

Tracking Neuron Reconstruction in Calcium Imaging using Markov Random Field

Sarun Gulyanon

April 30, 2017

1 Introduction

Understanding the relationship between neuron activities and movements of an organism gives us insight into neurons' functional mechanism from the application point of view. The calcium imaging reveals the spatio-temporal information of activities in neurons at the single-cell level. However, the amount of data is massive so the automated analysis system is required. The first step towards this goal is to track the location of individual neurons, including dendrites and soma, from time-lapse calcium images. However, one unique characteristic of calcium images is that they produce low responses when there are no activities on neurons so parts of dendrites become invisible. This property makes previous tracking systems failed because they will produce errors in those frames when dendrites disappeared and the error will propagate to the rest of the image sequence.

Tracking and detection have been one of the most studied topic in computer vision and many techniques have been proposed.

One way to solve the problem based solely on detection is to automatically trace neurons in each frame independently using neuron reconstruction methods like in [1, 2]. Then, the registration technique (e.g., [3]) can be applied to improve the temporal smoothness, increase robustness to noises, and obtain the trajectory of the neuron. Ones can even extend this by applying both registration and tracing simultaneously using method in [4]. However, calcium image sequences contain frames with low intensity so these techniques will fail to track neurons because they are unable to detect neurons in those frames and the error will propagate to following frames.

Another way is to use the optical flow computed by methods such as [5, 6]. These methods produce dense motion field over the entire image. New optical flow methods are capable to handle occlusions and large displacement motions by incorporating the feature matching technique [6]. Most optical flow methods follow the brightness constancy constraint equation [7], which assumes that pixel intensity remains constant during the displacement. However, this assumption is not true in the calcium imaging whose response changes with the activities on neurons. When the intensity changes, the optical flow starts tracking the

wrong dendrites or noises instead. Moreover, tracking over background regions, which occupied most of the area, is meaningless and distracting because images are noisy so optical flow will calculate the motion field of noises and regularized over the neurite regions.

We are interested only the location of neuron so rather than computing the motion field of the entire image, we only need the motion field of the neuron. Instead of relying solely on image appearance for tracking, the alternative is to integrate the domain knowledge on the neuron tree-structure shape and the object’s motion like the kinematic structure of articulated body such as methods in [8, 9]. It represented the human body by the loose-limbed model, which is the variation of the pictorial structure [10, 11] with elastic connection between adjacent parts. It formulated the graphical model where parts are sites and their configuration contains the position and orientation of parts. Edges encode the position and angle relationships between adjacent parts. The benefits of using the model like articulated body allow us to detect and track only the regions of interests as well as estimating the structure or the pose of the object, which is also the information we are interested in as it has various applications.

The previous works on pose estimation and tracking articulated body are not effective on neuron tracking because neuron images have different characteristics compared to human body or parts. First, the neuron has higher degrees of freedom compared to human body model because it has higher number of branch points or joints. Second, the number of degrees of freedom changes across neurons while it is fixed in human body. The main issue in neuron tracking is the massive search space due to larger degrees of freedom and unreliable image cues compared to self-occlusion and appearance variation due to clothing in human-body tracking so these previous techniques are infeasible or inappropriate for neuron tracking.

To address these problems, our method is based on the generalized pictorial structure that can be applied to any tree structure. We tackle the unreliability of image intensity by incorporating both image appearance and domain knowledge, including the shape and motion models. Here we describe 5 factors that drive our tracking system - a) image feature, b) dynamics model, c) transition similarity, d) part-based model, and e) shape smoothness. Each factor addresses the issue that occurs in neuron tracking. No frameworks in our knowledge have ever incorporate these number of factors before. They adopted only the partial of factors so our method prevails because they did not handle all issues like we do.

2 Method

Let the centerline of a neuron in each frame is represented by the pictorial structure model [10, 11], $\mathcal{G}^t = \{\mathbf{X}^t, \mathcal{E}_R\}$ (fig. 1). Let $\mathbf{X}^t = \{\mathbf{x}_i^t\}$ be a set of tree structure nodes $i \in \{1, \dots, n\}$ at time $t \in \{0, \dots, T\}$, and $\mathcal{E}_R = \{e_{ij}\}$ be a set of tree structure edges. Each node contains the Cartesian coordinate of the centerline, $\mathbf{x}_i^t = \{x, y\}$, where $x, y \in \mathbb{R}$. Let $K_n = \{\mathcal{V}, \mathcal{E}\}$ be the complete graph with n vertices and \mathcal{E}_G be a set of repulsive edges [12], which are not tree

structure edges, $\mathcal{E}_G = \mathcal{E} \setminus \mathcal{E}_R$.

