Morphological Analysis and Feature Extraction of Neurons from Mouse Cortices Multiscale 3D Microscopic Images

Alexis Zubiolo[1] Kawssar Harb[2] Michèle Studer[2] Éric Debreuve[1] Xavier Descombes[1]

***Abstract*— In this paper, we propose a framework to analyze the morphology of mouse neurons in the layer V of the cortex from 3D microscopic images. We are given 8 sets of images, each of which is composed of a 10x image showing the whole neurons, and a few (2 to 5) 40x images focusing on the somas. The framework consists in segmenting the neurons on both types of images to compute a set of specific morphological features, and in matching the neurons in the 40x images to their counterparts in the 10x images to combine the features we obtained, in a fully automatic fashion.**

I. Introduction

The mammalian neocortex is organized into six layers, in which neuronal populations can express different neuronal subtype-specific genes and have distinct morphology and connectivity patterns [1]. For instance, layer V neurons consist of different types of projection neurons (PNs): corticofugal (including subcerebral and corticostriatal), and corticocortical (including callosal). This motivates the need of a morphological analysis tool to characterize these neurons.

In this paper, we propose a method to match corresponding neurons from two different resolutions (confocal microscopy images taken at 10x and 40x zooms, see Figure 1). Instead of using registration techniques [2] that would be expensive on 3D stacks, we suggest to use nothing more than the position of the centroid of the somas and solve assignment problems to establish a correspondence map between the images at different resolutions. The framework consists of 3 parts. The first part is dedicated to the 40x images. Here, the goal is to segment the neurons and all their dendrites in order to compute various features such as the volumes of the somas and the numbers of dendrites (all of the chosen features are described in Section II). The second part (Section III) focuses on the 10x images, on which we segment the entire neurons, and more specifically their apical dendrite. The third part (Section IV) consists in establishing a correspondence map between the neurons found at both resolution to gather the information.

II. Analysis of the 40x images

The analysis of the 40x images is decomposed in two subparts: Firstly, we segment the neuron (Section II-A) in order to be able to compute in a second step the features describing each neuron (Section II-B).

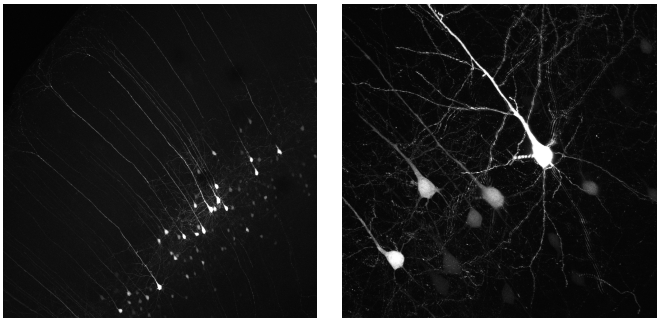


Fig. 1. Maximum intensity projections of the two different types of images dealt with: the 10x images showing the whole cortex (left), and a 40x image focusing on the somas (right).

A. Soma and dendrite starts detection

During segmentation of the neurons from the high resolution images, we are challenged by the noise in the data sets and the possible neurite connections between different neurons. We therefore adopt a two-step strategy that consists in first detecting the soma and the start of the dendrites and then reconstructing the remaining part of the dendrites. The soma and start of dendrites are segmented using a hysteresis threshold followed by a reconstruction step to recover the full dendrite. The resulting algorithm is described in Algorithm 1 and the different steps of the neuron detection are illustrated on Fig. 2.

