Computer vision profiling of neurite outgrowth morphodynamic phenotypes

September 19, 2013

We developed an image processing pipeline to segments nuclei, somata and neurites and to track them from Time-Lapse High-Content Screens. In the following, we first describe the data and introduce some notations, then we give a detailed description of our processing pipeline. Finally, we describe our evaluation methodology to assess the quality of the segmentation results.

1 Data description and notations

We worked with sequences of two-channel images, one in which the cytoskeleton is marked with Lifeact-GFP. In the other, the nuclei are marked with NLS-mCherry, see the first column of figure 2.

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. 2.2 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, and a set of J neurites $N_t^i = \{n_t^{i,1}, \ldots, n_t^{i,j}, \ldots, n_t^{i,J}\}$. Thus, a complete neuron i at time step t is described by $X_t^i = \{c_t^i, s_t^i, N_t^i\}$.

2 Neuron segmentation and tracking

A three stage neurone segmentation and tracking approach is proposed. First, as described in section 2.1, we segment nuclei and the associated somata. Since each neuron contains only one nucleus, there is a one-to-one mapping between each valid nucleus and soma, yielding to a neurone cell body. Second, the cell body tracking step described in section 2.2, associates valid cell bodies across time steps while rejecting spurious ones. Finally, using the valid tracked cell bodies, neurites are segmented and tracked over time.

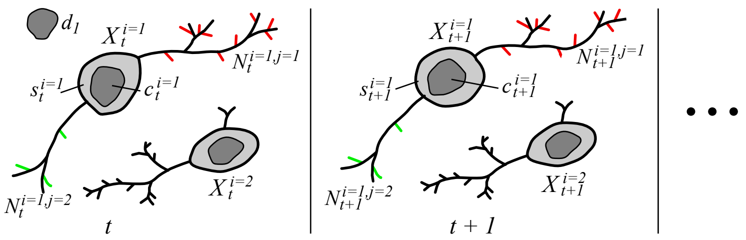

Figure 1: TODO TODO: no filopodia Neuron tracking notation. At time t a neuron i detection $X_t^i = \{c_t^i, s_t^i, N_t^i\}$ contains a nucleus c_t^i, a soma s_t^i, and a set of neurite-filopodia tuples $N_t^i = \{(n_t^{i,1}, F_t^{i,1}), \dots, (n_t^{i,j}, F_t^{i,j}), \dots, (n_t^{i,J}, F_t^{i,J})\}$ 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 $\mathcal{X}^i = \{X_a^i, \dots, X_t^i, \dots, X_b^i\}$. The tracking returns a set $\mathcal{X}^i$ for each neuron.

2.1 Nuclei and Somata segmentation

The first step in our approach is to extract a set of nucleus detections $\{d^1, \dots, d^K\}$ 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[1]. Default parameters of the MSER were used to segment the nuclei except the minimal and maximal size of a nuclei at the given resolution, which were fixed to 70 and 170 [2]. The main advantage of MSER compared to the thresholding approach [2] is its robustness and insensitivity to contrast change.

Using the nuclei as seed regions, somata are segmented using a region growing and region competition algorithm on the Lifeact-GFP channel, called the *green* channel. This is done by launching a propagating front from all the detected nuclei simultaneously. For a given image frame, let $\{d^1, \cdots, d^K\}$ be the set of detected nuclei. To segment the somata, 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 $\nabla \mathcal{U}$ is the gradient of a distance $\mathcal{U}$ to be computed and

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

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 respectively the

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

[2]Nuclei are considered to have circular shapes with radius varying between 4 and 8 pixels

mean and standard deviation of the green intensities of the pixels describing d_k. Parameter β, is a multiplicative factor of the intensity standard deviation σ_k, and represents a tolerance of variation between the local foreground (somata) intensities and the local background intensities.

$\mathcal{U}$ defines a geodesic 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 algorithm to compute the geodesic distance $\mathcal{U}$, is a variant of the Fast Marching algorithm [3, 4] introduced in [5].

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 2. For all our experiments, we took $A = 1e7$, $\beta = 1.5$, $\mathcal{T}_g = 2e-6$ and $\mathcal{T}_e = 7$.

At this point, cell body detections $d_t^i = (c_t^i, s_t^i)$ have been obtained form the whole sequence. A filtering step is applied to these detections to keep only the most reliable ones. First, detections that are too close to the image boundary (with distance less than 10 pixels) are ignored. Second, the minimal tolerated circularity is 0.2. Finally, the maximal accepted eccentricity is 0.85.