Let $\mathbf{V}^t = \{\mathbf{v}_i^t\}$ be a set of mapping vectors that transforms \mathbf{X}^{t-1} to \mathbf{X}^t , so $\mathbf{v}_i^t = \mathbf{x}_i^t - \mathbf{x}_i^{t-1}$. The objective function is formulated as the conditional distribution over the set of neuron's shape $\mathbf{S}^t = \{\mathbf{X}^t, \mathcal{E}\}$ for each frame, given the image sequence $\mathbf{I} = \{\mathbf{I}^t\}$ (fig. 1),

$$P(\mathbf{S}^t|\mathbf{I}) \propto P(\mathbf{X}^t|\mathbf{I}, \mathbf{S}^{t-1}, \mathbf{S}^0) \quad (1)$$

where \mathbf{S}^0 is the neuron's shape initialization. The trace of the neuron in frame t is tracked by solving the following optimization problem:

$$\hat{\mathbf{S}}_{MAP}^t = \arg \max_{\mathbf{X}^t} P(\mathbf{X}^t|\mathbf{I}, \mathbf{S}^{t-1}, \mathbf{S}^0) \quad (2)$$

Here we assume that the topology of the neuron \mathcal{E}_R does not change between frames, which means \mathcal{G}^t is isomorphic for every time t and only the local shape of the neuron deforms. So we optimize the objective function by modifying the position of the centerline \mathbf{X}^t . We modeled the probabilistic distribution of the centerline \mathbf{X}^t using the second-order multi-label MRF, where sites are nodes $\mathbf{x}_i^t \in \mathbf{X}^t$ and their configuration is every possible position.

$$P(\mathbf{X}^t|\mathbf{I}, \mathbf{S}^{t-1}, \mathbf{S}^0) = \frac{1}{Z} \exp \left\{ - \sum_{\mathbf{x}_i^t \in \mathbf{X}^t} E(\mathbf{x}_i^t|\mathbf{I}, \mathbf{S}^{t-1}, \mathbf{S}^0) \right\} \quad (3)$$

where Z is the partition function.

$$\begin{aligned} E(\mathbf{x}_i^t|\mathbf{I}, \mathbf{S}^{t-1}, \mathbf{S}^0) &= A(\mathbf{x}_i^t|\mathbf{I}, \mathbf{S}^0) + \sum_{e_{ij} \in \mathcal{E}} I(\mathbf{x}_i^t, \mathbf{x}_j^t|\mathbf{S}^{t-1}) \\ &\quad + \sum_{e_{ip}, e_{iq} \in \mathcal{E}_R, p \neq q} H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t) \end{aligned} \quad (4)$$

where e_{ij} is the edge incident to the node \mathbf{x}_i^t . A is the unary potential function (sec. 2.1), I the pairwise smoothness (sec. 2.2), and H is the tertiary smoothness. Section 2.3 will show that the careful choice of the tertiary term allows us to decompose it into a pairwise functions without adding an auxiliary node; therefore, the graphical model can be described by the graph $\mathbf{S}^t = \{\mathbf{X}^t, \mathcal{E}\}$.

2.1 Unary Potential

The data penalty term or unary potential assigns the position of the centerline based on the image appearance and the initial neuron's shape. It composes of image feature and dynamics model terms.

$$A(\mathbf{x}_i^t|\mathbf{I}, \mathbf{S}^0) = A_{img}(\mathbf{x}_i^t|\mathbf{I}) + A_{model}(\mathbf{x}_i^t|\mathbf{I}, \mathbf{S}^0) \quad (5)$$

Image feature assigns the position of node \mathbf{x}_i^t based on the response from the Frangi filter [13] for dendrites and the Hessian-based function for branch