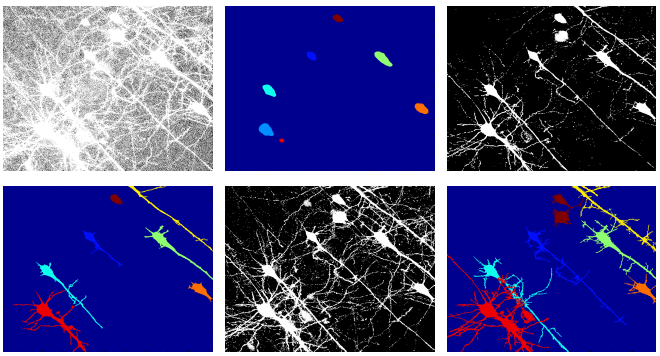


Fig. 2. Neuron segmentation (we show the maximum intensity projections). Initial image (top left), Labeled first threshold (top center), Second threshold (top right), First segmentation (bottom left), Third threshold (bottom center) and Final result (bottom right).

This work was supported by the French Government (National Research Agency) through the "Investments for the Future" LABEX SIGNALIFE: program reference ANR-11-LABX-0028-01.

[1]Alexis Zubiolo, Xavier Descombes and Éric Debreuve are within team Morpheme (I3S Laboratory, University of Nice Sophia Antipolis, Inria Sophia Antipolis, CNRS). alexis.zubiolo@i3s.unice.fr, xavier.descombes@inria.fr, eric.debreuve@cnrs.fr

[2]Kawssar Harb and Michèle Studer with the Institute of Biology Valrose kawssar.harb@unice.fr, michele.studer@unice.fr

Algorithm 1 The three steps of the neuron detection

1: **Thresholding:** Let us denote by I the initial image and consider three thresholded images $T_1(I)$ (resp. $T_2(I)$ and $T_3(I)$) with the corresponding thresholds t_1 (resp. t_2 and t_3) manually set such that $t_1 > t_2 > t_3$

2: **Hysteresis thresholding:** Consider the connected components of the first two thresholded images denoted $\{C_1(i), i = 1, \dots, n_1\}$ and $\{C_2(j), j = 1, \dots n_2\}$, the detected neurons are initialized as follows:

$$N = \{N_k : \exists i, j, N_k = C_2(j), N_k \cap C_1(i) \neq \emptyset\} \quad (1)$$

3: **Neurite Reconstruction:** Initialize $u = 0$, $\forall k, N_k^{(0)} = N_k$ iterate the following process until no change appear: for $k = 1, \dots, n_k$

$$N_k^{(u+1)} = D(N_k^{(u)}, b_1) \cap \left(T_3(I) / \cup_{v=1}^{k-1} N_v^{(u+1)}\right) \quad (2)$$

where $D(A, b_1)$ is the dilation of A by the unit sphere

B. Feature extraction

The segmentation previously obtained enables us to compute the first features describing the neurons. Let us consider a segmented neuron.

We start by the soma analysis. The rough soma detection of the somas described in Section II-A should be refined to obtain an accurate estimation of its volume. Indeed, depending on its size, the dendrites may have been included in the soma segmentation at this stage. It is important to distinguish the soma from the dendrites in a robust way to guarantee a fair feature estimation between all the neurons. As a consequence, we need to properly define what a soma is, *i.e.* determine where the soma ends and where dendrites start. To do so, we draw concentric spheres centered at the centroid of the first soma segmentation. We progressively increase the radius of the sphere (starting from 0), intersect this sphere with the final neuron segmentation, and consider the connected components. When the sphere starts being bigger than the soma, the number of connected components increases and when the circle reaches a dendrite, its intersection with the dendrite gives us a section of the dendrite. The size (*i.e.* the number of voxels) of this intersection gives an approximation of the surface of this section. Let us consider this surface as a function of the radius for each connected component. At some point, this function starts decreasing (the intersection is still in the soma), before becoming relatively constant (the intersection is in the dendrite). The breaking point between these two consecutive behaviors gives us the critical radius that separated the soma from the dendrite. Note that the critical radius can differ from one dendrite to another. This more accurate segmentation enables us to identify the position of all the dendrites.

