Computer vision profiling of neurite outgrowth morphodynamic phenotypes

March 6, 2013

In this supplementary document, we describe and evaluate the computer vision pipeline that allowed to automatically segment and track the soma and neurites in each frame of the timelapse datasets.

1 Data description and ground truth

1.1 Input data description and notations

The input to our approach is a series of T images $\mathcal{I} = \{I_1, \ldots, I_t, \ldots, I_T\}$ from which we extract K nucleus detections d_t^k. The tracking step described in Sec. 3.1 associates valid detections across time steps while rejecting spurious detections. Since each neuron contains only one nucleus, there is a one-to-one mapping between each valid nucleus detection c_t^i and a neuron X_t^i. Thus, the tracking task is to provide a set of neuron detections $\mathcal{X}^i = \{X_a^i, \ldots, X_t^i, \ldots, X_b^i\}$ defining an individual neuron i from time $t = a$ to $t = b$. As depicted in Fig. 1, each neuron detection X_t^i is composed of a nucleus c_t^i, a soma s_t^i, a set of J neurites $\{n_t^{i,1}, \ldots, n_t^{i,j}, \ldots, n_t^{i,J}\}$, and a set of L filopodia associated with each neurite $F_t^{i,j} = \{f_t^{i,j,1}, \ldots, f_t^{i,j,l}, \ldots, f_t^{i,j,L}\}$ so that $N_t^i = \{(n_t^{i,1}, F_t^{i,1}), \ldots, (n_t^{i,j}, F_t^{i,j})\}$. Thus, a complete neuron i at time step t is described by $X_t^i = \{c_t^i, s_t^i, N_t^i\}$.

1.2 Ground truth

In order to evaluate our automatic algorithm, we manually annotated cell bodies and neurites using appropriate tools as described hereinafter. We clearly distinguished between two annotation tasks:

- segmentation and tracking of nuclei and somata from dynamic sequences,
- segmentation of neurite trees from static images.

We clearly made this distinction because we could not find an appropriate semi-automatic or manual software for tracking the neurites. Therefore, even if our automatic pipeline is able to track neurites, only the quality of neurite trees at a given time frame will we assessed.

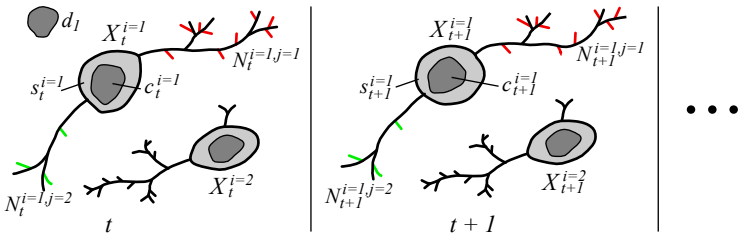

Figure 1: Neuron tracking notation. At time t a neuron i detection $X_t^i = \{c_t^i, s_t^i, N_t^i\}$ contains a nucleus c_t^i, a soma s_t^i, and a set of neurite-filopodia tuples $N_t^i = \{(n_t^{i,1}, F_t^{i,1}), \ldots, (n_t^{i,j}, F_t^{i,j}), \ldots, (n_t^{i,J}, F_t^{i,J})\}$ which contains J neurites and their associated filopodia shown in red for $j = 1$ and green for $j = 2$. A spurious nucleus detection d_1 is also shown. A neuron i is defined by a time-series of neuron detections $\mathcal{X}^i = \{X_a^i, \ldots, X_t^i, \ldots, X_b^i\}$. The tracking returns a set $\mathcal{X}^i$ for each neuron.

Our computer vision pipeline have been applied to two different sets of sequences acquired using two different magnifications: `10x` and `20x`. Therefore, we made random selections from each magnification set, and conducted our evaluation on each of them independently.

1.2.1 Cell body ground truth

To manually segment and track nuclei and somata from dynamic sequences, we used `TrakEM2`[1], a plugin of FIJI / ImageJ. To ease the annotation step, the input sequences have been encoded in a single multipage tiff file for each channel, and an xml file have been generated for each sequence to encode the segmentation and tracking data structure (see supplemental material).

- 12 sequences have been randomly selected from the `20x` dataset. From these sequences, 79 cells have been tracked, representing 4709 annotated nuclei and somata.
- 3 sequences have been randomly selected from the `10x` dataset. From these sequences, 29 cells have been tracked, representing 2152 annotated nuclei and somata.