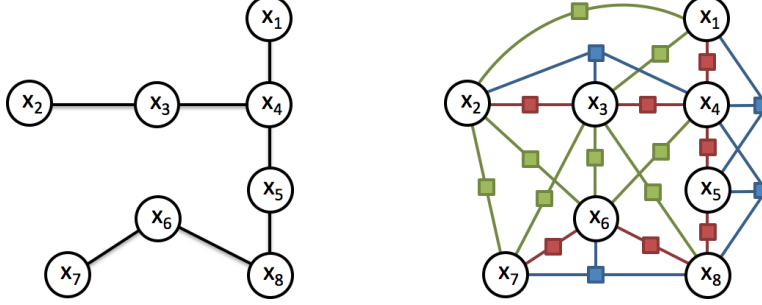


Figure 1: shows the tree structure or pictorial structure of the neuron (left) and our corresponding graphical model (right). Red cliques \mathcal{E}_R form the tree structure of neuron. Green cliques \mathcal{E}_G represent the repulsive edges, and blue cliques are the tertiary potential. Some of green and blue cliques were left out for the sake of visualization.

points. Frangi filter is a vesselness-filter designed to detect only the line-like structure, so a different function is required for the branch point (fig. 2). In 2D, the branch point appears like a blob so we design a function similar to Frangi filter that has high value only when both eigenvalues of Hessian matrix are high.

$$A_{img}(\mathbf{x}_i^t | \mathbf{I}) = \begin{cases} \exp\left(-\frac{R_B^2}{2\beta^2}\right) \left(1 - \exp\left(-\frac{S^2}{2c^2}\right)\right) & , \text{if } \deg(\mathbf{x}_i^t) \leq 2 \\ \left(1 - \exp\left(-\frac{R_A}{2\beta^2}\right)\right) \left(1 - \exp\left(-\frac{S^2}{2c^2}\right)\right) & , \text{otherwise} \end{cases} \quad (6)$$

where $\deg(\mathbf{x}_i^t)$ is the degree of the node \mathbf{x}_i^t , which is less than or equal to two if they are dendrite nodes, otherwise, they are branch points or joints between dendrites. $R_B = \lambda_1/\lambda_2$ is the ratio accounts for the deviation from a blob-like to a line-like structures, where λ_k is the eigenvalue of the Hessian matrix at position \mathbf{x}_i^t with the k-th smallest magnitude. $R_A = \sqrt{\min(\lambda_1, 0) \cdot \min(\lambda_2, 0)}$ measures the blob-like structure similarity for bright object over dark background, which is the branch point. For bright neurite over dark background, eigenvalues have negative values. λ_1 has low magnitude while λ_2 has high magnitude over lines. On the other hand, both eigenvalues have high magnitude at branch points. $S = \lambda_1^2 + \lambda_2^2$ distinguishes the neurite from the background since noises produce eigenvalues with low magnitude. β and c are parameters controlling the sensitivity of R_A , R_B , and S [13].

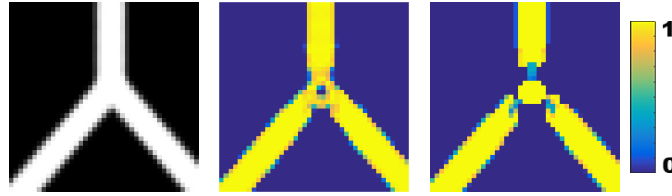


Figure 2: shows the synthetic image of a branch point (left) with its corresponding Frangi filter (middle) and our Hessian-based function (right) from eq. 6. Frangi filter gives low response at the center of the branch point while our Hessian-based function gives high response at the branch point and low value to surroundings.

Dynamics model assigns the position of node \mathbf{x}_i^t such that the node's position resembles to its position from our motion model, which is governed by the soma's location. Pairwise and tertiary potentials ensure that tracked neurons inherit shape characteristics of neurons, and they depend on the image feature to guide a neuron trace to the correct position. Image feature alone is not enough to locate dendrites because the whole dendrites can disappear in some frames of calcium images. The dynamics model solves this problem by introducing the motion model or the initial guess of the centerline location, which can be seen as the prior, to complement the image feature. Although dendrites are difficult to detect, on the other hand, the soma is very clear and easy to locate. In this work, we assume \mathbf{S}^0 is given, then our motion model is obtained by shifting the initial trace so that the locations of soma agree,

$$A_{model}(\mathbf{x}_i^t | \mathbf{I}, \mathbf{S}^0) = \exp(\alpha_d \cdot \|\mathbf{x}_i^t - T_{0,t}(\mathbf{x}_i^0)\|) \quad (7)$$

where $\|\cdot\|$ is the Euclidean distance. α_d is the parameter with negative value that tunes the sensitivity of the dynamics model. T_{t_1, t_2} is the translation operator that shifts the soma's position at time t_1 to its position at time t_2 , so $T_{t_1, t_2} = \sum_{t=t_1+1}^{t_2} \mathbf{v}_0^t$, where $\mathbf{v}_0^t = \mathbf{x}_0^t - \mathbf{x}_0^{t-1}$ and \mathbf{x}_0^t are the mapping vector and position of the soma at time t , respectively. The positions of the soma is located by the Circle Hough Transform technique [14, 15] and tracked along the optical flow computed by the FullFlow method [5].