Using this well-defined soma segmentation, it is possible to compute our first features: The volume and the shape of the soma. Its volume is simply the number of voxels multiplied by the volume of a voxel, *i.e.* $0.346 \times 0.346 \times 1 = 0.1197\,\mu\text{m}^3$. As for the shape description, we used Hu's seven moment invariants [3]. The other features are related to the dendrites: their number, the thickness of the apical dendrite, and the angular distribution of the basal dendrites. The number of dendrites and the thickness of the apical dendrite are easily computed thanks to the soma segmentation.

III. Analysis of the 10x images

To extract features from the 10x images, we segment the neuron and especially its apical dendrite (Section III-A) and then we compute the distance from the soma to the first apical branching based on this detection.

A. Apical dendrite detection

Apical dendrites are approximately parallel as they go from the neuron to the pia. Besides, the resolution and the noise in data make tracking algorithms inappropriate. We propose to detect the dendrite sections in the plane perpendicular to its axis and merge them using a data association procedure. We will consider the length between the soma and the first branch on the apical dendrite as a characteristic of the neuron [1]. The detection uses a Marked Point Process on a configuration space composed of small rectangles [4]. Let $r = \{r_i, i = 1, \dots, n\}$ be a set of rectangles. We consider the energy function defined as:

$$\mathcal{U}(r) = \sum_i \mathcal{U}_1(r_i) + \sum_{i,j} \mathcal{U}_2(r_i, r_j) \quad (3)$$

where $\mathcal{U}_2$ defines a non-overlapping prior ($\mathcal{U}_2(r_i, r_j) = \infty$ if r_i intersects r_j and 0 otherwise) and $\mathcal{U}_1$ is a data term that measures the contrast between the object and the background:

$$\mathcal{U}_1(u) = \left(exp\left(-\frac{d_B(u) - d_0}{100}\right) - 1\right) \delta(d_B(u) \geq d_0) \quad (4)$$

where:

$$d_B(u) = \frac{(\mu_1 - \mu_2)^2}{4\sqrt{\sigma_1^2 + \sigma_2^2}} - \frac{1}{2} log \frac{2\sigma_1\sigma_2}{\sigma_1^2 + \sigma_2^2} \quad (5)$$

and d_0 is a threshold defining the minimum acceptable contrast for an object, μ_1 (resp. σ_1^2) and μ_2 (resp. σ_2^2) are the means (resp. variances) of pixels within the objects and in the outer border of the object. To obtain the set of objects minimizing the energy defined in (3), we consider the Multiple Birth and Cut algorithm described in [5]. Once the sections have been detected we remove the noise using a morphological opening and merge the sections using a morphological closing in the direction of the dendrite. Results of this segmentation are given in Fig. 3.

B. Soma to first branching distance

The neuron segmentation previously computed enables us to estimate the distance between the soma to the first branching of the apical dendrite. This is done using the perpendicular slices (as in Section III-A) of the segmented image, starting with the soma of the dendrite (detected as in Section IV). For each neuron, we count the number of connected components for each slice: we have 1 connected

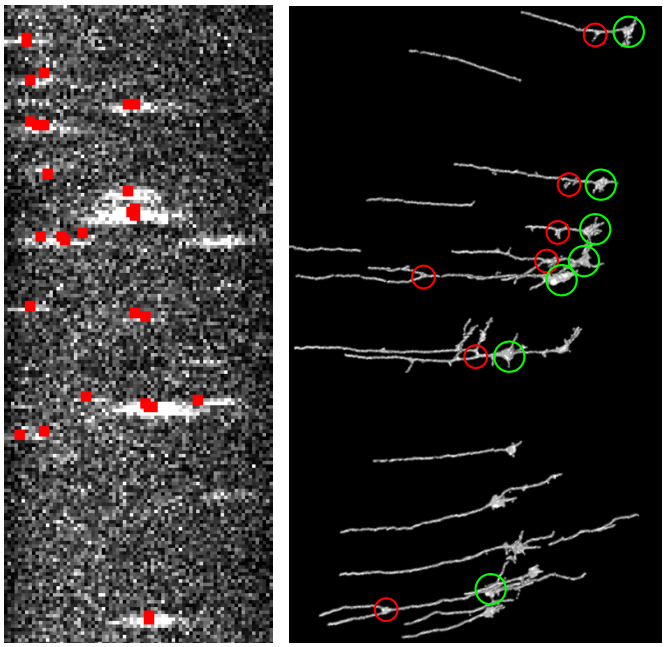


Fig. 3. The two steps of the neuron reconstruction: The dendrites sections detection (left) and the dendrite reconstructions (right). The right image is a snapshot from a 3D view of the reconstructed image where the somas are circled in green and the first branching in red.

