THREE-DIMENSIONAL TRACING OF NEURITES IN FLUORESCENCE MICROSCOPY IMAGES USING LOCAL PATH-FINDING

M. Gedda
Uppsala University
Centre for Image Analysis
Box 337, SE-751 05, Uppsala, Sweden

P. Vallotton
CSIRO Mathematical and Information Sciences
Locked Bag 17, North Ryde, NSW 1670, Australia

ABSTRACT

Neurite tracing in 3D neuron images is important when it comes to analysing the growth and functionality of nerve cells. The methods used today are either of high complexity, limiting throughput, or semi-automatic, i.e., requiring user interaction. This makes them unsuitable for analysis where high throughput is needed. In this work we propose a method designed for low complexity and void of user interaction by using local path-finding. The method is illustrated on both phantom and real data, and compared with a widely used commercial software package with promising results.

Index Terms— Tracing, neurite, path-finding, 3D

1. INTRODUCTION

In high-content screening (HCS) of neurons, neurite tracing is an important subject. Tracing neurites can give information of neuron growth and connections to other neurons. Like in most biological HCS, neurite tracing is a tedious task when performed manually. Automatic methods are therefore vital for conducting large scale experiments in HCS. Improvements in imaging techniques and increase in computation speed have made rapid acquisition of three-dimensional (3D) images possible. In its natural state, a neuron extends spatially into all three dimensions. Since 2D images of neurons lack spatial information in the direction normal to the viewing plane, 3D images generally provide a more accurate structural description. Automatic methods for analysing neurons in 3D images are, thus, vital for HCS of neuron images acquired using 3D imaging techniques. However, most existing methods for analysing neurites operate only on 2D images [1] and/or require user interaction [2, 3]. For analysing linear features in 3D images, such as images of neurons and vasculature, [4] covers the three most common approaches. The first is based on skeletonisation [5]. Among the skeletonisation approaches, [6] computes the skeleton through erosion and [7] uses a 3D wavelet transform. In [8], sparseness of the structures is assumed and the method requires an additional interaction step for reconnecting fragments. The second approach is based on edge enhancement and identifying contours by chaining edge voxels together. However, both the skeletonisation and edge enhancement approaches scale poorly with size since they require processing of every voxel in the image with numerous operations per voxel. The third approach, generally referred to as tracing, is exploratory by extracting an initial point and then tracing the structures recursively based on local image properties. A method using the exploratory approach is described in [4] where template matching of a generalised 3D cylinder model guide the recursive tracing of a neurite. Template matching operates locally and scales well with image size. However, many degrees of freedom makes template matching a computationally heavy approach to tracing.

Here we present an exploratory method similar to the one in [4]. Instead of using a template matching approach, the method automatically finds seeds in a local region and uses path-finding to connect them. The path-finding is based on grey-weighted distance using the local Hessian [9]. The method is presented in Section 2. In Section 3 we determine the feasibility of using this path-finding approach for tracing neurites, and in Section 4 we discuss the results.

2. METHOD

The tracing of neurites is done automatically by first localising the nucleus and extracting seed points from where the neurites attach to the nucleus (see Section 2.2). These initial seeds are then used as starting points for the tracing. Each step in the tracing selects one seed in a first-in-first-out (FIFO) manner, searches for new seeds automatically in a local region around that seed (see Section 2.3), and traces a path from the seed to each of the new seeds (see Section 2.4). This procedure is repeated for all seeds until no seeds are left to process, i.e., until no new seeds can be been found. The Sections below describe the details of the method.
2.1. Linear features from Hessian

Extraction of linear features using the Hessian have previously been used successfully in [3] to trace neurites in 2D, and in [9] for 3D vessel enhancement filtering. The Hessian responds to cylindrical features and is computed by convolving the image with the second-order derivatives of a Gaussian kernel,

$$f_{ij}(p) = (f \ast G_{ij})(p), \quad \text{with} \quad G_{ij} = \left( \frac{\partial^2}{\partial i \partial j} G \right)(p),$$

where * denotes spatial convolution, $p$ denotes the point in $Z^n$, and $i$ and $j$ are the derivative directions. The Hessian is then defined by

$$H_f(p) = \begin{bmatrix} f_{xx}(p) & f_{xy}(p) & f_{xz}(p) \\ f_{yx}(p) & f_{yy}(p) & f_{yz}(p) \\ f_{zx}(p) & f_{zy}(p) & f_{zz}(p) \end{bmatrix}.$$  

The eigenvalues of the Hessian matrix can be combined to a value indicating how strong the linear feature is for each voxel. In [9] a value referred to as the vesselness value is constructed from the eigenvalues of the Hessian. Here we use the same construct, however, as the linear features in this work represent neurites, we refer to it as neuriteness instead of vesselness. The input parameter $\sigma$ is the standard deviation of the Gaussian kernel $G$ above.