2.2 Pairwise Potential

The pairwise smoothness assigns the position of node such that the mapping function is smooth and the transformed neuron maintains the global shape. It consists of the transition similarity and the part-based model energies. These two types of energies applied on two different types of edges, \mathcal{E}_R and \mathcal{E}_G . It is defined as,

$$I(\mathbf{x}_i^t, \mathbf{x}_j^t | \mathbf{S}^{t-1}) = \begin{cases} \alpha_p \cdot \exp(\alpha_t \cdot \|\mathbf{v}_i^t - \mathbf{v}_j^t\|) & , \text{if } e_{ij} \in \mathcal{E}_R \\ \left[\|\mathbf{x}_i^t - \mathbf{x}_j^t\| > \langle \|\mathbf{x}_k^t - \mathbf{x}_l^t\| \rangle_{e_{kl} \in \mathcal{E}_R} \right] & , \text{if } e_{ij} \in \mathcal{E}_G \text{ and} \\ & br(\mathbf{x}_i^t) \neq br(\mathbf{x}_j^t) \\ 0 & , \text{otherwise} \end{cases} \quad (8)$$

where $[\cdot]$ is the indicator function taking value 1 if a given predicate is true, otherwise, value 0. $\langle \cdot \rangle$ is the expectation value. $br(\mathbf{x}_i^t)$ is the branch number in which the node \mathbf{x}_i^t is. $\alpha_p = (2 \langle \|\mathbf{v}_i^t - \mathbf{v}_j^t\| \rangle)^{-1}$ is the parameter that ensures this term to scale properly between small and large vector difference [16], while α_t scales the sensitivity of the exponential term.

Transition similarity enforces the smoothness of mapping vectors between adjacent nodes. It is applied over tree structure edges $e_{ij} \in \mathcal{E}_R$. It is defined by the function of the mapping vector difference between adjacent nodes so it encourages adjacent nodes to have the same or similar mapping vector.



Figure 3: shows that although the curvature and our decomposition are different, they share the same optimum when $\overline{\mathbf{x}_p^t \mathbf{x}_q^t} = \overline{\mathbf{x}_p^t \mathbf{x}_i^t} + \overline{\mathbf{x}_i^t \mathbf{x}_q^t}$ (left). They discourage the dendrite from bending when $\overline{\mathbf{x}_p^t \mathbf{x}_q^t} < \overline{\mathbf{x}_p^t \mathbf{x}_i^t} + \overline{\mathbf{x}_i^t \mathbf{x}_q^t}$ (right).

Part-based model constrains the morphology of the neuron structure using the repulsive edges [12]. It makes dendrites to repel each other and prevents them from creating any unnecessary crossover and deforming the morphology of tracked neuron. Repulsive edges \mathcal{E}_G are added between nodes from different branches. They give higher energy when dendrites overlap than when they don't. In this work, it is computed by thresholding the distance between branches.

2.3 Tertiary Potential

The tertiary smoothness incorporates the neuron shape characteristics into the model. It is decomposed into pairwise terms so the efficient inference method can be applied.

Shape smoothness ensures the smoothness of dendrites by encouraging low curvature and short length like in the active contour model [17]. Computing curvature requires the tertiary potential,

$$H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t) = (\mathbf{x}_p^t - \mathbf{x}_i^t)^2 + (\mathbf{x}_p^t - 2\mathbf{x}_i^t + \mathbf{x}_q^t)^2 \quad (9)$$

