated vasculature is affected when a cube-shaped

⁸ SimSET: <http://depts.washington.edu/simset/html/>.

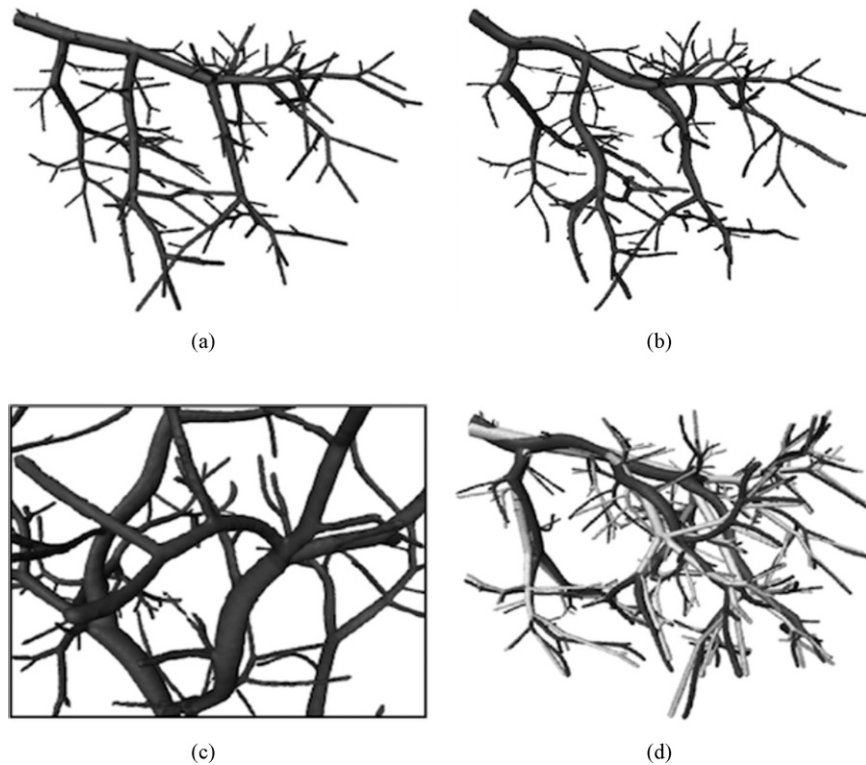


Fig. 7. Vasculature with (a) linear branches used to generate (b) curved vasculature via non-rigid, deformable B-spline-based spatial warps. (c) shows a close-up of (b). (d) Linear (light) and curved (dark) vasculature overlaid.

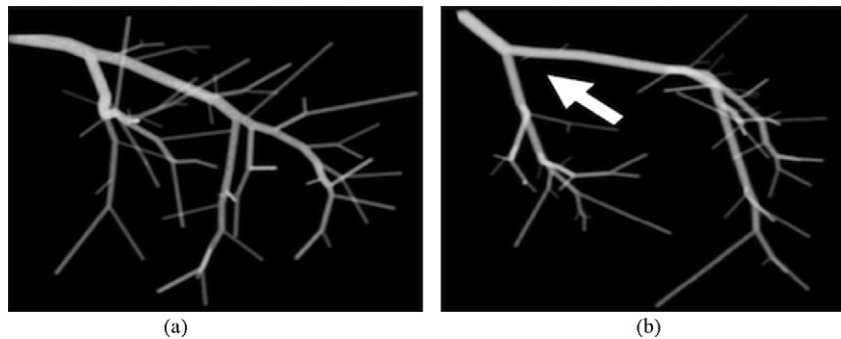


Fig. 8. Varying the demand map. (a) Vasculature supplying a spatially invariant oxygen demand. (b) Spatially variant demand, introduced via a zero-demand, cube-shaped obstruction in the center of the volume (white arrow), causes a noticeable gap between the two main branches.

obstruction with zero oxygen demand is centered in the volume (Fig. 8(b)). Fig. 9 shows generated vasculature for a variety of image artifacts.

