Graph-Based Neuron Tracking in Calcium Images

Sarun Gulyanon

June 24, 2019

In this neuron tracking problem, we are given the calcium image stack sequence, $\mathcal{I} = \{\mathcal{I}^{(t)}\}$, where $t = 0, 1, \ldots, T$. To make the problem feasible, we require the initial morphology of the neuron, $\mathcal{G} = \{V_\mathcal{G}, E_\mathcal{G}\}$.

1 Definition

A graph is denoted by $G = \{V, E\}$, where V is the set of nodes (also called *vertices*) and $E \subset V \times V$ (also defined as $E \subset [V]^2$) is the set of edges (also known as lines) of graph G.

2 Inexact Graph Matching

The problem of inexact graph matching (GM) or error-tolerant/correcting GM is finding the mapping of nodes between two graphs that optimizes a certain affinity or distortion criterion, where nodes do not need to have a correspondence in another graph.

3 Graph Edit Distance

A commonly used measurement of affinity criterion is the graph edit distance (GED). A graph can be transformed to another one by a finite sequence of graph edit operations. Each operation has a cost, which is defined differently in various algorithms. GED, $\delta(G_i, G_j)$, is defined by the least-cost graph edit operation sequence, P, that are needed to transform one graph, $G_i = \{V_i, E_i\}$, into another, $G_j = \{V_j, E_j\}$.

$$\delta(G_i, G_j) = \min_P \sum_{(u \to v) \in P} c(u \to v) \tag{1}$$

where nodes $u \in V_i$ and $v \in V_j$ are a correspondence of each other. $c(u \to v)$ is the cost of the graph edit operation that maps node u to v. The value of the cost depends on the affinity between u and v.

4 Method

In this work, we tackle the neuron tracking problem over calcium images as the inexact graph matching (GM) problem [1] computed by minimizing the GED.

Given the calcium image sequence, first convert every frame into the "frame" graph whose nodes are superpixels. Second, compute the "template" graph, which is the subgraph of the "frame" graph corresponded to the neuron trace, using the inexact graph matching. Finally, align each dendrite to the image cues, and repeat throughout the image sequence.

4.1 Creating "Frame" Graph

Due to the imaging system limitation, the maximum intensity projection of calcium image stack, $I_t = f(\mathcal{I}^{(t)})$, is used instead of the raw input volume.

To convert a frame into a graph, first compute the superpixel segmentation over I_t using techniques like Linear Spectral Clustering Superpixel [2, 3][1]. The result is the set of superpixels V_t.

Let the "frame" graph be $G_t = \{V_t, E_t\}$, where the existence of edges between superpixels E_t are computed using Delaunay Triangulation over the centroid of superpixels in V_t.

4.2 Initial "Template" Graph

The trace describes the neuron morphology at the structural level, while the input image stacks describe the neuron at the appearance level. We propose the "template" graph that bridges between two levels of representations.

We define the "template" graph at time $t = 0$, $G'_0 = \{V'_0, E'_0\}$, as the subgraph of G_0 corresponded to the initial morphology $\mathcal{G}$. V'_0 is the set of patches overlapping with $\mathcal{G}$,

$$V'_0 = \{\, v \mid v \text{ overlaps } \mathcal{G}, v \in V_0 \,\} \tag{2}$$

E'_0 is the set of edges between adjacent nodes in V'_0,

$$E'_0 = \{\, (u, v) \mid (u, v) \in E_0;\ u, v \in V'_0 \,\} \tag{3}$$

4.3 Tree Search Algorithm

A tree (state space) search algorithm is used to find the GED because it is simple, allows the many-to-many mapping function, and makes no assumptions about the problem. Tracking is done by computing the GED between the previous "template" graph and the current "frame" graph, $\delta(G'_{t-1}, G_t)$. The state space consists of a number of possible mappings and the corresponding cost. The root of the state space represents the graph edit operation sequence with

[1] https://jschenthu.weebly.com/projects.html,
https://www.mathworks.com/matlabcentral/fileexchange/65684-linear-spectral-clustering-superpixel

no correspondences. The first layer of states represents mappings with one correspondence, and so on. The algorithm terminates when every node in G'_{t-1} has at least one correspondence in G_t.

A state is produced by adding a new correspondence to the parent state. Let ψ be a function that determine, for a given state/mapping p, a set of possible new correspondences that can be added to form new child states.

$$\psi(p) = \{\check{V}_{p,t-1} \times \check{V}_{p,t}\} \cup \{\check{V}_{p,t-1} \times V_{p,t}\} \cup \{V_{p,t-1} \times \check{V}_{p,t}\} \tag{4}$$

where $\check{V}_{p,t}$ is the set of nodes that are not currently mapped and are adjacent to the set of mapped nodes $V_{p,t}$, $\check{V}_{p,t} = \{\, v \mid (u,v) \in E_t, u \in V_{p,t}, v \notin V_{p,t} \,\}$.