The method in [18] decomposed the second derivative term into pairwise terms with an auxiliary node. We avoid adding auxiliary nodes by simplifying the curvature using the triangle inequality, where the clique of three nodes $\{\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t\}$ is decomposed into three pairwise cliques with no extra auxiliary nodes - $\{\mathbf{x}_p^t, \mathbf{x}_i^t\}$, $\{\mathbf{x}_i^t, \mathbf{x}_q^t\}$, and $\{\mathbf{x}_p^t, \mathbf{x}_q^t\}$. Consider a triangle whose corners are defined by nodes \mathbf{x}_p^t , \mathbf{x}_i^t , and \mathbf{x}_q^t . The optimized curvature value is obtained when the triangle forms the straight line, $\overline{\mathbf{x}_p^t \mathbf{x}_q^t} = \overline{\mathbf{x}_p^t \mathbf{x}_i^t} + \overline{\mathbf{x}_i^t \mathbf{x}_q^t}$, so the simplification has the same optimum as the potential in eq. 9 (fig. 3), which is defined as,

$$H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t) = \alpha_s \cdot \left\{ \|\mathbf{x}_p^t - \mathbf{x}_i^t\|^2 + \|\mathbf{x}_i^t - \mathbf{x}_q^t\|^2 - \frac{\|\mathbf{x}_p^t - \mathbf{x}_q^t\|^2}{4} \right\} \quad (10)$$

where α_s is the parameter with negative value that tunes the sensitivity of the shape smoothness. The quadratic terms encourage the short distance between nodes. This term encourages both rigidity and elasticity like the active contour model [17]. The third term of the decomposition is divided by 4 to ensures that $H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t) \leq 0$ (the proof in appendix A).

3 Implementation

The position of centerline \mathbf{X}^t is discretized so the problem becomes feasible. Algorithm 1 shows the summary of our method given the image sequence \mathbf{I} and

the neuron trace initialization \mathbf{S}^0 . In line 6, our energy function in eq. 4 is non-submodular so MRF is optimized using the α -expansion [19] with QPBO [20, 21], also known as fusion moves [22].

Algorithm 1 Neuron tracking

```

1: procedure TRACKING(  $\mathbf{I}, \mathbf{S}^0$  )
2:   for  $t = 1$  to  $T$  do
3:     Compute  $A(\mathbf{x}_i^t | \mathbf{I}, \mathbf{S}^0)$  using eq. 6 and 7
4:     Compute  $I(\mathbf{x}_i^t, \mathbf{x}_j^t | \mathbf{S}^{t-1})$  using eq. 8
5:     Compute  $H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t)$  using eq. 10
6:      $\mathbf{S}^t \leftarrow \arg \max_{\mathbf{X}^t} P(\mathbf{X}^t | \mathbf{I}, \mathbf{S}^{t-1}, \mathbf{S}^0)$ 
7:   return  $\mathbf{S}$ 

```

For efficiency, the search space is limited by reducing the number of candidate positions and pruning the redundant edges from the graphical model \mathbf{S}^t .

To limit the number of candidate positions, we ignore locations that are far away from the neuron and consider only the regions around the starting location (in our experiment, within 20 pixels), which indicates the possible position based on previous trace. In this work, the starting location of each node at time t is computed by applying the translation operator that shifts soma’s location from previous time-step to the current position, $\tilde{\mathbf{x}}_i^t = T_{t-1,t}(\mathbf{x}_i^{t-1})$. Then, the possible configuration of node \mathbf{x}_i^t is limited to regions around the starting location $\tilde{\mathbf{x}}_i^t$.

Redundant edges are repulsive edges linking nodes from the same branch or nodes that are far apart. They link nodes that are unlikely to overlap so removing these edges makes no difference to the model. To remove these edges, \mathbf{S}^t is pruned such that there is at most one edge between a node and a branch, where the shortest edge is selected. Other nodes are considered far apart and edges incident to those nodes are redundant. With this scheme, we limit the number of repulsive edges $|\mathcal{E}_G| = \mathcal{O}(N)$. The total the number of edges in our model is also $\mathcal{O}(N)$ since $|\mathcal{E}_R| = \mathcal{O}(N)$ due to the tree structure of the neuron and the number of tertiary cliques is limited to three per node if every bifurcation has only two children [3].

4 Experiments

In this section, we apply our method on time-lapse calcium images. In this dataset, each sequence captures a few motions of neurons and the duration of each motion is around 10-20 frames. The ground truth was produced by tracing neurons manually in some frames of the image sequences, especially when there are movements over neurons. The manual reconstructions of neurons were produced using the software called *neuTube* [23]. Three dissimilarity metrics were used in this work, which are the spatial distance (SD), substantial SD (SSD), and the percentage of sample points used in SSD (%SSD). $SD(A,B)$ is defined by the mean of the average Euclidean distance of nodes in A to the