Fig. 10 shows the effect of varying several simulation parameters. Note how higher perforation flow (or lower pressure) yields thicker branches (and vice versa) (Fig. 10(a) and Fig. 10(b)). Note also the result obtained when γ (the radius exponent in (3)) takes on a value outside the range recommended in [20] (Fig. 10(d)) compared to an allowable value (Fig. 10(c)). The effect of changing μ and λ (used in the objective function (8)) is shown in Fig. 10(e) and Fig. 10(f). Note how Increasing λ favors thinner branches and increasing μ favors shorter branches. The result of varying D_B is shown in Fig. 10(g) and Fig. 10(h). Note the change in the minimum distances between branches (Section 3.4). Fig. 10(i) and Fig. 10(j) show the result of a fourfold increase in the number of terminal nodes. Varying the number k (Section 3.5) of closest branches examined as candidates to bifurcate is shown in Fig. 10(k) and Fig. 10(l).

Finally, a variety of vasculature graph visualization examples, for increasing number of terminal nodes, is shown in Fig. 11.

5. Discussion and conclusion

We describe a method for generating synthetic vasculature and embedding them into volumetric images. The generated volumes are used to simulate more realistic medical images by including different typical imaging artifacts, such as blurring, noise, image intensity (or contrast) dropouts, etc. We create vasculature with curved branches by applying a non-rigid deformable B-spline warp. The automatically generated graph representation encodes the tree properties using the XML-based Graph eXchange Language, which provides a standardized, text-based file format that is easy to import into a variety of software for validating image analysis algorithms or for visualizing the resulting hierarchy, e.g. via GraphViz. The method allows the control of several parameters, including the location of the root of the tree, the number of desired terminal (or leaf) nodes, and the oxygen demand map, which favors the creation of more terminal nodes wherever there is more demand. The simulation follows a realistic, physical model of fluid perfusion, and hence additional, detailed parameters can also be set, includ-

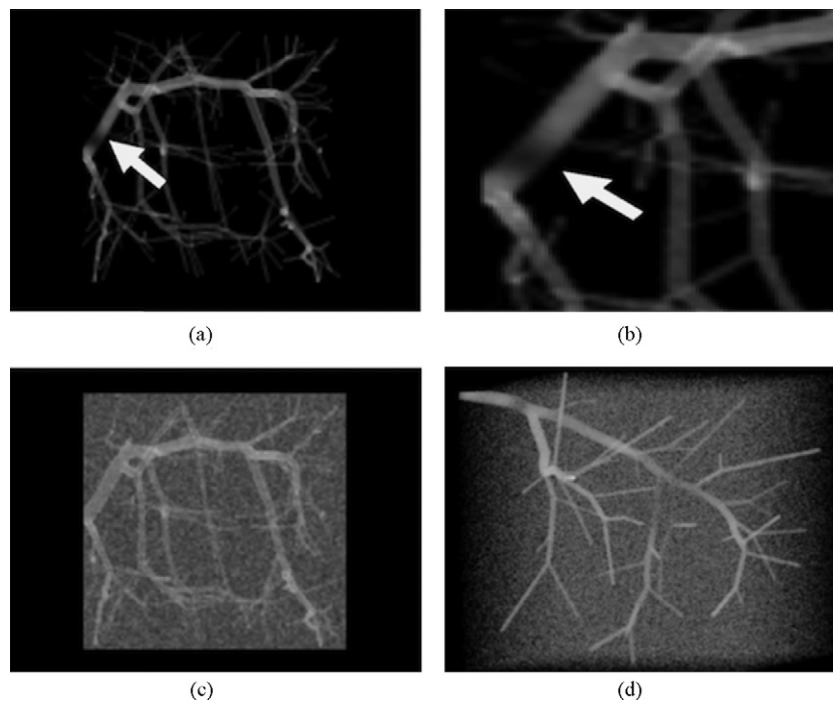


Fig. 9. Simulating imaging artifacts. (a) Drop out (shadow) artifact degrading the vasculature (close-up in (b)), which typically causes difficulties for vessel segmentation techniques. (c) Volume corrupted by Gaussian and (d) impulse (salt and pepper) noise.

ing fluid viscosity and perfusion pressure, term pressure, terminal flow, and others, to provide realistic, physically based vasculature.