4.4 Coarse Tracking

Let P be the graph edit operation sequence with the optimal GED; and $P(V_t)$ denote the set of nodes V_t mapped by the sequence P. Then, the "template" graph at time t is $G'_t = \{V'_t, E'_t, \alpha\}$, where

$$V'_t = P(V_t) \tag{5}$$

And E'_t is the set of edges between adjacent nodes in V'_t,

$$E'_t = \{\, (u,v) \mid (u,v) \in E_t;\; u,v \in V'_t \,\} \tag{6}$$

5 Optimization

A tree search algorithm in Section 4.3 that explores the whole state space is inefficient because the size of the state space is $|V'_{t-1}||V_t|$ and many states have no meaningful interpretation, e.g., mappings that produce unconnected dendrites.

In this work, the computation of GED is improved by using the A*-beamsearch [7] with node grouping, that only explores the most likely and meaningful states.

5.1 A* Search

The A* or the best-first search algorithm finds an optimal path from a state space based on a heuristic function, which estimates the expected costs of the best route from the root through the current state (partial solution) to a leaf state (complete solution). At each step during tree state space traversal, the most promising state — the one with the lowest heuristic cost — from the set of child states is chosen. Hence, at any state p, A* algorithm selects the path that minimizes the following cost:

$$f(p) = g(p) + h(p) \tag{7}$$

where $g(p)$ is the cost of the optimal path from the root to the current state p and $h(p)$ is the estimated cost from state p to a leaf state. [8]

Here, $h(n)$ is defined by the average cost of $g(p)$ times the number of remaining unmapped nodes.

5.2 Beam Search

In A* search, we may end up expanding all successor nodes in the search tree. In beam search, only a fixed number, b called *beam width*, of states in each layer are explored. We pick the b states with the lowest costs from eq. 7.

This means only those states with the most promising partial mappings are explored. Since a graph edit operation sequence between similar graphs has lower cost than the one between dissimilar graphs [7].

6 Cost Function

The cost function, $c(u \to v)$, of the GED $\delta(G'_{t-1}, G_t)$ takes a correspondence and returns a single value measuring its similarity based on the tissue deformation, local image feature, neuronal morphology, and local characteristics of neurons.

Edge comparison is not meaningful here since edge captures the relationship between nodes but the correspondent nodes are not guaranteed to be the same. However, we still need the neighborhood information that edges have. Thus, we aggregate the edge descriptors and embed them in the node descriptor. Next, we describe the cost function between two nodes (one-to-one function) and between a node and a group of nodes (many-to-one/one-to-many functions)

6.1 Node to Node

Average intensity of any node $v_i \in V_t$ is defined by,

$$I_t(v_i) = \frac{1}{|S(v_i)|} \sum_{x \in S(v_i)} I_t(x) \tag{8}$$

where $S(v_i)$ is the set of pixels in the superpixel corresponded to node v_i.

Average eigenvector from Frangi filter [4],

$$F(v_i) = \frac{1}{|S(v_i)|} \sum_{x \in S(v_i)} \hat{\mathbf{u}}_{x,1} \tag{9}$$

where Let $\lambda_{s,x,k}$ denote the eigenvalue corresponding to the k-th normalized eigenvector $\hat{\mathbf{u}}_{s,x,k}$ of the Hessian matrix of the image $\mathcal{H}_{x,k}$ all computed at scale s on pixel x.

$$\lambda_{x,k} = \max_{s_{min} \leq s \leq s_{max}} \lambda_{s,x,k} \tag{10}$$

will be the eigenvalue with the k-th smallest magnitude ($\lambda_{x,1} < \lambda_{x,2}$) and $\hat{\mathbf{u}}_{x,k}$ are their corresponding eigenvectors.

To quantify the **relation to neighbors**, first compute the location L_{ij} and magnitude M_{ij} based on the average intensity similarity for each neighbor v_j of v_i, $(v_i, v_j) \in E_t$,

$$\begin{aligned} L_{ij} &= C(v_j) - C(v_i) \\ M_{ij} &= 1 - \|I_t(v_j) - I_t(v_i)\| \end{aligned} \tag{11}$$

where $C(v_i)$ is the centroid of the superpixel corresponded to node v_i denoted by the xy-coordinate, (x_i, y_i). This approximate the geodesic distance between two nodes since their corresponding superpixel contains pixels with similar intensity.

Then, a histogram is created to encode the relation to neighboring superpixels, similar to the orientation histogram in SIFT [5]. In this histogram, the 360 degrees of L_{ij} direction are broken into 8 bins (each 45 degrees). The amount added to the bin depends on the magnitude projected in the bin's direction.

To achieve rotation independence, make the orientation relative to the keypoint's orientation by subtracting the keypoint's rotation from each orientation. The keypoint is the bin with the highest magnitude.

To achieve illumination dependence, threshold the magnitudes that are too big.

