5-2018

Model and Appearance Based Analysis of Neuronal Morphology from Different Microscopy Imaging Modalities

Sarun Gulyanon
Purdue University

MODEL AND APPEARANCE BASED ANALYSIS OF NEURONAL MORPHOLOGY FROM DIFFERENT MICROSCOPY IMAGING MODALITIES

A Dissertation

Submitted to the Faculty

of

Purdue University

by

Sarun Gulyanon

In Partial Fulfillment of the

Requirements for the Degree

of

Doctor of Philosophy

May 2018

Purdue University

West Lafayette, Indiana

THE PURDUE UNIVERSITY GRADUATE SCHOOL
STATEMENT OF DISSERTATION APPROVAL

Dr. Gavriil Tsechpenakis, Co-Chair

Department of Computer and Information Science, IUPUI

Dr. Voicu Popescu, Co-Chair

Department of Computer Science, West Lafayette

Dr. Mihran Tuceryan

Department of Computer and Information Science, IUPUI

Dr. W. Daniel Tracey Jr.

Department of Biology, IU Bloomington

Approved by:

Voicu Popescu by William J. Gorman

Head of the Graduate Program

Dedicated to my parents,

Tanongsak and Pannipa Gulyanon

ACKNOWLEDGMENTS

I would first like to thank Dr. Gavriil Tsechpenakis for guiding me throughout the last five years with my research. His support helped foster my ability to learn and grow as an individual. I am forever grateful for his guidance. I would also like to extend a special gratitude to Dr. Akira Chiba, Dr. Nima Sharifai, Dr. Dan Tracey, Dr. Liping He, and their colleagues for their hard work on collecting data used in this work and insightful comments on topics outside of my expertise.

I would like to thank my committee members, Dr. Voicu Popescu, Dr. Mihran Tuceryan, Dr. Dan Tracey, and Dr. Daniel Aliaga for providing me constructive feedback in the course of my dissertation.

I am thankful to the Computer Science Department. They have such a wonderful and friendly environment that makes me feels at home. Additionally, I like to thank all my collegues, especially Sepehr Farhand, Nathan Hammes, Ruwan Janapriya Egoda Gamage, Siripriya Morusu, Huiwen Cheng, Abdulmecit Gungor, and Sarkhan Badirli, for your friendship and support throughout this journey.

Finally, I must express my very profound gratitude to my family and Siripha Rungsrisutthiwong for their endless love and support over the years of my study, especially my father, who suggested me to go to graduate school in the United States. Without their continuous encouragement, this accomplishment would not have been possible. I am grateful and fortunate to have them in my life.

This work was supported by NSF/DBI [#1252597]: "CAREER: Modeling the structure and dynamics of neuronal circuits in the Drosophila larvae using image analytics", and NSF/DBI [#1062405]: "ABI Innovation: Modeling the Drosophila Brain with Single-neuron Resolution using Computer Vision Methods", awarded to G. Tsechpenakis.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Figure Page

Figure Page

ABBREVIATIONS

CNS	Central Nervous System
CRF	Conditional Random Field
CSF	Center Surround Filters
DRF	Discriminative Random Fields
EM	Expectation Maximization
FFD	Free-Form Deformation
FLIM	Fluorescence Lifetime Imaging Microscope
FN	False Negative
FP	False Positive
FRET	Förster Resonance Energy Transfer
GFP	Green Fluorescent Protein
GPR	Gaussian Process Regression
HCF	Highest Confidence First
HGD	Histogram of Gradient Deviation
ICM	Iterative Conditional Mode
LoG	Laplacian of Gaussian
MAE	Mean Absolute Error
MAP	Maximum A Posteriori
MRF	Markov Random Field
MST	Minimum Spanning Tree
NCC	Normalized Cross Correlation
NCT	Neuron Circuit Tracer
OOF	Optimally Oriented Flux
PCT	Principal Curve Tracing

PDE	Partial Difference Equation
PSF	Point Spread Function
Q-MIP	Quadratic Mixed Integer Program
QPBO	Quadratic Pseudo-Boolean Optimization
ROI	Region of Interest
SNR	Signal-to-Noise Ratio
SRS	Simultaneous Registration and Segmentation
TED	Tree Edit Distance
TP	True Positive
VNC	Ventral Nerve Cord
2D	2-dimensional
3D	3-dimensional
4D	4-dimensional

ABSTRACT

Gulyanon, Sarun Ph.D., Purdue University, May 2018. Model and Appearance Based Analysis of Neuronal Morphology from Different Microscopy Imaging Modalities. Major Professor: Gavriil Tsechpenakis.

The neuronal morphology analysis is key for understanding how a brain works. This process requires the neuron imaging system with single-cell resolution; however, there is no feasible system for the human brain. Fortunately, the knowledge can be inferred from the model organism, *Drosophila melanogaster*, to the human system. This dissertation explores the morphology analysis of *Drosophila* larvae at single-cell resolution in static images and image sequences, as well as multiple microscopy imaging modalities. Our contributions are on both computational methods for morphology quantification and analysis of the influence of the anatomical aspect. We develop novel model-and-appearance-based methods for morphology quantification and illustrate their significance in three neuroscience studies.

Modeling of the structure and dynamics of neuronal circuits creates understanding about how connectivity patterns are formed within a motor circuit and determining whether the connectivity map of neurons can be deduced by estimations of neuronal morphology. To address this problem, we study both boundary-based and centerline-based approaches for neuron reconstruction in static volumes.

Neuronal mechanisms are related to the morphology dynamics; so the patterns of neuronal morphology changes are analyzed along with other aspects. In this case, the relationship between neuronal activity and morphology dynamics is explored to analyze locomotion procedures. Our tracking method models the morphology dynamics in the calcium image sequence designed for detecting neuronal activity. It follows

the local-to-global design to handle calcium imaging issues and neuronal movement characteristics.

Lastly, modeling the link between structural and functional development depicts the correlation between neuron growth and protein interactions. This requires the morphology analysis of different imaging modalities. It can be solved using the part-wise volume segmentation with artificial templates, the standardized representation of neurons. Our method follows the global-to-local approach to solve both part-wise segmentation and registration across modalities.

Our methods address common issues in automated morphology analysis from extracting morphological features to tracking neurons, as well as mapping neurons across imaging modalities. The quantitative analysis delivered by our techniques enables a number of new applications and visualizations for advancing the investigation of phenomena in the nervous system.

1 INTRODUCTION

Studying how a brain works is one of the greatest challenges for modern science. It requires an understanding of the structure and function of the nervous system during development at the molecular, cellular, and systemic levels [1]. To monitor how the human brain works, many tools are invented, e.g., Computerized Axial Tomography (CAT) Scan, Magnetic Resonance Imaging (MRI), Positron Emission Tomography (PET) Scan, and especially Electroencephalogram (EEG) and Functional MRI (fMRI), which are designed to measure brain activity. EEG is a non-invasive method for recording electrical activity of the brain emanated from the ionic current within the neurons [2], whilst fMRI indirectly measures brain activity by detecting changes of the local blood supply [3]. However, these are tools for studying neural connectivity via responses rather than the structural mechanisms that create these responses. Without tools capable of monitoring the structural properties of neurons that generate these responses, it is impossible for neurologists to identify the links between neurons and brain mechanisms. That is why we carry out this study in the 'bottom-up' approach, which builds up knowledge starting from individual neurons, the basic element of the nervous system, towards the complete map of the neuronal connectivity over time.

Our focus is on the neuronal morphology because the morphological attributes of the axonal and dendritic arborizations are the main components of the neuronal phenotype [4]. In addition, neuronal arborization patterns also influence the synaptic connectivities, which in turn determine neuronal functions [5]. For these reasons, neurobiologists are interested in studying the structures of neurons at the single-cell resolution *in vivo*. Nonetheless, we do not know whether it is the interaction network of proteins that raises the probability of physical association between neurons, or if it is the other way around where the morphogenesis drives the synaptic connectivities.

To address this problem, the multi-modality imagery is used for studying the causality between protein interactions (proteome) and cellular complexity (phenome), which have been studied independently in previous experimentations [6, 7].

Model organism. Imaging the human brain at the single-cell resolution *in vivo* is challenging because the human brain has a vast complexity with billions of neurons and trillions of synapses [8, 9], which make such a system practically infeasible with current computational power. Fortunately, the knowledge can be inferred from some other species, the model organism, to the human system; therefore, the model organism larval *Drosophila melanogaster* is studied instead. The complete neuronal map of the *C. elegans* nervous system with 302 neurons was created successfully [10]. Hence, we select *Drosophila* larvae whose central nervous system contains approximately 10,000 neurons [11], which is the next step from the nematode worm (*C. elegans*). A brain of an adult fruit fly (*Drosophila*) has around 100,000 neurons [12] and a mouse (*Mus musculus*) has a much more intricate system with approximately 4,000,000 neurons [13]. Larval *Drosophila* is our best option because it is simpler than other model organisms. Meanwhile, it is complex enough that the automation is required, unlike *C. elegans*. The manual reconstruction of the neuronal map of *Drosophila* larvae is not feasible due to a large number of neurons and even larger number of synapses. The task becomes highly labor intensive and tedious for human operators, resulting in error-prone results. Thus, the automated morphology analysis is needed. Its major bottleneck is the quantification of neuronal morphology.

