Point-set registration framework with Conditional Random Fields for automatic tracking of neurons in C. elegans whole-brain videos

Shivesh Chaudhary, Georgia Institute of Technology
Hang Lu, Georgia Institute of Technology
Point-set registration framework with Conditional Random Fields for automatic tracking of neurons in C. elegans whole-brain videos
Point-set registration framework with Conditional Random Fields for automatic tracking of neurons in C. elegans whole-brain videos

Shivesh Chaudhary, Hang Lu
Department of Chemical and Biomolecular Engineering
Georgia Institute of Technology, Atlanta, GA 30332
hang.lu@gatech.edu

Abstract

Advanced microscopic techniques combined with microfluidics allow fast collection of whole brain functional recordings in C. elegans. However generating important neuroscience insights from whole-brain videos is still limited by the processing time of the videos. Significant head deformations during recordings make tracking of neurons throughout the video a challenging task, thus slowing down the process of extracting neuron activity traces from the videos. In this paper we present a framework for automatic tracking of neurons.

1 Introduction

C. elegans as a model organism offers two unique advantages that can be utilized to investigate the relationship between complex behavior and whole-brain dynamics. First, quantitative analysis of complex behavior is possible e.g. [? ?]. Second, functional activity of majority of neurons can be recorded with high spatiotemporal resolution [22]. Compared to the fast collection rate of whole-brain recordings, processing videos to generate important neuroscience insights is slow. A major bottleneck in processing these videos is tracking neurons throughout the video length to accurately extract neuron activity traces. Large non-rigid head deformations during recording make tracking a challenging task. Errors in segmenting densely packed neurons and dim neurons generates outliers in some frames and missing neurons in others thus making tracking further difficult. In this work we address these issues and present a framework for automatic tracking of neurons.

2 Related Work

Object tracking in fluorescence microscopy images is achieved by either sequential correspondence estimation between objects in consecutive frames [1, 7, 11, 20] or global optimization [24, 3, 8, 37, 17, 28, 29]. These methods fall into two categories. Methods that estimate only correspondence [1, 26] etc. and registration methods that estimate correspondence as well as optimal spatial transformation to register objects in consecutive frames e.g. [49]. Iterative estimation of correspondence in registration methods provide better results when large deviations are present in frames. Among general class of registration algorithms, soft-assign approaches [8] and gaussian mixture model (GMM) based methods [21, 31, 42] are well known. Further, several improvements have been proposed to registration algorithms both for improving correspondence estimation [10, 53, 36, 35, 52, 25] as well as improving spatial transformation [27, 35, 33, 13, 15, 6].

Sequential pairwise registration methods mentioned above are not suitable for tracking when the number of objects detected in each frame vary due to presence of outliers or missed detection. Joint registration methods which register multiple frames simultaneously [44, 43, 12] can handle such
We use modified Expectation Conditional Maximization algorithm [18] to account for outliers (misidentified neurons in a frame), we add an extra component with third, these methods assume independence of tracks [37]. The contribution of this work is two fold. First, we present a joint registration framework for video in which neurons are to be tracked. We treat tracking problem as labelling problem where j

E-step calculates the expected value of complete data log likelihood equivalent to calculating the smoothness of the transformation [31, 27]. Reproducible Kernel Hilbert Space (RKHS) with gaussian reproducing kernel and ensures spatial

g kernel given by T

G

i

\rightarrow

3 Formulation

Let X_i = \{x_{i1}, x_{i2}, \ldots, x_{iN_i}\} \in \mathbb{R}^{D \times N_i} be a matrix forming the the i th point-set that represents N_i neurons detected in the i th frame of the video. The set X = \{X_i\}_{i=1}^N represents N frames of video in which neurons are to be tracked. We treat tracking problem as labelling problem where neurons detected in each frame are to be assigned a globally consistent label. Our algorithm consists of two steps. In the first step we build a global reference point-set and use this set to generate coarse correspondence between all point-sets. In the second step we assign a globally consistent label to each point using spatial and temporal contextual features.

3.1 Step - 1 : Joint registration

In joint registration framework [12] each point-set X_i is generated by applying a spatial transformation \Phi_i(G_i, W_i) : \mathbb{R}^3 \rightarrow \mathbb{R}^3 to a realization of a GMM. The component centroids of the GMM are given by a global reference point-set Y = \{y_1, y_2, \ldots, y_M\} \in \mathbb{R}^{D \times M}. We assume isotropic covariances and equal membership probabilities for each GMM component. To account for outliers (misidentified neurons in a frame), we add an extra component with prior probability \omega. Hence the probability of realization of jth neuron in i th frame is given by P(x_{ij}) = \frac{\omega}{N_i} + \frac{1-\omega}{M} \sum_{k=1}^{M} \exp \left( - \frac{\|T_i(x_{ij}, W_{ij}) - y_k\|^2}{2\sigma_i^2} \right). Spatial transformation on x_{ij} is parameterized as T_i(x_{ij}, W_i) = x_{ij} + \sum_{k=1}^{N_i} G_i(i, k) W_{ik}. Here G_i \in \mathbb{R}^{N_i \times N_i} is a gaussian kernel given by G_i(p, q) = \exp \left( - \frac{\|x_{ij} - x_{pq}\|^2}{2\sigma_i^2} \right) and W_{ik} is the kth row of transformation parameters matrix W_i \in \mathbb{R}^{N_i \times D} for ith point-set. Gaussian kernel constrains the transformation to lie in a Reproducible Kernel Hilbert Space (RKHS) with gaussian reproducing kernel and ensures spatial smoothness of the transformation [31, 27].

