Published in final edited form as:
Proc SPIE. 2012 February 23; 8314: . doi:10.1117/12.912237.

Automated Reconstruction of Neural Trees Using Front Re-initialization

Amit Mukherjee and **Armen Stepanyants**
Department of Physics and Center for Interdisciplinary Research on Complex Systems, Northeastern University, Boston MA 02115

Abstract

This paper proposes a greedy algorithm for automated reconstruction of neural arbors from light microscopy stacks of images. The algorithm is based on the minimum cost path method. While the minimum cost path, obtained using the Fast Marching Method, results in a trace with the least cumulative cost between the start and the end points, it is not sufficient for the reconstruction of neural trees. This is because sections of the minimum cost path can erroneously travel through the image background with undetectable detriment to the cumulative cost. To circumvent this problem we propose an algorithm that grows a neural tree from a specified root by iteratively re-initializing the Fast Marching fronts. The speed image used in the Fast Marching Method is generated by computing the average outward flux of the gradient vector flow field. Each iteration of the algorithm produces a candidate extension by allowing the front to travel a specified distance and then tracking from the farthest point of the front back to the tree. Robust likelihood ratio test is used to evaluate the quality of the candidate extension by comparing voxel intensities along the extension to those in the foreground and the background. The qualified extensions are appended to the current tree, the front is re-initialized, and Fast Marching is continued until the stopping criterion is met. To evaluate the performance of the algorithm we reconstructed 6 stacks of two-photon microscopy images and compared the results to the ground truth reconstructions by using the DIADEM metric. The average comparison score was 0.82 out of 1.0, which is on par with the performance achieved by expert manual tracers.

Keywords

Medial axis; Neuron Tree; Fast Marching Method; Eikonal Equation

1. INTRODUCTION

One of the greatest challenges in the field of neuroscience is to uncover the relationship between the structure of the neural circuits and the function of the nervous system [1]. This knowledge at the microscopic as well as the systems level is needed for the quantitative understanding of information processing in the brain, learning, and memory formation. To achieve this goal, medical imaging must be combined with quantitative analyses of neuron morphologies [2-5].

Automated tracing algorithms are used to extract digital reconstructions of axonal and dendritic arbors of labeled neurons from light microscopy stacks of images. A trace is a compact linear representation of neural trees embedded in the image, often used for statistical analyses of neural shapes and simulations of neural activity. Over the past three decades, researchers have addressed the tracing problem using diverse image processing techniques. These techniques can be broadly classified into three categories: segmentation based methods, sequential tracking methods, and minimum cost path methods [6]. In this

study we describe a method which belongs to the latter category. In general, methods based on the minimum cost path technique use a speed image, computed from the image intensity values, to find the trace with the smallest cumulative cost. Cohen et al. [7] have shown that the global minimum cost path can be obtained by solving the Eikonal equation with the Fast Marching Method. Overall, the minimum cost path methods are advantageous compared to the segmentation and sequential tracking approaches because they utilize both local image intensity and neural shape related information. This is accomplished by accumulating voxel level information in the form of cumulative costs of all paths and then finding the path with the minimum cost. While the accumulation of information makes this approach robust to noise and local intensity variations, it may also make the algorithm blind to small "jumps" through the background as shown in the phantom demonstration in Figure 1. This paper proposes a novel technique to overcome this limitation by re-initializing the front after it has traveled a predetermined distance. Front re-initialization offers the following advantages: a) it makes it possible to monitor the quality of small sections of the trace, b) it speeds up computations and reduces memory requirements, and c) it allows propagation of higher level curve geometry information, similar to tracking based methods.

2. METHODS

We propose a method for detecting the traces of neural trees contained in stacks of images. Given an image I ($R^3 \rightarrow R$), a trace $\mathbf{x_0}$ ($\mathbf{x_0} \in R^3$) is defined as a set of connected locations representing the centerline, or the medial axis, of the neural tree. Our method consists of two steps. First, the image stack is processed to generate a medialness measure of the neurites. Next, this medialness measure is used as the speed image for front propagation and minimum cost path computation to obtain the trace. These steps are described in detail below.

