Dendrite Tracking in Microscopic Images using Minimum Spanning Trees and Localized E-M

François Fleuret and Pascal Fua
CVLAB – EPFL

March 27, 2006

Technical Report EPFL/CVLAB2006.02

Abstract

We describe in this document our preliminary results regarding the tracking of dendrites spreading from a neuron in confocal microscope images. When using a small number of image layers, we obtain good results by combining a EM-based local estimate of the probability that an image pixel belongs to a neuron filament with the global tree properties of the complete set of dendrites. The optimal tree is obtained with a modified minimum-spanning tree procedure. We will argue that this approach extends naturally to the complete data volume and should give even better results.

1 Introduction

Full reconstruction of neuron morphology is of fundamental interest for the analysis and understanding of their functioning. So far, most commercial products such as Neurolucida[1], Imaris[2] or Metamorph[3] provide with powerful interfaces to reconstruct dendritic trees but relies heavily on manual operations for initialization and re-initialization of the procedures able to follow neuron filaments along in the tissue.

In its most basic form, the problem consists of processing a stack of pictures produced by a confocal microscope, each of them showing a slice

[1] http://www.microbrightfield.com/prod-nl.htm
[2] http://www.bitplane.com/products/imaris/imaris_product.shtml
[3] http://www.moleculardevices.com/pages/software/metamorph.html

of the same piece of tissue at a different depth. It is transparent enough so that these pictures can be acquired by simply changing the focal plane. Since dye was injected into the cell to be reconstructed, it became opaque to light, while the rest of the tissue retains its original color. Finally, the neuron and its extension appear on the picture as dark filaments as shown inFig. 1(a).

To automatically delineate the dendrites in such images, the main difficulty comes from the multiple discontinuities in the filaments, which are due to inhomogeneities in their thickness. For example, the presence of synapses locally increases the diameter of a filament and results in series of dots such as those that are clearly visible in Fig. 2(a).

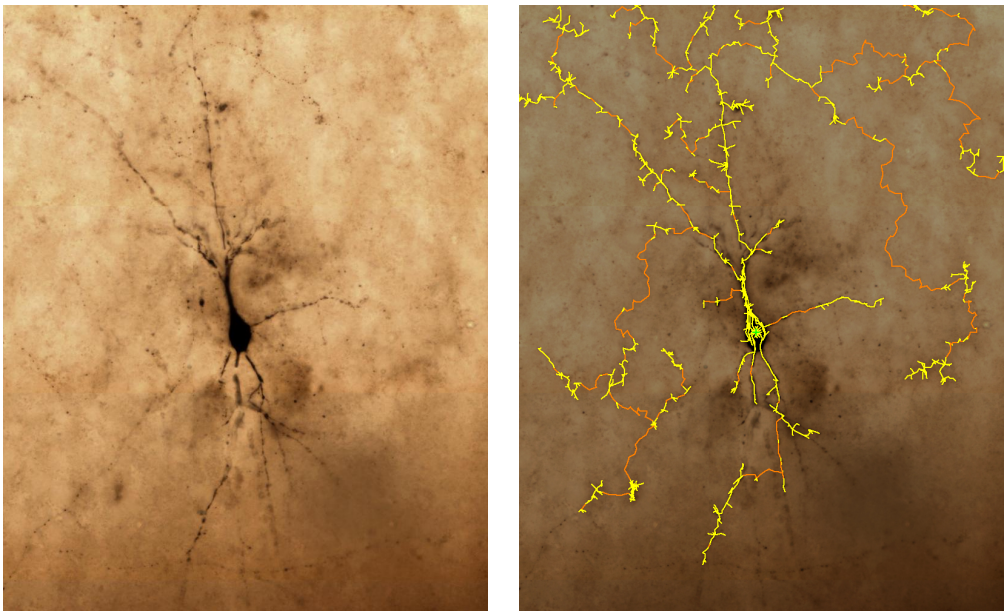
Thus, simply following the darkest paths in the picture cannot be expected to yield good results. A successful strategy has to involve both the local characterization of the filament-like parts of the pictures and more global modeling of dendritic networks.