components at the beginning and we browse the slices until we find two connected components. If we are unable to find the two connected components by the end of the image, we force the distance between the soma and the first branching to zero, which means that the feature could not be computed.

IV. Correspondence Mapping Between 40x and 10x Images

A. Description of the proposed method

Let us consider a set of images with a 10x image and p 40x images. As we have two different types of images representing the same data (*i.e.* the same neurons in our case), we need to establish a correspondence map between the two resolutions of images: if we consider a neuron N_j^k in the kth 40x image ($1 \leq k \leq p$), which neuron N_i in the 10x image does it correspond to?

To answer this question, we could have registered the 40x images on the 10x image using a maximum correlation principle [2]. However, such pixel-based methods would be a computationally expensive task. Besides, image registration is not required as we only need to match neurons between 40x and 10x images. Therefore, we would rather consider an approach based on geometric pattern matching scheme as in [6], [7], and [8]. We use a bipartite graph matching model based on a soma detection. That is performed on both types of images, using Algorithm 1. Denote a bipartite graph $G_k = (U, U_k, E_k)$, where U is the set of neurons in the 10x image, U_k is the set of neurons in the kth 40x image, and E_k is the set of weighted edges between the nodes (*i.e.* the neurons) of U and U_k. Note that, in our case, all the neurons in U_k are found in U.

To define the weights of the edges in G_k, we propose an approach based on the pairwise distances between neurons. We define the distance $d_{ii'}$ (resp. $d_{jj'}$) between two neurons N_i and $N_{i'}$ (resp. N_j and $N_{j'}$) of the same image as the distance between their centroids c_i and $c_{i'}$ (resp. c_j and $c_{j'}$).

Therefore, each neuron N_i (resp. N_j) of the 10x images (resp. the 40x images) can be described by a pairwise distance vector (in µm) $d_i = (d_{ii'})_{1 \leq i' \leq n}$ (resp. $d_j = (d_{jj'})_{1 \leq j' \leq n_k}$) where n is the number of detected neurons on the 10x image (resp. n_k the number of detected neurons on the kth 40x image).

Let us consider a pair of neurons (N_i, N_j) where N_i is in the 10x image and N_j in the kth 40x image. The idea is the following: if two neurons correspond to each other, then they have common (or close) values in their respective distance vectors. Consequently, we decide to match their respective distance vectors by considering another bipartite graph matching problem. Given the correspondence matrix solution X of this matching problem (*i.e.* $X_{i'j'} = 1$ if $d_{ii'}$ and $d_{jj'}$ match, and 0 otherwise), we define the weight term between these two neurons as $e_{ij}^k = C^k(d_i, d_j, X)$, where

$$C^k(d_i, d_j, X) = \sum_{i'=1}^{n} \sum_{j'=1}^{n_k} |d_{ii'} - d_{jj'}| \, X_{i'j'}$$

The optimal matching X can be obtained as the solution of an integer linear programming problem (ILP):

$$e_{ij}^k = \min_{X \in \{0,1\}^{n \times n_k}} C^k(d_i, d_j, X) \qquad (\mathrm{P}_{ij}^k)$$

$$\text{s.t.} \quad \forall j' \in [\![1, n_k]\!], \sum_{i'=1}^{n} x_{i'j'} = 1 \qquad (6)$$

$$\text{and} \quad \forall i' \in [\![1, n]\!], \sum_{j'=1}^{n_k} x_{i'j'} \leq 1 \qquad (7)$$