1.2.2 Neurite ground truth

To segment neurite trees from static images, we used an improved version of the Simple Neurite Tracer (SNT) plugin. The SNT plugin[2], a FIJI plugin, is a semi-automatic tool meant to reduce user' interaction for neuronal tree tracing . Recently, an improved version of SNT have been released `http://cvlab.epfl.ch/software/delin/index.php`, allowing a better description of the neurite branches, including their width estimate and more accurate centeline extraction as shown in figure 2.

[1] publicly available at `http://www.ini.uzh.ch/~acardona/trakem2.html`
[2] `http://fiji.sc/wiki/index.php/Simple_Neurite_Tracer`

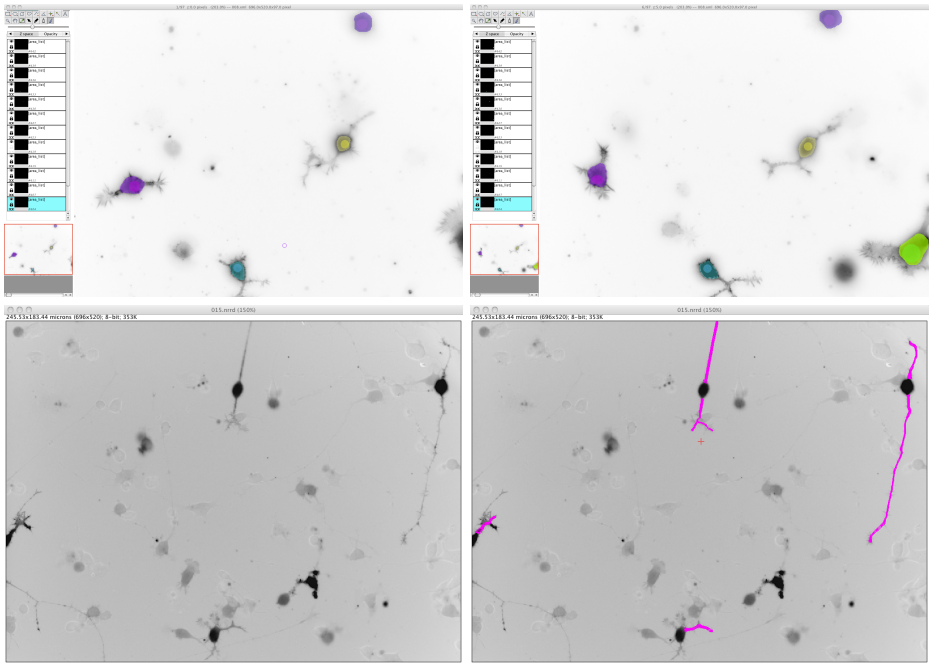

Figure 2: Ground truth annotation. First row, neuron tracking annotation using the `TrakEM2` plugin. Two different time frames are displayed. Second row, Neurite trees annotation using the Simple Neurite Tracer plugin. On the left, the original image, and on the right the manual annotation overlaid on the image.

Similarly to the dynamic dataset, we randomly selected images from the two different magnification sets, and annotated all visible neurite trees:

- 30 images from the 20x dataset have been randomly selected. 223 neurite trees or neurite branches have been annotated.
- 27 images from the 10x dataset have been randomly selected. 257 neurite trees or neurite branches have been annotated.

Filopodia have been annotated only on images from the 20x dataset.

2 Nuclei and somata segmentation

2.1 Nuclei Detection

The first step in our approach is to extract a set of nucleus detections $\{d^1, \ldots, d^K\}$ over the image series. We worked with two-channel images where the cytoskeleton is marked with Lifeact-GFP and nuclei are marked with NLS-mCherry. The nuclei can be reliably detected as a Maximally Stable Extremal Region

(MSER) [1] of the NLS-mCherry channel, and performing a morphological filling operation. The MSER detector finds regions that are stable over a wide range of thresholds of a gray-scale image. For that, we used the `VLFeat` implementation of MSER[3]. Default parameters of the MSER were used to segment the nuclei except the minimal and maximal size of a nuclei at the given resolution. The main advantage of MSER compared to the thresholding approach [2] is its robustness and insensitivity to contrast variations.