Similarly, **relation to mapping** is a histogram of mapped neighbors' orientation weighted by the inverse of distance D_{ij}, where $v_j \in V_{p,t}$.

$$D_{ij} = \frac{1}{\|L_{ij}\|} \tag{12}$$

Smooth deformation keeps the change in deformation low over space. For $u_i \in \check{V}_{p,t-1}$ and $v_i \in \check{V}_{p,t}$, the deformation of the correspondence i is defined as,

$$t_i = C(u_i) - C(v_i) \tag{13}$$

For $(u_i, u_j) \in E'_{t-1}$ and $(v_i, v_j) \in E_t$, where u_i and v_i are correspondence we are considering; $u_j \in V_{p,t-1}$ and $v_j \in V_{p,t}$ are the adjacent correspondences we found earlier. Then, the change in deformation at correspondence i is:

$$T_i = \sum_{t_j} \|t_j - t_i\| \tag{14}$$

Hence, the labeling function $\alpha(v_i)$ is defined as the tuple of 19 numbers: average intensity $I_t(v_i)$, the orientation of the average eigenvector $F(v_i)$ in radian, smooth deformation T_i, and the values in both histograms' bins.

The cost of substituting node $(u, v) \in \{\check{V}_{p,t-1} \times \check{V}_{p,t}\}$ is defined based on cosine similarity as follows:

$$c(u \to v) = 1 - \frac{\sum_{i=1}^{N} \alpha_i(u)\alpha_i(v)}{\sqrt{\sum_{i=1}^{N} \alpha_i^2(u)}\sqrt{\sum_{i=1}^{N} \alpha_i^2(v)}} \tag{15}$$

6.2 Node to Group of Nodes

The node grouping is needed for finding many-to-many relationship between frames in an efficient way. Node grouping similar to [6].

The cost of substituting node $(u, v) \in \{\check{V}_{p,t-1} \times V_{p,t}\}$ is defined based on cosine similarity as before but the labeling function of u must take into account other correspondences of v.

Let $\hat{u}$ is the set of nodes correspondence to v, $\hat{u} = \{\, w \mid (w \to v) \in p \,\} \cup \{u\}$. Then, the labeling function $\alpha(\hat{u})$ is also defined for a group of nodes. **Average intensity** and **average eigenvector** of a group of nodes are defined as,

$$\begin{aligned} I_t(\hat{u}) &= \frac{1}{|S(\hat{u})|} \sum_{x \in S(\hat{u})} I_t(x) \\ F(\hat{u}) &= \frac{1}{|S(\hat{u})|} \sum_{x \in S(\hat{u})} \hat{\mathbf{u}}_{x,1} \end{aligned} \tag{16}$$

The **relation to neighbors** and **relation to mapping** are quantified from the neighbors of the group of nodes, ignoring the internal edges. Using the centroid of the group of nodes and edges between a groupd of nodes and other.

The cost of substituting node $(u, v) \in \{V_{p,t-1} \times \check{V}_{p,t}\}$ follows the same suit.

7 Fine Tracking: Aligning Dendrites

Consider a set of nodes, $V_d \subset V_{\mathcal{G}}$, that constitutes a dendrite/branch d in the neuron morphology. The corresponding nodes in the "template" graph at time $t = 0$ is

$$V'_{d,0} = \{\, v \mid v \text{ overlap with } V_d, v \in V_0 \,\} \tag{17}$$

While, the corresponding nodes in the "template" graph at time t is

$$V'_{d,t} = P(V_{d,t-1}) \tag{18}$$

The alignment between dendrite at consecutive time step is done by registering patches from $V'_{d,t}$ and $V_{d,t-1}$ using techniques like free-form deformation (FFD).

References

[1] X. Gao, B. Xiao, D. Tao, and X. Li, "A survey of graph edit distance," *Pattern Analysis and applications*, vol. 13, no. 1, pp. 113–129, 2010.

[2] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 1356–1363, 2015.

[3] J. Chen, Z. Li, and B. Huang, "Linear spectral clustering superpixel," *IEEE Transactions on Image Processing*, vol. 26, no. 7, pp. 3317–3330, 2017.

[4] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *International conference on medical image computing and computer-assisted intervention*, pp. 130–137, Springer, 1998.

[5] D. G. Lowe *et al.*, "Object recognition from local scale-invariant features.," *ICCV*, vol. 99, no. 2, pp. 1150–1157, 1999.

[6] P. Morrison and J. J. Zou, "Inexact graph matching using a hierarchy of matching processes," *Computational Visual Media*, vol. 1, no. 4, pp. 291–307, 2015.

[7] M. Neuhaus, K. Riesen, and H. Bunke, "Fast suboptimal algorithms for the computation of graph edit distance," in *Joint IAPR International Workshops on Statistical Techniques in Pattern Recognition (SPR) and Structural and Syntactic Pattern Recognition (SSPR)*, pp. 163–172, Springer, 2006.

[8] P. E. Hart, N. J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE transactions on Systems Science and Cybernetics*, vol. 4, no. 2, pp. 100–107, 1968.