2.1 Computation of the speed image

We use a modified version of the Gradient Vector Flow (GVF) method proposed by Xu and Prince [8] to obtain a regularized vector field **v** from image *I*. The GVF essentially extends a virtual potential field induced by the edge map, $f = \|\nabla I\|$, into the interior regions of the image. The vector field **v** is computed as the equilibrium solution of the following vector diffusion equation on the edge map:

$$\begin{aligned} &\frac{\partial \mathbf{v}(t)}{\partial t} = H\left(\frac{d\|\mathbf{v}(t)\|}{dt}\right)\left[g\nabla^2\mathbf{v}(t) - f\|\nabla f\|\left(\mathbf{v}(t) - \nabla f\right)\right]. \\ &\mathbf{v}(0) = \nabla f \end{aligned} \quad (1)$$

In this equation ∇^2 is the Laplacian operator and g is the diffusion constant. Eq. (1) is similar to the original GVF equation used in [8] except for an additional Heaviside function which ensures that the gradient vectors do not decrease in magnitude as the diffusion progresses, preventing the erosion of faint edges. In solving Eq. (1) numerically we start with the initial condition and update **v** only at locations **x** in the image where the update leads to an increase in the magnitude of **v**. An example of the GVF field is shown in Figure 2a. As can be seen from this figure, locations where the vector field lines meet trace out the medial axes of the neurite.

Next, the average outward flux $\varphi(\mathbf{x})$ is calculated for every **x** by taking the inner product of the vector field with the radial vector $(\mathbf{x}_i - \mathbf{x})$ [9]:

$$\varphi(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{n}\frac{\mathbf{v}(\mathbf{x}_i)^{\mathbf{T}}(\mathbf{x}_i - \mathbf{x})}{\|\mathbf{v}(\mathbf{x}_i)\|\|\mathbf{x}_i - \mathbf{x}\|}. \quad (2)$$

Vectors $\mathbf{x}_i$ in this expression denote the positions of voxels in the n-neighborhood of $\mathbf{x}$ (n = 8 in 2D and n = 26 in 3D).

The speed image is then generated in such a way that locations in the image where both the intensity and the average outward flux are high are assigned high speed values:

$$F(\mathbf{x}) = \exp(\gamma [1 - \varphi(\mathbf{x})] I(\mathbf{x})) - 1. \quad (3)$$

Defined this way, the speed image, F, is guaranteed to have positive intensity values, with high intensities near the medial axes of the neurites and low intensities everywhere else (Figure 2b). Adjustable parameter γ in Eq. (3) was introduced to enhance the contrast in speeds between voxels with inward and outward GVF fluxes. This parameter should generally remain in the 1-2 range and γ = 1.5 was used throughout this study.

2.2 Computing candidate extensions

The next step of the algorithm uses the Fast Marching Method to compute of the minimum cost paths in the speed image. The Fast Marching Method solves a boundary value problem,

$$\begin{aligned} F(\mathbf{x}) |\nabla T(\mathbf{x})| &= 1 \\ T(\mathbf{x}_0) &= 0, \end{aligned} \quad (4)$$

by approximating the viscosity solution of the Eikonal equation [10]. The arrival time map, $T(\mathbf{x})$, represents the time it takes the front, originating from the initial location $\mathbf{x}_0$ and propagating with the speed specified by F, to arrive at the location $\mathbf{x}$. Because the front propagation speed is highest in the vicinity of the medial axes of neurites (Figure 2b) the paths corresponding to the shortest arrival times (minimum cost) reveal the centerlines of structures contained in the image [7]. In practice however, erroneous paths me be created due to the proximity of neural branches and the presence of noise. In the former case, the path may travel through the background, erroneously connecting adjacent branches, and in the latter case, the path may extend into the background region following chains of noisy voxels. We address these problems by interrupting and re-initializing front propagation at regular intervals.