In this ILP, (6) means that each distance from the kth 40x image has exactly one corresponding distance in the 10x image; (7) means that each distance in the 10x image has at most one corresponding distance in the kth 40x image (all the neuron from the 10x image do not appear in the kth 40x image); and (P_{ij}^k) means that we want to minimize the differences between matched distances. Finally, once all the e_{ij}^k are computed, the set of weighted edges E_k is defined and the neuron mapping is the result of a bipartite graph matching. This problem has to be solved for all $k \in [\![1, p]\!]$ in order to match the neurons from all the p images. The results of the algorithm are illustrated in Fig. 4. Note that, given the problem we propose to solve, the algorithm is likely to fail if a neuron detected in the kth 40x image is not detected in the 10x image. Indeed, this neuron would be paired with a non-corresponding neuron (to satisfy (6)); And as a consequence the corresponding weight e_{ij}^k would not be close to 0.

B. Remarks on the proposed method

Using this method, we end up solving $n \times n_k$ ILPs to match a 40x image with its corresponding 10x image, but the resulting complexity is low as the IPLs are low dimensional.

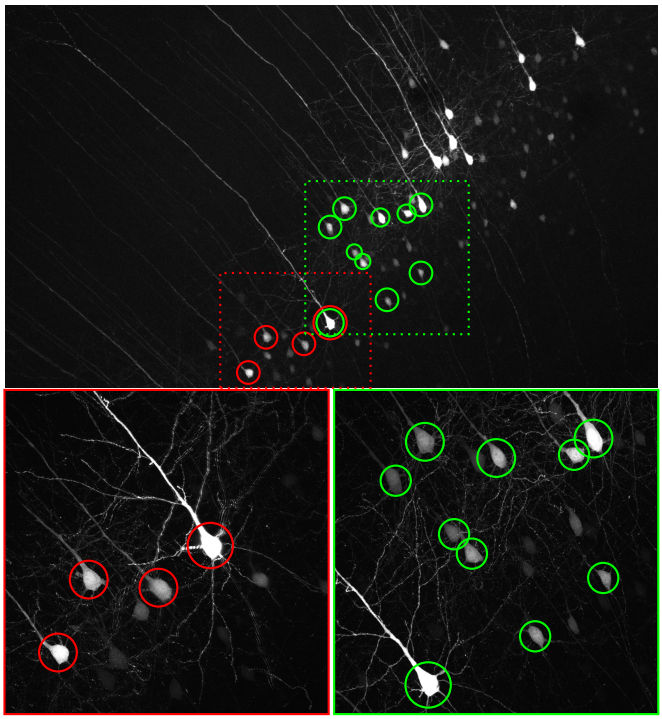


Fig. 4. Example of the resultant correspondence map. The corresponding neurons are circled with the same colors on a 10x image (top, cropped) and two of the corresponding 40x images (bottom). Note that the two chosen 40x images are overlapping and have a neuron in common.

Let us consider the case where two sets of neurons in two different 40x images can exhibit the same pairwise distances configuration. The algorithm, as it has been described in Section IV-A would then possibly fail at matching the correct pair. This situation is extremely rare (and has yet to occur in our experiments) and detectable by means of the two penalty terms, e_{ij} and $e_{ij'}$, which would be equal or close to 0. To overcome this possible issue, we could use a correlation criteria to chose between j and j' to make the method robust.

The ILPs P_{ij}^{k} can be solved with any ILP solver, such as the simplex (using the total unimodularity property of the constraint matrix [9]) or the Hungarian algorithm [10]. As the Hungarian algorithm is initially designed to solve assignment problems where $n = n_k$, it is slightly slower than the simplex if n_k is significantly lower than n. In our case, the difference is insignificant as the numbers of neurons in the images are low, but it is a detail worth considering with bigger datasets.