What is neuronal morphology? It is the attributes related to the form, shape, and structure of neurons, i.e., the soma or cell body, and particularly neurite (the term denoting axon and dendrite). They are essential for understanding how connectivity patterns are formed within neural circuits and how interconnections among neural circuits are related with higher-order cognitive functions [1, 4]. Image stacks for the study of neuronal morphology are generally acquired by confocal laser scanning microscopy. Computing numerical features for morphology quantification usually involves the neuron reconstruction process, which extracts the boundary and/or

centerline of neurons from input image stacks. These numerical features are a set of values measuring either low- or high-level attributes of neuronal morphology. They can be categorized into two main types based on the scope of neuronal morphology used in the feature extraction: global and local. Global features take into account the topology of the whole tree data structure and produce the overall properties without any specification of individual neuron branches, while local features describe certain properties of the individual neuron branches. Since local features normally involve a large number of values, the summarized data produced by applying statistical techniques are often preferred over raw data [14].

Examples of frequently used global features are the total height, width, depth, length, volume, bifurcation count, and branch tip count of a neuron [15] (Fig. 1.1). The spatial distribution of the neuron is usually measured by Sholl analysis [16] or its variation. This analysis divides the space around the neuron with concentric circles (in 2D) or spheres (in 3D) of increasing radii centered at the soma cell or a reference

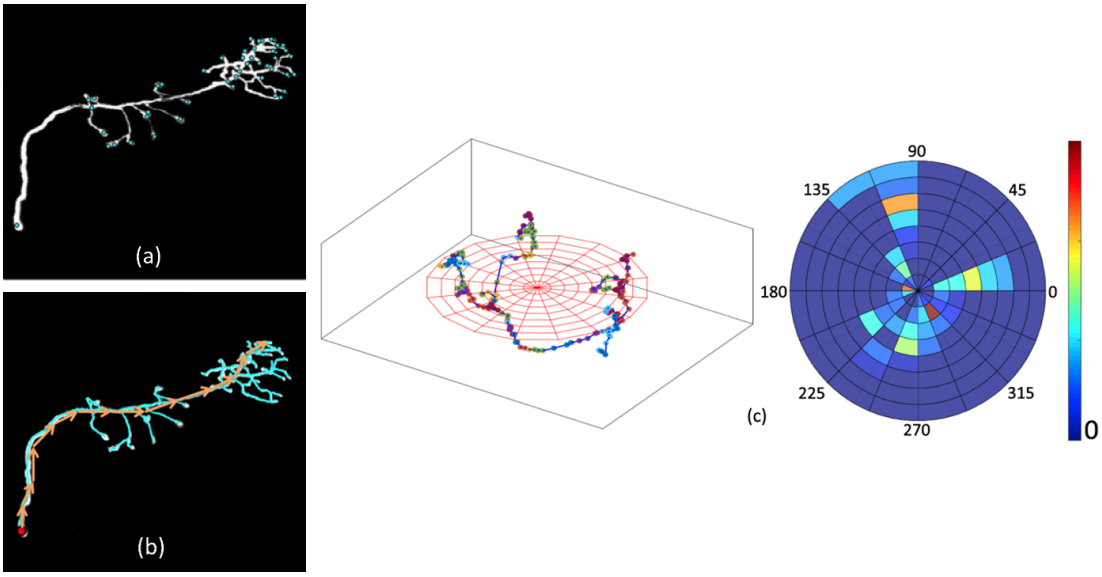


Figure 1.1. Examples of global features: (a) the number of branch tips (in cyan), (b) the length of axon, and (c) the shape histogram of neuron's length and its corresponding color-coded plot.

point. Sholl profile measures the specific property of a neuron within the regions bounded by these concentric circles/spheres. One notable variation is the shape histogram, which also divides the surrounding space with equally spaced slices on top of concentric circles/spheres. Other common global features are the total number of primary, secondary, and tertiary dendrites; the average path length from dendrite tips to the cell body; and the radial distances, which is the Euclidean distance from dendrite tips to the cell body. Primary dendrites are dendrites directly issued from the soma, while secondary dendrites are branches emerging from a primary dendrite and tertiary dendrites emanate from secondary dendrites [17].

For local features, some examples seen in the literature include branch length, ratios between adjacent branches, bifurcation angles, and branch curvature [18]. With high-resolution optical imaging, dendritic spines can be detected. Spines are normally categorized into three common types based on their shape: thin, mushroom, and stubby. Examples of spine quantification include length, diameter, orientation, volume, count, and density [1].

In this dissertation, our focus is on the single-cell morphology analysis for both static images and image sequences. We develop model-and-appearance-based methods to solve four fundamental tasks in neuronal morphology quantification from different microscopy imaging modalities, i.e., segmentation, tracing, tracking, and part-wise volume segmentation with artificial templates. Most previous works rely solely on the appearance information, which is derived from the input neuron volumes [19–22]. Some works employ the simple model information, i.e., local shape of neurons, that is manually crafted or learned using the training data [23–25]. On the contrary, our methods combine the appearance and model information together, and adopt the more sophisticated model encoding the global shape and the structure of the whole neuron. To illustrate the significance of our contributions, our methods are applied on the three following studies: modeling the structure and dynamics of neuronal circuits, finding the patterns of the morphology dynamics, and modeling the connection between structural and functional development.

1.1 Modeling the Structure and Dynamics of Neuronal Circuits

Neurons do not operate in isolation [26]. There are interactions among neurons, where neuronal signal transduction occurs [27]. They form circuits that process specific types of information through synaptic connectivity, which is dictated by neuronal morphology [5]. That is why it is desirable to create the entire neuronal map of individual neurons and their synapses at the single-cell level *in vivo*, called connectome. Moreover, the temporal prediction model of neuronal maps must be constructed as well to observe the development in connectome. The first steps towards this goal are the extraction of neuronal morphology and the computation of a compact, numerical representation; the whole process is called neuron reconstruction.

Some computer scientists might consider the automated neuron reconstruction problem is solved since commercial and academic tools, e.g., Neurolucida 360[1], NeuronStudio[2], and Neuromantic[3], are available and claim success [1]. However, their automatic functionality operates properly only for certain datasets. Some methods try to alleviate the problem by allowing the user input to correct the automatic traces, resulting in the lower amount of work required by human operators. Biologists still struggle with the neuronal analysis because of a bottleneck in the reconstruction process caused by the lack of a generally applicable tool. Thus, some institutes are inspired to organize the competition or collaboration like in DIADEM challenge[4] [28] or BigNeuron project[5] [29]. These organizations encourage the development of new methods in order to advance the field by creating a large collection of 3D neuron morphology data, and developing the standard for the variety of available automated neuron reconstruction protocols.

Here, we develop novel neuron reconstruction methods for extracting neuronal morphology from static neuron volumes. They can be classified as boundary-based

[1] `http://www.mbfbioscience.com/neurolucida360`
[2] `http://research.mssm.edu/cnic/tools-ns.html`
[3] `http://www.reading.ac.uk/neuromantic/body_index.php`
[4] `http://diademchallenge.org/`
[5] `http://alleninstitute.org/bigneuron`

and centerline-based approaches (Fig. 1.2). These two approaches illustrate the neuron reconstruction results at two different levels, which provide different perspectives for morphology analysis. The boundary-based method describes the neuronal morphology at the margin level, whereas the centerline-based method describes the neuronal morphology at the structure level.

- **Boundary-based approach** involves the neuron segmentation to extract the position and outline of a neuron. The segmentation problem is concerned with neurite detection and classifying voxels as either neurite or background. This aspect delineates the perimeter of a neuron, which is suitable for low-level morphology analysis.

- **Centerline-based approach** involves the neuron tracing to obtain the underlying model (i.e., tree data structure) of the neuronal topology. The tracing problem is focused on centerline extraction and neuronal topology recovery. This approach extracts the topological skeleton of a neuron, which contains the relevant structural information that allows easy computation of high-level measurements of biological variables [1].

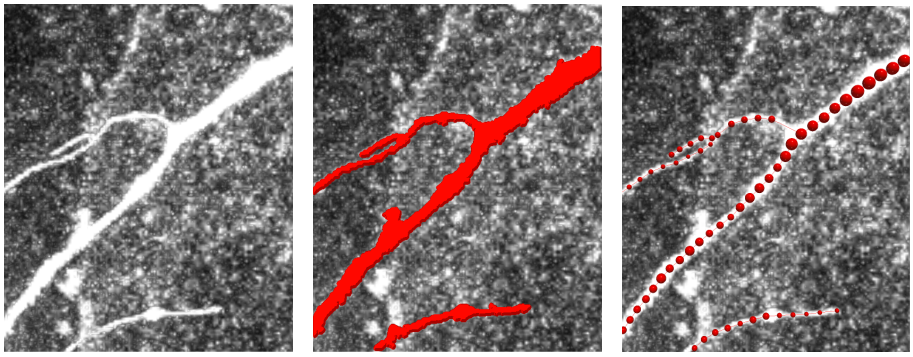


Figure 1.2. Neuron reconstruction methods of the input image stack (left) can be categorized into two groups: boundary-based (middle) and centerline-based (right) approaches.