In this report, we propose such a strategy, similar to (He et al. 2003), which combines local color-based estimate of the probability that an image pixel belongs to a filament with the global tree properties of the complete set of filaments. We avoid having to de-noise the image by keeping a probabilistic semantic in the result of the segmentation step. Our method produces results such as those of Fig. 1(b), which compare favorably to manual delineation. So far, we have only tested our approach in the 2–D case using a small number of layers. However, we expect our approach to naturally extend to the full data volume, where it should perform even better.

2 Related Work

Reconstruction of large networks of filaments is an important subject at the interface between medical imaging and computer vision. Its two main applications are the reconstruction of brain vascular networks (Kirbas & Quek 2003, Krissian et al. 2004) and neuron dendritic trees.

For clean high-resolution images, techniques can rely on the continuity of the structures to follow and take advantage of sophisticated 3D models of the filaments (Shen et al. 2001, Al-Kofahi et al. 2002). However, since we want to deal here with pictures of lower quality, where structures may appear to be discontinuous due to noise, we propose to use a technique based on the minimal spanning tree technique. This optimization scheme has been used in computer vision since the 80s, for instance for road-tracking in satellite images (Fischler et al. 1987). Our approach is



(a) (b)

Figure 1: Delineating the dendritic network in the 2–D image corresponding to a particular depth. (a) Original image. (b) Resulting fully automatic delineation (the original image has been darkened to make the delineation more visible)

globally similar to (He et al. 2003), but we avoid the deconvolution step and rely on a log-odd ratio computed during the pre-processing to score the candidate trees. This scoring scheme is consistent under a legitimate model of the image, and our probabilistic formulation gives flexibility for improving the models of the filament appearance or tree geometry while keeping the global optimization framework.

3 Approach

We formulate the problem as Bayesian inference. The hidden state we have to infer is the actual location of the dendritic tree in the tissue volume, and the visible state is the tissue color. The prior on the hidden state is that it is a tree as opposed to a graph, which means that there are no cycles or unconnected parts. The prior on the color, given the tree, is that pixels located on the filaments are significantly darker than pixels that are on the rest of the tissue.

Our algorithm involves the two following algorithmic steps:

Table 1: Notation used in Sections 3.1 and 3.2

$I(x, y)$
: Image intensity at location (x, y);

$Y(x, y)$
: Boolean random process standing for the presence of a filament at pixel (x, y);

$W(x, y)$
: Boolean random process standing for the visibility of filament (if present) at pixel (x, y);

$Z(x, y) = Y(x, y)\, W(y, x)$
: Boolean random process standing for the actual neuron-color coloration of pixel (x, y);

$\xi(x, y) = P(Z = 1 \mid I(x, y))$
: Probability that a pixel is on a visible part of a filament given its color;

$\delta = P(Y = 1)$ and $\epsilon = P(W = 1)$
: The prior probability of the filament presence and filament visibility.

1. Estimating the probability for every pixel to be on a filament, given only its intensity;
2. Building the optimal tree.

In the two following sections, we describe these individual steps.

3.1 Estimating the Probability of Belonging to a Filament

Our algorithm's first step is to estimate the probability for every pixel to belong to a filament given its color. As stated above, our basic assumption is that a filament pixel is, in general, darker than a non-filament one. Note, however, that this is a local property as opposed to a global one. For example, in the images we experimented with, some areas were globally brighter than others, presumably due to imaging artifacts. As a result,