V. Results

We are given 3D microscopic grayscale images of mice cortices, in which the layer V subcerebral PNs are characterized by the Thy1-eYFP-H protein [11]. The images are spread into 8 sets, each of which is composed of a 10x image and a few 40x images. The 10x images show the whole cortex, including the apical dendrite. The voxel size is $1.38\,\mu\mathrm{m} \times 1.38\,\mu\mathrm{m} \times 1\,\mu\mathrm{m}$. The 40x images focus on a few somas and show more details including the secondary dendrites (see Fig. 1). The voxel size is $0.346\,\mu\mathrm{m} \times 0.346\,\mu\mathrm{m} \times 1\,\mu\mathrm{m}$. For both types of images, the number of pixels per slice is 1024×1024 and the number of slices varies between 30 and 100.

The proposed framework gives a feature table describing a total of 317 neurons. From the 8 sets of images given, all of the neurons have been successfully detected in both the 10x and 40x images. However, all features could not be computed for all the neurons: A few neurons (especially the dendrites) are barely detectable (even by an expert) as they showed limited reaction to the Thy1-eYFP-H protein.

We have compared the distance from the soma to the first branching given by the algorithm described in Section III with those of a semi-automatic segmentation made by an expert using the software Imaris from Bitplane followed by hand tracing on one of the eight images set. 8 apical branching have been detected by the expert (the others are not contrasted enough) whereas our automatic approach could get 7 of them. Besides, the matching algorithm proposed matched all the detected neurons accurately.

VI. Conclusion and Further Work

In this paper, we propose a fully automatic framework to compute features describing neurons from two different types of 3D images and a method to match objects taken from different images only based on pairwise distances. This method can be used as far as the pairwise distances between objects are considered as a criterion robust enough to describe the set of objects.

The algorithm will be tested on other sets of images from different rat populations (*e.g.* mutant and non-mutant) to study links between the morphological aspects of the neurons and a certain gene mutation. Moreover, unsupervised classification algorithms will be used to determine neuron classes based on the morphological features computed.

References

[1] A.M. Hattox and S.B. Nelson, Layer V Neurons in Mouse Cortex Projecting to Different Targets Have Distinct Physiological Properties, Journal of neurophysiology, vol. 98, pp. 3330–3340, 2007.

[2] W. Pratt, Correlation techniques of image registration, IEEE Transactions on Aerospace and Electronic Systems, pp. 353–358, 1974.

[3] M.K. Hu, Visual pattern recognition by moment invariants, IEEE IRE Transactions on Information Theory, vol. 8, pp 179–187, 1962.

[4] X. Descombes, Stochastic geometry for image analysis, John Wiley & Sons, 2013.

[5] A. Gamal Eldin, X. Descombes, G. Charpiat and J. Zerubia, Multiple Birth and Cut Algorithm for Multiple Object Detection. Journal of Multimedia Processing and Technologies, 2012.

[6] P.L. Chew, M.T. Goodrich, D.P. Huttenlocher, K. Kedem, J.M. Kleinberg, D. Kravets, Geometric pattern matching under Euclidean motion, Computational Geometry, vol. 28, pp. 113–124, 1997.

[7] D. Aiger and K. Kedem, Geometric pattern matching for point sets in the plane under similarity transformations, Information Processing Letters, vol. 109, pp. 935–940, 2009.

[8] D. Aiger and K. Kedem, Approximate input sensitive algorithms for point pattern matching, Computational Geometry, vol. 43, pp. 153–159, 2010.

[9] A. Schrijver. Combinatorial optimization: polyhedra and efficiency Springer Science & Business Media, 2003.

[10] H.W. Kuhn, The Hungarian Method for the Assignment Problem, Naval research logistics quarterly, vol. 2, pp. 83–97, 1955.

[11] G. Feng, R.H. Mellor, M. Bernstein, C. Keller-Peck, Q.T. Nguyen, M. Wallace, J.M. Nerbonne, J.W. Lichtman and J.R. Sanes, Imaging Neuronal Subsets in Transgenic Mice Expressing Multiple Spectral Variants of GFP, Neuron, vol. 28, pp. 41–51, 2000.