2.2 Somata detection

Using the nuclei as seed points, somata are segmented using a region growing and region competition algorithm on the Lifeact-GFP channel. This is done by launching a propagating front from all the detected nuclei simultaneously. For a given image frame, let $\{d^1, \cdots, d^p\}$ be the set of detected nuclei. To segment the semata, we first compute a solution of the Eikonal equation

$$\|\nabla \mathcal{U}\| = \mathcal{P} \quad \text{such that} \quad \mathcal{U}(d^k) = 0, \tag{1}$$

where

$$\mathcal{P}(x) = \frac{1}{A \exp\left(-\frac{(I(x)-\mu_k)^2}{2\beta^2\sigma_k^2}\right) + 1}, \tag{2}$$

k being the index of the closest nuclei detection d_k to the pixel location x, and $I(x)$ being the associated green intensity, μ_k and σ_k being the mean and standard deviation of the green intensities of pixels describung d_k.

[3]publicly available at `http://www.vlfeat.org/`

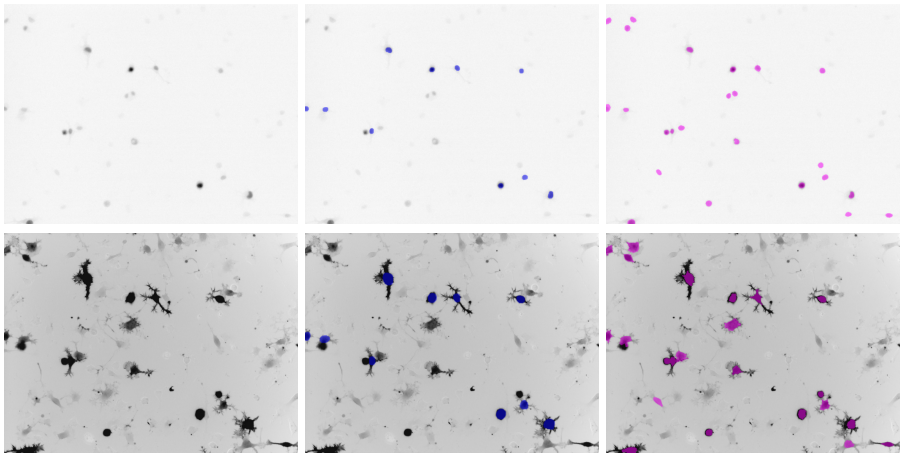

Figure 3: Cell body detection. On the first row, nuclei detections overlaid on top of the NLS-mCherry channel. On the second row, somata detections overlaid on top of the Lifeact-GFP channel. From left to right: the original image, the manual annotations, and the automatic detections.

$\mathcal{U}$ defines a distance to the detections d^k. It combines the local intensity differences and the euclidean distance. From equation 2, one can see that the more the green intensity of a pixel $I(x)$ is different from the mean intensity of the closest detected nuclei μ_k, the higher the potential $\mathcal{P}$ would be, and from equation 1, the higher $\mathcal{U}$ would be.

The somata segmentations are finally obtained by thresholding both $\mathcal{U}$ and the euclidean distance to the nuclei (those 2 shresholds are denoted $\mathcal{T}_g$ and $\mathcal{T}_e$ repectivelly). An example for nuclei and somata segmentation is depicted on figure 3. For all our experiments, we took $A = 1e7$, $\beta = 1.5$, $\mathcal{T}_g = 2e - 6$ and $\mathcal{T}_e = 7$ for the 10x magnification and $\mathcal{T}_e = 12$ for the 20x magnification.

2.3 Evaluation

A detection, either a nucleus or a somata, is considered positive if there exist a ground truth object (of the same kind) overlapping sufficiently with it. More formally, a detection d is considered as a positive detection if there exist a ground truth object g such that $\frac{g \cap d}{g \cup d} > 80\%$.

- On the 10x dataset, and over the 3 annotated sequences, only 28 nuclei have not been detected. That represents **1.3%** of miss detections.
- On the 20x dataset, and over the 12 annotated sequences, 101 nuclei have not been detected. That represents **2.14%** of miss detections.

Since the ground truth is not complete, in other words, some visible cells have not been annotated, then we do not count the false positive detections.