Segmentation and tracing problems are tightly related since they both solve the neuron reconstruction problem. The improvement in one task would result in the improvement of another, so they are usually solved simultaneously. Another benefit of the neuron reconstruction is that it requires less memory storage for storing neuronal morphology information, compared to image stacks. The neuron segmentation encodes boundaries of neurons using binary volumes or run-length encoding, while the neuron tracing encodes structures of neurons into the compact numerical representation (see details in App. A).

After the reconstruction of neuronal morphology, the next step is modeling the dynamics of neuron circuits, which could be solved using the neuron subtype recognition. Neuron subtypes play the key role in providing the overview of the evolution of the nervous system because they are related to the underlying principles of synaptic connectivity in a motor neuron circuit [5]. Since neuron subtypes can be modeled based on the morphological stereotype [30], the branching details computed by our methods could be used to improve the existing neuron subtype classifiers [30–33]. Finally, with the availability of a neuron subtype recognition system, the neuron circuits could be modeled based on the ensembles of neuron subtypes.

1.2 Finding the Patterns of the Morphology Dynamics

Neuronal morphology controls how neurons respond to stimuli but the underlying mechanism of how topology and its changes affect relevant circuits is still elusive [34]. To study this phenomenon, the temporal prediction model of morphology changes, or morphology dynamics, must be reconstructed and patterned over the appropriate (domain-specific) time intervals. Topology changes captured by neuron image sequences have to be recorded along with other properties in order to understand their association that controls the neuronal mechanisms. In this case study, we explore the relationship between neuronal activity and morphology dynamics to study the locomotion behaviors of *Drosophila* larvae.

It is known that the locomotion of an organism is controlled by proprioceptive neurons but their relationship is unknown. Understanding this relationship will not only give insights into the neural mechanism for locomotion, but also reveal the locomotion procedure. To study the mechanism of how the sensory neurons operate and coordinate the locomotion, we investigate the relationship between deformation of morphology and timing of neurotransmission.

In the *Drosophila* larvae, the optogenetic excitation [35] followed by cell-specific calcium imaging [36] enables the observing of the link between the micrometer-scale neuronal connectome [37] with the millimeter-scale locomotion behavior [38]. The neuron reconstruction over calcium image sequence is required to quantify both morphology dynamics and neuronal activity simultaneously (Fig. 1.3). One main issue of calcium images is that they produce low responses when there are no neuronal activities. Another issue is that the neuron morphology reconstruction of static image stack is inadequate for this task because it cannot capture characteristics of morphology dynamics during the locomotion. To address these issues, we develop our neurite tracking method for calcium image sequences in order to capture the changes occurred in the nervous systems during the locomotion process.

Our method tracks the movement and deformation of neurons using the local-to-global design: neuron tracking is guided by local features derived from input volumes,

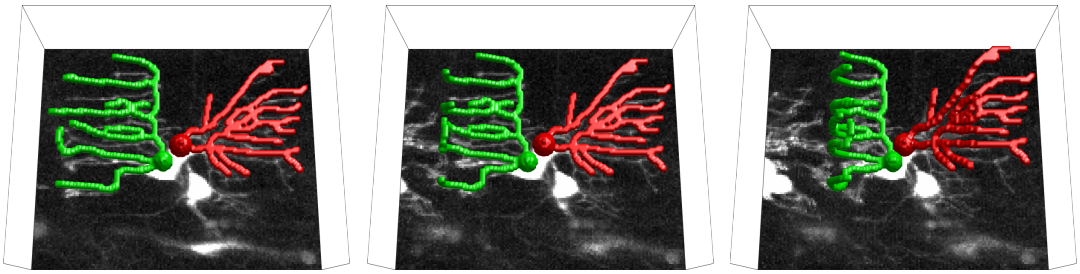


Figure 1.3. Morphology dynamics of two neurons (red and green colors) over calcium time-lapse sequence at three consecutive time steps from left to right. Neuronal activity is directly proportional to calcium responses.

while the global model constrains the search space based on the motion model of the morphology dynamics. The local features provide the probability to the reconstructed neuron proposal over the image sequence. Then, the global model limits the search space to increase efficiency and probability of finding an acceptable, optimal solution. These two components are then combined in the principled manner through the graphical model.

The neuron reconstruction and morphology dynamics over the calcium image sequence computed by our tracking method can be linked with the neuronal activity

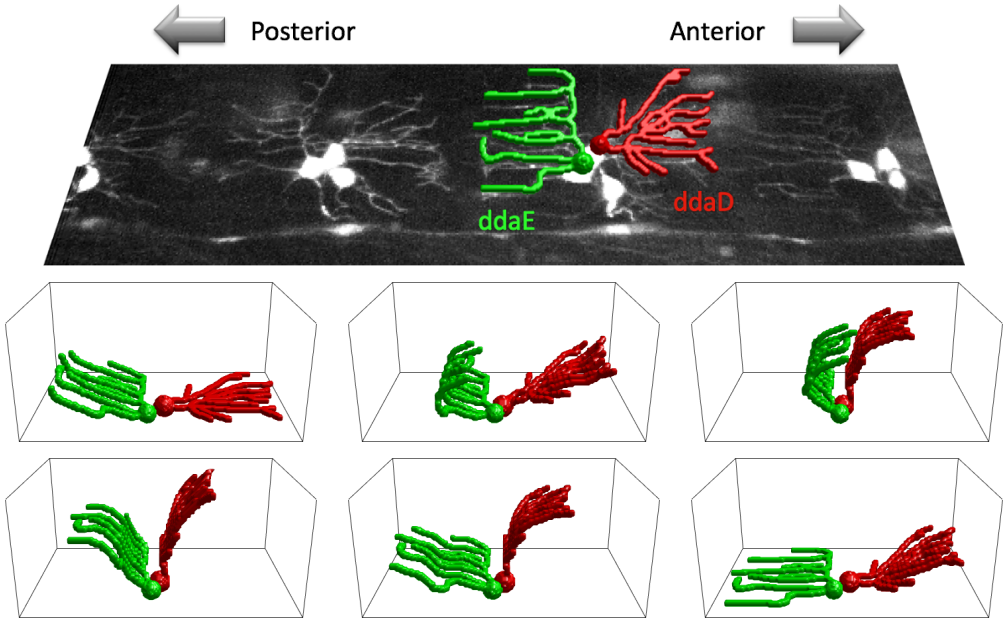


Figure 1.4. Patterning morphology dynamics of proprioceptors. ddaD (red) and ddaE (green) neurons usually appear together as a cluster of neurons (top). ddaE neurons expand towards the posterior of the larvae, while ddaD neurons branch out towards the anterior. The maximum intensity projection of an input volume is displayed for the purpose of visualization. The deformation of these neurons occurs in a cycle that composes of two main processes: contraction (middle) and relaxation (bottom). From left to right, 3D neuron rendering shows three time steps of the contraction and relaxation processes in a cycle during the forward movement. The phase of this muscle contraction cycle is used as the reference for comparison between larval movements.

information directly. In our experiments, we apply our neuron tracking method to model the morphological changes of proprioceptors, i.e., ddaD and ddaE neurons, during the movement of the *Drosphila* embryo. The deformation occurs in tandem with a similar pattern composed of two processes: contraction and relaxation (Fig. 1.4). This cycle is used as the reference for comparison between larval movements and patterning the changes in morphology along with the associated neuronal activity. This scheme creates the new technique for visualizing the connection between the neuronal activities indicated by transmission of nerve impulse, and the morphology of neurons during the locomotion process.

1.3 Modeling the Link between Structural and Functional Development

Studying the neuronal morphology provides only the anatomical aspect of neurons; however, there are other aspects such as neuron functionality influencing the neuron mechanisms. The analysis must take into account all aspects of neurons altogether in order to fully understand the whole picture of the neuron mechanisms. Therefore, neuroscientists are interested in the study of the connection between neuron functionality at the molecular level and neuron structure at the cellular level. Studying different aspects requires different imaging systems, i.e., FRET microscopy acquires the nanometer-scale protein network, and confocal laser scanning microscopy acquires the micrometer-scale cellular complexity.

Unfortunately, combining information from multiple modalities is not a trivial task since the information is described in different scales. The information across modalities must be aligned into the same coordinate to enable the relevant integration across modalities. Furthermore, the alignment must be carried out according to the neuron structures usually described qualitatively by neuroscience experts to provides the biological interpretation. Therefore, the conventional registration techniques are inadequate because they register images based on the appearance and ignore the neuron shape.