We use modified Expectation Conditional Maximization algorithm [18] to learn parameters. The E-step calculates the expected value of complete data log likelihood equivalent to calculating the posterior probability that neuron j in i th frame is generated from kth spatial component.

\begin{equation}
\alpha_{ijlk}^{old} = \frac{\exp \left( - \frac{\|T_i(x_{ij}, W_{ij}^{old}) - y_k\|^2}{2\sigma_i^2} \right)}{\sum_{k=1}^{M} \exp \left( - \frac{\|T_i(x_{ij}, W_{ij}^{old}) - y_k\|^2}{2\sigma_i^2} \right) + (2\pi\sigma_i^2)^{\frac{D}{2}} \frac{1}{1-\omega N_i}}.
\end{equation}
In the first M-step we obtain the spatial transformation parameters by maximizing the complete data log likelihood.

\[ L = - \sum_{i,j,k} \alpha_{ijk}^{old} \left\| T_i(x_{ij}, W_i) - y_k \right\|^2 / 2\sigma_i^2 - \sum_i N_i \left( \frac{\alpha}{2} tr(W_i^T G_i W_i) + \frac{\lambda}{2} tr(T_i^T M_i T_i) \right) + c \]  

(2)

where \( c = \sum_{i,j,k} N_i \alpha_{ijk}^{old} \left[ \frac{1 - \omega}{2\sigma_i^2} \right] + \sum_{i,j,k} N_i \alpha_{ijk}^{old} \left[ \frac{1 - \omega}{2\sigma_i^2} \right] \). The second term in equation (2) minimizes the norm of \( \Phi \), in RKHS [27], thus regularizes its smoothness. Further, the third term puts a constraint on the transformation to preserve the local topology of the point-set [15]. \( \Phi_j = (I - L_j)^T(I - L_j) \) where \( j \)th column of \( L_j \) consists of weights of \( K \) nearest neighbors of \( x_{ij} \) that reconstruct \( x_{ij} \) [38]. These weights are obtained in least squares sense by minimizing \( \| x_{ij} - \sum_{k=1}^K x_{ik} L_{ik} \|^2 \). We solve for \( W_i \) and \( \sigma_i^2 \) by setting the derivative of equation (2) to zero.

\[ (d(P;1)G_i + \alpha\sigma_i^2 I + \lambda\sigma_i^2 M_i G_i)W_i = P_i^T Y - (d(P;1) + \lambda\sigma_i^2 M_i)X_i^T \]  

(3)

\[ \sigma_i^2 = \frac{1}{N_{pi}D} tr(X_i^T Y d(P;1)^T) \]  

(4)

Here \( d(a) \) is diagonal matrix formed by vector \( a \), \( N_{pi} = \sum_{i,j,k} \alpha_{ijk}^{old} \) and \( P_i \in \mathbb{R}^{M \times N_i} \) is the matrix of posterior probabilities for \( i \)th point-set with \( P_i(k,j) = \alpha_{ijk}^{old} \). We repeat the E-step and the first M-step iteratively to obtain optimal transformation parameters \( W_i \).

After estimating all \( W_i \), in the second M-step, we update the GMM centroids keeping other parameters fixed. Setting \( T = [T_1, T_2, \ldots, T_N] \) and \( P = [P_1, P_2, \ldots, P_N] \).

\[ Y = TP^T d(P;1)^{-1} \]  

(5)

We initialize GMM centroids as neuron locations in a randomly selected frame. Since certain neurons may not have been detected in this frame, the number of GMM components may be fewer than the total number of neurons detected in all frames. Therefore, we update the number of components before the second M-step. We assign neurons in each frame to their maximum posterior probability component using the posterior probabilities \( P_i \), such that each component is assigned to only one neuron. Subsequently, we use kernel density estimation to estimate the distribution of unassigned neurons in all frames and initialize more components at local maxima of the estimated distribution.