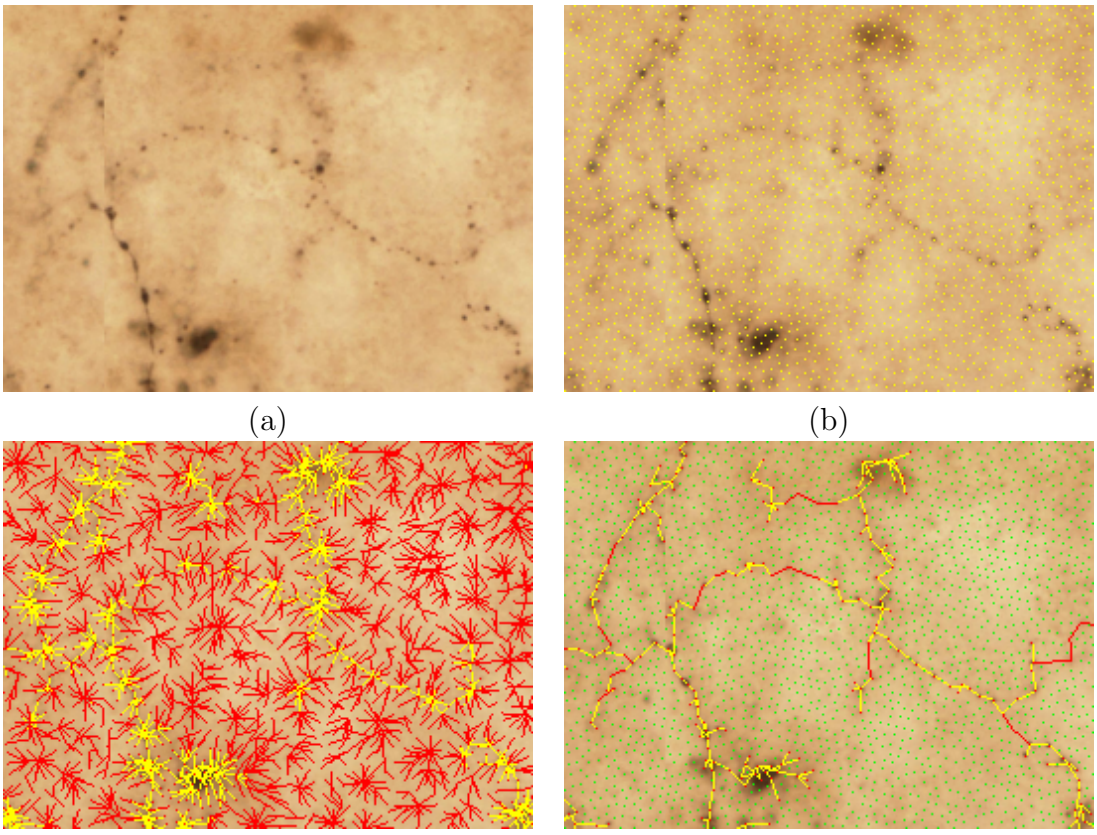
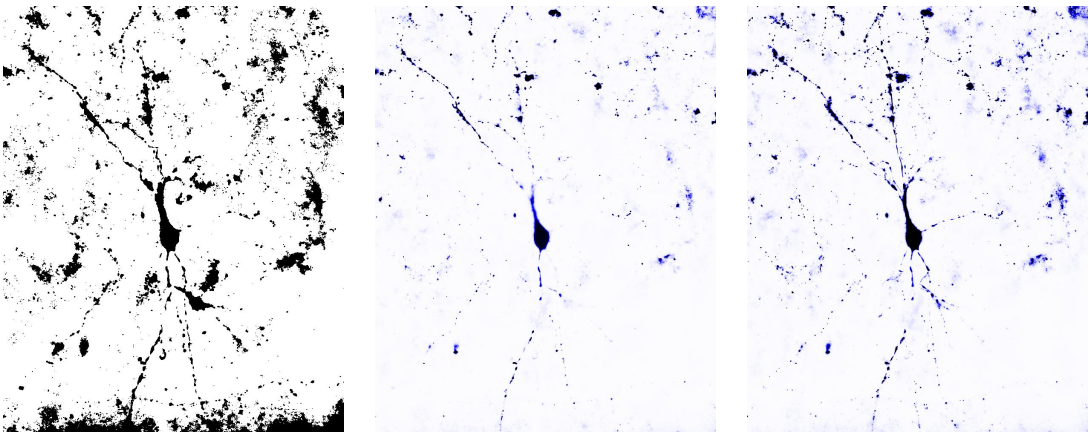


Figure 2: Building and pruning the spanning tree. (a) Detail of the image of Fig. 1 , (b) Anchor points after decimation, (c) Full tree before pruning and (d) Final tree after pruning. The red segments do not appear to be filaments but they are retained to connect subtrees.