2.2. Selecting start points

Since neurites grow from the nucleus, the local region around the nucleus provides a natural place to look for initial seeds to use as starting points for the tracing. To be able to do this, the method first has to locate the nucleus. This is done by eroding the image by a spherical structuring element with radius specified by a neurite radius parameter $N_r$. Another parameter, $K_i$, specifies the nucleus threshold intensity in the eroded image. Each voxel in the eroded image which has a greylevel value higher than or equal to $K_i$ is marked as a nucleus voxel, and the set of all nucleus voxels make up the nucleus mask. Both the erosion of the image and the localisation of the nucleus is done in the same pass through the image.

Once the nucleus is found, a local region around it is extracted for localising the initial seeds. The size of the local region is determined by a parameter $K_m$ which specifies how far from the nucleus mask the initial seeds will be sampled. This is done by dilating the nucleus mask with a sphere of radius $K_m$, producing a surface $K_m$ voxels out from the nucleus mask. The initial seed points are then extracted as the local maxima on the neuriteness transform over this surface. To decrease the impact of noise in seed selection, an input parameter $N_t$ specifying a neuriteness intensity threshold is used. Hence, a voxel cannot be selected as a seed unless its neuriteness value is greater than or equal to $N_t$.

2.3. Tracing the neurites

The tracing is done iteratively by expanding one seed at a time. Expanding a seed means localising new seeds in a spherical neighbourhood around that seed and finding a path from the current seed through the neurite to each new seed. A 2D illustration of the spherical region is shown in Figure 1, where $R_1 = r_1N_r$ is the distance margin for new seeds, and $R_2 = r_2N_r$ is the outer radius. The valid region $\mathcal{S}$ for locating new seeds (referred to as the sensor region) is the volume enclosed by $R_1$ and $R_2$ in Figure 1. The input parameters $r_1$ and $r_2$ can be chosen by the user to set the size of $\mathcal{S}$. If the current seed is not one of the initial seeds selected in Section 2.2, a cone is cut out from the sensor to prevent back-tracking (see Figure 1). The cone has its apex at the seed and its axis running through the previous seed on the path.

![Fig. 1. The sensor area (yellow) used when finding new seeds.](image)

The local image region containing the sensor is a cube with edge length $2R_2$. Using this local approach makes the method scale well with size. New seed candidates are chosen as local maxima on the neuriteness map of the sensor region $\mathcal{S}$ and the path to each candidate is traced by using the path-finding described in Section 2.4. Each candidate has to fulfil certain criteria to be selected as a new seed.

1. Neuriteness value larger than $N_t$.
2. No more than $L$ consecutive voxels with neuriteness lower than $N_t$ on path to candidate.
3. Path to candidate cannot be covered by more than 70% by a path to a candidate which fulfills all criteria and has a longer path.

The first criterion will suppress local maxima in areas of low neuriteness and the second criterion prevents the tracing from spanning across large gaps. The third criterion stems from the fact that candidates can have overlapping paths. If a path is overlapped to a large extent by another path, the candidate with the shorter path is considered a spurious response along the longer path and will be ignored. The value 70% is estimated roughly from visual inspection but is not a crucial value. Once a path has been traced to a seed, the seed is marked for expansion. The tracing terminates when there are no more seeds left to expand.
2.4. Path-finding

Each voxel in the local region around the seed is considered as a node on a graph and the shortest path to each voxel is calculated by finding the lowest cost path from the seed. Consider finding the shortest path from a voxel $u$ to a voxel $v$. Each step $[t_i, t_{i+1}]$ of the path from $t_0 = u$ to $t_m = v$ is attributed a cost value

$$w_i = q_1(1 - \frac{1}{2}(\rho(t_i) + \rho(t_{i+1}))) + q_2||t_i - t_{i+1}||$$ (1)

where $t_i$ and $t_{i+1}$ are neighbouring voxels, $\rho(\cdot)$ is the neuriteness, $||t_i - t_{i+1}||$ is the spatial distance between the voxels, and the input parameters $q_1, q_2 \in [0, 1]$ control the impact of each term. The global cost of the path from $u$ to $v$ is the sum of all the costs of the local steps: $W(P_{uv}) = \sum_i w_i$, and the grey-weighted distance is the minimum of the costs of all the paths: $d_{uv} = \min\{W(P_{uv})\}$. The minimum-cost path is, thus, the path of consecutive voxels $\{t_0 = u, \ldots, t_m = v\}$ on the path $P_{uv}$ which result in the minimum cost $d_{uv}$. All paths are calculated using the well known Dijkstra’s algorithm.