The proposed front re-initialization is an iterative scheme for updating the boundary condition of the Fast Marching method, $T(\mathbf{x}_0) = 0$. At each iteration, we allow the fastest advancing tip of the front to travel a geometric distance r. Then the best possible extension to the existing neural tree is determined by performing the gradient descent on $T(\mathbf{x})$, starting at the tip of the front and ending on $\mathbf{x}_0$. The gradient descent gives a set of locations on the shortest arrival time path, which we refer to as the candidate extension and denote it by $\mathbf{x}_1$. Due to the monotonic update of $T(\mathbf{x})$ in the Fast Marching algorithm, $\mathbf{x}_1$ is guaranteed to be the best possible extension to the neural tree. After the candidate extension $\mathbf{x_1}$ is obtained it is subjected to the likelihood ratio test. If the likelihood ratio test qualifies $\mathbf{x}_1$ for re-initialization, we append $\mathbf{x}_1$ to the neural tree, $\mathbf{x_0} = \mathbf{x_0} \cup \mathbf{x_1}$, and assign $T(\mathbf{x}_1) = 0$. It can be easily shown that the additional boundary condition due to the inclusion of the candidate extension $\mathbf{x_1}$ does not increase the value of $T(\mathbf{x})$ anywhere in R^3. Specifically, the value of $T(\mathbf{x})$ reduces in the vicinity of $\mathbf{x}_1$, while it remains the same everywhere else. This property can be used to make the front propagation computationally efficient. After a re-initialization, the new front originating from $\mathbf{x}_1$ is allowed to propagate until it merges with the existing arrival time map, at which point front propagation for $\mathbf{x_0}$ is continued. This initial restriction of the update of $T(\mathbf{x})$ to the locations near $\mathbf{x}_1$ increases the efficiency of the algorithm.

Figure 3 shows an illustration where a front is initialized at one point and the geometric distance traveled by the fastest advancing tip of the front is monitored. When the tip of the

front reaches a specified distance r (from the start point) front propagation is interrupted and the minimum cost path is obtained by tracing back from the front tip to the start point. This minimum cost path forms a candidate extension to the existing tree. It is appended to the tree only after qualifying in the robust likelihood ratio test as described below.

2.3 Robust likelihood ratio test

Robust hypothesis testing is a general methodology for choosing the most plausible explanation of an event from two or more alternative explanations. Here, this test is performed to determine if the candidate extension is likely to correspond to a neurite structure. This is accomplished by comparing the likelihoods that the voxels on the candidate extension $\mathbf{x}_1$ belong to the background or the foreground intensity distributions [11]. The intensity distribution of the foreground region is computed by using voxels on $\mathbf{x}_0$, and the intensity distribution of the background is estimated based on voxels with intensities less than the mean intensity of the image. Candidate extensions which do not pass the robust likelihood ratio test are eliminated (Figure 4).

Pseudo-code for the algorithm

Input : F, x_{root}
Output : Neuron tree originating at x_{root}
Initialize : $T(\mathbf{x}) \leftarrow \infty \; \forall \; \mathbf{x} \neq \mathbf{x}_{root}$
$T(\mathbf{x}_{root}) \leftarrow 0$
$Q \leftarrow x_{root}$
while *size* $(Q) > 0$ **do**
  $\mathbf{x} \leftarrow \mathbf{argmin}_{x \in Q} T(\mathbf{x})$
  if $D(\mathbf{x}) > r$ **then**
    $m \leftarrow$ TraceBackPath $(\mathbf{x})$
    if m Qualifies LRT **then**
      Reinitialize (m,T)
    endif
  endif
  Relaxation $(\mathbf{x}, Q)$
end while

3. RESULTS, CONCLUSIONS, AND FUTURE WORK

The algorithm proposed in this study produces fast and reliable reconstructions of neural arbors. It is robust with respect to imaging noise and variations in image intensity arising from imperfect labeling. The algorithm was tested on one of the DIADEM datasets [12] (6 image stacks), for which the ground truth traces were obtained by expert manual tracers. The results are presented in Table 1, where the DIADEM metric was used to measure the fidelity of the automated reconstructions to the corresponding ground truth traces. The average comparison score was 0.82 ± 0.07 (mean ± SD) which is not significantly different from the average score of 0.78 ± 0.1 ($n = 8$) achieved by expert manual tracers [12]. Figure 5 illustrates the algorithmic steps leading to the reconstruction of one of the image stacks.

In our future work we would like to incorporate higher level geometrical information in the likelihood ratio test. This information may include branch thickness, tortuosity, branching pattern, presence of dendritic spines and axonal boutons. We are also planning to test the

automated reconstruction algorithm presented in this study on a variety of images covering a wider range of neuron classes, labeling techniques, and imaging modalities.

