

IMPROVED AUTOMATIC CENTERLINE TRACING FOR DENDRITIC STRUCTURES

David Jiménez^{1,2}, Manos Papadakis^{1,2}, Demetrio Labate¹, Ioannis A. Kakadiaris²

¹Department of Mathematics, University of Houston

²Computational Biomedicine Lab, University of Houston

ABSTRACT

Centerline tracing of dendritic structures in confocal images of neurons is an essential tool for the construction of geometrical representation of a neuronal network from its coarse scale up to its fine scale structures. In this paper, we propose a novel algorithm (ORION 2) for centerline extraction that is both highly accurate and computationally efficient. The main novelty of the proposed method is the use of a small set of Multiscale Isotropic Laplacian filters for a fast and efficient binarization of the dendritic structure. The performance of this algorithm, which is validated on data from the DIADEM set, is shown to be very competitive against other state-of-the-art algorithms.

1. INTRODUCTION AND PREVIOUS WORK

The recent advances in confocal and multiphoton microscopy have opened up tremendous opportunities to study fundamental changes in neuronal morphology which, in the brain, are typically associated with learning and memory formation. To this goal, it is essential to develop accurate and efficient tools to segment neuronal structures and quantify their associated morphological characteristics. Deducing the graph connectivity for the centerline of a dendrite and using this graph as the backbone for representing the geometry of the tubular structure of a dendrite is essential for building a geometrical representation of a neuron. Although the problem of extracting this neuronal mapping still requires significant human intervention, methods for the extraction of the graph connectivity of dendrites have been extensively studied, including methods for tracing dendritic centerlines and identifying their branching and terminal points.

A commonly used strategy to determine the centerline in binarized images is *skeletonization* (e.g., [10]), where the binary volume undergoes an iterative thinning process. However, skeletonization is not particularly robust to noise. Other methods for centerline extraction require the selection of a *point source* (also called, a *seed point*), that may be automatically detected [8], randomly selected [2], or user-provided [7]. This point source initializes a process of wave propagation [4] or a fast marching algorithm [7] which proceeds to draw the centerline until it reaches some *terminal points*. While this approach is computationally efficient, its

main drawback is that its performance tends to be quite sensitive to the choice of the point source, leading to the risk of missing complete branches of the segmented volume.

In this paper, we propose a novel algorithm, called ORION 2, for the semi-automatic centerline extraction from 3D volumes, that is very competitive in terms of accuracy and computational efficiency and significantly reduces the need for training. This new efficient algorithm extracts the graph connectivity of a neuronal network. The main novelty of the proposed method is the use of a small set of Multiscale Isotropic Laplacian filters for a quick and efficient binarization of the dendritic structure combined with the application of a simple 3D finite-length filter which *automatically* detects the seed points for the centerline extraction. Although the strategy of our approach bares some similarity with the strategy recently proposed by Xie *et al.* [9], our method is different. The first step of the algorithm by Xie *et al.* segments the solid by applying a global threshold. This thresholding strategy is commonly used due to speed and simplicity. Nevertheless, this binarization method is not robust with respect to segmentation errors [6]. In contrast, as it will be explained below, our approach provides a more reliable and accurate binarization. The proposed seeding and tracing algorithms have a simpler mathematical justification, a much more streamlined implementation, and avoid the forward and backward propagation algorithms used in ORION [5, 7] and FARSIGHT [8], reducing computational cost.

2. METHODS

2.1. Training of the Classifier

The training of the classifier consists of three stages: feature selection, sample selection and parameter optimization. For computational efficiency, the features to consider are generated by a set of filters F_1, \dots, F_N , combining isotropic low pass band filters $\widehat{F}_i^{(LP)}(\xi) = P_n(C_{n,k}\|\xi\|^2) e^{-C_{n,k}\|\xi\|^2}$, to capture the low scale detail, high pass band filters $\widehat{F}_i^{(HP)}(\xi) = P_n(C_{n,k_1}\|\xi\|^2) e^{-C_{n,k_1}\|\xi\|^2} - P_n(C_{n,k_2}\|\xi\|^2) e^{-C_{n,k_2}\|\xi\|^2}$ to capture higher scale detail, and Laplacian filters $\widehat{F}_i^{(L)}(\xi) = \|\xi\|^2 P_n(C_{n,k}\|\xi\|^2) e^{-C_{n,k}\|\xi\|^2}$ that act as second order