For this reason, we tackle the problem of analyzing different microscopy imaging modalities as the part-wise neuron volume segmentation problem with artificial templates (Fig. 1.5). To provide the reference for biologically meaningful comparison across modalities, we create the 'prototype', which is the standardized representation of a neuron regardless of modality. This artificial template is generated based on the input from domain expertise, which is a common practice in biology and life science. It contains the pre-determined partition based on context-specificity, e.g., topology regarding the natural environment, neuron shape, and function of compartments, which provides the biological interpretation. Hence, we develop the simultaneous registration and segmentation method for solving the part-wise neuron volume segmentation problem with artificial templates. In addition, our method also computes

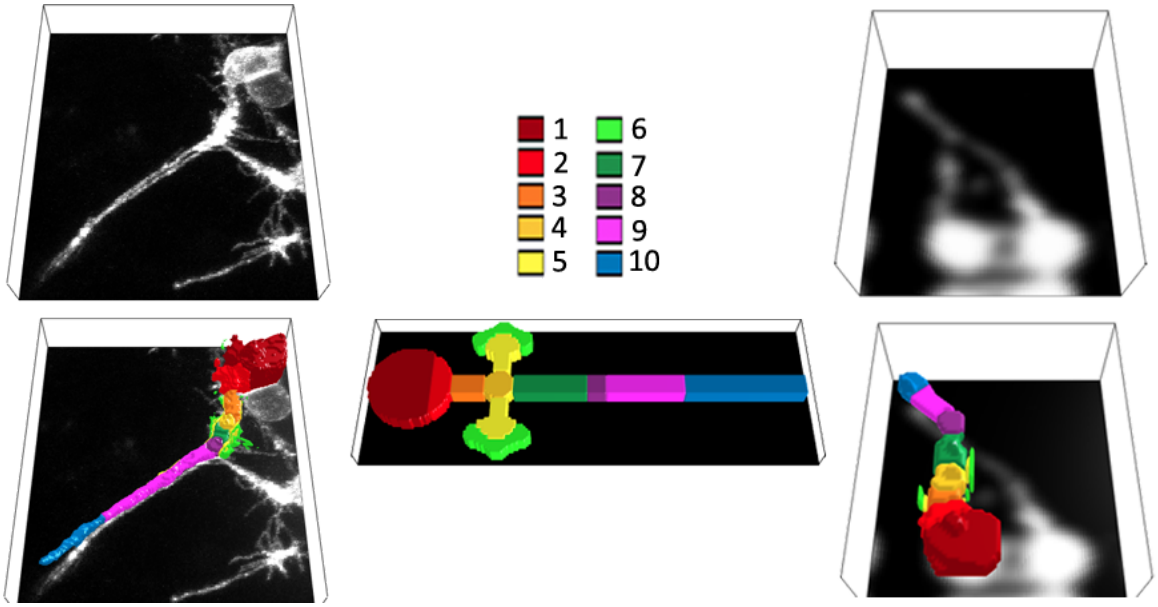


Figure 1.5. Part-wise volume segmentation problem with artificial templates. Given image stacks with different spatial resolutions (top) and the template (middle) partitioned according to the input from domain expertise. The goal is to separate input neuron volumes from their surroundings and partition them accordingly (bottom). Different colors correspond to different compartments.

the mapping function between compartments to combine information about protein interaction and morphology, and facilitate the study of the link between the functionality and structure of neurons. Our method follows the global-to-local approach: the template acts as the global feature, which constrains the part-wise segmentation derived from local features to ensure the interpretability.

Given the registration across multiple modalities, we can visualize the link between protein interaction and morphology/cellular complexity. In our exploration of aCC volumes, the protein interaction concentration map, which has low spatial resolution, can be mapped to the high resolution image of a neuron using our method (Fig. 1.6). As a result, we obtain the estimation of the protein interaction at the branch level. This provides the tool for neuroscientists to study the correlation and causality between protein interactions (proteome) and cellular complexity (phenome).

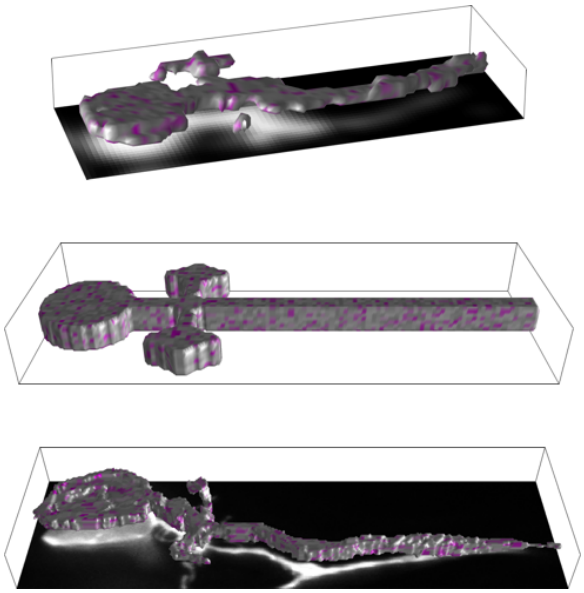


Figure 1.6. Mapping protein interaction across modalities through the artificial template. The protein interaction localization in the low resolution volume (top) is mapped to the artificial template (middle) using our part-wise segmentation method. Then, this step is repeated to map the template to the high resolution volume with branching details (bottom).

1.4 Challenges

Neuron reconstruction has been viewed as one of the most challenging tasks for computational neuroscience. The major challenge of automated neuron reconstruction for single image stack is mainly influenced by four factors: device limitations, noise, structural complexity, and structure variability [39].

Dendrites and spines are small compared to the achievable resolution from the imaging systems, so they are usually just a few voxels wide in the image stack. Their boundaries are often blurred because of the diffraction. In addition, the resolution along the depth of image stacks is usually low, especially for *in vivo* samples; otherwise, the integrity of the image stack might be compromised because image slices are taken in sequence.

The signal-to-noise ratio (SNR) of input image stacks can be poor. The neuronal structure can be discontinuous or occluded by undesired structures like parts of other neurons or body walls. The contrast inhomogeneity may occur within images, which is influenced by many causes such as non-uniform illumination and non-uniform intracellular distribution of fluorophores.

The structure of the neurons could be complex, especially when neurites are intertwined together. It is tricky to separate the adjacent neurites in such situations (e.g., crossovers and bifurcations). With the combination of complex structures and low resolution images, neurites could appear like knots that are highly ambiguous even to the human operators.

Finally, the high variability in the neuronal topology and appearance makes it difficult to model neurites. The number of bifurcations could range from a few to hundreds, while the thickness and length of neurites could vary within the same neuron. Due to the variation in both neuronal structure and appearance, it is difficult to create a robust mathematical model with the flexibility to represent all possible neurons.

Furthermore, the tracking problem in the time-lapse sequence of calcium images is challenging due to the characteristics of neuron's locomotive behavior and calcium imaging. Time-lapse image stacks contain image noise and out-of-focus regions, while neurons exhibit the inherent structural complexity and variability of neurites. In addition, calcium signals are sensitive to neuronal activities so neurons become invisible when they are stationary. Another issue is the locomotion characteristics of larvae, which produce movement pauses followed by severe deformations of neurons.

In the part-wise segmentation problem with artificial templates, the reference image is an artificially generated neuron model so the intensity of the reference image is unavailable. Although this is the segmentation problem, the partition must be done according to the template. Hence, the registration techniques can be applied between the segmentations of input images and the template to ensure that the constraint is complied. However, there are appearance differences between the reference and target images. So the conventional registration techniques cannot be applied here because they assume that input images have similar appearance.

1.5 Thesis Statement

The model-and-appearance-based analysis combines high- and low-level features derived from input volumes and domain knowledge to accurately extract neuronal morphology properties from multi-modal microscopy images.

1.6 Contributions and Dissertation Organization

This dissertation is organized into seven chapters including the introduction, background, four chapters that constitute our contributions, and summary. Our work contributes to both the computer science and biology communities. In the computer vision area, we develop novel model-and-appearance-based methods for solving four main tasks in the automated morphology analysis: segmentation, tracing, tracking, and part-wise segmentation with artificial templates. These methods enable the ad-

vance of neuroscience research by creating new techniques for computationally quantifying the anatomical aspect and new visualizations for links between properties of neurons. Moreover, their applications allow neuroscientists to investigate a nervous system at the most fundamental level, the single neuron resolution. The datasets used in this dissertation and details of imaging preparation of our datasets are described in Appendix B. The brief overview for each of these four chapters are provided below.

Chapter 3 introduces a novel segmentation method for time-lapse image stacks based on the co-segmentation principle. Our method aggregates information from multiple stacks to improve the segmentation task of individual stack, using a neurite model and a tree similarity term. The neurite model takes into account branching characteristics, such as local shape smoothness and continuity, while the tree similarity term exploits the local branch dynamics across image stacks. Our approach improves accuracy in ambiguous regions, handling successfully out-of-focus effects and branching bifurcations. It also detects the topological changes on neurons over the image sequence. We validate our method using *Drosophila* sensory neuron datasets and made comparisons with existing methods.

