INEXACT GRAPH MATCHING FOR NEURON TRACKING IN CALCIUM IMAGING

K. V. Busum[1], *W. D. Tracy*[2], *L. He*[2], *S. Gulyanon*[3], *and G. Tsechpenakis*[1]

[1]Computer and Information Science Department, Indiana University-Purdue University Indianapolis, USA;
[2] Department of Biology, Indiana University, USA;
[3]Data Science and Innovation Program, Faculty of Science and Technology, Thammasat University, Thailand;

ABSTRACT

Calcium imagery captures both the structure and functionality of neurons. Tracking neurons on such images gives us a glimpse of how the locomotion in Drosophila larvae is executed and gain insights into the relationship between the two aspects of neurons. This problem is challenging due to (a) only active neurons are visible in calcium images, and (b) severe deformation from peristaltic locomotion. We propose the novel neuron tracking method based on the inexact graph matching over a set of graphs, each representing the neuron at one timestep. Computed using A*-beamsearch, the optimal match minimizes the graph edit distance. It incorporates the tissue deformation, local image feature, neuronal morphology, and local characteristics of neurons. Unlike existing methods, ours has no hard shape constraints so it requires no meticulous conditions for each type of neurons. Our results show the performance of our method and compare it with existing frameworks using sample calcium volumes of larval Drosophila sensory neurons.

Index Terms— neuron tracking, calcium images, inexact graph matching, graph edit distance, A*-beamsearch

1. INTRODUCTION

High speed confocal imaging of calcium dynamics in proprioceptive neurons of larval Drosophila allows us to observe both the morphology and neuronal activity simultaneously. These sensory neurons lie directly beneath the cuticle and epidermis, which are optically transparent, so they are observable *in vivo*. The responses of calcium images are proportional to neuronal activities on neurites. By measuring displacements of proprioceptive sensory dendrites and changes in activity levels over neurites, we are able to relate the direction of locomotion to these two properties and gain insights into the underlying molecular mechanisms that determine the responses of the proprioceptive cells [1].

Neuron tracking *in vivo* is a challenging problem by itself due to the crawling locomotion of Drosophila larvae. It produces a movement pattern that alternates between stationary poses and severe deformations. Second issue is related to the unique characteristic of calcium images, where low levels of neuron activation yield low intensity. This gives difficulty in distinguishing between the neuron volume and background because of low contrast. It also introduces changes in intensity during the locomotion and inhomogeneous responses over dendrites from cellular mechanism. Other issues are noisy calcium image sequences and low resolution along the Z-axis owing to imaging system limitations.

Conventional tracking approaches, i.e., tracking-by-detection [2, 3] and optical flow-based [4] methods, fail to detect and track neurites under spatially inhomogeneous intensity ambiguities of calcium images and wave-like movement patterns of Drosophila larvae because these methods assume temporal intensity smoothness constraint [5]. Previous approaches address these problems by enforcing hard shape constraints, e.g., the proximity between dendrites, to avoid invalid tracking results through techniques like deformable models. For example, the method in [5] tracks parts of the articulated body that represents the neurite centerlines, while authors in [6] use the accordion model, a variant of an articulated surface, to capture the tissue deformation of the surface to which the neurons are attached. The hard constraints for topology produce robust tracking results that are visually pleasing at the cost of poor accuracy under abrupt deformations.

In this paper, we tackle the neuron tracking problem over calcium images as the *inexact graph matching* (IGM) problem, where the segmentation of neuron in each timestep is represented by a graph. IGM finds the mapping of nodes between two graphs of the same neuron at consecutive timesteps by optimizing a certain affinity criterion, the customized *graph edit distance* (GED). It takes into account the tissue deformation, local image feature, neuronal morphology, and local characteristics of neurons. This formulation utilizes the soft shape constraints embedded in the GED instead of the hard shape constraints so the rigorous topology conditions are no longer needed. Techniques for GED computation are reviewed in [7, 8] and A*-beamsearch is suitable for our task since it is the simple but effective optimization algorithm that can accommodate many-to-many relationship [9, 10].