derivatives at the direction of the gradient, which is in principle perpendicular to the centerline. Here, P_n is the Taylor polynomial of degree n associated with the exponential function e^x , and $C_{n,k} = (2n + 1)/(\pi\sigma_k)^2$. The volume $F_i * I$ is computed, where I is the input volume. A normalizing constant α_i is defined such that $\max_{\mathbf{v} \in I} \{\alpha_i |(F_i * I)\mathbf{v}|\} = 1$. For each voxel \mathbf{v} in I , we define $(\alpha_1(F_1 * I)\mathbf{v}, \dots, \alpha_N(F_N * I)\mathbf{v})$ to be the feature vector corresponding to \mathbf{v} .

The output of a confocal or multiphoton microscopy scan of a neuron can be approximated by a finite sum of the form

$$I = \sum_{i=1}^n \sum_{k=1}^K \sum_{j_1=1}^{J_1} \sum_{j_2=1}^{J_2} T_{x_i} R_k a_{i,k,j_1,j_2} f_{\sigma_{j_1}, \sigma_{j_2}}$$

where $f_{\sigma_1, \sigma_2}(x, y, z) = e^{-\frac{x^2}{2\sigma_1^2}} e^{-\frac{y^2+z^2}{2\sigma_2^2}}$, $x, y, z, \in \mathbb{R}$, $R_k \in SO(3)$, $T_{x_i}g(x) = g(x - x_i)$, and $a_{i,k,j_1,j_2} > 0$. According to this model, the surface of the solid is identified by the inflexion points of the sum. Thus, theoretically, we expect that the Laplacian of I would be negative in the interior of the structure and positive in the exterior.

Noise present in real volumes may introduce spike-values in its Laplacian. These spikes have high frequency content, and therefore such noise is mitigated by the application of a band-pass filter. Hence, voxels corresponding to larger positive or negative values have a higher likelihood of being background or solid, respectively.

A human operator chooses regions R_i inside the solid to be used for training, and parameters p_i and n_i , that correspond to the percentage of the total number of positive and negative values that will be considered as samples (typically 0.5% – 2%). The operator can visually inspect the sample sets, and either approve them or adjust the parameters p_i and n_i . The operator’s goal is to generate sample sets as representative as possible. Once sample sets and respective feature vectors are obtained, a grid search for the SVM parameters is performed via a four-fold validation process. Upon completion of the search, the classifier is generated. This SVM training procedure was inspired and improved from ORION [5, 7].

2.2. Binarization

Often, in a neuronal volume, it is expected that the number of object voxels will be considerably smaller than the number of the background voxels, with the latter having lower intensity values. If we compute μ , the average intensity of the volume I , voxels with intensity values lower than μ can safely be considered to belong to the background. This preliminary crude segmentation step reduces the computational overhead of the binarization. To segment the remaining voxels we use the SVM classifier previously trained utilizing the proposed features.

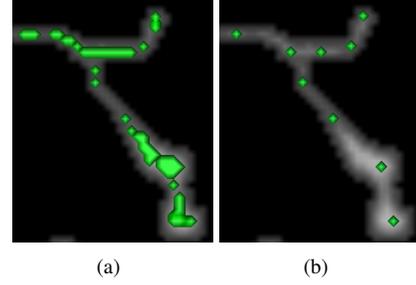


Fig. 1. Process of seed generation. (a) Depiction of seed candidates obtained from the filtering process, prior to decimation. (b) seeds remaining after decimation.

2.3. Selection of Seed Points

The selection of seed points from the binary volume is performed in three steps: filter seeding, seed decimation, and compensatory seeding.

Consider B to be the set of voxels in the segmented solid after the completion of the binarization. Each voxel \mathbf{v} is identified by its integer reference coordinates (x, y, z) . We define $d(\mathbf{v}) = \min\{\|\mathbf{v} - \mathbf{p}\| : \mathbf{p} \notin B\}$, where $\|\cdot\|$ is the Euclidean norm.