(a) (b) (c)

Figure 3: Estimating the probability that a pixel belongs to a filament based on its color. (a) Initial estimate. (b) Refined estimate using a single layer. (c) Refined estimate using seven layers.

some pixel filaments in the bright image areas can actually be lighter than non-pixel filaments in dark areas.

Also, as shown in Fig. 2(a), very thin portions of a filament may not be visible, resulting in what appears to be set of unconnected fragments. We must therefore allow the tree to spread to light parts of the picture as necessary to connect those fragments. To model this situation we introduce a boolean variable $Y(x, y)$ which is equal to 1 on filaments and a variable $W(x, y)$ equal to 1 when the filament at that location is visible, assuming there is actually a filament. The product of the two $Z(x, y) = Y(x, y) * W(x, y)$ modulates the darkness of the pixels: pixels are darker when Z is equal to one. We therefore write the probability we want to estimate as $\xi(x, y) = P(Z = 1 \,|\, I(x, y))$. This model takes explicitly into account invisible parts of filaments and relaxes the penalty for the tree spanning such areas, instead of relying for instance on the variance of a Gaussian model to account for such anomaly.

To evaluate ξ, we make the assumption that in any sub-image of size $\Delta \times \Delta$ the intensity of pixels is a mixture of two Gaussian distributions, one standing for the intensity on the visible filaments, and one for the intensity on the rest of the tissue. As said earlier intensity on filaments or tissues varies greatly in different part of the image, however the assumption of being locally constant seems reasonable experimentally.

We thus have developed a variation of the classical E-M algorithm. We alternatively estimate the probability $\xi(x, y)$ for each pixel to be neuron-colored given an *a priori* conditional distribution of pixel intensities in its

neighborhood, and then re-estimate this conditional intensity distribution given which pixels are considered to belong to filaments and which are not.

Thus, this can be divided in three parts:

1. **Initialize ξ**
   This is done by setting this value to 1 where the intensity is more than one standard deviation from the mean, where both mean and standard deviation have been computed over the whole image;
2. **Update the local models**
   Compute for each pixel the conditional expectation and standard deviation of the intensity inside and outside neuron filaments in a neighborhood of size $\Delta \times \Delta$. The cost of this estimation can be made independent from Δ by using integral images;
3. **Update ξ**
   Re-estimate ξ according to a Bayesian rule and Gaussian models. Go back to 2. until convergence.

Fig. 3(a) shows the initialization values. Fig. 3(b) depicts the resulting refined probability image. To further improve the result, we can perform this computation on several layers of thin stack of images and take ξ to be the maximum value at each (x, y) location, which yields the result of Fig. 3(c).

3.2 Computing the Optimal Tree

Let T be the set of pixels on the actual filament tree. We show in Appendix A that the log probability of observing the specific image I we use as input, given T, can be evaluated as

$$\log P(I \,|\, T) = \Psi_0 \; + \sum_{(x,y) \in T} \Psi(x, y)$$

where

$$\Psi(x, y) = \log \frac{P(I(x, y) \,|\, Y(x, y) = 1)}{P(I(x, y) \,|\, Y(x, y) = 0)}$$

which can be computed from the estimate of $\xi(x, y) = P(Z(x, y) = 1 \,|\, I(x, y))$ and the priors $\delta = P(Y = 1)$ and $\epsilon = P(W = 1)$ (see Appendix B).

