Threshold Relaxation is an Effective Means to Connect Gaps in 3D Images of Complex Microvascular Networks

John Kaufhold
SAIC Intelligent Systems Division
4001 N. Fairfax Drive, Ste. 600, 22203
kaufholdj@saic.com

Phil Tsai, Pablo Blinder, David Kleinfeld
UCSD Physics Dept.
9500 Gilman Drive, 92093

Abstract

All optical histology (AOH) uses femtosecond pulse plasma mediated laser ablation in conjunction with two-photon laser scanning microscopy (TPLSM) to produce large anatomical volumes at micrometer-scale resolution. Specifically, we use AOH to produce ~1mm^3 datasets of cerebral vasculature with the goal of modeling its structural and physiological relationship to neuronal cells. Generating a binary mask of the cerebral vasculature is a first step towards this goal, and many methods have been proposed to segment such 3D structures. However, many analyses of the tubular vascular network (e.g., average vessel segment length, radii, point-to-point resistance and cycle statistics) are more efficiently computed on a vectorized representation of the data, i.e. a graph of connected centerline points. Generating such a graph requires sophisticated upstream algorithms for both segmentation and vectorization. Occasionally, the algorithms form erroneous gaps in the vectorized graph that do not properly represent the underlying anatomy. We present here a method to connect such gaps via local threshold relaxation. The method A) fills gaps by relaxing a binarization threshold on the grayscale data volume in the vicinity of each gap (found using the vectorization), B) computes a "bridging" strand for each gap, and C) produces a confidence metric for each "bridging strand". We show reconnection results using our method on real 3D microvasculature data from the rodent brain and compare to a tensor voting method.

1. Introduction

Understanding the fine details of the brain's vascular structure has recently received renewed interest [1-3]. Positron emission tomography (PET), magnetic resonance imaging (MRI), and intrinsic imaging exploit the neurovascular coupling between neurological activity, the ensuing oxygen and energy consumption, and increased blood perfusion to the activated brain regions. Although this relationship between neuronal activity and blood perfusion has been used to image brain activity, the microscopic details of the vascular response remains poorly understood, and investigators continue to debate which specific aspects of the neuronal activity elicits these observable changes [4, 5]. Furthermore, it has been found that the spatial extent of the imaged response extends beyond the anatomical limits of its corresponding neuronal origin[6, 7], a phenomenon likely related to the anatomical properties of the nearby vasculature.

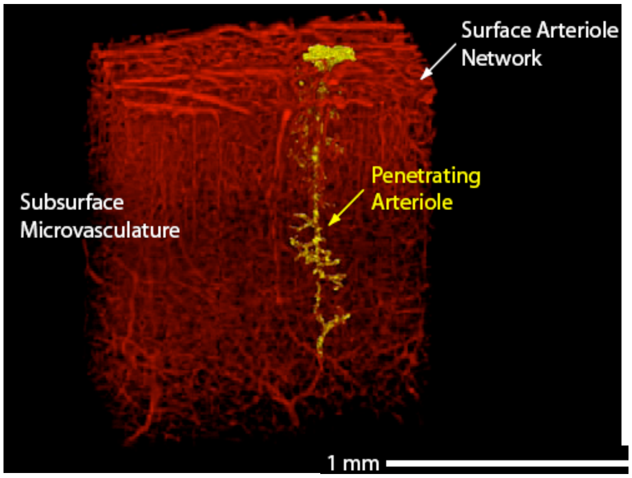

Figure 1: A 1mm x 1mm x 1mm section of mouse cortical microvasculature. The pia is at the top and the white matter is at the bottom. The local branching of one "Penetrating Arteriole" is shown in yellow.

A set of stroke studies provides an example of this link between vascular topology and its function; these studies demonstrate distinct topological organizations across the cortical vasculature. Three distinct networks could be distinguished (Figure 1): a network of surface arterioles, a set of penetrating arterioles, and a subsurface network of microvasculature that includes the capillary beds. The surface arterioles (40-150 µm diameter vessels) constituting the surface branches of the anterior, posterior and middle cerebral arteries, form a 2-D network at the pial surface of the brain. Here, the presence of anastomoses, i.e., interconnections between vessels to form loops, ensures that blood flow can be re-routed to bypass potential blockages, thus providing a robust continuous blood supply [8]. From the surface a set of penetrating arterioles (~30-100 µm diameter vessels) plunge into the brain and connect the surface arterioles to the subsurface microvasculature. Some of these penetrating arterioles can traverse the entire depth of the cortex with little to no branching (preliminary observation) as they provide blood to the deep layers of cortex. Penetrating arterioles form “bottlenecks” to flow, in that an occlusion of a single penetrating arteriole has devastating consequences as blood supply is drastically diminished in a ~500 μm diameter cylinder around the vessel [9]. The third network consists of microvessels (<7μm in diameter) that form a densely-packed, 3-D subsurface network. As in the surface arterioles, loops within the microvasculature network allow for rerouting of blood flow around an occlusion [10].