We compute the distance transform of the structure through the use of an efficient off-the-shelf algorithm. The resulting distance volume is convolved with T , which is a $3 \times 3 \times 3$ filter with $T_{0,0,0} = 1$, and $T_{i,j,k} = -1/26$ for all $(i, j, k) \neq (0, 0, 0)$. Let $N(\mathbf{v})$ be the set of immediate neighbors of \mathbf{v} . Note that $(T * d)(\mathbf{v}) = d(\mathbf{v}) - \frac{1}{26} \sum_{\mathbf{w} \in N(\mathbf{v})} d(\mathbf{w})$ is the difference between the distance of \mathbf{v} and the average of distance all neighboring voxels. The solid obtained is normalized by dividing by the maximum of its values, and a threshold equal to 0.5 is applied. The set of seed candidates may contains voxel that are not in the centerline. Our second step, *decimation*, intends to eliminate these voxels and it is performed by Algorithm 1.

Algorithm 1 Seed Decimation

Input:

Voxel \mathbf{v} and set $N(\mathbf{v})$.

Output:

Decision S : \mathbf{v} is not a seed

- 1: Set $d_M(\mathbf{v}) = \max\{d(\mathbf{w}) | \mathbf{w} \in N(\mathbf{v})\}$
 - 2: Set $d_m(\mathbf{v}) = \min\{d(\mathbf{w}) | \mathbf{w} \in N(\mathbf{v})\}$
 - 3: **if** $d_m(\mathbf{v}) = 0$ and $d(\mathbf{v}) < d_M(\mathbf{v})$ **then**
 - 4: $S = \text{true}$
 - 5: **else if** $\sqrt{d(\mathbf{v})^2 + 1} < d_M(\mathbf{v})$ **then**
 - 6: $S = \text{true}$
 - 7: **else**
 - 8: $S = \text{false}$
 - 9: **end if**
-

It is possible that no seeds were generated in some re-

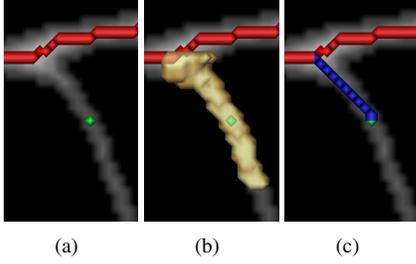


Fig. 2. Tracing Process. Following an iteration of the algorithm 2, (a) depicts current path P in red, and next seed s_k in green. (b) depicts in yellow the region R_k where the path search will be performed. The region containing the subpath from the new seed to the current path is identified. (c) shows in blue the subpath P_k generated between P and s_k .

regions of the solid, creating the potential to miss partial or entire branches during tracing. To mitigate this effect, we identify these portions as the complement in the solid of the set $\{\mathbf{v} | \exists \mathbf{s} \in \mathcal{S}, \|\mathbf{v} - \mathbf{s}\| \leq C \cdot d(\mathbf{s})\}$ where C is twice the z -smear factor of the input, and \mathcal{S} is the current set of seeds. The new seeds are computed by iteratively obtaining the maximum value $d(\mathbf{v})$ of the voxels in such regions, and mark the voxel \mathbf{v} in question as a seed, and remove from such regions the voxels \mathbf{w} such that $\|\mathbf{v} - \mathbf{w}\| \leq C \cdot d(\mathbf{v})$. This process is significantly slower than the first seed selection process, and therefore it is not used to generate seeds on the entire solid, but it is a reasonable alternative for few significantly smaller portions of the solid.

2.4. Tracing

Once the seeds are generated, the centerline is traced by iteratively identifying and connecting seeds to to the current path, according to Algorithm 2. To compute the subpath from the new seed to the previous path, we propose a variation of Dijkstra’s algorithm [3], to be applied to search the minimum path between s_k and each voxel in $P \cap R_k$. Unlike the standard Dijkstra’s algorithm, this algorithm uses an ordered pair of weights $W(e) = (W_1(e), W_2(e))$ for the directed edge $e = (\mathbf{v}, \mathbf{w})$ between neighboring voxels \mathbf{v} and \mathbf{w} . The term $W_1(e) = (d(\mathbf{v}))^{-1} + (d(\mathbf{w}))^{-1}$ is introduced to maximize the distance from the path elements to the background, and $W_2(e) = 1 - \max \left\{ \frac{\langle \mathbf{v} - \mathbf{w}, \mathbf{w} - \mathbf{p} \rangle}{\|\mathbf{v} - \mathbf{w}\| \cdot \|\mathbf{w} - \mathbf{p}\|} \mid \mathbf{p} \in P \cap R_k \right\}$ is specified to attenuate the z -smear effect by placing higher relevance to the path with lower directional variation. We consider W to be ordered by lexicographical ordering.