2.5. Addressing distorted images

The image acquisition introduce distortions every once in a while which make the original assumption of cylindrical shaped neurites invalid. Instead of discarding such images, the problem can be addressed by an additional step in the method which is only used when such distortions arise. The invalidity of the cylindrical assumption will produce multiple responses along the neurites due to multiple peaks on the neuriteness map. Here we solve this by dilating the paths using a spherical structuring element with radius $r$. This makes paths occupying the same neurite grow together, creating a binary image of thick branches. We then extract a medial representation only one voxel thick using skeletonisation [5] with the tracing start and end points as anchor points. This increases the complexity by applying a global method but is only used on distorted images.

3. EXPERIMENTS

The method was tested on a phantom image of a neuron with a single neurite and a noisy fluorescence microscopy image of a neuron from a rat brain. The rat neuron image was chosen specifically to illustrate the concept under difficult conditions and here the step from Section 2.5 was applied. Since this work aims to determine the feasibility of tracing neurites by using the path-finding approach described in Section 2, rather than comparing computational speeds to existing methods, the method was implemented using MATLABTM (The MathWorks, Natick, MA, USA) v7.7 for rapid development and flexibility in testing.

The parameters used for the two cases are listed in Table 1. The standard deviation, $\sigma$, was selected to give high neuriteness responses for the neurites. The parameters $N_r$, $N_l$, $K_l$, $K_m$, $L$, $r_1$, and $r_2$ were all derived from observing the tracing behaviour. Since images acquired from similar specimen using the same microscope setup have similar properties, the parameters can be estimated from just a few images representative of a large data set and then be used throughout the analysis. The parameters $q_1$ and $q_2$ was chosen to favour neuriteness over spatial proximity in the cost function, and, thus, prevent the tracing from taking short-cuts.

The phantom image is a 70×130×70 volume containing a phantom neuron. The phantom is composed of a sphere, 21 voxels in diameter, and a sinusoidal tail, 5 voxels in diameter. Zero-mean Gaussian white noise with intensity-dependent variance of 0.4 was added to the image and it was filtered using Gaussian smoothing ($\sigma=1.2$) to make it resemble a real life setting. A slice of the phantom image is shown in Figure 2 (a), and the tracing is shown in Figure 2 (b).

The rat neuron image is a 200×200×200 volume composed from in vitro Wistar rat brain slices (400 $\mu$m thick). The volume contains a part of a singular Biocytin-filled neuron. The slices were imaged at a magnification of $\times 650$ using a camera lucida mounted on an Olympus BX50 microscope. A maximum intensity projection (MIP) of the rat neuron image is shown in Figure 2 (c), and the tracing is shown in Figure 2 (d). The same data set was also traced with the commercial neuron reconstruction software AutoNeuron\(^1\) for visual comparison. The tracing from AutoNeuron is shown in Figure 2 (e).

The result from the phantom image (see Figure 2 (b)) shows that the tracing follows the linear feature very well and gives a fair representation of the synthetic neurite without any short cuts or spurious responses. The result from the noisy rat neuron image (see Figure 2 (d)) shows that the concept works for distorted images as well even though the tracings tends to be slightly jagged in parts where the neurites are disconnected into small segments. The apparent advantage over the result from AutoNeuron is that the method produces connected tracings, unlike the AutoNeuron result which requires user interaction to connect the disconnected segments in a post-processing step, thus, seriously limiting throughput.

---

\(^1\)MBF Bioscience, http://www.mbfbioscience.com/autoneuron

---

### Table 1. Parameters used in the experiments

<table>
<thead>
<tr>
<th></th>
<th>$\sigma$</th>
<th>$N_r$</th>
<th>$N_l$</th>
<th>$K_l$</th>
<th>$K_m$</th>
</tr>
</thead>
<tbody>
<tr>
<td>phantom</td>
<td>2.0</td>
<td>4</td>
<td>0.10</td>
<td>30</td>
<td>15</td>
</tr>
<tr>
<td>real</td>
<td>2.5</td>
<td>5</td>
<td>0.03</td>
<td>40</td>
<td>10</td>
</tr>
<tr>
<td>$q_1, q_2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>phantom</td>
<td>1.0</td>
<td>0.2</td>
<td>5</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>real</td>
<td>1.0</td>
<td>0.1</td>
<td>7</td>
<td>2</td>
<td>4</td>
</tr>
</tbody>
</table>
4. DISCUSSION

The novelty of the method proposed is doing automatic tracing with low computational complexity from using a local approach. This initial proof of concept show that 3D neurite tracing by local path-finding is feasible. The method traces paths in three dimensions even under difficult conditions where the neurites more resemble chopped up flat structures than homogeneous cylinders. The natural progression from here is to do a more detailed comparison with existing methods. Future work also includes further addressing noise issues from the image acquisition. What has been done so far really show that the approach has large potential to become a method competitive to the more computationally heavy methods as well as the semi-interactive methods used today.

5. ACKNOWLEDGEMENTS

Dr Jenny Gunnersen, Howard Florey Institute, University of Melbourne, Australia, is acknowledged for providing the rat neuron image.

6. REFERENCES