3 Cell body tracking

3.1 Fast Greedy Tracking of Nucleus Detections

The tracking algorithm searches through the full set of nuclei detections and iteratively associates the most similar pairs of detections, returning lists of valid detections corresponding to each neuron $\mathcal{X}^i$. This is accomplished by constructing a graph $\mathcal{G} = (\mathcal{D}, \mathcal{E})$ where each node $d_t^k \in \mathcal{D}$ corresponds to a detection. For each detection d_t^k in time step t, edges $e \in \mathcal{E}$ are formed between d_t^k and all past and future detections within a time window W. A weight w_e is assigned to each edge according to spatial and temporal distances, and a shape measure $w_e = \alpha||d_{t1}^k - d_{t2}^l|| + \beta|t1 - t2| + \gamma f(\nu_{t1}^k, \nu_{t2}^l)$ where $e^{k,l}$ connects d_t^k and d_t^l, and ν^k is a shape feature vector containing d_t^k's area, perimeter, mean intensity, and major and minor axis lengths of a fitted ellipse. f evaluates differences between a feature a extracted from d_t^k and d_t^l as $f(a^k, a^l) = \frac{|a^k - a^l|}{|a^k + a^l|}$. The tracking solution corresponds to a set of edges $\mathcal{E}' \subset \mathcal{E}$ that minimizes the cost $\sum_{e \in \mathcal{E}'} w_e$.

To minimize this cost function, we adopt a greedy selection algorithm outlined in Table 1 and summarized in Fig. 4 that iteratively selects an edge with

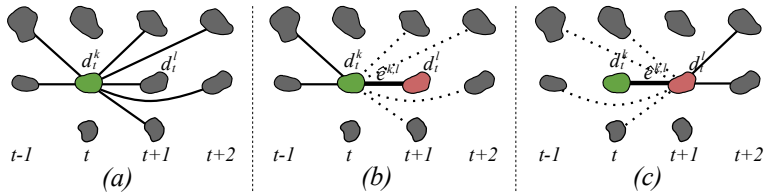

Figure 4: *Greedy Tracking. (a)* The algorithm begins with each detection fully connected to all future and past detections within a time window W. Above, only d_t^k's edges are shown. *(b)* Each iteration, the edge $\hat{e}^{k,l}$ with minimum cost $\hat{w}_e$ is added to $\mathcal{E}'$. Edges connecting d_t^k to future detections are removed from $\mathcal{E}$. *(c)* Edges connecting d_t^l to the past are removed from $\mathcal{E}$. The process is repeated until $\hat{w}_e > T$.

minimum cost $\hat{w}_e$ and adds it to the set $\mathcal{E}'$, removing future and past connections from the detections $e^{k,l}$ connects. The algorithm iterates until the minimum cost $\hat{w}_e$ is greater than a threshold T. The track for neuron i is extracted from $\mathcal{E}'$ by traversing the graph $(\mathcal{G}, \mathcal{E}')$ and appending linked nucleus detections to $\mathcal{X}^i$.

Algorithm 1 Greedy tracking association algorithm

Start with an empty set $\mathcal{E}'$.
repeat
  Find edge $\hat{e}^{k,l}$ with minimum cost $\hat{w}_e$.
  Add $\hat{e}^{k,l}$ to $\mathcal{E}'$, linking detections d_{t1}^k and d_{t2}^l.
  Remove $\hat{e}^{k,l}$ from $\mathcal{E}$.
  if $t1 < t2$ **then**
    Remove edges between d_{t1}^k and *future* detections (where $t > t1$) from $\mathcal{E}$
    Remove edges between d_{t2}^l and *past* detections (where $t < t2$) from $\mathcal{E}$
  else
    Remove edges between d_{t1}^k and *past* detections (where $t < t1$) from $\mathcal{E}$
    Remove edges between d_{t2}^l and *future* detections (where $t > t2$) from $\mathcal{E}$
  end if
until $\hat{w}_e > T$

3.2 Evaluation: TODO

4 Neurites detection

4.1 Probabilistic Neuron Segmentation and Neurite Tree Extraction

Given an image I_t and the set of somata present in it $S_t = \{s_t^1 \dots s_t^m\}$, our goal is to associate to each pixel u a label $J_t(u)$ that indicates to which soma it belongs. The probability of $J_t(u)$ can be deduced using Bayes' rule,

$$P(J_t(u) = i | S_t, I_t) = \frac{P(S_t, I_t | J_t(u) = i)}{\sum_{\eta=1}^{m} P(S_t, I_t | J_t(u) = \eta)}, \tag{3}$$

where we have assumed a uniform distribution on $P(J_t(u))$. The numerator is modeled as the probability of the path L that connects maximally the voxel u to the soma s_t^i, $P(S_t, I_t|J_t(u) = i) = \max_{L:u\to s_t^i} \prod_{\{l_r\}\in L} P(I_t(r)|l_r)$, where l_r are indicator variables for the locations forming the path L. We chose this model since an optimal maxima can be found by minimizing its negative likelihood using geodesic shortest path [3] and because it produces connected components.