The motivation for this work is the provision of an abundance of 3D images of vasculature with known ground-truth analyses. The ability to generate as many volumetric images as needed with the corresponding underlying ground-truth segmentation (i.e. binary labelled image) serves as a benchmark for evaluating and validating the vessel segmentation algorithms continuously being proposed (e.g. [5,11,8,31]), but seldom compared. Further, the generated data is useful for validating image analysis techniques that provide high level analysis of the vascular tree and not only the binary segmentation (e.g. [7,9]). This includes, for example, discovering the tree hierarchy, the lengths of branches, and the locations of bifurcation points. This validation of analysis algorithms is enabled since a high level representation of the tree hierarchy in the form of an attributed graph is generated along with every synthetic image volume. The graph's vertices and edges represent, respectively, the bifurcation points and their locations in the volume, and the branches with their corresponding radii, length, and flow values.

The steps for validating image segmentation and analysis algorithms using the simulated results are known in the medical image analysis community and generally proceed as follows. First, our approach is used to generate (i) a ground-truth binary segmentation volume (with 1 voxels within the vascular tree and 0 otherwise); (ii) a degraded version of the volume mimicking imaging artifacts, e.g. blurring and noise, or more elaborate medical image acquisition simulators such as POSSUM or SimSET; and (iii) a set of ground-truth measurements and analyses including tree topology, branch radii, lengths, or other measures such as DM, ICM, and SOAM. Second, the automatic or semi-automatic algorithm under evaluation is applied to the degraded volume ((ii) above) to produce the segmentation and other analysis results. Finally, the results from the algorithm under evaluation are compared to the ground-truth results ((i) and (iii) above). Comparing the segmentation results is typically carried out using the well-known Dice Similarity Coefficient [32] or the Jaccard index [33].

Vessel segmentation algorithms can also be evaluated by measuring how well they localize the medial axes of the vessels, e.g. by calculating the Hausdorff distance between the medial axes of the vessels extracted from the segmentation results of the algorithm and from the ground-truth binary volume. Errors in bifurcation location and lengths or radii of vessels can also be reported given that the simulation generates their ground-truth values. However, if branch correspondence is unknown then more elaborate tree matching algorithms must be applied [34]. Validating the topology of the extracted tree can be performed by measuring the graph edit distance [35].

Another useful application of the large numbers of physically based simulated data sets is training machine learning algorithms. In medical imaging, learning techniques suffer from the problem of high-dimensional, small sample size datasets ("the curse of dimensionality"), in which even the smallest of typical 2D scalar medical images (few hundred pixels, squared) can be seen as samples in tens-of-thousands dimensional space. The ability to generate very large numbers of realistic data sets with corresponding ground-truth analyses therefore holds a promise for addressing this problem.

The results presented are intended to showcase the capabilities of the algorithm by using a wide variety of parameter values. A researcher who is interested in a particular organ, e.g. brain, kidney, or lungs, must customize the parameters (e.g. number of terminal nodes, oxygen demand map) accordingly. Similar investigation may be carried in the future aiming at customizing the method for venous rather than arterial vasculature. We note that there is inherent randomness in the tree generation procedure of our algorithm (Section 3.4). Therefore, it will be highly unlikely that the tree generated by our method will be exactly equal to a real tree. One possible future work is to segment a large number of real trees and analyze their characteristics (e.g. number of branching points, histograms of branch lengths, etc.) and compare this to the results generated by our algorithm.

Our current algorithm is designed to create a vascular tree to meet a particular demand for nutrients and therefore relies