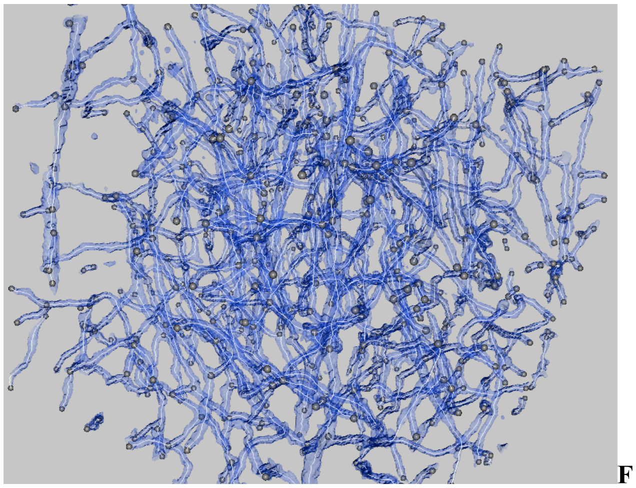

Figure 2: Vectorized network of a small part of the volume in Figure 1 (strand endpoints and bifurcations are marked with black spheres). The white polylines indicate the “strands” defined between the black spheres. The vessel mask is the blue isosurface.

A detailed knowledge of both the cellular and vascular spatial organization at the micrometer scale is crucial to understanding the neurovascular dynamics both under normal and pathological conditions. More precisely, a complete high resolution –gap-free♠– *vectorized* (i.e. graph) representation of the vasculature accompanied by all cell nuclei locations (both neurons and non-neurons) in a sufficiently large cortical volume would enable such a study. The vectorized representation is required to move from more rudimentary morphological statistics to a system level approach where network properties per se can be measured, not estimated from isolated pieces of information. For example, such a study could identify the presence of repeating microvascular motifs and establish whether the microvasculature is organized as a continuum or as a set of connected microdomains.

In addition to interest by neuroscientists, 3D tubular structures in general are of interest for many applications, including finding/measuring vessels and airways in lung Computer Aided Detection (CAD) for lung abnormalities, estimation of stenoses in medical images, generating virtual colonoscopy fly-through paths, generating 3D articulable models for graphics, and nonrigid anatomical registration using vessel trees as fiducials [11].

1.1. 3D Vectorized Tubular Networks

Many methods exist to segment 3D vessels from raw data [12]. Multiscale eigenanalyses of local Hessian operators can enhance local rod-like shapes of varying radii [13,14], e.g.. Many methods also exist to extract centerlines from binary images of tubes. Skeletonization methods can accomplish this, but due to noise or real bulges and the ill-conditioned medial axis transform (MAT), many small branches develop which are unrelated to the larger objects the MAT is meant to represent. In 3D, MATs can also develop “medial surfaces” which are not centerlines at all. Curve evolution methods and morphological operators, e.g., have been introduced to mitigate these issues [15,16,17].

Recent work in vectorizing 3D microvascular networks includes [1,3,15,18]. Related work in connectomics also requires strategies to connect gaps [19]. Though vectorization methods differ, all resulting vectorizations consist of a set of “strands” (called *segments* in [20]). As in [20] a strand is a 1D graph “defined between two bifurcations, between one bifurcation and one [endpoint], or between two [endpoints]” (see Figure 2). Note that all endpoints are connected to exactly one strand.

Though both 3D segmentation and vectorization of tubular networks are fairly well-studied, as noted in [20], the post-processing step of connecting gaps in the vectorization is not. The focus of this paper is finding and connecting such gaps.

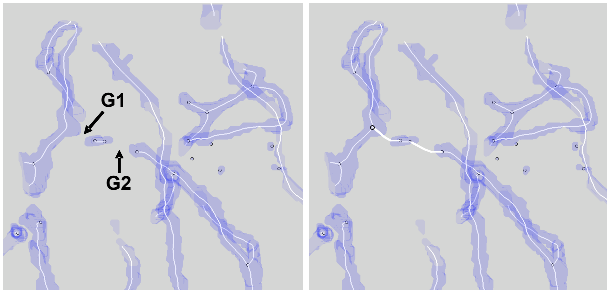

Figure 3: Gaps G1 and G2 (left: unconnected, right: illustrating the desired output “bridging strands” in white. The original mask, B_V, is shown in blue.

1.2. Gaps in Graphical Representations of 3D Tubular Networks

The left panel of Figure 3 illustrates gaps G1 and G2 in a small volume of interest interior to the volume in Figure 2. Due to one or more upstream causes including staining, imaging, segmentation, and/or vectorization, the vectorization

---

♠ Artificial gaps can be introduced by the stitching, segmentation or vectorization algorithms, but gaps may also reflect ongoing angiogenesis –the process of new vessel formation– yet a qualitative survey of our *in vivo* data on animals of the same age range as the ones used here does not support this hypothesis.