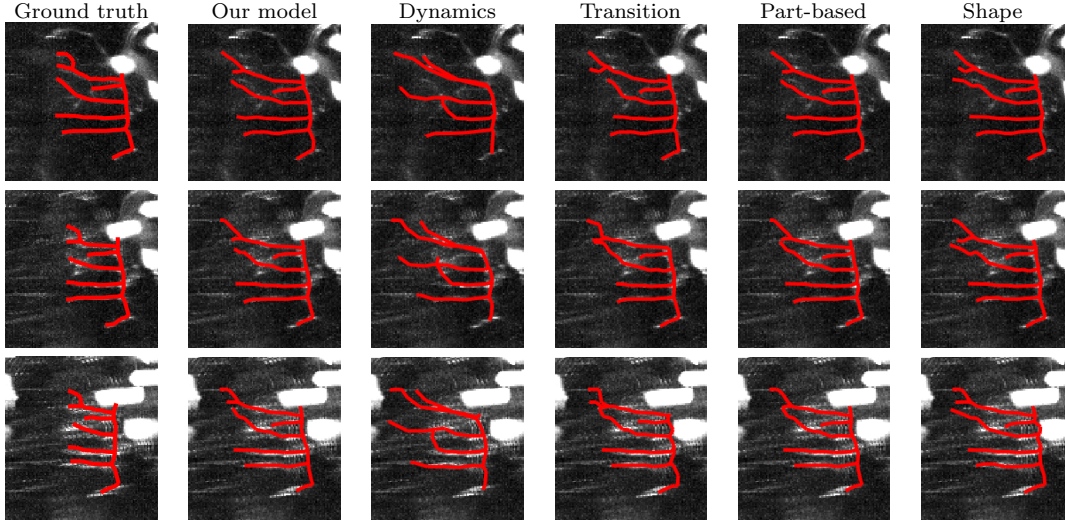


Figure 4: Tracking results (in red color) on 3 adjacent frames in each row. The first and second columns show the ground truth and our results, while the next four columns are our results excluding one of the factors - dynamics model, transition similarity, part-based model, and shape smoothness respectively.

closest point in B and the other way round. Nodes in A and B are resampled such that the distance between adjacent nodes is one voxel to remove the bias from trace node sampling. When most parts of neurons overlap, SD gives low values and offsets parts of the neuron that are totally out-of-place. To emphasize the magnitude of errors, SSD is defined as the SD over nodes that are apart from the other neuron at least 2 voxels. The last metric is the percentage of nodes that are apart from the other neuron. It indicates the inconsistency between two neurons. The detailed definition of these criteria is given in [24]. We compared our method with the optical-flow-based neuron tracking computed by the FullFlow method [5]. The experiment was carried out on Mac Pro (2×2.66 GHz 6-Core Intel Xeon, 20GB 1333MHz DDR3 ECC); the run time of our method was around one second per frame per neuron. The dimension of each frame is around 512×192 pixel. Tracking was done on 2D maximum projection of 3D images, albeit the availability of image stack sequences, to eliminate the color bleach effect that is more likely to occur over Z-axis due to image acquisition process, where image slices were taken in sequence.

Fig. 4 shows the importance and effects of each factor, we display the tracking results when all factors are integrated and when one of the factors is excluded. Dynamics model is required to compensate for the unreliability of image intensity. Without this factor, nodes move towards the pixel with high intensity regardless of neuron’s shape. Transition similarity prevents rapid changes in adjacent nodes resulting in smooth mapping. Part-based model ensures dendrites avoid overlapping. Finally, shape smoothness smooths out the dendrites; otherwise, crooked curves are detected due to dip in intensity over neurites.

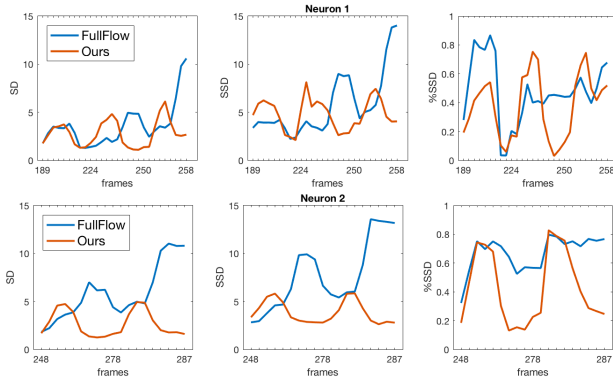


Figure 5: shows three dissimilarity scores - SD, SSD, and %SSD, of FullFlow [5] and our method. Rows display the tracking results of two different neurons. The first row shows the easy case where the SNR is high. The second row shows the difficult case where the SNR is low.