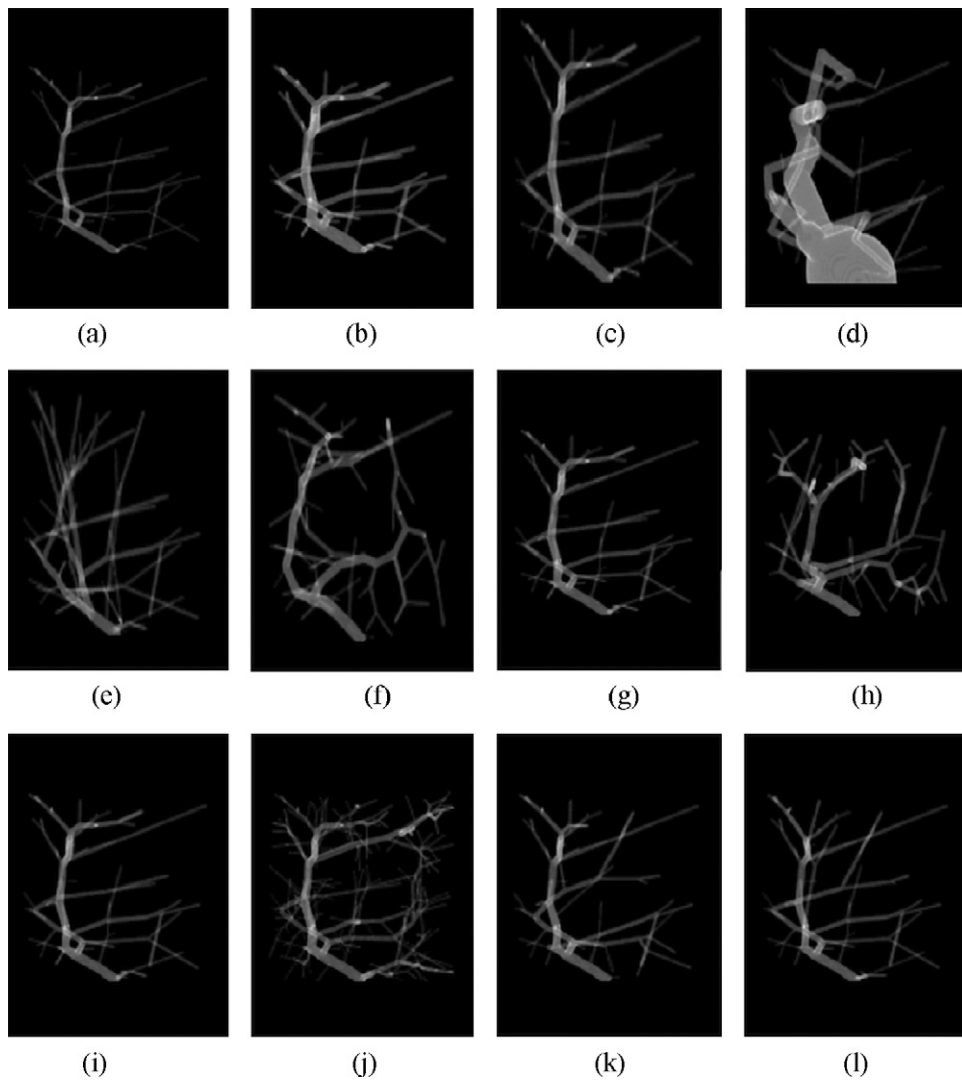


Fig. 10. Effect of varying different simulation parameters. (a) $Q_{perf} = 3.33$ and (b) $Q_{perf} = 28.33$, with $P_{term} = 83$ mmHg. Note the increase in radii. Similar images to (a) and (b) are obtained when decreasing P_{perf} from 233 to 100 mmHg, changing P_{term} from 50 to 100 mmHg for $P_{perf} = 133$ mmHg, or changing η from 0.016 to 0.066. (c) $\gamma = 2.55$ (a value within the recommended range?) and (d) $\gamma = 1.0$ (outside the range). (e) $\lambda = 3$, $\mu = 1$ and (f) $\lambda = 1$, $\mu = 3$. Note the shorter and thicker branches in (f). (g) $D_B = 10$ and (h) $D_B = 1$. Note how D_B controls the distance between branches. Number of terminal nodes (i) $N = 50$ and (j) $N = 200$. Varying the number of closest branches k examined as candidates to bifurcate: (k) $k = 3$ and (l) $k = 10$.

on non-uniform (based on demand) random sampling. A possible extension of our current work is to make the algorithm more specific to a deterministic branching pattern of a particular anatomy (e.g. the airways in the lungs have a particular branching pattern). Therefore, it may be a worthwhile future work to modify the algorithm so that it respects such prior knowledge when available, e.g. by presenting the algorithm with additional geometrical and topological rules that can not be violated, while allowing some degree of randomness controlled by the user. A by-product of this extension will be the generation of anatomical labels for branches (e.g. superior/inferior lobar bronchi in lung airway trees or cerebral/communicating in brain vasculature) for validating tree-labeling algorithms.