Thus finding the maximum likelihood tree T reduces to finding the tree t that maximizes

$$\sum_{(x,y)\in t} \Psi(x,y) \quad .$$

To this end, we have developed a greedy algorithm, which is a variation of the maximum spanning tree (Kruskal Jr. 1956, Prim 1957, Gower & Ross 1969, Zahn 1971). We first find a set of *anchor points*, which are local maxima of the $\xi(x,y)$ probability defined in Section 3.1. We then build the maximum likelihood tree that spans them.

1. **Subsampling the set of pixels**

   The purpose of this initial step is to reduce the cardinality of the set of pixels to process. We call the pixels remaining after that step *anchor points* and they are chosen to be sufficiently dense on the areas of interest so that the optimal tree can be defined as a tree spanning a subset of these points.

   To select anchor points, we build a ranked list of all the pixel of the image according to their Ψ value. We take the first element of the list to be an anchor and remove from the list all other points within a radius ρ. We then iterate this process until the list is empty, which yields results such as the one if Fig. 2(b).

   This process dramatically reduces the number of pixels to be considered, while retaining all pixels at locations likely to be on a filament.

2. **Building the Tree**

   We build the set of all edges of length less than 3ρ between anchors and rank them according to the *average value* of Ψ along the segment joining them. We then consider them sequentially and add to the tree any edge which does not create a cycle in the resulting graph.

   This yields trees such as the one of Fig. 2(c). Note that due to the 3ρ distance threshold, we may not be able to build a tree spanning all anchors. Non-connected anchors are considered as belonging to pieces of pieces of filaments unrelated to the considered neuron.

3. **Pruning optimally**

   Given the optimal full tree spanning all the anchor points, we can compute the global optimal subtree by removing all the anchors such that the integral of Ψ on the subtree it holds is negative (see figure 2 (c) and (d) and 4). This can be done by ranking the anchors according to their discrete distance to the starting point, measured by how many intermediate anchors between it and the starting point there are, and updating that integral score by considering them sequentially.

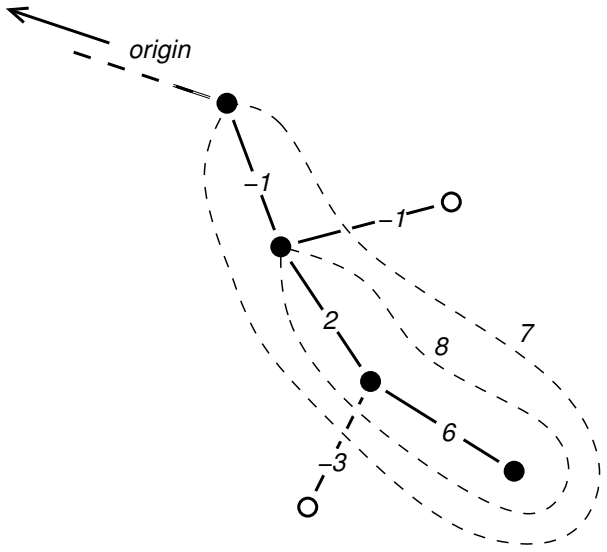


Figure 4: Sub-trees which negative score are pruned. Solid dots denote vertices that are retained and hollow dots vertices that are removed.

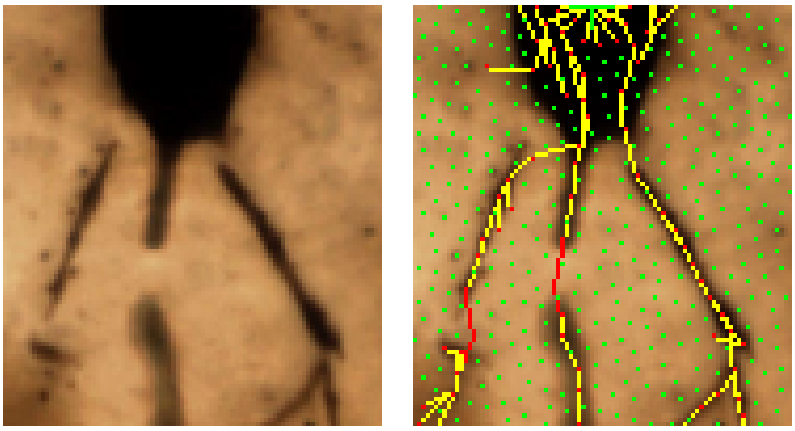
> Such an algorithm keeps an edge with a very negative scores when it connects a subtree with a high score to the origin. This property ensures a good behavior at gaps, as shown in Fig. 5.