We evaluate our method and competitor using the dissimilarity criteria on image sequences with low and high signal-to-noise ratio (SNR) to illustrate the advantage of our method in fig. 5 and 6. For each frame, three scores are computed on our results and competitor’s (fig. 5). They show that our method gives high error when neurons move, especially %SSD that peaks at the middle of the motions. The error arises from the out-of-focus problem because the imaging system acquires blurry images when the organism moves, as well as the neuron. Another advantage is that even though our method may produce wrong tracked neuron, the distance error from the ground truth is low as shown by SSD scores, which is around 5 pixels on the testing data.

To get the summary of performance, the box plot of SD score is displayed in fig. 6. It shows that our results are comparable to the competitor when SNR is high but our results outperforms the other method when SNR is low. This observation is supported by the qualitative results in fig. 7. It illustrates that optical-flow-based method fails to recover the motion field, resulting in poor tracked neuron centerline. While our method integrates shape and motion information with intensity to produce the robust tracking results.

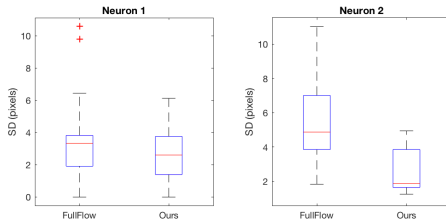


Figure 6: shows the box plot of SD scores produced by our method and FullFlow [5] when SNR is high (left) and low (right). In a box plot, the red line indicates the median. The bottom and top of the rectangle indicate the first and third quartiles respectively. The vertical line shows the range of value and outliers are plotted individually.

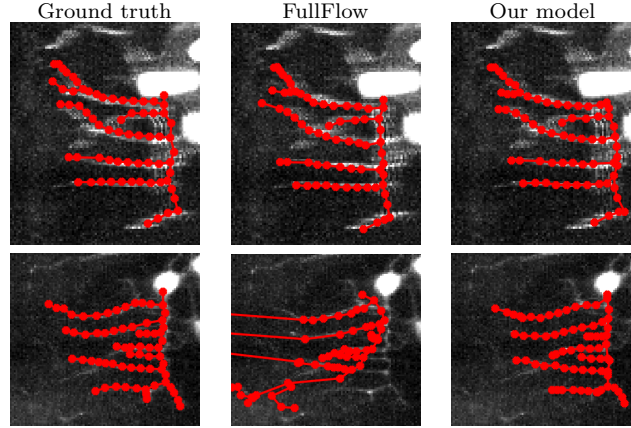


Figure 7: shows the ground truth (left), results from FullFlow (middle), and our results (right) in red color. Rows show tracking results when SNR is high (top) and low (bottom).

References

- [1] S. Gulyanov, N. Sharifai, S. Bleykhman, E. Kelly, M. D. Kim, A. Chiba, and G. Tsechpenakis, “Three-dimensional neurite tracing under globally varying contrast,” *ISBI*, pp. 875–879, 2015.
- [2] S. Gulyanov, N. Sharifai, M. D. Kim, A. Chiba, and G. Tsechpenakis, “Crf formulation of active contour population for efficient three-dimensional neurite tracing,” *ISBI*, pp. 593–597, 2016.
- [3] S. Farhand, S. Gulyanov, N. Sharifai, M. D. Kim, A. Chiba, and G. Tsechpenakis, “Temporal neurite registration using hierarchical alignments,” *ISBI*, pp. 334–338, 2016.
- [4] P. Glowacki, M. A. Pinheiro, A. Mosinska, E. Turetken, D. Lebrecht, A. Holtmaat, R. Sznitman, J. Kybic, and P. Fua, “Reconstructing evolving tree structures in time lapse sequences by enforcing time-consistency,” *TPAMI*, vol. PP, no. 99, pp. 1–1, 2017.
- [5] Q. Chen and V. Koltun, “Full flow: Optical flow estimation by global optimization over regular grids,” *CVPR*, pp. 4706–4714, 2016.
- [6] C. Bailer, K. Varanasi, and D. Stricker, “Cnn-based patch matching for optical flow with thresholded hinge loss,” *CVPR*, 07 2017.
- [7] D. Fortun, P. Bouthemy, and C. Kervrann, “Optical flow modeling and computation: A survey,” *CVIU*, vol. 134, pp. 1 – 21, 2015.
- [8] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard, “Tracking loose-limbed people,” *CVPR*, vol. 1, pp. I–421–I–428 Vol.1, 2004.
- [9] T. X. Han and T. S. Huang, *Articulated Body Tracking Using Dynamic Belief Propagation*, pp. 26–35, Springer Berlin Heidelberg, Berlin, Heidelberg, 2005.
- [10] M. A. Fischler and R. A. Elschlager, “The representation and matching of pictorial structures,” *IEEE Trans. Comput.*, vol. C-22, no. 1, pp. 67–92, 1973.