This work is supported by NSF/DBI [#1252597]: 'CAREER: Modeling the structure and dynamics of neuronal circuits in the *Drosophila* larvae using image analytics' awarded to G. Tsechpenakis.

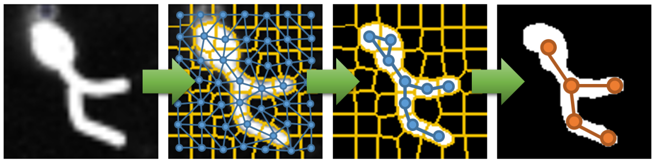

Fig. 1. From left to right. Input image, the "frame" graph, the "template" graph, and the aligned neuron trace. Yellow lines show the boundary of superpixels. Blue color shows the "frame" graph and "template" subgraph. Orange color shows the trace result.

2. METHOD

Let $\mathcal{I} = \{\mathcal{I}^{(t)}\}$ denote the calcium image stack sequence, where $t = 0, 1, \ldots, T$. To make the problem feasible, we require the initial morphology of the neuron, $\mathcal{G} = \{V_\mathcal{G}, E_\mathcal{G}\}$. A *graph* is denoted by $G = \{V, E\}$, where V is the set of nodes and $E \subset V \times V$ is the set of edges of graph G. Every frame is represented by the "frame" graph, where nodes are the non-overlapping regions in the image. The segmentation of the neuron is represented by the "template" graph, the subgraph of "frame" graph depicting the neuron.

Let G'_{t-1} and G'_t are the "template" graphs of the same neuron at consecutive timesteps. We formulate the tracking problem as finding G'_t and a sequence of graph edit operations, P, that transforms G'_{t-1} into G'_t, with the minimum GED, $\delta(G'_{t-1}, G'_t)$:

$$\delta(G'_{t-1}, G'_t) = \min_{P} \sum_{(u \to v) \in P} c(u \to v) \qquad (1)$$

where nodes $u \in V'_{t-1}$ and $v \in V'_t$ are a correspondence; and $c(u \to v)$ is the cost of the graph edit operation that maps node u to v, which depends on the affinity between u and v described in Sec. 4. Finally, we recover the trace in every frame by aligning dendrites of the initial morphology, $\mathcal{G}$, to the binary image of neuron (fig. 1).

2.1. Creating "Frame" Graph

The "frame" graph, $G_t = \{V_t, E_t\}$, of the image stack at time t is computed by the superpixel segmentation [11, 12]. 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. The set of nodes, V_t, is the superpixels over I_t, while the set of edges, E_t, are computed using Delaunay triangulation over the centroids of superpixels in V_t.

2.2. Initial "Template" Graph

A trace describes the neuron morphology at the structural level, while an image stack describes the neuron at the appearance level. We propose the "template" graph that bridges between two levels of representations.

The "template" graph at time $t = 0$, $G'_0 = \{V'_0, E'_0\}$, is the subgraph of G_0 corresponded to the initial morphology $\mathcal{G}$. $V'_0 = \{\, v \mid v \text{ overlaps } \mathcal{G}, v \in V_0 \,\}$ is the set of node depicting the initial morphology $\mathcal{G}$. E'_0 is the set of edges between adjacent nodes in V'_0, $E'_0 = \{\, (u, v) \mid (u, v) \in E_0;\ u, v \in V'_0 \,\}$.

2.3. Tree Search Algorithm

A tree (state space) search algorithm [9] 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, $\delta(G'_{t-1}, G_t)$. Since the IGM is error tolerant, nodes of the graph do not need to have a correspondence in another graph and the match do not need to be exactly the same. Hence, this computes both G'_t and P simultaneously as G'_t is the subgraph of G_t that has the correspondence in G'_{t-1} with respect to P and P describes the steps to transform from G'_{t-1} to both G_t and G'_t.