4 Future Work

This short study shows that a Bayesian formulation combined with a classical tree-optimization method is effective to delineate dendritic branches in thin stacks of confocal microscope images.

All the results presented were obtained using either a single or a small number of image layers, which we refer to as the 2–D case. This gives rise to ambiguities that may evaporate if one considers the full data volume instead. This typically happens when a filament is above another. In a single layer, they may appear to intersect even though, in 3–D, they do not.

Fortunately, all the algorithms described above extend naturally to the 3–D case and should therefore work even better than in 2–D. The only serious difficulty we expect to encounter is the vastly increased computational cost, especially to handle the list of all possible edges of Section 2. Fortunately, this can be dealt with in a pure algorithmic way by using lazy algorithms and sparse representation, such as hash table or Kd-trees. Furthermore, should the need arise, the algorithm could be parallelized to



(a) (b)

Figure 5: Bridging gaps. (a) Image detail with substantial gaps between filaments. (b) The pruning algorithm may retain edges whose score is bad, such as those shown in red, if they connect high scoring subtrees.

take advantage of EPFL's BlueGene computer.

Also, the estimation of $P(Y(x) \mid I)$, as described in Section 3.1, is based on the individual pixel intensities, which is very crude. It does not take advantage of all the information that can be extracted from the intensity patterns around the pixels. For example, the neighborhood of a filament is cylindrical and therefore locally isotropic. We could clearly take this into account by using statistical learning applied to pattern recognition.

Given a few stacks of images that have been labeled by an expert, we will collect many image patches located on those filaments to build a training set. We will use it create a non-parametric model—decision tree, naive Bayesian, or SVM—to predict whether or not a new image patch of image is likely to be located on a neuron filament. Such an approach has the potential to account for shape and color regularities in small cubes of brain matter. This prediction will be easy to convert into a directly usable *a posteriori* probability.

5 Acknowledgement

We thank Professor Wulfram Gerstner for his support, and Sonia Garcia for her help and comments.

A Tree score

$$\begin{aligned}
\log P(I \,|\, T=t) &= \sum_{(x,y)} \log P(I(x,y) \,|\, T=t) \\
&= \sum_{(x,y)} \log P(I(x,y) \,|\, Y(x) = 1_{(x,y)\in t}) \\
&= \sum_{(x,y)} \log P(I(x,y) \,|\, Y(x)=0) + \sum_{(x,y)\,\in\, t} \underbrace{\log \frac{P(I(x,y) \,|\, Y(x,y)=1)}{P(I(x,y) \,|\, Y(x,y)=0)}}_{\Psi(x,y)} \\
&= \Psi_0 + \sum_{(x,y)\in t} \Psi(x,y)
\end{aligned}$$