Acknowledgments

This work was supported by the NIH grants NS063494.

REFERENCES

1. Ramón Y, Cajal S. Sur la structure de l'écorce cérébrale de quelques mammifères. La Cellule. 1891; 7
2. Ascoli, G. Computational neuroanatomy: Principles and methods. Humana Press Inc; 2002.
3. Bjaalie J. Understanding the brain through neuroinformatics. Frontiers in neuroscience. 2008; 2(1): 19. [PubMed: 18982101]
4. Stepanyants A, Chklovskii D. Neurogeometry and potential synaptic connectivity. Trends in neurosciences. 2005; 28(7):387–394. [PubMed: 15935485]
5. Stepanyants A, Hof P, Chklovskii D. Geometry and structural plasticity of synaptic connectivity. Neuron. 2002; 34(2):275–288. [PubMed: 11970869]
6. Meijering E. Neuron tracing in perspective. Cytometry Part A. 2010; 77(7):693–704.
7. Cohen L, Kimmel R. Global minimum for active contour models: A minimal path approach. International Journal of Computer Vision. 1997; 24(1):57–78.
8. Xu C, Prince J. Snakes, shapes, and gradient vector flow. Image Processing, IEEE Transactions on. 1998; 7(3):359–369.
9. Vasilevsky A, Siddiqi K. Flux maximizing geometric flows. IEEE Trans. Pat. Anal. Mach. Intel. 2002; 24(12):1565–1578.
10. Sethian J, et al. Level set methods and fast marching methods. Journal of Computing and Information Technology. 2003; 11(1):1–2.
11. Poor HV. An introduction to signal detection and estimation. Springer. 1994:5–28.
12. Gillette TA, Brown KM, Ascoli GA. The DIADEM metric: comparing multiple reconstructions of the same neuron. Neuroinformatics. 2011; 9(2-3):233–245. [PubMed: 21519813]

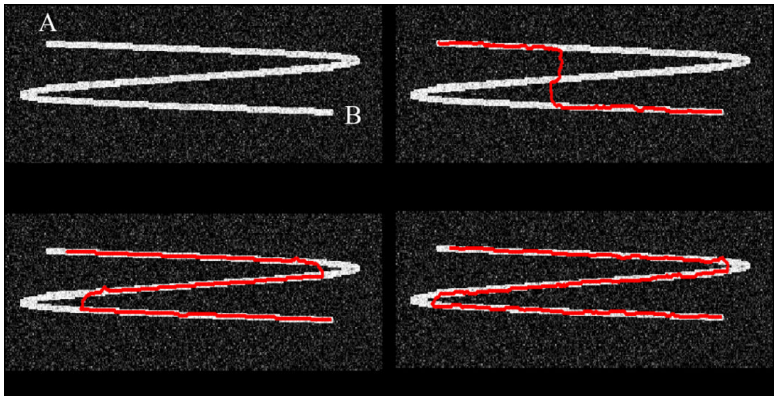

Figure 1.
Front re-initialization. In a phantom image (a) the minimum cumulative cost path between the terminal points A and B takes a short-cut by travelling through the background (b). By iteratively re-initializing the front after it has travelled a certain distance, the local cost of the path is minimized at each iteration, reducing the travel through the background. Front reinitialization is done after a travel distance of 30 pixels in (c) and 15 pixels in (d).

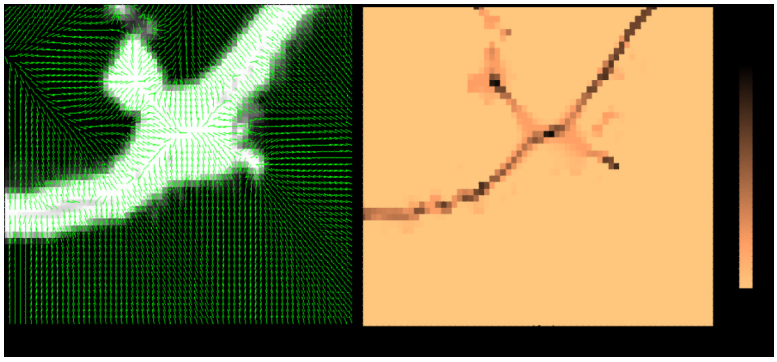

Figure 2.
The GVF field and the speed image. (a) An image containing a section of a neuron with complex boundary morphology. The field induced by the GVF is overlaid in green. (b) Average outward flux of the GVF field is used to compute the speed image.