2.5. Centerline Post-processing

It may be the case that after the decimation step, some seeds remain that are not in the centerline, and therefore, these seeds may have been traced as false branches. These *false seeds* are,

Algorithm 2 Tracing

Input:

Seed set S and solid voxels B .

Output:

Centerline path voxel set P .

- 1: Select seed s_1 so that $d(s_1) = \max\{d(\mathbf{s}) | \mathbf{s} \in S\}$.
 - 2: $P \leftarrow \{s_1\}$.
 - 3: $k \leftarrow 1$.
 - 4: **while** $S \setminus P \neq \emptyset$ **do**
 - 5: $k \leftarrow k + 1$.
 - 6: Select seed s_k so that $d(s_k) = \max\{d(\mathbf{s}) | \mathbf{s} \in S \setminus P\}$.
 - 7: Set $R_k = \{s_k\}$.
 - 8: **while** $R_k \cap P = \emptyset$ **do**
 - 9: $R_k \leftarrow (R_k \cup N(R_k)) \cap B$.
 - 10: **end while**
 - 11: $P_k \leftarrow \text{dijkstraVariation}(s_k, R_k, P \cup R_k)$.
 - 12: $P \leftarrow P \cup P_k$.
 - 13: **end while**
-

nevertheless, close to the real centerline, thus it is reasonable to expect these *false branches* to be fairly short.

To address this potential source of errors, we identify from the graph structure of the centerline, the terminal and branching points. We compute the length of the branch from the terminal point \mathbf{u} to its closest branching point \mathbf{v} . If this length is smaller than $d(\mathbf{v})$, we eliminate all the voxels in this branch but \mathbf{v} .

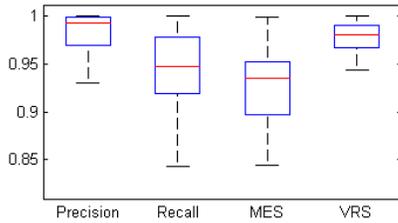
3. EXPERIMENTS, RESULTS AND COMPARISONS

To evaluate the accuracy of the proposed method, we used all of the Olfactory Projection Fibers and the first set of Neuromuscular Projection Fibers (NPF) datasets from the DIADEM challenge [1]. To compare our results, we take the Gold Standard Reconstructions provided by DIADEM. For each of these data sets, a single classifier was trained in one solid only. Some of the Gold Standard Reconstructions provided for the NPF are incomplete. An estimate of the percentage of solid traced was computed for each of the 152 solids in this dataset by linearly interpolating the annotations, constructing a synthetic solid from these annotations, and comparing these synthetic solids with a user selected binarization thresholding. This tests shows that only for thirty solids the manual annotations cover over 99% of the binary solid. We used these solids for our analysis. For our binarization step, we considered a single training with nine features: three Laplacian, with $\sigma \in \{0.25, 0.5, 0.75\}$, three low pass band filters, with $\sigma \in \{0.3, 0.5, 0.7\}$, and three high pass filters with $(\sigma_1, \sigma_2) \in \{(0.4, 0.1), (0.6, 0.3), (0.8, 0.5)\}$. We set $n = 60$, where n is the degree of P_n .

To quantitatively validate the performance of our algorithm, we used the metrics employed in [8] and [9]. These include *Precision*, *Recall*, and the *Miss-Extra-Score* (MES)

Table 1. Metrics for Neuromuscular Projection Fiber stacks.