To optimize this function, we first compute geodesic distances from the tracked somata using the fast marching algorithm [4, 3], applied on $P(I_t(v)|v)$, where $P(I_t(v)|v)$ represents the probability that a neurite traverses a node v and is obtained by applying a sigmoid function to the output of the tubularity filter of [5]. The parameters of the sigmoid function are estimated using maximum likelihood.

Finally, we define the set of neurite pixels U_n^t as those that connect to any soma with a higher probability than ϵ. We predict their labels as the ones that maximize Eq. 3. The set of pixels associated to neuron X_t^i is the union of the neurites and the soma associated with i, $U_i^t = \{u \in U_n^t | J_t(u) = i\} \cup s_t^i$. To reduce the neurite segmentation to a tree, we skeletonize the neuron and define as root node the pixel of the skeleton closest to the centroid of the nucleus. We instantiate a Minimum Spanning Tree from the root and create a neurite tree every time the the spanning tree exits the soma.

4.2 Evaluation: TODO

5 Neurite Tracking and Filopodia Detection

Neurites are tracked by applying the algorithm described in Sec 3.1 using the centroids of the neurite trees instead of nucleus centroids, with the additional

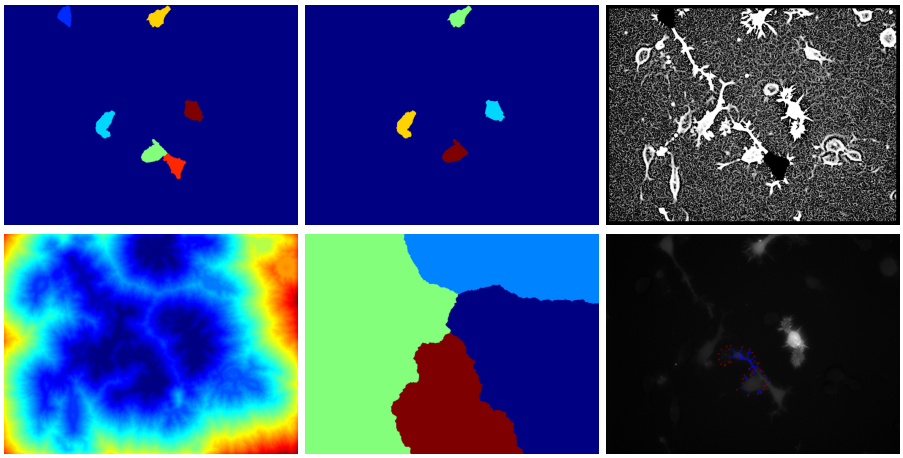

Figure 5: TODO.

constraint that edges may only exist between neurites that emanate from the same soma. Filopodia are detected by starting at each leaf node in a neurite and traversing the tree until a branch point is reached. If the distance traversed is less than a threshold T_f, the traversed locations are considered to be filopodia.

6 Filopodia detection

References

[1] Nistér, D. & Stewénius, H. Linear time maximally stable extremal regions. In *Proceedings of the 10th European Conference on Computer Vision: Part II*, ECCV '08, 183–196 (Springer-Verlag, Berlin, Heidelberg, 2008).

[2] Gonzalez, G., Fusco, L., Pertz, O. & Smith, K. Automated quantification of morphodynamics for high-throughput live cell imaging datasets. *EPFL Technical Report* (2011).

[3] Cohen, L. & Kimmel, R. Global Minimum for Active Contour Models: A Minimal Path Approach. 666–673 (1996).

[4] Sethian, J. *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science* (Cambridge University Press, 1999).

[5] Frangi, A. F., Niessen, W. J., Vincken, K. L. & Viergever, M. A. Multiscale Vessel Enhancement Filtering. *Lecture Notes in Computer Science* **1496**, 130–137 (1998).