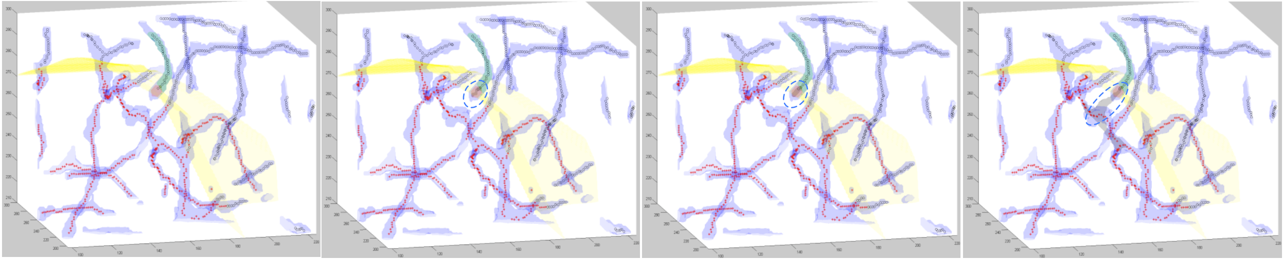

Figure 3: An illustration of the threshold relaxation process. The original mask with a gap is shown with a blue isosurface. The neighborhood of the endpoint to connect, P_{Ei}, is shown with a red isosurface. Points in G_V on the same strand as P_{Ei} are highlighted with a green isosurface. The yellow isosurface separates points that are closer to P_{Ei} than to other points on the same strand. The resulting candidate connection points in G_V are shown with red *'s; excluded points in G_V are shown with black o's. On the far left, the threshold, T_z, admits no points in the gap connecting mask, B_z. Moving right, the threshold is relaxed (lowered) and the mask contiguous with P_{Ei} (B_z) is illustrated with a black isosurface. Moving right, as the threshold is relaxed further, more points near P_{Ei} are added to B_z. At the far right, P_{Ei} is connected to a number of candidate points in G_V by B_z.

is not a single connected graph of circuits as would be anatomically expected (modulo edge effects). In the right panel of Figure 3, gaps are connected by "bridging strands" (in thicker white). Thus our goal is to find gaps, G_i, and compute a bridging strand, S_i, for each.

1.3. Published Gap-connection Methods

The problem of connecting gaps is well-studied in 2D as weak edge linking downstream of edge detection has frustrated automated edge detection and image analysis for decades [21, 22]. Some recent results on edge linking highlighting different linking strategies can be found in the references [23,24,25,26], but because the literature on 2D edge linking (especially for road network inference) is enormous, we omit the references and assert some combination of 2D methods may conceptually map to the 3D method presented here.

Though the problem is well studied in 2D, far fewer results have been collected for the analogous 3D problem [27,28]. One promising connection method grounded in the formalism of tensor voting uses the vectorized graph alone to infer gap connections [20]. More recently, the method has been shown to perform favorably to mathematical morphology and an Ising model for the same task [29]. Though promising, the method in [29] relies only on the graph, and thus cannot use the underlying grayscale vessel data to inform the gap filling method.

2. Gap Connection via Threshold Relaxation

The gap connection method presented here 1) exploits both the topology of the vectorized graph for gap-finding as well as the underlying grayscale data to infer connections, 2) is not limited in connection size, 3) prevents backtracking, 4) is conceptually simple, modular, and extensible.

Threshold Relaxation Summary

The method accepts as input a grayscale image volume, E_V, the corresponding binary segmentation, B_V, and its vectorization, G_V. The vectorization is a graph, $G_V=(V_{Gv},E_{Gv})$. Specifically, $V_{Gv}=\{P_0,P_1,\ldots,P_N\}$, where each vertex, P_i, is a 3D location. Edges, E_{Gv}, indicate which vertices are connected to which other vertices. The method can be summarized as a 2-step process, which we discuss next.

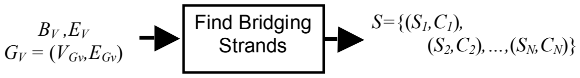

Figure 4: Algorithm accepts as input a binary mask of the microvasculature, B_V, a continuous-valued volume, E_V, and a graph, G_V and produces a set of "bridging strands", S_i, and their confidence levels, C_i.

Step 1: Finding a Connecting Point

Every gap presumably originates at an endpoint vertex, P_{Ei}, in the graph, G_V. In a local bounding box about P_{Ei}, we relax a threshold, T_z, on the grayscale volume, E_V to produce a new binary mask, B_z (defined as $E_V > T_z$). B_z is then trimmed to disallow backtracking to centerline points that fall "behind" P_{Ei}, including those on the originating strand. The threshold, T_z, is relaxed until a connection is made between P_{Ei} and at least one other point in G_V through B_z. If more than one vertex becomes connected, then the connection point P_{Ci}, is chosen so as to minimize the pathlength, constrained along B_z, between P_{Ei} and P_{Ci}. This process is illustrated in Figure 4.

Step 2: Computing the Bridging Strand

The revised binary mask, B_z, can be large and include many points irrelevant to finding the 3D path between P_{Ei} and P_{Ci}. Therefore, we further refine B_z. using a binary search over thresholds to tighten the mask to include the fewest voxels while still linking P_{Ei} and P_{Ci}. We then use a "paired pathlength distance transform" to eliminate all points in the mask except those most likely to participate in the path, producing a new, smaller mask in the vicinity of the gap, B_G. Dijkstra's algorithm then produces the output strand, S_i, connecting P_{Ei} to P_{Ci} constrained to B_G.

In § 2.1 we discuss the two different distance transforms used in the algorithm. We sketch both steps of the gap connection algorithm as pseudocode in § 2.1 and § 2.2.

2.1. Distance Transforms

Euclidean Distance Transform

A 2D binary mask, B, consisting of bright and dark pixels, with values 1 and 0 respectively, is shown in the left panel Figure 6. The standard Euclidean distance transform, $D_r(B)$, yields the distance from every bright pixel to its closest dark pixel, as shown in the center panel of Figure 6. Note that the distance transform is 0 everywhere outside the mask.