Chapter 4 presents two novel methods for three-dimensional neurite tracing using a population of open-curve active contour models: joint probability and conditional random field methods. Our aim is to increase the robustness under spatially varying neurite-background contrast. While most existing active contour model methods perform tracing by evolving snakes in a sequential manner, ours implement a simultaneous evolution. Our methods combine the active contour model with the probabilistic framework to incorporate the local shape model, image statistics, and the global shape encoded by the pairwise interaction among snakes based on their spatial proximity and shapes. This interaction helps resolve the connectivity of active contour population, resulting in increased accuracy in ambiguous regions (e.g., low contrast, neurite bifurcations and crossovers, etc.). We illustrate the performance of our methods and compare them with existing frameworks on publicly available datasets: wild-type sensory neurons in the larval *Drosophila*, DIADEM, and FlyCircuit datasets.

In Chapter 5, we present the novel neuron tracking method based on the local-to-global approach for time-lapse calcium images. Locomotion is controlled by sensory neurons, yet a fundamental, unanswered question is how this happens. We use time-lapse calcium imagery to observe and model the link between locomotion and sensory neuron activity in larval *Drosophila*. The main issue with calcium images is that they produce low responses when there is little or no neuronal activity. Here we use a neurite centerline tracking technique to tackle this issue, even under significant deformations during movement. Our method incorporates both local features (image appearance and local shape) and the global model (global shape and motion) in a Markov random field (MRF) framework. The objective function is optimized using Quadratic Pseudo-Boolean Optimization (QPBO) with α-expansion, which is also known as *fusion moves*. In our experiments, we illustrate how our method can track neurites in time-lapse calcium images under severe local intensity ambiguities.

Chapter 6 illustrates how to analyze multiple image stacks with different modalities by solving part-wise segmentation problem with artificial templates. We analyze neuronal morphology during development over volumes with different spatial resolutions using artificially created templates. Such templates serve as input from the domain expertise: a standardized representation of the neuron, independent from imaging modalities and resolutions, is what a neurobiologist can provide from knowledge and qualitative observations under the microscope. This is the first method that uses the totally artificial templates rather than the data-driven ones since it is a common practice in biology or life science for experts to describe knowledge qualitatively based on their observations. Moreover, computing data-driven template requires training data that captures sufficient variance of neuronal morphology, which is usually not true for the available neuron datasets. The artificial template is divided into context-specific (topology, shape, and/or function) compartments to provide the interpretable compartmentalization and comparison among neurons; and our task is to segment input neuron volumes from their surroundings and partition them accordingly. We solve this problem using our novel simultaneous registration

and segmentation method following the global-to-local approach. We first employ a global transformation that serves as part-wise alignment of the template with the input volume. Then, we apply the local, deformable compartment shape registration technique, the MRF-based free-form deformations (FFD). We validate our results using aCC motorneuron image stacks from larval *Drosophila*, at multiple developmental instances and different spatial resolutions.

2 BACKGROUND

In this chapter, we review a number of basic concepts for solving four fundamental problems in neuronal morphology analysis: segmentation (Sec. 2.1), tracing (Sec. 2.2), tracking (Sec. 2.3), and part-wise segmentation with artificial templates (Sec. 2.4). There has been extensive research in these four areas so the aim of this chapter is to provide the overview and contrast of related concepts, where a more comprehensive survey is provided in the following chapters. We also discuss the differences between existing concepts and our methods to illustrate the significance of our contributions.

2.1 Neuron Segmentation

The neuron segmentation is a boundary-based neuron reconstruction that detects the outline of a neuron. Given a temporal sequence of image stacks $\mathbf{X} = \{\mathbf{X}^0, ..., \mathbf{X}^T\}$ and a set of corresponding parameters, $\theta = \{\theta^0, ..., \theta^T\}$, the objective function for the segmentation problem over time-lapse data volume can be defined by,

$$E(\mathbf{X}) = \sum_t E_{spatial}\left(f(\mathbf{X}^t, \theta^t)\right) + E_{temporal}\left(f(\mathbf{X}, \theta)\right) \tag{2.1}$$

where f is the feature extraction function from input volumes. $E_{spatial}$ is the energy within an individual image stack, while $E_{temporal}$ is the energy across image stacks. Here we discuss three existing concepts for segmentation over time-lapse image stacks: graph registration, image registration, and co-segmentation.

Graph registration exploits the underlying tree data structure of neurons by formulating the neuron segmentation problem as the registration problem over graphs [40–44]. This scheme adopted a function f that converts the input volume $\mathbf{X}^t$ into a graph. $E_{spatial}$ acts as the measurement of how well the dendritic tree model fits the

input volume, while $E_{temporal}$ defines the similarity criteria for aligning tree structures across image stacks.

Image registration selects the mapping function that maximizes the neuron image similarity, measured by $E_{temporal}$, among segmented neuron images, produced by $E_{spatial}$. A choice of similarity criteria, such as residual sum of squares or normalized cross-correlation [45], depends on the type of algorithm, which can be feature-based or intensity-based or the combination of both. In feature-based algorithms, a function f extracts feature points from the neuron volume. Instead, the function f in intensity-based methods computes features from the image intensity of every pixel/voxel.

Co-segmentation is normally defined as the task of jointly segmenting 'similar things' in a given set of images [46]. Similarly, neuron co-segmentation is defined as the joint segmentation problem of similar neurites over a set of neuron images. In this scheme, $E_{spatial}$ generates neurite-like region proposals, while $E_{temporal}$ ensures the co-existence of neurites across images. The co-segmentation problem can be solved using Markov random field (MRF) [19] or conditional random field (CRF) [47] frameworks.

MRF is the undirected graphical model following the probabilistic generative framework (Fig. 2.1). It models the joint probability of the observed data and the corresponding labels. Given the observed data like image stack in our case, we divide the observed data into sites. For example, the image stack can be divided into one voxel per site, where each site has the corresponding label or configuration. In the neuron co-segmentation problem, the site's label is either neurite or background.

Let $\mathbf{X} = \{x_i\}_{i \in S}$ denote the observed data divided into a set of site S, where x_i is the data from ith site. Let $\mathbf{Y} = \{y_i\}_{i \in S}$ denote the set of the corresponding labels. The MRF framework follows the Bayes' rule,

$$P(\mathbf{Y}|\mathbf{X}) \propto P(\mathbf{X}, \mathbf{Y}) = P(\mathbf{Y})P(\mathbf{X}|\mathbf{Y}) \tag{2.2}$$

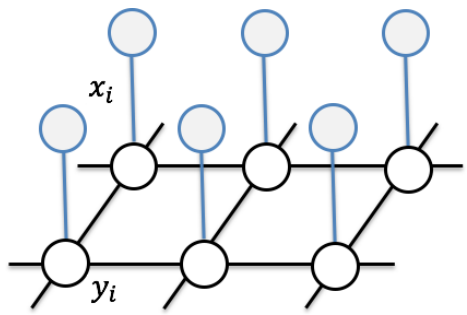


Figure 2.1. Markov random field. The observed data is divided into sites x_i (blue circles) with the corresponding label or configuration y_i (black circles).

where the likelihood model $P(\mathbf{X}|\mathbf{Y}) = \prod_{i \in S} P(x_i|y_i)$ is factorized by sites. While the prior $P(\mathbf{Y})$ is modeled by the product of potential function ψ_C over the maximal cliques $\mathbf{y}_C$ of the graph [48],

$$P(\mathbf{Y}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{y}_C) \tag{2.3}$$

where Z is the partition function that ensures the distribution $P(\mathbf{Y})$ is correctly normalized. Then, we substitute the likelihood and the prior into eq. (2.2),

$$P(\mathbf{X}, \mathbf{Y}) = \frac{1}{Z} \prod_C \psi_C(\mathbf{y}_C) \prod_{i \in S} P(x_i|y_i) \tag{2.4}$$

Given the energy function E_C of the maximal clique, $\psi_C(\mathbf{y}_C) = \exp\{-E_C(\mathbf{y}_C)\}$, the above equation can be rewritten as:

$$P(\mathbf{X}, \mathbf{Y}) = \frac{1}{Z} \exp\left\{ -\sum_C E_C(\mathbf{y}_C) + \sum_{i \in S} \log P(y_i|x_i) \right\} \tag{2.5}$$

Following the Hammersley-Clifford theorem [49], we assume only up to pairwise clique potentials to be nonzero [48]. We can rearrange the equation into the following:

$$P(\mathbf{Y}|\mathbf{X}) \propto \frac{1}{Z} \exp\left\{ \sum_{i \in S} A(x_i, y_i) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} I(y_i, y_j) \right\} \tag{2.6}$$

where $\mathcal{N}_i$ is the set of neighbors of y_i. A is the association potential or the data cost term, and I is the interaction potential or the smoothing term [50].