### 3.2 Step - 2 : Labelling by Conditional Random Fields

In gaussian mixture models, labelling is achieved by assigning each neuron to its maximum posterior probability component. Such labelling uses only spatial location information. To achieve more spatially and temporally coherent labelling, we model the conditional distribution of labels of neurons given the registered point-sets as Conditional Random Field. Each neuron \( T_i(x_{ij}) \) can take a label \( l_{ij} \in \{1, 2, \ldots, M\} \). Therefore the joint distribution of labels is given as

\[ P(l_{11}, \ldots, l_{NN}) = \frac{1}{Z} \exp \left( \sum_{ij} N \Phi(l_{ij}) + \sum_{i,j,k} \Phi(l_{ij}, l_{ik}) + \sum_{i,j,k} \Phi(l_{ij}, l_{i+1,k}) \right) \]  

(6)

Here \( Z \) is the partition function. We include three kinds of potentials in our model that specify different spatial and temporal contextual constraints. The unary potentials \( \Phi(l_{ij}) \) specify the likelihood that neuron \( j \) in frame \( i \) is assigned a label \( k \in \{1, 2, \ldots, M\} \) on the basis of featural differences between \( j \) and \( y_k \). Various types of features can be used such as shape context [2], graph centrality measures [10] etc. We define \( \Phi(l_{ij}) \) on the basis of spatial affinity of registered neuron \( j \) to component \( k \).

\[ \Phi(l_{ij} = k) = \exp(-\lambda_u \| T_i(x_{ij}) - y_k \|^2) \]  

(7)

We impose the local neighborhood structural constraint that labels assigned to neighboring neurons should preserve the distance between the labels [6, 41]. For this we include Potts like pairwise potentials \( \Phi(l_{ij}, l_{ik}) \) between each neuron \( j \) and its \( k \) spatial neighbors in \( N_{ij}^{s} \). Zheng et al. [53] proposed such constraint however CRF framework can generalize this to more complex constraints.

\[ \Phi(l_{ij} = m, l_{ik} = n) = \exp(-\lambda_s \| d(T_i(x_{ij}), T_i(x_{ik})) - d(y_m, y_n) \|^2), k \in N_{ij}^{s} \]  

(8)
To ensure temporal consistency of labelling, we build temporal neighbourhood graph by sequential pairwise registration of frame $i$ to frame $i+1$ and adding edges between neurons in frame $i$ to their closest match in frame $i+1$. Further we include Ising like pairwise potentials $\Phi(l_{ij}, l_{i+1,k})$ that favor that temporal neighbors are assigned same labels.

$$\Phi(l_{ij} = m, l_{i+1,k} = n) = \exp(-\lambda t \|T_i x_{ij} - x_{i+1,k}\|^2) 1\{m = n\}, k \in N_{ij}^t$$ (9)

Finally marginal distribution of each neuron label is inferred by Loopy Belief Propagation algorithm [30] and neurons are assigned to the maximum marginal component.

### 4 Results

We set the following values for hyperparameters in our model - outlier ratio $\omega = 0.1$, gaussian kernel scale $\beta = 3$, smoothness regularizer $\alpha = 3$, local topology regularizer $\lambda = 10$. CRF parameters for unary and pairwise potentials were set as $\lambda_u = 0.3, \lambda_s = 1, \lambda_t = 1$. $\omega$ was set on the basis of average segmentation error rate in frames. $\lambda$ was set high to preserve the local rigidity of transformation. This helps in maintaining the neighborhood constraint specified in CRF. $\lambda_u$, $\lambda_s$ and $\lambda_t$ were chosen on the basis of relative importance of respective constraint in labelling. Accuracy of the algorithm was assessed by comparing tracking results with a manually annotated video. Specifically, we analyzed how accurately each neuron was tracked throughout the video and number of incorrectly tracked neurons in each frame. We found that most of the neurons, 70%, were incorrectly tracked in very few frames 5% (Figure 1(a)). Also most of the frames in video, 88%, had less than 14% incorrectly tracked neurons (Figure 1(b)). Figure 1(c) shows qualitative results of labels assigned to neurons in a frame sequence. Here color encodes the label assigned to neuron.

![Figure 1](image)

Figure 1: a) Neuron tracking accuracy b) Frame tracking accuracy c) Qualitative tracking results

### 5 Conclusion

In this work we presented an algorithm for automatic tracking of neurons in C. elegans whole-brain videos. Validation on a manually annotated video showed that the algorithm provides promising accuracy. The algorithm is robust against very large deformations in consecutive frames but temporal constraint in CRF needs to be relaxed in this case. Also, large outlier ratio due to errors in segmenting densely packed neurons degrades accuracy. Further work is required for extensive validation on more videos and comparison with other methods. We speculate our algorithm to achieve better accuracy as compared to directed graphical models such as Kalman or Particle filtering [16, 19] that do sequential tracking. Further, CRF model does not assume independence of observations thus non-independent, higher order contextual features [24] can be specified. Hence our model is more general as compared to many cost optimization tracking methods e.g. nearest neighbor [40, 5], temporal association [47] methods. Our algorithm should improve the processing speed of whole-brain videos of C. elegans. Fast extraction of neuron activity traces from tracked videos will help in fully utilizing the large amount of whole-brain data being collected.
References