Pathlength Distance Transform

The pathlength distance transform, $D_p(B,s)$, from a chosen starting point, s, is shown in the right panel of Figure 6. The pathlength distance, D_p, is defined as the geodesic, e.g., shortest path, from one point in the mask to another point in the mask constrained such that all intervening edges are also in the mask. By definition, the pathlength distance between points on the mask and points outside the mask is ∞, i.e., they are not connected. Many methods can be used to compute the pathlength distance transform, including fast marching methods and Chamfer methods, e.g.. In this work, we use the Chamfer$_{3,4,5}$ method for pathlength computation as defined in [15].

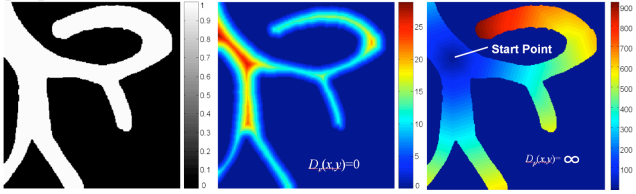

Figure 5: A 2D binary mask, *B* (left), its Euclidean distance transform, $D_r(B)$ (middle), and a pathlength distance transform, $D_p(B)$, where the pathlength distance is computed from the "Start Point", *s* (right).

2.2. Finding a Connection via Local Threshold Relaxation

A small 2D cross section through the enhanced grayscale volume, E_V, is shown in Figure 7. In the same figure, the blue overlaid outline indicates the corresponding binary mask, B_V. See Figures 2&3 for examples of a corresponding 3D graph, G_V.

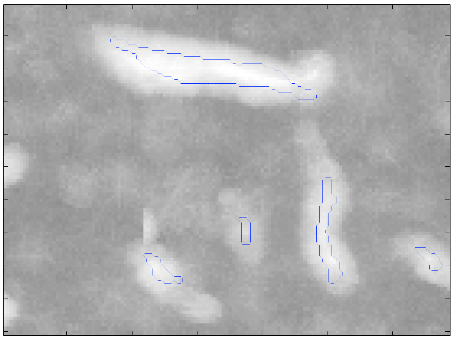

Figure 6: Continuous-valued "vessel network" volume, E_V, cross-section (grayscale) and outline of in-slice vessel network mask, B_V, (blue contour).

The pseudocode to find the connection points from the mask and graph inputs is given below.

ThresholdRelaxation

```
1:  Given $B_V, E_V, G_V$, Find $S, C$
2:  $P_E$= { all interior degree 1 vertices in $G_V$ }
3:  for all $P_{E_i} \in P_E$ do
4:      GetBoundingBox($P_{E_i}$): $B_V \rightarrow B_B, E_V \rightarrow E_B, G_V \rightarrow G_B$
5:      $P_{S_i} = \{$Points on strand containing $P_{E_i} \setminus P_{E_i}\}$
6:      $B_R = D_r(B_W \setminus P_{E_i}) < D_r(B_W \setminus P_{S_i})$
7:      $\mu_{BG} = \mu(E_B \mid B_B = 0)$
8:      $\sigma_{BG} = \sigma(E_B \mid B_B = 0)$
9:      $V_C = \{V_{G_B} \setminus P_{E_i}\}$
10:     initialize $z = \frac{\max(E_V) - \mu_{BG} + \Delta}{\sigma_{BG}}$
11:     while $P_{L_i} = \emptyset$ do
12:         $T_z = \mu_{BG} + z\sigma_{BG}$
13:         $L_z = (E_B > T_z) \wedge B_R$
14:         $B_z = L_z$ contiguous with $P_{E_i}$
15:         $P_{L_i} = \{V_C \mid B_z(V_C) = 1\}$
16:         if $P_{L_i} \neq \emptyset$; $T_{C_i} = T_z$; return
17:         else $T_{U_i} = T_z$; relax $z$
18:     end while
19:     $D_C = D_P(B_z, P_{E_i})$
20:     $P_{C_i} = \arg\min_{P_{L_{ij}}} D_C(P_{L_{ij}})$
21:     $[S_i, C_i]$ = GetBridgingStrand($E_B, P_{E_i}, P_{C_i}, T_{C_i}, T_{U_i}$)
22: end for
```

The following notation applies to the **ThresholdRelaxation** pseudocode: Interior vertices are those vertices further from the edges of V by a distance $\geq \Delta E$. $X_V \rightarrow X_B$ restricts the volume X to only the bounding box, B, from the entire volume, V. $B_{W\setminus Pj}$ means the binary mask of all 1s except at locations Pj. The function $\mu(E|B)$ returns the mean of E where B is true; similarly, $\sigma(E|B)$ returns the standard deviation of E where B is true. The vertices of graph G are located at V_G.

In **ThresholdRelaxation:7-8**, the mean and standard deviation of the local volume, E_B, is computed where B_B=0. An example of the background and vessel distributions for one bounding box is shown in Figure 8.

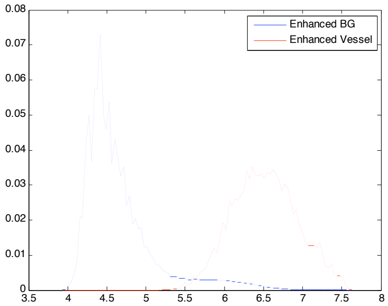

Figure 7: Separation of E_B "on/off" histograms by z-score. The blue curve is a histogram of $E_B(B_B = 0)$ and the red curve is a histogram of $E_B(B_B = 1)$.