- [11] P. F. Felzenszwalb and D. P. Huttenlocher, “Pictorial structures for object recognition,” *IJCV*, vol. 61, no. 1, pp. 55–79, 2005.
- [12] V. Ferrari, M. Marin-Jimenez, and A. Zisserman, “Pose search: Retrieving people using their pose,” *CVPR*, pp. 1–8, 2009.
- [13] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, “Multiscale vessel enhancement filtering,” *MICCAI*, pp. 130–137, 1998.
- [14] R. O. Duda and P. E. Hart, “Use of the hough transformation to detect lines and curves in pictures,” *Commun. ACM*, vol. 15, no. 1, pp. 11–15, Jan. 1972.
- [15] T. J. Atherton and D. J. Kerbyson, “Size invariant circle detection,” *Image and Vision Computing*, vol. 17, no. 11, pp. 795 – 803, 1999.
- [16] Y. Y. Boykov and M. P. Jolly, “Interactive graph cuts for optimal boundary and region segmentation of objects in n-d images,” *ICCV*, vol. 1, pp. 105–112, 2001.
- [17] M. Kass, A. Witkin, and D. Terzopoulos, “Snakes: Active contour models,” *IJCV*, vol. 1, no. 4, pp. 321–331, 1988.
- [18] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon, “Global stereo reconstruction under second-order smoothness priors,” *TPAMI*, vol. 31, no. 12, pp. 2115–2128, 2009.
- [19] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” *TPAMI*, vol. 23, no. 11, pp. 1222–1239, 2001.
- [20] E. Boros and P. L. Hammer, “Pseudo-boolean optimization,” *Discrete Appl. Math.*, vol. 123, no. 1-3, pp. 155–225, Nov. 2002.
- [21] V. Kolmogorov and C. Rother, “Minimizing nonsubmodular functions with graph cuts—a review,” *TPAMI*, vol. 29, no. 7, pp. 1274–1279, 2007.
- [22] V. Lempitsky, C. Rother, S. Roth, and A. Blake, “Fusion moves for markov random field optimization,” *TPAMI*, vol. 32, no. 8, pp. 1392–1405, 2010.
- [23] L. Feng, T. Zhao, and J. Kim, “neutube 1.0: a new design for efficient neuron reconstruction software based on the swc format,” *eNeuro*, 2015.
- [24] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers, “V3d enables real-time 3d visualization and quantitative analysis of large-scale biological image data sets,” *Nat Biotech*, vol. 28, no. 4, pp. 348–353, 2010.

A Appendix A

Here we prove that $H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t) \leq 0$ (eq. 10). From the triangle inequality,

$$\begin{aligned} \|\mathbf{x}_p^t - \mathbf{x}_i^t\| + \|\mathbf{x}_i^t - \mathbf{x}_q^t\| &\geq \|\mathbf{x}_p^t - \mathbf{x}_q^t\| \\ (\|\mathbf{x}_p^t - \mathbf{x}_i^t\| + \|\mathbf{x}_i^t - \mathbf{x}_q^t\|)^2 &\geq \|\mathbf{x}_p^t - \mathbf{x}_q^t\|^2 \end{aligned}$$

Let $a = \max\{\|\mathbf{x}_p^t - \mathbf{x}_i^t\|, \|\mathbf{x}_i^t - \mathbf{x}_q^t\|\}$, then

$$\begin{aligned} (2a)^2 &\geq \|\mathbf{x}_p^t - \mathbf{x}_q^t\|^2 \\ a^2 &\geq \frac{\|\mathbf{x}_p^t - \mathbf{x}_q^t\|^2}{4} \\ \|\mathbf{x}_p^t - \mathbf{x}_i^t\|^2 + \|\mathbf{x}_i^t - \mathbf{x}_q^t\|^2 &\geq \frac{\|\mathbf{x}_p^t - \mathbf{x}_q^t\|^2}{4} \end{aligned}$$

α_s has negative value so $H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t) \leq 0$.