The initial simulation generates straight branches optimized to meet the demand for nutrients, whereas, in reality, branches may not perfectly straight segments. One possible future work is to find the optimal curved branches that meet the demand for nutrients. However, optimizing curves introduces many additional degrees of freedom. Clearly, a straight line segment in 3D is described by only 6 parameters (the coordinates of the starting and ending points) whereas a curve requires many more parameters (e.g. the coordi-

nates of several control points of a spline). Therefore, optimizing curves rather than straight lines increases the search space and the computational complexity of the simulation. In this work, we trade off optimality of curved branches for lower computational complexity, by spatially warping the generated straight-branched tree to introduce curvedness in the branches. Simulating realistic curved branches (even if not perfectly optimal) is useful for validating vessel tracking algorithms, since tracking straight vessels is easier than tracking curved vessels. Optimizing for curved branches or quantifying how the branch curving process affects the oxygen supply is left for future exploration.

The current algorithm produces a model of the vasculature and an image containing the vascular tree only, without any surrounding organs or structures. This is not unrealistic as several angiographic medical imaging modalities produce similar images (i.e. suppress image contrast due surrounding tissue and enhance only the contrast of vasculature), as in magnetic resonance angiography (MRA) or computed tomography angiography (CTA). In future work, we intend to embed other geometrical objects, e.g. spheres, ellipsoids, or even more realistic organs (e.g. segmented from real medical images), in such a way that these objects are close