Stack	Precision	Recall	MES	ADE (σ)	VRS
1	1.00	1.00	1.00	0.71 (0.66)	0.97
2	1.00	0.98	0.98	0.89 (1.02)	1.00
3	0.85	0.88	0.82	1.31 (0.89)	0.85
4	0.99	1.00	0.99	0.95 (1.07)	0.99
5	0.79	0.94	0.81	1.37 (0.97)	0.88
6	0.95	1.00	0.96	0.76 (1.08)	0.98
Average	0.93	0.97	0.93	1.0 (0.95)	0.87

**Fig. 3.** Boxplots of metrics for the NPF reconstructions.

defined as $Precision = S_C/S_T$, $Recall = S_C/(S_C + S_{miss})$, $MES = (S_G - S_{miss})/(S_G + S_{extra})$, where S_C is the total length of the *correctly* traced segments, S_T is the total length of the traced centerline, S_G is the total length of all segments in the ground-truth trace, S_{miss} and S_{extra} are the total lengths of missing and extra segments in the computed trace, respectively. The second metric is the *Average-Displacement-Error* (ADE) which is defined as the average displacement of the matched components. The third and last metric to be computed is *Volume-Reached-Score* (VRS), defined as the ratio in the cardinalities of the set $\{\mathbf{v} \in B | \exists \mathbf{c} \in C \text{ s.t. } \|\mathbf{v} - \mathbf{c}\| \leq d(\mathbf{c})\}$, and B , where B is the set of voxels in the binarized solid, and C is the set of voxels in the computed centerline.

Our results compare very favorably with the published literature. Specifically, Xie *et al.* [9] reported an average MES of 0.86 for a subset of five out of the six solids in Table 1. Our algorithm’s average MES is 0.93 in the entire collection, and no less than 0.91 on any subcollection of five of these solids. Compared with the FARSIGHT algorithm [8] report on a subset of 26 solids from the Neuromuscular Projection Fibers datasets, average *Precision* of 0.98 and *Recall* of 0.95, after manual editing. From the subset analyzed in this paper, we obtain an average *Precision* of 0.98 and *Recall* of 0.94, with unedited traces, and without additional training. The experiments were performed using a PC with a 2.79GHz i5 quad-core processor and 16 GB of RAM memory, using MATLAB R2011b. We used the LIBSVM library as our Support Vector Machine engine.

4. REFERENCES

- [1] K.M. Brown, G. Barrionuevo, A.J. Canty, V.D. Paola, J.A. Hirsch, G.S.X.E. Jefferis, J. Lu, M. Snippe, I. Sugihara, and G.A. Ascoli. The DIADEM data sets: representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9(2-3):143–157, 2011.
- [2] P. Chothani, V. Mehta, and A. Stepanyantas. Automated tracing of neurites from light microscopy stacks of images. *Neuroinformatics*, 9(2-3):263–278, 2011.
- [3] E. Dijkstra. A note on two problems in connection with graphs. *Numerische Mathematic*, 1:269–271, 1959.
- [4] M.S. Hassouna and A.A. Farag. Robust centerline extraction framework using level sets. In *IEEE Proc. Conference on Computer Vision and Pattern Recognition*, volume 1, pages 458–465, June 2005.
- [5] B.E. Losavio, Y. Liang, A. Santamaria-Pang, I.A. Kakadiaris, C.M. Colbert, and P. Saggau. Live neuron morphology automatically reconstructed from multiphoton and confocal imaging data. *Journal of Neurophysiology*, 100:2422–2429, 2008.
- [6] E. Meijering. Neuron tracing in perspective. *Cytometry Part A*, 77(7):693–704, 2010.
- [7] A. Santamaria-Pang, C.M. Colbert, B. Losavio, P. Saggau, and I.A. Kakadiaris. Automatic morphological reconstruction of neurons from optical images. In *Proc. International Workshop in Microscopic Image Analysis and Applications in Biology*, Piscataway, NJ, Sep. 3-4 2007.
- [8] Y. Wang, A. Narayanaswamy, C.-L. Tsai, and B. Roysam. A broadly applicable 3-D neuron tracing method based on open-curve snake. *Neuroinformatics*, 9(2-3):193–217, 2011.
- [9] J. Xie, T. Zhao, T. Lee, E. Myers, and H. Peng. Anisotropic path searching for automatic neuron reconstruction. *Medical Image Analysis*, 15(5):680–689, 2011.
- [10] Y. Zhou and A.W. Toga. Efficient skeletonization of volumetric objects. *IEEE Transactions on Visualization and Computer Graphics*, 5(3):196–209, July-September 1999.