The state space consists of a number of possible mappings with different cost (fig. 2). 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 determines, for a given state/mapping p, a set of possible new correspondences that can be added to form new child states.

$$\psi(p) = \{\check{V}_{p,t-1} \times \check{V}_{p,t}\} \cup \{\check{V}_{p,t-1} \times V_{p,t}\} \cup \{V_{p,t-1} \times \check{V}_{p,t}\} \quad (2)$$

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

2.4. Fine Tracking: Aligning Dendrites

Consider a set of nodes, $V_d \subset V_\mathcal{G}$, that constitutes a dendrite d in the neuron trace. The nodes corresponding to the dendrite d in the "template" graph at time $t = 0$ is $V'_{d,0} = \{\, v \mid v \text{ overlap with } V_d, v \in V_0 \,\}$. For time t, $V'_{d,t} = P(V'_{d,t-1})$. The alignment between dendrite at consecutive timesteps is done by registering patches from $V'_{d,t-1}$ to $V'_{d,t}$ using techniques like free-form deformation (FFD) [**?**].

3. OPTIMIZATION

A tree search algorithm in Sec. 2.3 must be efficient to make this method feasible since there are $|V'_{t-1}||V_t|$ possible correspondences. Fortunately, many states/mappings have no meaningful interpretation, e.g., mappings that produce unconnected dendrites, and neurons usually occupy small regions in the image stacks. Hence, there are a limited number of candidate mappings to need to be explored.

In this work, the computation of GED uses the A*-beamsearch algorithm [10] with node grouping, that only explores the most likely and meaningful states and allows many-to-many mapping solution.

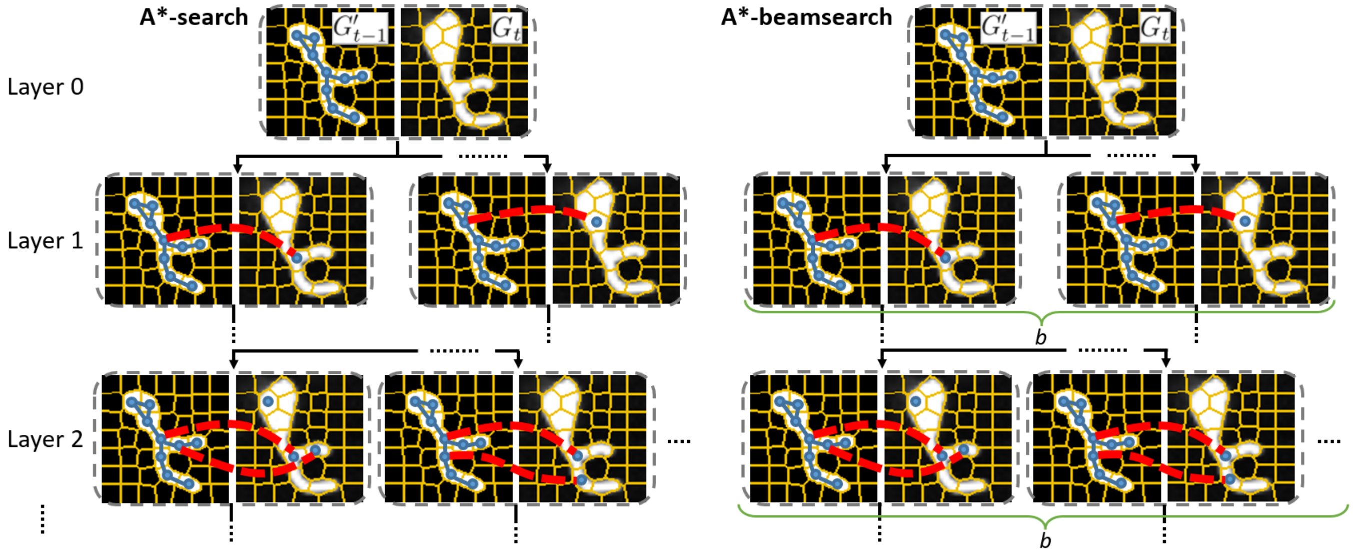

Fig. 2. Tree state space search using A*-search (left) and A*-beamsearch (right). The root of the state space represents the mapping with no correspondences. The first layer of states represents mappings with one correspondence, and so on. Red lines show pairs of nodes that are a correspondence. In A*-beamsearch, only a b number of states with the lowest heuristic costs in each layer are explored.