Given the above formulation, the inference is the problem of finding the optimal labels $\mathbf{Y}$ of all sites with respect to the cost function in eq. (2.6). There are many methods for inference in MRF, e.g., ICM [51], HCF [52] graph cuts [53], QPBO [54], and fast-PD [55].

CRF is another undirected graphical model, similar to MRF, but the difference is that CRF is the discriminative model that directly models the conditional distribution over labels, i.e., $P(\mathbf{Y}|\mathbf{X})$. One advantage of CRF over MRF is that dependencies among the input variables $\mathbf{X}$ do not need to be explicitly represented because the model is conditional, which allows factors on both the individual sites and edges to depend on rich global features of the input [56] (Fig. 2.2). Lafferty et al. in 2001 [57] gave the following formal definition:

> Let $G = \{S, E\}$ be a graph such that $\mathbf{Y} = \{y_i\}_{i \in S}$, so that $\mathbf{Y}$ is indexed by the vertices of G. Then $(\mathbf{X}, \mathbf{Y})$ is a "conditional random field" in case, when conditioned on $\mathbf{X}$, the random variables y_i obey the Markov property with respect to the graph: $P(y_i \mid \mathbf{X}, y_j, i \neq j) = P(y_i \mid \mathbf{X}, y_j, i \sim j)$, where $i \sim j$ means that i and j are neighbors in G.

According to the above definition, CRF can be written as the conditional distribution. Let $\mathbf{X}$ denote the observed data and $\mathbf{Y} = \{y_i\}_{i \in S}$ denote the set of the

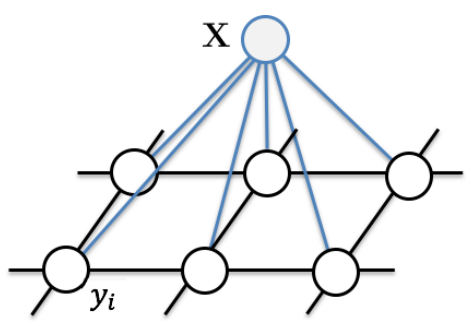


Figure 2.2. Conditional random field. All labels or configurations y_i (black circles) depends on the observed data $\mathbf{X}$ (blue circle).

corresponding labels, as either neurite or background for the neuron co-segmentation problem. Let G be a factor graph over $\mathbf{Y}$, then the conditional probability is the product of factors $\Psi_a \in G$ [56],

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z} \prod_{\Psi_a \in G} \Psi_a(x_a, y_a) \tag{2.7}$$

where x_a and y_a are data sites and their corresponding labels in the factor Ψ_a, and Z is the partition function that ensures the distribution $P(\mathbf{Y}|\mathbf{X})$ is correctly normalized. Similarly to MRF, using the Hammersley-Clifford theorem [49], the CRF framework is defined as the conditional distribution over the label given the observed data [50],

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z} \exp\left\{ \sum_{i \in S} \left(A(y_i, \mathbf{X}) + \sum_{i \in S} \sum_{j \in \mathcal{N}_i} I(y_i, y_j, \mathbf{X}) \right) \right\} \tag{2.8}$$

There are two main differences between CRF in eq. (2.8) and MRF in eq. (2.6). First, the association potential in MRF at any site y_i is a function of the associated observation x_i, while in CRF, it is a function of all observation $\mathbf{X}$. The second difference is that the interaction potential for each pair of sites in MRF is a function of their corresponding labels, whereas in CRF, the interaction potential also includes the observation $\mathbf{X}$ along with labels of a pair of sites. The discriminative approach allows the modeling of arbitrary interactions between observed data and labels in a principled manner, which enables arbitrary potentials dependent on the whole observation [50].

In CRF, the inference is the problem of finding the optimal labels $\mathbf{Y}$ over the sites with respect to the cost function in eq. (2.8). Most inference methods for MRF can be modified for the inference in CRF as well.

Our MRF-based neuron co-segmentation. Our contributions in the neuron segmentation task are the introduction of the novel function f and energy function $E_{temporal}$. Our function f exploits the local branch characteristics of neurons to improve neurite detection, while our energy function $E_{temporal}$ enforces the coherency across images at the structure level, unlike conventional methods that ensure the coherency at the appearance level. As a result, our structure level coherency improves the accuracy over branching bifurcations.

2.2 Neuron Tracing

The neuron tracing is the centerline-based neuron reconstruction that extracts the tree model of the neuronal topology. Here we discuss two main types of tracing scheme for data volumes with spatially varying contrast: global and local methods.

2.2.1 Global Methods

Global methods extract underlying tree data structures of neurons using the signal distribution of the entire input images [58]. Given the input image stack $\mathbf{X}$ and the parameter θ, global methods optimize the following objective function,

$$E(\mathbf{X}) = E_{global}\big(f(\mathbf{X}, \theta)\big) \tag{2.9}$$

where E_{global} is the energy derived from the global signal distribution. Global methods can be categorized into skeletonization, minimal path, and graph-based methods.

Skeletonization is the task of computing the binary image of the neuron centerline from input volumes. In this scheme, a function f produces the segmented image on which the energy function E_{global} applies the medial axis transform to extract the topological skeleton [59].

Minimal path-based methods extract neurite traces by computing the geodesic path between seed points, usually solved by the fast marching method [60]. In this framework, the function f extracts seed points from input volume, while the energy function E_{global} connects these seed points using minimal paths to recover the tree structure of neurons.

Graph-based methods solve the neurite tracing problem using graph theory on the tree-like structure of a neuron. According to this scheme, the function f converts $\mathbf{X}$ into a graph, where the energy function E_{global} selects the optimal graph with respect to some criteria based on the neuron's model [61, 62].

2.2.2 Local Methods

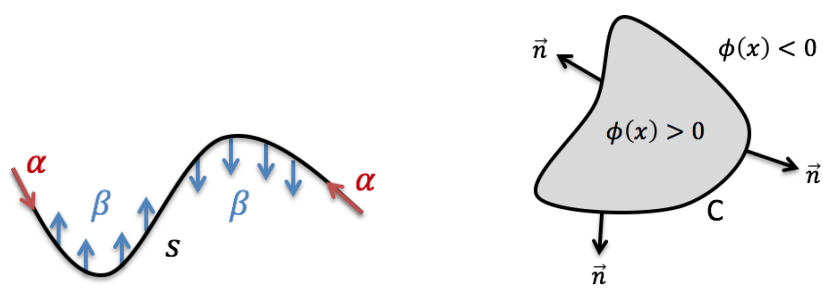
Local methods carry out the neuron tracing process based on the signal distribution of local regions. Given the subregion of image stack, $\mathbf{X}_i \subset \mathbf{X}$, and its corresponding parameter θ_i, the objective function of local methods is defined by,

$$E(\mathbf{X}) = \sum_i E_{local}\big(f(\mathbf{X}_i, \theta_i)\big) \tag{2.10}$$

where E_{local} is the energy derived from the signal distribution of local regions.

Sequential tracing is the repetitious framework that traces the vicinity of seed points iteratively until the neuronal morphology is obtained. This framework employs a function f that measures the compatibility of the neuron segment model and the subvolume, while the energy function E_{local} selects the best-fit parameter θ_i [58, 63].

Deformable models select the neuron's boundary based on local image features that satisfies constraints with respect to the shape of the neuron. There are two main types of deformable models based on the curve representation: parametric [64–66] and geometric [67–69].



(a) (b)

Figure 2.3. Parametric (left) and geometric (right) deformable models. (a) The open-snake s has α and β parameters to regularize the 'elasticity' and 'rigidity' of the curve. (b) The level set ϕ implicitly represents the curve C and it is evolved in the direction of the normal unit vector $\vec{n}$.