2.3. Computing a Bridging Strand

Using the endpoint, connection point, thresholds, and local volumes found in **ThresholdRelaxation:1-20**, step 2 of the algorithm is **ThresholdRelaxation:21,** which can be written functionally as in **GetBridgingStrand** below.

In **GetBridgingStrand:1**, the tight threshold, T_T, is chosen via a binary search of thresholds between T_{Ci} and T_{Ui} such that $B_T(P_{Ei}) = 1$, $B_T(P_{Ci}) = 1$ and $D_P(B_T, P_{Ei}, P_{Ci}) < \infty$ (i.e. P_{Ei} and P_{Ci} are connected via B_T).

```
Function $[S_i, C_i]$ = GetBridgingStrand($E_B, P_{E_i}, P_{C_i}, T_{C_i}, T_{U_i}$)
1: $T_T$ = ChooseTightThreshold($E_B, P_{E_i}, P_{C_i}, T_{C_i}, T_{U_i}$)
2: $B_T = E_B > T_T$
3: $C_i = z(T_T) = \frac{T_T - \mu_{BG}}{\sigma_{BG}}$
4: $D_G = D_P(B_T, P_{E_i}) + D_P(B_T, P_{C_i})$
5: $B_G = D_G > \min(D_G) + \Delta$
6: $S_i$ = Dijkstra($B_G, P_{E_i}, P_{C_i}$)
7: Center $S_i$ inside $B_G$*
     * step omitted in results presented here
```

In the above pseudocode, D_G is the continuous-valued "paired pathlength distance" in the gap between P_{Ei} and P_{Ci}. B_G is a binary mask indicating where that distance, D_G, is smaller than the minimum paired pathlength distance plus some tolerance, Δ. An example of B_G is shown in Figure 9 as a green isosurface — note that B_G was derived from B_T, shown as a gray isosurface. The i^{th} "bridging strand", S_i, is computed via Dijkstra's shortest path algorithm from P_{Ei} to P_{Ci} constrained to B_G, and is depicted as a thick black polyline in Figure 9. The paired pathlength distance transform reduces Dijkstra's search space from the larger volume, B_T — depicted in gray, to the smaller volume, B_G — depicted in green. Lattice edges available to the search are shown in red in Figure 9.

The outputs of the algorithm are: $\{(S_1,C_1), (S_2,C_2), \ldots,(S_N,C_N)\}$, where each S_i consists of a list of coordinates that bridges a single gap between one vertex in V_G (P_{Ei}, e.g.) to another vertex in V_G. The coordinates in each strand, S_i, are compiled in sequential order, i.e, the first coordinate is connected to the 2^{nd}, the 2^{nd} to the 3^{rd}, etc.. This produces a monofilament graph with n points and n-1 edges. The confidence metric, C_N, is the z-score for the tight threshold, T_T, connecting all those points.

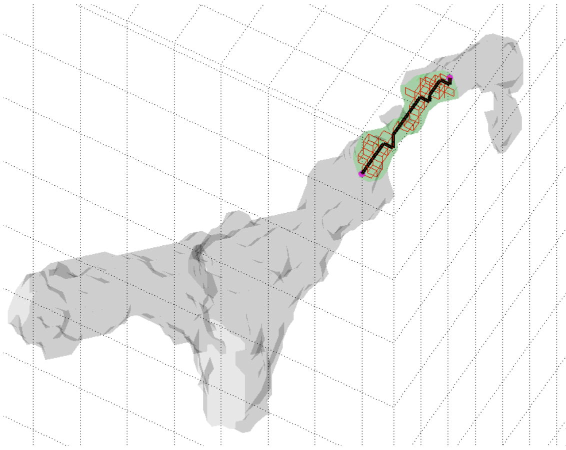

Figure 8: The tight mask that bridges the gap between P_{Ei} and P_{Ci} (red dots) is shown in gray. The gap mask, B_G, is shown in green. The candidate voxel lattice edges for the Dijkstra bridging strand search are shown in red, and the minimum path from P_{Ci} to P_{Ei} constrained to that lattice, the bridging strand, S_i, is shown in black.

3. Gap Connection Results

An example bridging strand, as computed by the algorithm described above, is shown in Figure 10 (in magenta). Additional results can be found at the end of the paper in Figure 11.

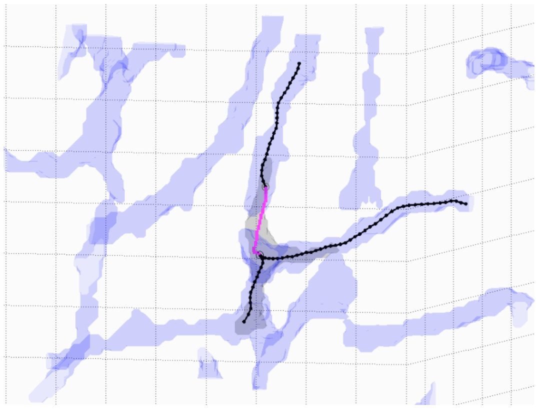

Figure 9: Outputs: Bridging strand, S_i (magenta). Only adjacent strands to the bridging strand are shown in black. The original mask, B_B, is shown in blue, the gap mask, B_T, is shown in gray.