B Value of Ψ

$$\begin{aligned}
\Psi(x,y) &= \frac{P(I \,|\, Y=1)}{P(I \,|\, Y=0)} \\
&= \frac{P(I,\, W=0 \,|\, Y=1) + P(I,\, W=1 \,|\, Y=1)}{P(I,\, W=0 \,|\, Y=0) + P(I,\, W=1 \,|\, Y=0)} \\
&= \frac{P(I \,|\, W=0,\, Y=1)\, P(W=0 \,|\, Y=1) + P(I \,|\, W=1,\, Y=1)\, P(W=1 \,|\, Y=1)}{P(I \,|\, W=0,\, Y=0)\, P(W=0 \,|\, Y=0) + P(I \,|\, W=1,\, Y=0)\, P(W=1 \,|\, Y=0)} \\
&= \frac{P(I \,|\, W=0,\, Y=1)\, P(W=0) + P(I \,|\, W=1,\, Y=1)\, P(W=1)}{P(I \,|\, W=0,\, Y=0)\, P(W=0) + P(I \,|\, W=1,\, Y=0)\, P(W=1)} \\
&= \frac{P(I \,|\, YW=0)\, P(W=0) + P(I \,|\, YW=1)\, P(W=1)}{P(I \,|\, YW=0)\, P(W=0) + P(I \,|\, YW=0)\, P(W=1)} \\
&= \frac{P(I \,|\, YW=0)\, P(W=0) + P(I \,|\, YW=1)\, P(W=1)}{P(I \,|\, YW=0)} \\
&= P(W=0) + \frac{P(I \,|\, YW=1)}{P(I \,|\, YW=0)}\, P(W=1) \\
&= P(W=0) + \frac{P(YW=1 \,|\, I) P(YW=0)}{P(YW=0 \,|\, I) P(YW=1)}\, P(W=1) \\
&= P(W=0) + \frac{\xi}{1-\xi}\, \frac{1-P(YW=1)}{P(YW=1)}\, P(W=1) \\
&= (1-\epsilon) + \frac{\xi}{1-\xi}\, \frac{1-\delta\epsilon}{\delta\epsilon}\, \epsilon \\
&= (1-\epsilon) + \frac{\xi}{1-\xi} \left(\frac{1}{\delta} - \epsilon \right)
\end{aligned}$$

References

Al-Kofahi, K. A., Lasek, S., Szarowski, D. H., Pace, C. J., Nagy, G., Turner, J. N. & B., R. (2002), 'Rapid automated three-dimensional tracing of neurons from confocal image stacks', *IEEE Transactions on Information Technology in Biomedicine* **6**(2), 171–187.

Fischler, M., Tenenbaum, J. & Wolf, H. (1987), 'Detection of Roads and Linear Structures in Low-resolution Aerial Imagery Using a Multisource Knowledge Integration Technique', *Computer Vision, Graphics, and Image Processing* **15**(3), 201–223.

Gower, J. & Ross, G. (1969), 'Minimum spanning trees and single linkage cluster analysis', *Appl. Stat.* **18**, 54–64.

He, W., Hamilton, T. A., Cohen, A. R., Holmes, T. J., Pace, C., Szarowski, D. H., N, J. & and, T. (2003), 'Automated three-dimensional tracing of neurons in confocal and brightfield images', *Microscopy and Microanalysis* **9**(4), 296–310.

Kirbas, C. & Quek, F. (2003), Vessel extraction techniques and algorithms: A survey, *in* 'Proceedings of the Third IEEE Symposium on BioInformatics and BioEngineering (BIBE'03)', p. 238.

Krissian, K., Kikinis, R. & Westin, C.-F. (2004), Algorithms for extracting vessel centerlines, Technical Report 0003, Department of Radiology, Brigham and Women's Hospital, Harvard Medical School, Laboratory of Mathematics in Imaging.

Kruskal Jr., J. (1956), On the shortest spanning subtree of a graph and the travelling salesman problem, *in* 'Proc. American Math. Soc.', Vol. 7, pp. 48–50.

Prim, R. (1957), 'Shortest connection networks and some generalizations', *Bell Syst. Tech. J.* **36**, 1389–1401.

Shen, H., Roysam, B., Stewart, C. V., Turner, J. N. & Tanenbaum, H. L. (2001), 'Optimal scheduling of tracing computations for real-time vascular landmark extraction from retinal fundus images', *IEEE Transactions on Information Technology in Biomedicine* **5**(1), 77–91.

Zahn, C. (1971), 'Graph-theoretical methods for detecting and describing gestalt clusters', *IEEE Trans. Comp.* **C20**, 68–86.