3.1. A* Search

The A* or the best-first search algorithm [14] 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 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:

$$f(p) = g(p) + h(p) \tag{3}$$

where $g(p)$ is the cost of the optimal path from the root to the current p and $h(p)$ is the estimated cost from state p to a leaf state. Here, $h(p)$ is defined by the average cost of $g(p)$ times the number of remaining unmapped nodes.

3.2. Beam Search

A* search may end up expanding all successor nodes in the search space. 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. 3 (fig. 2), so only states with the most promising partial mappings are explored. Hence, the cost function must reflects the graph similarity, the more similar the graphs are the lower the cost is [10].

4. COST FUNCTION

The cost function, $c(u \to v)$, of the state p takes a correspondence and returns a single value measuring the compatibility between node u and v 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 information and embed them in the node descriptor.

The match is inexact so a node may have multiple correspondences. Thus, we must define 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).

4.1. Node to Node

The first part of the node descriptor is the image feature considered through intensity and local orientation. The **average intensity** of node $v_i \in V_t$ is defined by,

$$I_t(v_i) = \frac{1}{|S(v_i)|} \sum_{x \in S(v_i)} I_t(x) \tag{4}$$

where $S(v_i)$ is the set of pixels in the superpixel corresponded to node v_i. The **average eigenvector** of node $v_i \in V_t$ indicates the local neurite orientation via eigenvalue analysis of the Hessian matrix [15],

$$F(v_i) = \frac{1}{|S(v_i)|} \sum_{x \in S(v_i)} \hat{\mathbf{u}}_{x,1} \tag{5}$$

where $\hat{\mathbf{u}}_{x,1}$ is the eigenvector with the smallest eigenvalue at pixel x. To include the local characteristics, the **relation to neighbors** measures the intensity similarity between a node and its neighborhood. To compute, first find the location L_{ij} and magnitude M_{ij} based on the average intensity similarity for each neighbor v_j of v_i, $(v_i, v_j) \in E_t$,

$$\begin{aligned} L_{ij} &= C(v_j) - C(v_i) \\ M_{ij} &= 1 - \|I_t(v_j) - I_t(v_i)\| \end{aligned} \tag{6}$$

where $C(v_i)$ is the centroid of the superpixel corresponded to node v_i denoted by the xy-coordinate, (x_i, y_i).

Then, a histogram is created to encode the relation to neighboring superpixels, similar to the orientation histogram in SIFT [16]. 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** indicates the proximity between v_i and mapped nodes. This helps distinguish nodes along ambiguous regions like dendrites since points along the dendritic line segment have similar properties. It is a histogram of mapped neighbors' orientation weighted by the inverse of distance H_{ij}, where $v_j \in V_{p,t}$.

$$H_{ij} = \frac{1}{\|L_{ij}\|} \tag{7}$$

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

$$d_i = C(u_i^{t-1}) - C(v_i^t) \tag{8}$$

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

$$D_i = \sum_{d_j} \|d_j - d_i\| \tag{9}$$

Hence, the node descriptor $\alpha(v_i)$ 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:

$$c(u \to v) = 1 - \frac{\sum_{i=1}^{N} \alpha_i(u)\alpha_i(v)}{\sqrt{\sum_{i=1}^{N} \alpha_i^2(u)}\sqrt{\sum_{i=1}^{N} \alpha_i^2(v)}} \tag{10}$$

4.2. Node to Group of Nodes

The node grouping is needed for finding many-to-many relationship between frames in an efficient way, similar to [9].

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 descriptor 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, $\alpha(\hat{u})$ denotes the group-of-node descriptor. **Average intensity** and **average eigenvector** of a group of nodes are defined as,

$$\begin{aligned} I_t(\hat{u}) &= \frac{1}{|S(\hat{u})|} \sum_{x \in S(\hat{u})} I_t(x) \\ F(\hat{u}) &= \frac{1}{|S(\hat{u})|} \sum_{x \in S(\hat{u})} \hat{\mathbf{u}}_{x,1} \end{aligned} \tag{11}$$

The **relation to neighbors**, **relation to mapping**, and **smooth deformation** are quantified from the neighbors of the group of nodes, ignoring the internal edges within the group; while the centroid is defined as the middle point of the group.