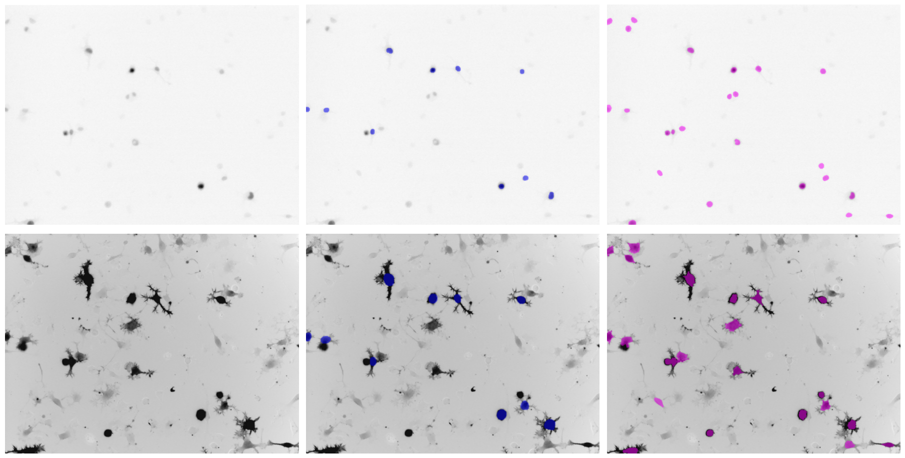

Figure 2: Cell body detection. On the first row, nuclei detections overlaid on top of the NLS-mCherry channel. On the second row, somata detections overlaid on top of the Lifeact-GFP channel. From left to right: the original image, the manual annotations, and the automatic detections.