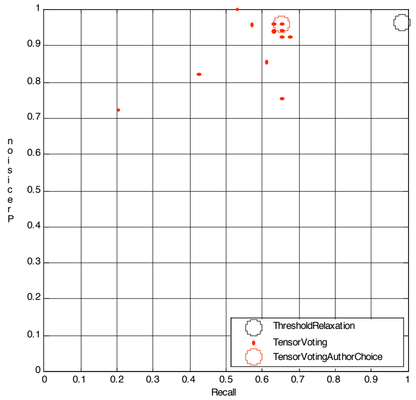

Figure 10: Precision-Recall comparison of tensor voting and threshold relaxation methods for gap connection.

Gaps were closed using both the threshold relaxation method and the tensor voting method in [20]. Run times were ~1 hour on a 2GHz machine for Threshold Relaxation and ~4 hours for Tensor Voting. For both algorithms, we visually examined the gap-connected graphs and classified the gaps visually. If either algorithm found a bridging strand in the vicinity of a real gap, it was given credit for a "true positive". Otherwise that gap was scored as a miss. Spurious connections outside the locations of real gaps were scored as false positives. Because only interior endpoints are connected in threshold relaxation, only misses in the interior were counted (and all false alarms were counted). Tensor voting was scored similarly, although it was given credit (true positives counted) for connections made near edges if it found them. Tensor voting requires two parameters, a characteristic gap length and an angle that controls allowable gap curvature; we screened parameters between 20 and 100 for each input parameter. We also requested a parameter set chosen by the author of the tensor voting method; the author was given access to our graph with gaps to determine how to set parameters. The resulting Precision-Recall curve is shown in Figure 11.

In general, both methods perform well on precision, but the threshold relaxation method outperforms on recall, meaning it does a better job of connecting all gaps. For some gaps, the threshold relaxation method connected true gaps larger than 30 voxels which the tensor voting method did not connect—these are false negative connections (i.e. misses) for the tensor voting method. In the PR curve shown, all bridges found by threshold relaxation bridges were allowed, regardless of the confidence score. In a small number of cases, this resulted in spuriously connected endpoints that did not correspond to real gaps (i.e. threshold relaxation false positives). This false positive rate can be reduced, e.g., by applying an acceptance limit to the confidence metric discussed above in § 2.3.

4. Discussion

The threshold relaxation method presented here enjoys a number of desirable characteristics: 1) It exploits the topology of the vectorized graph for gap-finding, 2) It exploits the underlying intensity data to guide connections, 3) It prevents backtracking, 4) It can connect potentially large gaps, 5) It is conceptually simple, 6) It is modular, and 7) it can be extended to incorporate more sophisticated search strategies (e.g., tensor voting). By visual examination, the algorithm performed well on reconnection tests with real data. In practice, most gaps can be connected by choosing a marginally relaxed (lower) threshold in the vicinity of the gap. In these cases, the gap connection algorithm intuitively finds that new lower threshold that will connect the gap through a vessel segment that is, in fact, represented in the original grayscale data, albeit at a lower intensity.

Though empirical tests bear out the relaxation method, the current implementation of the algorithm has limitations: The algorithm can only connect gaps in the vicinity of at least one endpoint. Conversely, if there is an endpoint associated with a "true" gap that does not merit reconnection, the relaxation process may lower the threshold excessively, leading to a spurious bridging of the gap through noise voxels. However, in the case where the threshold is lowered excessively, the confidence on the bridging strand can be used to reject such connections (not shown). Furthermore, since spurious endpoints are the ultimate cause of spurious connections, upstream improvements to vectorization that recognize morphological noise (i.e. "bumps" in the vessel mask) also mitigate this limitation.

Compared to a more sophisticated gap connection method like tensor voting [20], threshold relaxation method presented connected nearly all the gaps that tensor voting connected. It also connected larger gaps that were missed by tensor voting, but it also erroneously added some small loops. Theoretically, the tensor voting formalism is attractive because it takes into account the direction of vessels and makes incremental extensions in the direction of vessel axes more likely. The backtracking rejection mask, B_R, in the threshold relaxation method serves a similar purpose, but cannot discriminate small variations in direction.

Finally, the method presented both requires ***and exploits*** the underlying continuous-valued volume, E_V, corresponding to the vectorized graph, G_V, whereas the tensor voting method only requires the downstream vectorization, G_V. This final consideration, that the algorithm preferentially form bridges that are supported by grayscale data, represents either a limitation or a benefit, depending on the application at hand; our results clearly indicate that using the grayscale data helps. Hybrid methods, involving threshold relaxation and tensor voting or other methods are obviously attractive extensions.

5. Conclusion

The gap connection via threshold relaxation method is a simple tool to connect potentially large gaps in vectorizations of tubular networks. The method presented only computes the bridging strand, but moving forward, the bridging strand must be incorporated into the larger vectorization, G_V. We have tested the straightforward method of using the bridging strand to generate a "gap connecting mask" which should then connect the original mask, B_V. This new reconnected mask is then revectorized. Of course, only one vectorization step is necessary, and one can also incorporate all S_i directly into G_V and make the corresponding adjustments to other strand definitions via a low level re-indexing of G_V.

With accurate vectorizations, as discussed in §1, downstream tasks like microdomain identification, vascular network topology quantification, and anatomical statistic generation become tenable.