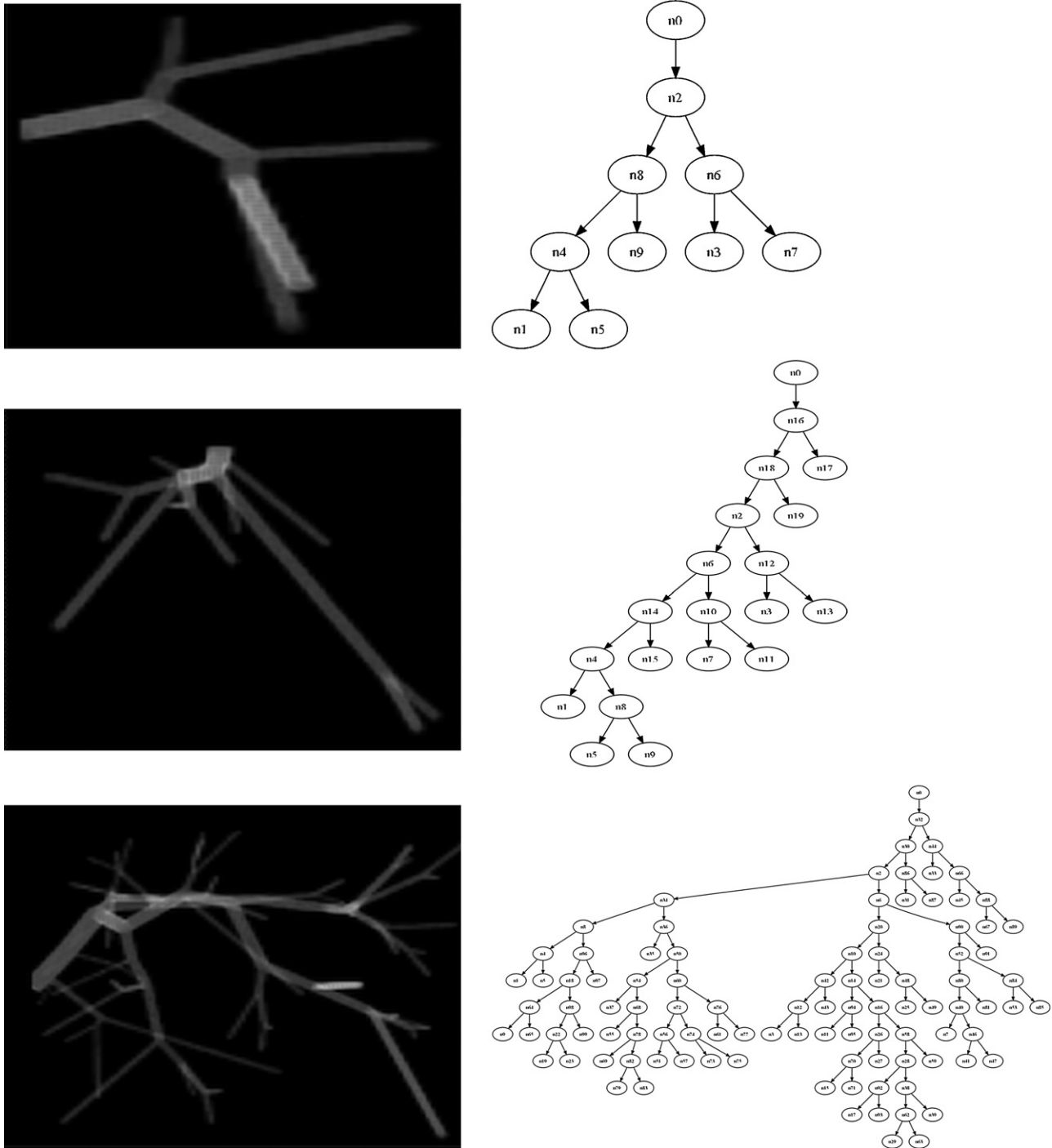


Fig. 11. Vascular volumes and corresponding graphs of varying complexities, i.e. with increasing N .

or even touch, but do not intersect with the generated vasculature. This will simulate more realistic embedding of the vasculature in background tissue and will create more challenging scenario for automated segmentation algorithms. To make the simulation even more realistic and to provide more challenging data sets for validating segmentation, the current approach must be augmented with models of pathology (e.g. stenoses and aneurysms) as well as other imaging artifacts (e.g. mimicking image blurring due to patient motion) or more extreme image degradation (e.g. using DeformIt [29]). More elaborate, modality-specific simulations can be obtained by using the binary image output of our algorithm as input to existing simulators, e.g. POSSUM [26] or SimSET [27].

Acknowledgments

We thank James Styles for code development and Lisa Tang for assistance in developing the deformable spatial warps. We thank Richard J. Wagner for provision of the Mersenne Twister random number. We used the Insight toolkit ITK (www.itk.org) to create the volumes, CMake (www.cmake.org) to build the project workspace, VolView (www.volview.org) to perform MIP visualization, GXL (www.gupro.de/GXL) to represent graph, and GraphViz (www.graphviz.org) to visualize the graphs. Partial funding for this research has been provided by the National Science and Engineering Research Council (NSERC) of Canada.