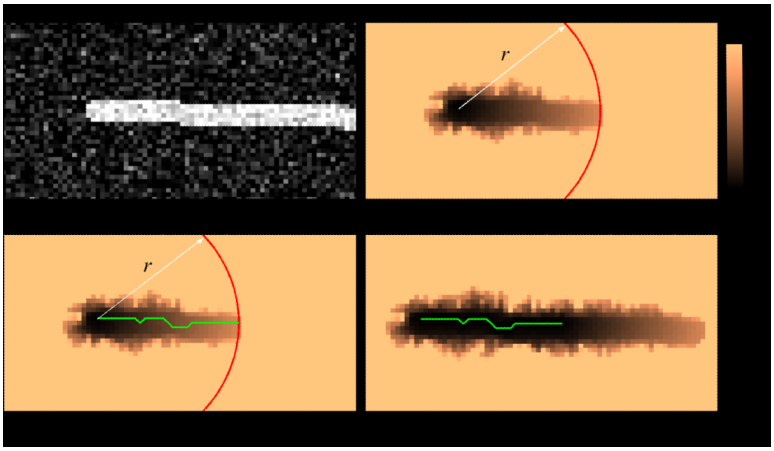

Figure 3.
Computation of candidate extensions. (a) Speed image. Brighter pixels correspond to higher speeds. (b) Arrival-time map of the front initiated at the root and travelling until a distance r is reached. (c) Trace-back path (green line) that minimizes the cumulative time of travel is obtained by gradient descent on the arrival-time map. (d) The arrival-time on the trace-back path is re-initialized to 0 and the front propagation is continued.

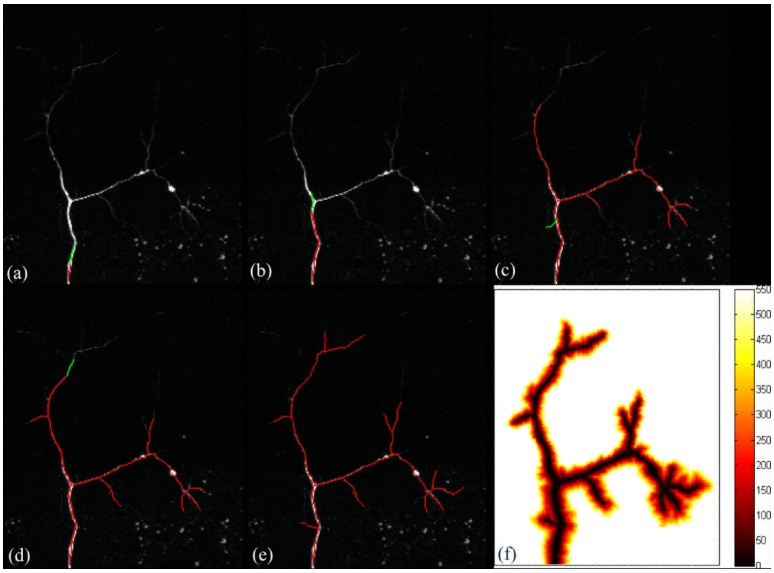

Figure 4.
Progression of the algorithm. (a - e) Stages of tree evolution. Reconstructed tree is shown in red and the candidate extensions are in green. (c) Example of an extension that failed the robust likelihood ratio test and is discarded in (d). (f) Arrival-time map obtained from the solution of the Eikonal equation.

Figure 5.
Automated reconstruction process. (a) 3D image stack of a neuron used in the DIADEM challenge. (b) Corresponding speed image, *F*. (c) Automated reconstruction of the neuron.

Table 1

Evaluation of the performance of the proposed algorithm using the DIADEM metric [11].

	DIADEM-metric score (Max score 1.0)
Image Stack 1	0.924
Image Stack 2	0.756
Image Stack 3	0.817
Image Stack 4	0.836
Image Stack 5	0.750
Image Stack 6	0.865