Parametric deformable model or active contour model, usually denoted by snake, was first introduced in [64]. It represents the region of interest (ROI) by the curve delineating its boundary. For the curvilinear structure like neuron centerline, the ROI is represented by the open curve s, called the open-snake (Fig. 2.3(a)). Given the snake curve s and the snake parameter encoding its normalized arc-length t, the corresponding position is $s(t) = \{x(t), y(t), z(t)\}$. The snake deforms towards the ROI, while it ensures the smoothness of the curve by minimizing the following energy function, where $\mathbf{S}$ is the set of snakes:

$$\begin{aligned} E_{local}(\mathbf{X}) &= \sum_{s \in \mathbf{S}} E_{snake}(s) \\ E_{snake}(s) &= \int_0^1 E_{int}(s)dt + \int_0^1 E_{ext}(s)dt \end{aligned} \tag{2.11}$$

The snake energy E_{snake} is composed of the internal energy E_{int} and the external energy E_{ext} [70]. The internal energy regularized the snake curve by enforcing the curve smoothness through the elasticity and rigidity, where α and β are the elasticity and rigidity weights:

$$E_{int}(s) = \underbrace{\alpha(t)|\nabla s(t)|^2}_{elasticity} + \underbrace{\beta(t)|\nabla^2 s(t)|^2}_{rigidity} \tag{2.12}$$

The elasticity term uses the first-order derivative to contract the snake when it is stretched. The rigidity employs the second-order derivative to restrict the snake from bending. While the external energy guides the snake curve towards the salient feature based on the edge detector function, $g\colon \Omega \to \mathbb{R}^+$, that gives high value only at the vicinity of the edge,

$$E_{ext}(s) = -|g(s(t))|^2 \tag{2.13}$$

The above formulation can be minimized using the gradient descent algorithm, where the derivative can be analytically solved using the variational calculus. The function E_{snake} can be redefined as,

$$E_{snake}(s) = \int_0^1 F(t, s, s', s'') \quad dt \tag{2.14}$$

where we abbreviate function s and its derivatives for readability: $s = s(t)$, $s' = \nabla s(t)$, $s'' = \nabla^2 s(t)$. Following these definitions, the change of the energy function E_{snake} with respect to s can be described by the Taylor expansion,

$$\delta E_{snake}(s) \approx \int_0^1 \left(\frac{\partial F}{\partial s} \cdot \delta s + \frac{\partial F}{\partial s'} \cdot \delta s' + \frac{\partial F}{\partial s''} \cdot \delta s'' \quad dt \right. \tag{2.15}$$

Then, we break down $\delta s'$ and $\delta s''$ by the chain rule,

$$\delta E_{snake}(s) \approx \int_0^1 \left(\frac{\partial F}{\partial s} \cdot \delta s + \frac{d}{dt}\frac{\partial F}{\partial s'} \cdot \delta s + \frac{d^2}{dt^2}\frac{\partial F}{\partial s''} \cdot \delta s \quad dt \right. \tag{2.16}$$

At minimum, $\delta E_{snake} = 0$,

$$\frac{\partial F}{\partial s} + \frac{d}{dt}\frac{\partial F}{\partial s'} + \frac{d^2}{dt^2}\frac{\partial F}{\partial s''} = 0 \tag{2.17}$$

In 3D space, assuming that $\alpha(t) = \alpha$ and $\beta(t) = \beta$ are constant, substituting eq. (2.14) in eq. (2.17) gives rise to the Euler equation,

$$\frac{\partial E_{ext}}{\partial s} + \alpha s'' + \beta s'''' = 0 \tag{2.18}$$

The derivatives of snakes are approximated using the finite difference method, $s(t) \approx v_i = \{x_i, y_i, z_i\}$ [64]. Then, eq. (2.12) can be rewritten as,

$$E_{int}(i) = \alpha_i \frac{|v_i - v_{i-1}|^2}{2h^2} + \beta_i \frac{|v_{i-1} - 2v_i + v_{i+1}|^2}{2h^4} \tag{2.19}$$

Thus, the Euler equation in eq. (2.18) is,

$$\begin{aligned} &\alpha_i |v_i - v_{i-1}| - \alpha_{i+1} |v_{i+1} - v_i| \\ &\quad + \beta_{i-1} |v_{i-2} - 2v_{i-1} + v_i| \\ &\quad - 2\beta_i |v_{i-1} - 2v_i + v_{i+1}| \\ &\quad + \beta_{i+1} |v_i - 2v_{i+1} + v_{i+2}| \\ &\quad + \frac{d}{dt} E_{ext} = 0 \end{aligned} \tag{2.20}$$

It can be rewritten in the matrix form into three equations as,

$$\begin{aligned} Ax + \frac{d}{dx} E_{ext} &= 0 \\ Ay + \frac{d}{dy} E_{ext} &= 0 \\ Az + \frac{d}{dz} E_{ext} &= 0 \end{aligned} \tag{2.21}$$

Following the gradient descent algorithm, the position of snake curve at time n given time $n-1$ is optimized by,

$$\begin{aligned} Ax_n + \frac{d}{dx}E_{ext}(x_{n-1}, y_{n-1}, z_{n-1}) &= -\gamma(x_n - x_{n-1}) \\ Ay_n + \frac{d}{dy}E_{ext}(x_{n-1}, y_{n-1}, z_{n-1}) &= -\gamma(y_n - y_{n-1}) \\ Az_n + \frac{d}{dz}E_{ext}(x_{n-1}, y_{n-1}, z_{n-1}) &= -\gamma(z_n - z_{n-1}) \end{aligned} \tag{2.22}$$

Rearranging the above equations gives the update function of the snake curve position for minimizing the snake energy function E_{snake} at each iteration n [23],

$$\begin{aligned} x_n &= (A+\gamma I)^{-1}\left(\gamma x_{n-1} - \frac{\partial E_{ext}(x_{n-1}, y_{n-1}, z_{n-1})}{\partial x}\right) \\ y_n &= (A+\gamma I)^{-1}\left(\gamma y_{n-1} - \frac{\partial E_{ext}(x_{n-1}, y_{n-1}, z_{n-1})}{\partial y}\right) \\ z_n &= (A+\gamma I)^{-1}\left(\gamma z_{n-1} - \frac{\partial E_{ext}(x_{n-1}, y_{n-1}, z_{n-1})}{\partial z}\right) \end{aligned} \tag{2.23}$$

On the other hand, if snakes are discretized then eq. (2.11) can be solved straightforwardly using dynamic programming [71] or greedy algorithm [72], where derivatives can be approximated using the finite difference method.

Geometric deformable model or level set method is originally introduced in [73]. It is built upon the parametric deformable model to overcome the mathematical challenge, where the boundary curve and input images have a different number of dimensions, i.e., snake splines are in 1D while neuron images are in 2D (or 3D for volumes). To avoid this issue, the curve C delineating the ROI is represented implicitly via a Lipschitz function ϕ, $C = \{x \mid \phi(x) = 0\}$ (Fig. 2.3(b)). There are numbers of variations of level set methods, but notable models are Mumford-Shah [74], Chan-Vese [68], and geodesic active contours [67]. The main difference between these models is their assumption about the ROI.

Given the function for extracting features from an image stack $f\colon \Omega \to \mathbb{R}$, the Mumford-Shah model [74] optimizes the following piecewise-smooth function,

$$\arg\min_{u,C} \mu \cdot Length(C) + \lambda \cdot \int_{\Omega} \big(f(x) - u(x)\big)^2 dx + \int_{\Omega \setminus C} |\nabla u(x)|^2 dx \tag{2.24}$$

where u is the solution image formed by smooth regions with constant intensity. μ and λ are model parameters regularizing the length of object and balancing between the regularization and image feature respectively.

On the contrary, the Chan-Vese model [68] follows the piecewise-constant formulation,

$$\begin{aligned} \arg\min_{c_1,c_2,C} \mu \cdot Length(C) + \nu \cdot Area(inside(C)) \\ + \lambda_1 \iint_{inside(C)} |f(x) - c_1|^2 dx \\ + \lambda_2 \iint_{outside(C)} |f(x) - c_2|^2 dx \end{aligned} \tag{2.25}$$

where c_1 and c_2 are two possible value of u. ν is the regularization parameter on the enclosed area. λ_1 and λ_2 are model parameters tuning the image feature inside and outside of C respectively.

The key differences between the Chan-Vese and Mumford-Shah models are the additional regularization on the enclosed area and the removal of the smoothness term, which is compensated by the simplification that u have only two possible values: c_1 and c_2 [75].

The geodesic active contour evolves the curve based on the edge information, unlike the Mumford-Shah and Chan-Vese models that incorporate only the region information. Its energy function is derived from the parametric active contour model, which is defined by the following [67]:

$$\min_{C} \iint_{0}^{1} g\big(C(t)\big)|C'(t)|\, dt \tag{2.26}$$

In this work, we follow the Chan-Vese model for representing the boundary of a neuron since it is simpler than the Mumford-Shah model [75]. The geodesic active contour can be integrated to incorporate the edge information [76]. Given the image stack, the objective function in eq. (2.25) is redefined using the level set function ϕ,

$$
\begin{aligned}
E_{local}(\mathbf{X}) = \quad & \mu \int_{\Omega} \delta\big(\phi(x)\big)|\nabla\phi(x)|dx + \nu H\big(\phi(x)\big) \\
& + \lambda_1 \int_{\Omega} |f(x) - c_1|^2 H\big(\phi(x)\big)dx \\
& + \lambda_2 \int_{\Omega} |f(x) - c_2|^2 \Big(1 - H\big(\phi(x)\big)\Big)dx
\end{aligned} \tag{2.27}
$$

where H is the Heaviside function and δ is the Dirac measure.

$$
\begin{aligned}
H(z) &= \begin{cases} 1 & , z \geq 0, \\ 0 & , z < 0, \end{cases} \\
\delta(z) &= \frac{d}{dz} H(z)
\end{aligned} \tag{2.28}
$$