2.2 Cell body tracking

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. 3 that iteratively selects an edge with 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 $\hat{w}_e > T$
```

The parameters of this algorithm have been fixed empirically as follows: $W = 4$, $\alpha = 1$, $\beta = 50$, $\gamma = 40$ and $T = 200$. Only tracks containing at east 20 frames are kept. In addition to these parameters, a spatial connection constraint is applied during the construction of the graph $\mathcal{G}$. In fact, only detections that are at distance less than 50 pixels are connected to create the set of edges $\mathcal{E}$.

Once the tracking is achieved, tracks are sorted according to their total cumulated green intensities, and only the 20 best tracks are kept.

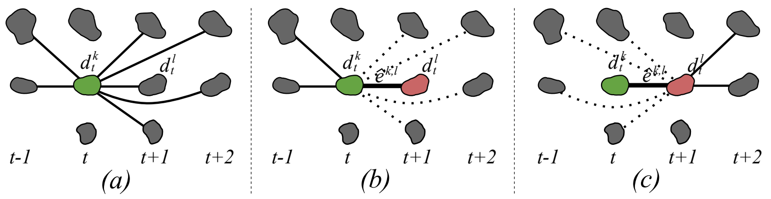

Figure 3: *Greedy Tracking. (a)* The algorithm begins with each detection fully connected to all future and past detections within a time window W. Above, only d_t^k's edges are shown. *(b)* Each iteration, the edge $\hat{e}^{k,l}$ with minimum cost $\hat{w}_e$ is added to $\mathcal{E}'$. Edges connecting 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 $\hat{w}_e > T$.

2.3 Neurites segmentation and association

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.

The extraction of neurites from a time frame I_t proceeds in the following stages:

- Compute tubularity measure T_t as in [6]. In addition, detected but not tracked somata are ignored.
- Estimate the parameters of a sigmoid functions which is applied to the tubularity measure to obtain a potential P_t that drives the Fast Marching algorithm [3, 4].
- Launch simultaneously the front propagation Fast Marching algorithm from all the tracked somata. This is done by solving the Eikonal equation $\|\nabla \mathcal{U}_t\| = P_t$. That yields a geodesic distance map $\mathcal{U}_t$ and the associated tessellation $\mathcal{V}_t$. The complexity of the algorithm, at this stage, does not depend on the number of tracked somata but depends only on the size of the image.
- Threshold the geodesic distance with a soft threshold $\mathcal{T}_s$, and then extract local maxima of $\mathcal{U}_t$ in each thresholded region[3]

[3]the local maxima are obtained using the Matlab function `imregionalmax`

- Using the Geodesic distance $\mathcal{U}_t$, back-propagate from all the local maxima, by keeping only points for which the value is above a hard threshold $\mathcal{T}_h$
- Finally, we instantiate a Minimum Spanning Tree from each root touching a soma, and create the associated neurite tree.

The different steps of our approach are illustrated in figure 4.

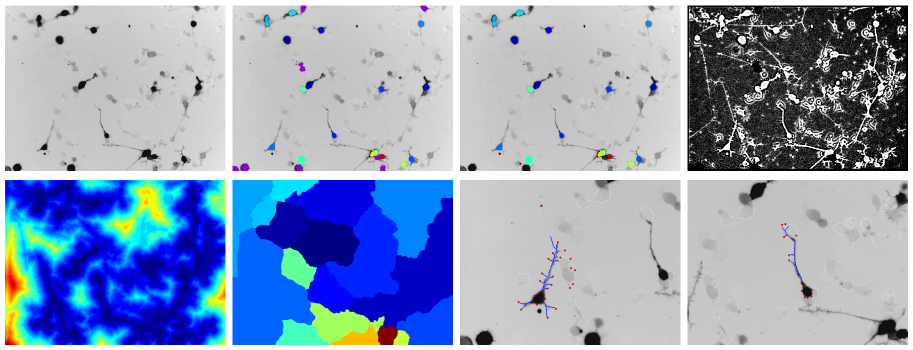

Figure 4: Neurites detection. From left to right, top to bottom : -Original image (only the green channel is displayed). -Both tracked and detected somata are overlaid on top of the original image. Detected but not tracked somata are coloured with dark violet. -Only tracked somata are overlaid on top of the original image. - Potential P_t, based on the tubularity measure of [6]. -Geodesic distance $\mathcal{U}_t$ computed from the tracked somata. Blue corresponds to low values and red to high values. - The Voronoi tessellation $\mathcal{V}_t$ associated to the geodesic distance. Each colour represent a region. -The last two figures are closeup looks of the original image illustrating the last steps of our neurite detection algorithm: first, local maxima of the soft thresholded regions are displayed in red, then the hard thresholded back propagation pixels are shown in blue.

Parameters of the neurite detection algorithm are as follows:

- Frangi [6] parameters are: `FrangiOpt.FrangiScaleRange = [1 2]`, `FrangiOpt.FrangiScaleRatio = 1`, `FrangiOpt.FrangiBetaOne = .5`, `FrangiOpt.FrangiBetaTwo = 15`,,
- Geodesic distance thresholds: $\mathcal{T}_s = -\log(10^{-4})$, and $\mathcal{T}_h = -\log(0.2)$.
- Finally, any neurite containing less than 10 pixels have been ignored.

2.4 Neurite Tracking

Neurites are tracked by applying the algorithm described in Sec 2.2 using the centroids of the neurite trees instead of nucleus centroids, with the additional constraint that edges may only exist between neurites that emanate from the same soma. The weight w_e of an edge connecting two neurites N_t^i and $N_{t'}^j$is assigned according to spatial distance and a shape measure $w_e = w_{TCL} f(\text{TotalCableLenght}(N_t^i), \text{TotalCableLenght}(N_{t'}^j)) + w_{\text{Centroid}} \|\text{Centroid}(N_t^i) -$

Centroid$(N_{t'}^{j})\| + w_{\text{SomaContact}}\|\text{SomaContact}(N_t^i) - \text{SomaContact}(N_{t'}^{j})\|$, where $w_{TCL} = 50$, $w_{\text{Centroid}} = 10$, $w_{\text{SomaContact}} = 5$, TotalCableLength$(N)$ is the total cable length of a neurite N, Cenrtoid(N) is its centroid, SomaContact(N) is its contact point with the soma, and $f(a^k, a^l) = \frac{|a^k - a^l|}{|a^k + a^l|}$.

During the neurite tracking stage, only neurites that are considered stable have been taken into account. Neurites are considered stable if their total cable length is above 30 pixels. The threshold parameter of the tracking algorithm (Algorithm 1), have been adapted to neurites by taking $T = 800$.

3 Evaluation

Intro::: TODO

3.1 Evaluating Cell body detection

3.2 Evaluating Cell body tracking

3.3 Evaluating Neurites detection

3.4 Evaluating Neurites association

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] Benmansour, F. & Cohen, L. D. Fast object segmentation by growing minimal paths from a single point on 2d or 3d images. *Journal of Mathematical Imaging and Vision* **33**, 209–221 (2009).

[6] 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).