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

5. RESULTS

We validated our method using time-lapse calcium volumes of larval Drosophila dda-E and dda-D sensory neurons. Larvae expressing the genetically encoded calcium sensor GCaMP6.0F in the proprioceptors were observed in 4-D confocal microscopy imagery [1].

6. CONCLUSION

7. REFERENCES

[1] L. He, S. Gulyanon, M.M. Skanata, D. Karagyozov, E.S. Heckscher, M. Krieg, G. Tsechpenakis, M. Gershow, and W.D. Tracey, "Direction selectivity in drosophila proprioceptors requires the mechanosensory channel Tmc," *Current Biology*, 29(6):945–956.e3, 2019.

[2] S. Gulyanon, N. Sharifai, M. D. Kim, A. Chiba, and G. Tsechpenakis, "CRF formulation of active contour population for efficient three-dimensional neurite tracing," *in proc. ISBI*, pp. 593–597, 2016.

[3] P. Głowacki, M. A. Pinheiro, A. Mosinska, E. Türetken, D. Lebrecht, R. Sznitman, A. Holtmaat, J. Kybic, and P. Fua, "Reconstructing evolving tree structures in time lapse sequences by enforcing time-consistency," *TPAMI*, 40(3):755–761, 2018.

[4] Q. Chen and V. Koltun, "Full flow: Optical flow estimation by global optimization over regular grids," *in proc. CVPR*, pp. 4706–4714, 2016.

[5] S. Gulyanon, L. He, W.D. Tracey, and G. Tsechpenakis, "Neurite tracking in time-lapse calcium images using MRF-modeled pictorial structures," *in proc. ISBI*, pp. 1564–1568, 2018.

[6] S. Gulyanon, L. He, W.D. Tracey, and G. Tsechpenakis, "Neuron tracking in calcium image stacks using accordion articulations," *in proc. ISBI*, 2019.

[7] J. Yan, X.-C. Yin, W. Lin, C. Deng, H. Zha, and X. Yang, "A short survey of recent advances in graph matching," *in proc. ICMR*, pp. 167–174, 2016.

[8] X. Gao, B. Xiao, D. Tao, and X. Li, "A survey of graph edit distance," *Pattern Anal Appl*, 13(1):113–129, 2010.

[9] P. Morrison and J.J. Zou, "Inexact graph matching using a hierarchy of matching processes," *Computational Visual Media*, 1(4):291–307, 2015.

[10] M. Neuhaus, K. Riesen, and H. Bunke, "Fast suboptimal algorithms for the computation of graph edit distance," *S+SSPR*, pp. 163–172, 2006.

[11] Z. Li and J. Chen, "Superpixel segmentation using linear spectral clustering," *in proc. CVPR*, pp. 1356–1363, 2015.

[12] J. Chen, Z. Li, and B. Huang, "Linear spectral clustering superpixel," *IEEE Trans Image Process*, 26(7):3317–3330, 2017.

[13] S. Farhand, S. Gulyanon, N. Sharifai, M. D. Kim, A. Chiba, and G. Tsechpenakis, "Temporal neurite registration using hierarchical alignments," *in proc. ISBI*, pp. 334–338, 2016.

[14] P.E. Hart, N.J. Nilsson, and B. Raphael, "A formal basis for the heuristic determination of minimum cost paths," *IEEE Trans Syst Sci Cybern*, 4(2):100–107, 1968.

[15] A.F. Frangi, W.J. Niessen, K.L. Vincken, and M.A. Viergever, "Multiscale vessel enhancement filtering," *in proc. MICCAI*, pp. 130–137, 1998.

[16] D.G. Lowe, "Object recognition from local scale-invariant features," *in proc. ICCV*, vol. 2, pp. 1150–1157, 1999.