6. Acknowledgements

Thanks to Laurent Risser for enlightening correspondence and making available a version of the Tensor Voting method for gap filling. Thanks also to Mahnaz Maddah for making available her 3D Chamfer$_{3,4,5}$ pathlength distance transform.

7. References

[1] L.Risser, F.Plouraboue, A.Steyer, P.Cloetens, G.Le Duc, "From homogeneous to fractal normal and tumorous microvascular networks in the brain," J. Cerebral Blood Flow and Metabolism (2007) 27, 293-303.

[2] S.Heinzer, G.Kuhn, T.Krucker, E.Meyer, A.Ulmann-Schuler, M.Stampanoni, M.Gassmann, H.H.Marti, R.Muller, J.Vogel, "Novel three-dimensional analysis tool for vascular trees indicates complete micro-networks, not single capillaries, as the angiogenic endpoint in mice overexpressing human VEGF(165) in the brain," Neuroimage, 39(4):1549-58, 2008.

[3] F.Cassot, F.Lauwers, C.Fouard, S.Prohaska, L.Valerie, "A novel three-dimensional computer-assisted method for a quantitative study of microvascular networks of the human cerebral cortex," Microcirculation, 13(1):1-18, January-February 2006.

[4] A.Devor, P.Tian, N.Nishimura, I.C.Teng, E.M.C.Hillman, S.N.Narayanan, I.Ulbert, D.A.Boas, D.Kleinfeld, A.M.Dale, "Suppressed neuronal activity and concurrent arteriolar vasoconstriction may explain negative blood oxygenation level-dependent signal," J.Neuroscience, 27(16):4452-4459, 2007.

[5] N.K.Logothetis, J.Pauls, M.Augath, T.Trinath, A.Oeltermann, "Neurophysiological investigation of the basis of the fMRI signal," Nature, 412(6843):150-157, 2001.

[6] A.Grinvald, H.Slovin, I.Vanzetta, "Non-invasive visualization of cortical columns by fMRI," Nature Neuroscience, 3(2):105-7, 2000.

[7] M.Fukuda, C.-H.Moon, P.Wang, S.-G.Kim, "Mapping iso-orientation columns by contrast agent-enhanced functional magnetic resonance imaging: reproducibility, specificity, and evaluation by optical imaging of intrinsic signal," J.Neuroscience, 26(46):11821-11832, 2006.

[8] C.B.Schaffer, B.Friedman, N.Nishimura, L.F.Schroeder, P.S.Tsai, F.F.Ebner, P.D.Lyden, D.Kleinfeld, "Two-photon imaging of cortical surface microvessels reveals a robust redistribution in blood flow after vascular occlusion," PLoS Biology, 4(2):0001-0013, 2006.

[9] N.Nishimura, C.B.Schaffer, B.Friedman, P.D.Lyden, D.Kleinfeld, "Penetrating arterioles are a bottleneck in the perfusion of neocortex," PNAS, 104(1):365-370, 2007.

[10] N.Nishimura, C.B.Schaffer, B.Friedman, P.S.Tsai, P.D.Lyden, D.Kleinfeld, "Targeted insult to subsurface cortical blood vessels using ultrashort laser pulses: three models of stroke," Nature Methods, 3(2):99-108, 2006.

[11] A.C.S.Chung, H.-M.Chan, S.C.H.Yu, W.M.W.III, "2d-3d vascular registration between digital subtraction angiographic (dsa) and magnetic resonance angiographic (mra) images," in ISBI, pages 708-711, 2004.

[12] C.Kirbas, F.Quek, "A review of vessel extraction techniques and algorithms," ACM computing surveys, 36(2):81-121, 2004.

[13] K.Krissian, G.Malandain, N.Ayache, R.Vaillant, Y.Trousset, "Model-based multiscale detection of 3d vessels," Proc. CVPR, pp. 722, 1998.

[14] A.F.Frangi, W.J.Niessen, K.L.Vincken, M.A.Viergever, "Multiscale vessel enhancement filtering," LNCS 1496/1998:130, 1998.

[15] M.Maddah, H.Soltanian-Zadeh, A.Afzali-Kusha, A.Shahrokni, Z.G.Zhang, "Three-dimensional analysis of complex branching vessels in confocal microscopy images," CMIG, 29(6):487-498.

[16] F.F.Leymarie, P.J.Giblin, B.B.Kimia, "Towards surface regularization via medial axis transitions," Proc. ICPR, Vol. 3, pp. 123-126, 2004.

[17] S.Bouix, K.Siddiqi, A.Tannenbaum, "Flux driven automatic centerline extraction," Medical Image Analysis, 9(3):209-221, June 2005.

[18] J.A. Tyrrell, B.Roysam, E.di Tomaso, R.Tong, E.B.Brown, R.K.Jain, "Robust 3-d modeling of tumor microvasculature using superellipsoids," In ISBI, pages 185-188, IEEE 2006.

[19] V.Jain, J.F.Murray, F.Roth, S.Turaga, V.Zhiqulin, K.L. Briggman, M.N.Helmstaedter, W.Denk, H.S.Seung, "Supervised learning of image restoration with convolutional networks," ICCV 2007.

[20] L.Risser, F.Plouraboue, X.Descombes, "Gap filling in 3d vessel like patterns with tensor fields," in Proc. ICCV, 2007.

[21] S.Casadei, S.K.Mitter, "Hierarchical curve reconstruction. Part I: Bifurcation analysis and recovery of smooth curves," ECCV (1), pp.199-208, 1996.