References

- [1] Tawhai MH, Burrowes KS. Modelling pulmonary blood flow. *Respiratory Physiology and Neurobiology* 2008;163(1–3):150–7. *Respiratory Biomechanics*, <http://dx.doi.org/10.1016/j.resp.2008.02.016>.
- [2] Bullitt E, Gerig G, Pizer S, Lin W, Aylward S. Measuring tortuosity of the intracerebral vasculature from MRA images. *IEEE Transactions on Medical Imaging* 2003;22(9):1163–71, <http://dx.doi.org/10.1109/TMI.2003.816964>.
- [3] Aylward S, Bullitt E. Initialization, noise, singularities, and scale in height ridge traversal for tubular object centerline extraction. *IEEE Transactions on Medical Imaging* 2002;21(2):61–75, <http://dx.doi.org/10.1109/42.993126>.
- [4] Kirbas C, Quek F. A review of vessel extraction techniques and algorithms. *ACM Computing Surveys* 2004;36(2):81–121, <http://dx.doi.org/10.1145/1031120.1031121>.
- [5] Wink O, Niessen W, Viergever M. Multiscale vessel tracking. *IEEE Transactions on Medical Imaging* 2004;23(1):130–3, <http://dx.doi.org/10.1109/TMI.2003.819920>.
- [6] Manniesing R, Viergever MA, Niessen WJ. Vessel enhancing diffusion: a scale space representation of vessel structures. *Medical Image Analysis* 2006;10(6):815–25, <http://dx.doi.org/10.1016/j.media.2006.06.003>.
- [7] McIntosh C, Hamarneh G. Vessel crawlers: 3D physicallybased deformable organisms for vasculature segmentation and analysis. In: *Proceedings of the IEEE computer society conference on computer vision and pattern recognition*. 2006. p. 1084–91, <http://dx.doi.org/10.1109/CVPR.2006.329>.
- [8] Yan P, Kassim AA. Segmentation of volumetric MRA images by using capillary active contour. *Medical Image Analysis* 2006;10(3):317–29, <http://dx.doi.org/10.1016/j.media.2005.12.002>.
- [9] Manniesing R, Viergever M, Niessen W. Vessel axis tracking using topology constrained surface evolution. *IEEE Transactions on Medical Imaging* 2007;26(3):309–16, <http://dx.doi.org/10.1109/TMI.2006.891503>.
- [10] Poon K, Hamarneh G, Abugharbieh R, Live-Vessel. Extending livewire for simultaneous extraction of optimal medial and boundary paths in vascular images. In: *Medical image computing and computer-assisted intervention*; 2007. pp. 444–451, http://dx.doi.org/10.1007/978-3-540-75759-7_54.
- [11] Li H, Yezzi A. Vessels as 4-D curves: Global minimal 4-D paths to extract 3-D tubular surfaces and centerlines. *IEEE Transactions on Medical Imaging* 2007;26(9):1213–23, <http://dx.doi.org/10.1109/TMI.2007.903696>.
- [12] Preim B, Oeltze S, Mlejnek M, Groeller E, Hennemuth A, Behrens S. Survey of the visual exploration and analysis of perfusion data. *IEEE Transactions on Visualization and Computer Graphics* 2009;15(2):205–20, <http://dx.doi.org/10.1109/TVCG.2008.95>.
- [13] Cocosco CA, Kollokian V, Kwan RKS, Pike GB, Evans AC. BrainWeb: online interface to a 3D MRI simulated brain database. *NeuroImage* 1997;5:425.
- [14] Reilhac A, Batan G, Michel C, Grova C, Tohka J, Collins D, Costes N, Evans A. Pet-sorteo: validation and development of database of simulated PET volumes. *IEEE Transactions on Nuclear Science* 2005;52(5):1321–8, <http://dx.doi.org/10.1109/TNS.2005.858242>.
- [15] Christensen GE, Geng X, Kuhl JG, Bruss J, Grabowski TJ, Pirwani IA, Vannier MW, Allen JS, Damasio H. Introduction to the non-rigid image registration evaluation project (NIREP). In: *Biomedical image registration*, vol. 4057 of lecture notes in computer science. Heidelberg: Springer Berlin; 2006. pp. 128–135, http://dx.doi.org/10.1007/11784012_16.
- [16] Gerig G, Jomier M, Chakos M. Valmet: a new validation tool for assessing and improving 3D object segmentation. In: *Medical image computing and computer-assisted intervention*; 2008. pp. 516–523, http://dx.doi.org/10.1007/3-540-45468-3_62.
- [17] Warfield S, Zou K, Wells W. Simultaneous truth and performance level estimation (STAPLE): an algorithm for the validation of image segmentation. *IEEE Transactions on Medical Imaging* 2004;23(7):903–21, <http://dx.doi.org/10.1109/TMI.2004.828354>.
- [18] Prusinkiewicz P, Hammel M, Mech R, Hanan J. The artificial life of plants. *Artificial Life for Graphics, Animation, and Virtual Reality*, SIGGRAPH. Course Notes 1995;7:1–38.
- [19] Schreiner W, Buxbaum P. Computer-optimization of vascular trees. *IEEE Transactions on Biomedical Engineering* 1993;40(5):482–91, <http://dx.doi.org/10.1109/10.243413>.
- [20] Karch R, Neumann F, Neumann M, Schreiner W. A three-dimensional model for arterial tree representation, generated by constrained constructive optimization. *Computers in Biology and Medicine* 1999;29(1):19–38, [http://dx.doi.org/10.1016/S0010-4825\(98\)00045-6](http://dx.doi.org/10.1016/S0010-4825(98)00045-6).