[22] J.Canny, "A computational approach to edge detection," PAMI, 8(6):679-698, 1986.

[23] E.Sharon, A.Brandt, R.Basri, "Completion energies and scale," PAMI, 22(10):1117-1131, 2000.

[24] X.Ren, C.C.Fowlkes, J.Malik, "Learning probabilistic models for contour completion in natural images," IJCV, 2007.

[25] C.Rothwell, J.Mundy, B.Hoffman, V.-D.Nguyen, "Driving vision by topology," INRIA TR No. 2444, 1994.

[26] C.Poullis, S.You, U.Neumann, "A vision-based system for automatic detection and extraction of road networks," WACV, 2008.

[27] A.Szymczak, A.Tannenbaum, K.Mischaikow, "Coronary vessel cores from 3d imagery: a topological approach," SPIE Med.Imag., vol. 5747, pp.505-513, 2005.

[28] T.Pock, C.Janko, R.Beichel, H.Bischof, "Multiscale medialness for robust segmentation of 3D tubular structures," Proc. Comp.Vis.Winter Workshop, Austria, 2005.

[29] L.Risser, F.Plouraboue, X.Descombes, "Gap filling of 3-D microvascular networks by tensor voting," Trans.Med.Imag. 27(5):674-687, May 2008.

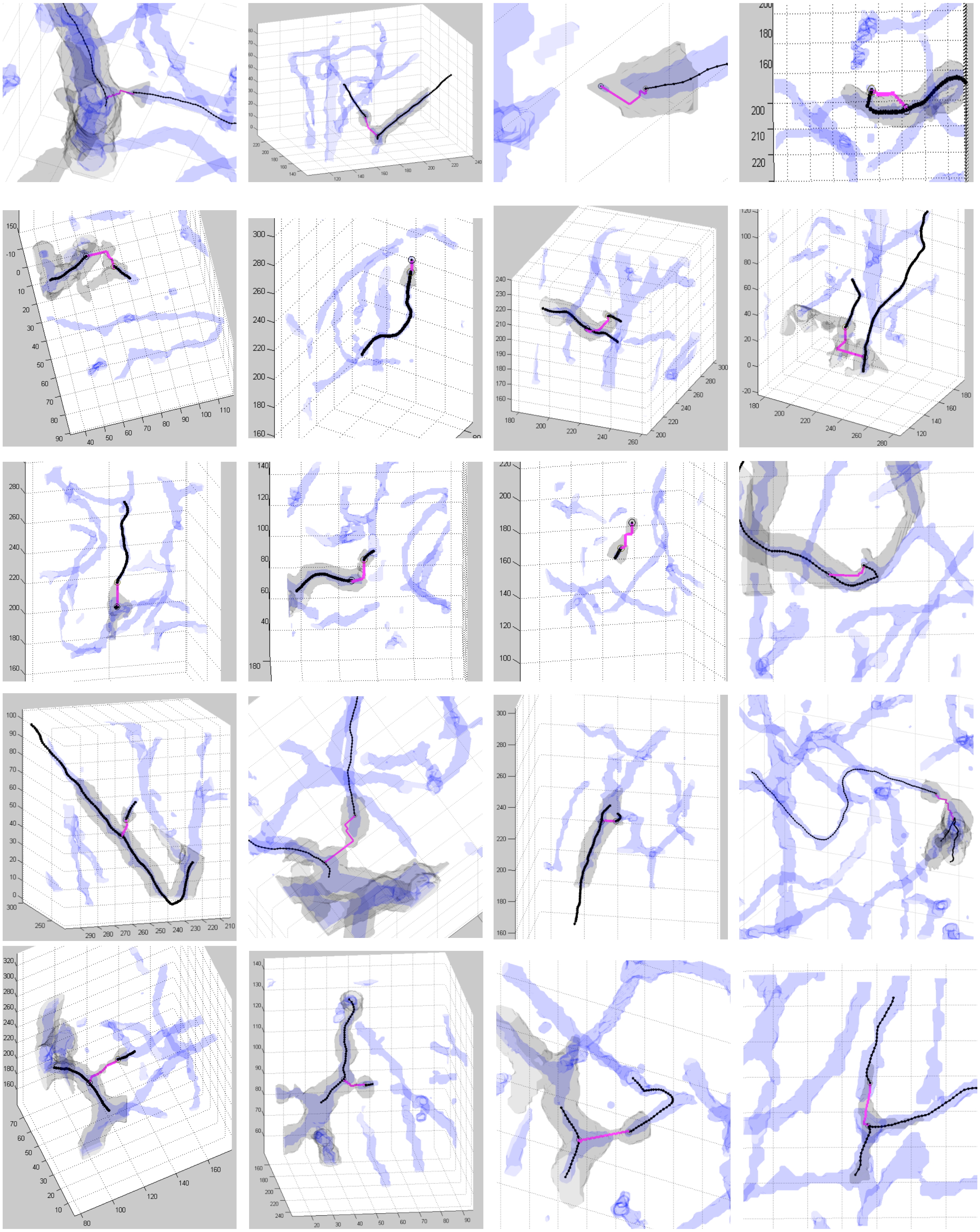

Figure 10: An illustration of the method on a small number of real gaps. Original mask is shown in blue, gap-filling mask in gray, bridging strand in magenta, and adjacent strands in black.