- [21] Kretowski M, Rolland Y, Bzy-Wendling J, Coatrieux J-L. Fast algorithm for 3-D vascular tree modeling. *Computer Methods and Programs in Biomedicine* 2003;70(2):129–36, [http://dx.doi.org/10.1016/S0169-2607\(01\)00200-0](http://dx.doi.org/10.1016/S0169-2607(01)00200-0).
- [22] Szczerba D, Szekely G. Macroscopic modeling of vascular systems. In: *Medical image computing and computer-assisted intervention*, vol. 2489; 2002. pp. 284–292, http://dx.doi.org/10.1007/3-540-45787-9_36.
- [23] Zamir M. Distributing and delivering vessels of the human heart. *The Journal of General Physiology* 1988;91(5):725–35, <http://dx.doi.org/10.1085/jgp.91.5.725>.
- [24] Matsumoto M, Nishimura T. Mersenne twister: a 623-dimensionally equidistributed uniform pseudo-random number generator. *ACM Transactions on Modeling and Computer Simulation* 1998;8(1):3–30, <http://dx.doi.org/10.1145/272991.272995>.
- [25] Winter A. Exchanging graphs with GXL. In: *Graph drawing*, vol. 2265 of lecture notes in computer science. Heidelberg: Springer Berlin; 2002. pp. 54–58, http://dx.doi.org/10.1007/3-540-45848-4_58.
- [26] Drobnjak I, Gavaghan D, Sul E, Pitt-Francis J, Jenkinson M. Development of a functional magnetic resonance imaging simulator for modeling realistic rigid-body motion artifacts. *Magnetic Resonance in Medicine* 2006;56(2):364–80, <http://dx.doi.org/10.1002/mrm.20939>.
- [27] Harrison R, Gillispie S, Lewellen T. Simulation System for Emission Tomography (SimSET): using simulation to research ideas in emission tomography (PET and SPECT). *Journal of Nuclear Medicine* 2008;49(Meeting Abstracts 1), 157P–b–.
- [28] Jensen J, Nikolov I. Fast simulation of ultrasound images. In: *IEEE ultrasonics symposium*, vol. 2; 2000. pp. 1721–1724, <http://dx.doi.org/10.1109/ULSYM.2000.921654>.
- [29] Hamarneh G, Jassi P, Tang L. Simulation of ground-truth validation data via physically- and statistically-based warps. In: *Medical image computing and computer-assisted intervention*; 2008. pp. 459–467, http://dx.doi.org/10.1007/978-3-540-85988-8_55.
- [30] Rueckert D, Sonoda L, Hayes C, Hill D, Leach M, Hawkes D. Non-rigid registration using free-form deformations: application to breast MR images. *IEEE Transactions on Medical Imaging* 1999;18(8):712–21, <http://dx.doi.org/10.1109/42.796284>.
- [31] Lesage D, Angelini ED, Bloch I, Funka-Lea G. A review of 3D vessel lumen segmentation techniques: models, features and extraction schemes. *Medical Image Analysis* 2009;13(6):819–45, <http://dx.doi.org/10.1016/j.media.2009.07.011>.
- [32] Dice LR. Measures of the amount of ecologic association between species. *Ecology* 1945;26(3):297–302, <http://dx.doi.org/10.2307/1932409>.
- [33] Jaccard P. Étude comparative de la distribution florale dans une portion des alpes et des jura. *Bulletin del la Société Vaudoise des Sciences Naturelles* 1901;37:547–79.
- [34] Charnoz A, Agnus V, Malandain G, Soler L, Tajine M. Tree matching applied to vascular system. In: *Graph-based representations in pattern recognition*; 2005. pp. 183–192, <http://dx.doi.org/10.1007/b107037>.
- [35] Tang WH, Chung ACS. Cerebral vascular tree matching of 3D-RA data based on tree edit distance. In: *Medical imaging and augmented reality*, vol. 4091 of lecture notes in computer science. Heidelberg: Springer Berlin; 2006. pp. 116–123, http://dx.doi.org/10.1007/11812715_15.

Ghassan Hamarneh received the bachelor's degree from the Department of Electrical Engineering at Jordan University in 1995, the master's degree, with distinction, in digital communications from Chalmers University, Sweden, in 1997, and the doctoral degree from the Department of Signals and Systems at Chalmers University of Technology, Sweden, in 2001. He joined the School of Computing Science at Simon Fraser University, Canada, as an assistant professor in 2003 and has been an associate professor since 2009. Previously, he was a postdoctoral fellow at the Mouse Imaging Centre at the Hospital for Sick Children and the Department of Medical Biophysics at the University of Toronto, Canada (2001–2003). He was a pre-doctoral research fellow in the Department of Computer Science at the University of Toronto, Canada (2000–2001). His primary research interest is in medical image analysis. He is the co-director and co-founder of the Medical Image Analysis Lab (MIAL) at Simon Fraser University. He is a senior member of the IEEE.

Preet Jassi received the bachelor's degree from the School of Computing Science at Simon Fraser University, Canada in 2009. His primary research interest is in medical image analysis, specifically physiologically inspired segmentation algorithms. He currently works for Yahoo! Inc as a Tech Yahoo! He is a member of the Medical Image Analysis Lab at Simon Fraser University.