The objective function is minimized with respect to c_1 and c_2 by keeping ϕ fixed,

$$
\begin{aligned}
c_1 &= \frac{\int_{\Omega} f(x) H\big(\phi(x)\big)dx}{\int_{\Omega} H\big(\phi(x)\big)dx} \\
c_2 &= \frac{\int_{\Omega} f(x)\Big(1 - H\big(\phi(x)\big)\Big)dx}{\int_{\Omega}\Big(1 - H\big(\phi(x)\big)\Big)dx}
\end{aligned} \tag{2.29}
$$

H needs to be a continuous function so the level set can be evolved using the gradient descent algorithm, and its derivatives can be computed using the variational calculus; the Heaviside function and the Dirac measure are thereby approximated by,

$$
\begin{aligned}
H_\epsilon(z) &= \frac{1}{2}\left(1 + \frac{2}{\pi}\arctan\left(\frac{z}{\epsilon}\right)\right) \\
\delta_\epsilon(z) &= \frac{\epsilon}{\phi(\epsilon^2 + z^2)}
\end{aligned} \tag{2.30}
$$

With these definitions, we update ϕ to minimize eq. (2.25) at time t of the level set curve evolution and solve its derivative in eq. (2.27) using the variational calculus,

$$
\begin{aligned}
\frac{\partial\phi}{\partial t} &= \delta_\epsilon(\phi)\left[\mu \operatorname{div}\left(\frac{\nabla\phi}{|\nabla\phi|}\right) - \nu - \lambda_1\big(f(\mathbf{X}) - c_1\big)^2 + \lambda_2\big(f(\mathbf{X}) - c_2\big)^2\right] \\
\text{subject to} \quad & \frac{\delta_\epsilon(\phi)}{|\nabla\phi|}\frac{\partial\phi}{\partial\vec{n}} = 0 \qquad \text{on the boundary}
\end{aligned} \tag{2.31}
$$

where $\vec{n}$ denotes the normal vector. There are many numerical methods [68, 77–79] for computing eq. (2.31).

Our neuron tracing method combines both local and global signal distributions through the set of parametric deformable models, and obtains the best of both worlds. The local features enable spatially adaptive filtering to enhance neuron detection in an efficient manner, while the global features derived from interactions among active contour models ensure the structural coherency to improve the topology recovery.

2.3 Neuron Tracking

The neuron tracking traces the neuron centerlines over time-lapse sequence in order to capture the changes in its morphology. Given a temporal sequence of image stacks $\mathbf{X} = \{\mathbf{X}^0, ..., \mathbf{X}^T\}$ and a set of corresponding parameters, $\theta = \{\theta^0, ..., \theta^T\}$, the objective function of tracking problem is defined by,

$$E(\mathbf{X}) = \sum_t \left\{ E_{app}\big(f(\mathbf{X}^t, \theta^t)\big) \right\} + E_{model}\big(f(\mathbf{X}, \theta)\big) \tag{2.32}$$

where E_{app} is the appearance energy of individual image stack, and E_{model} ensures temporal smoothness over the consecutive frames. Here we discuss three existing concepts for tracking over time-lapse calcium images: tracking-by-detection, articulated neuron, and optical flow.

Tracking-by-detection. This scheme traces every neuron image stack independently, e.g., [80, 81], and applies registration technique, e.g., [44], to enforce temporal smoothness. The energy function E_{app} is the tracing criteria and E_{model} is the registration metric for aligning neuron traces. In this framework, these two problems are decoupled and they are optimized in sequence.

Articulated neuron. In this framework, the neuron's tree is modeled by the kinematic articulated structure [82, 83]. A function f fits the neuron's model with configuration parameter θ to the data volumes. While E_{app} computes the optimal neuron's model and E_{model} ensures the smooth transition between consecutive time steps.

Optical flow is the estimation of a dense motion field corresponding to the displacement of each pixel in neuron volume [84]. It gives the low-level motion analysis of intensities in the image plane, which provides the foundation for a more sophisticated analysis. Optical flow usually follows the Brightness Constancy Constraint Equation (BCCE). At a given voxel $x \in \Omega$ at time t, BCCE is defined by,

$$\frac{df}{dt}(x,t) = 0 \tag{2.33}$$

where $f\colon \Omega \times \mathrm{T} \to \mathbb{R}$ is the feature calculated from image sequence. Ω is the image domain and T is the sampled time interval from the sequence. The discrete approximation of eq. (2.33) is given by,

$$f(x + w(x), t + 1) - f(x,t) = 0 \tag{2.34}$$

where $w\colon \Omega \to \mathbb{R}$ is the motion field at time t of the image $\mathbf{X}^t$. In this framework, E_{app} computes the optimal motion field w, while E_{model} enforces the temporal intensity smoothness constraint.

Our local-to-global approach models the neuronal morphology using the kinematic articulated structure with novel constraints, which take into account the motion and global shape of a neuron. The integration of motion into the energy function E_{model} helps detect neurons even when they are invisible, as occur in noisy calcium images. While local features are constrained by the global shape through E_{app} to cope with severe deformations from neuron's movement behavior.

2.4 Part-Wise Neuron Volume Segmentation with Artificial Templates

This task is to segment input neuron volumes from their surroundings and partition them according to templates. They are artificial and derived from the domain knowledge to provide biologically meaningful comparison and registration. The templates cannot be created using a data-driven methodology such as maximum a posteriori (MAP) framework [85] and active shape model [86, 87] because the available

data is insufficient to capture enough variation to build the acceptable templates. Fortunately, it is a common practice in biology or life science for experts to describe knowledge qualitatively, where biologists draw the neuronal morphology template based on their observations. Hence, the man-made templates are used in this problem instead of the data-driven ones.

This problem can be solved by the simultaneous registration and segmentation (SRS) techniques; however, the template is only the label function and the registration over segmentations must be consistent with the partition. Hence, given the input volume $\mathbf{X}$ and the artificial template $\mathbf{A}$, the objective function is formulated as,

$$E(\mathbf{X}, \mathbf{A}) = E_{seg}\big(f(\mathbf{X}, \theta)\big) + E_{reg}\big(L(\mathbf{X}), \mathbf{T}[L(\mathbf{A})]\big) \tag{2.35}$$

where E_{seg} is the segmentation criteria and E_{reg} ensures the consistency between the segmentations of the input volume and the template. f is the feature extraction function, L is the label function of the volume, and $\mathbf{T}$ is the transformation function, e.g., transformation matrix, vector field, or free-form deformation (FFD).

FFD is the geometric technique for modeling the non-linear deformation [88] by enclosing the object (e.g., a neuron) with the lattice and deforming the lattice to transform the object (fig. 2.4). The deformation is defined over the set of control

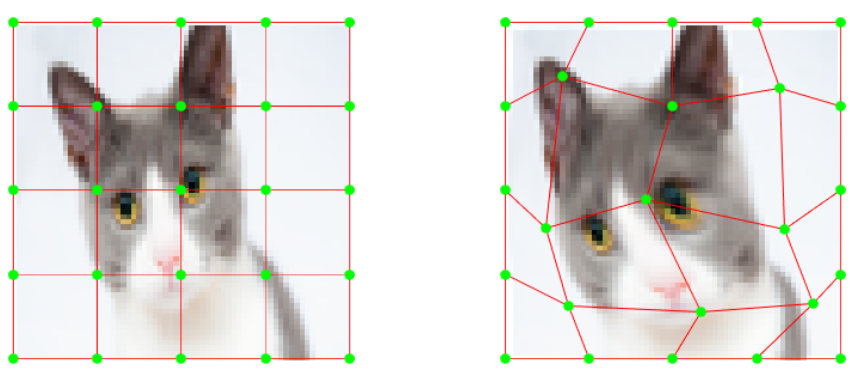


Figure 2.4. The 2D FFD of a cat image with 5×5 grid (lattice) control points (green dots). Before (left) and after (right) transformations.

points rather than every point in the space to reduce the number of parameters and ensure the space continuity. The popular configuration is the uniform cubic B-spline FFD because of its efficiency and flexibility, as well as a guarantee of continuity.

Let $\mathbf{T}(\mathbf{x})$ denote the 3D deformation at point $\mathbf{x}$, and $\mathbf{C}$ be the set of $n_x \times n_y \times n_z$ grid control points $\mathbf{c}_{i,j,k}$ with uniform spacing. Then, the transformation at $\mathbf{x}$ is defined by the combination of the grid control points as the tensor products of three 1D cubic B-splines,

$$\begin{aligned}
\mathbf{T}_n(\mathbf{x}) &= \sum_{\mathbf{c}\in\mathbf{C}} \eta(\mathbf{x},\mathbf{c})\mathbf{d}_{\mathbf{c}} \\
&= \sum_{p=0}^{3}\sum_{q=0}^{3}\sum_{r=0}^{3} B_p(u)B_q(v)B_r(w)\mathbf{c}_{i+p,j+q,k+r} \\
B_0(u) &= (1-u)^3/6 \\
B_1(u) &= (3u^3-6u^2+4)/6 \\
B_2(u) &= (-3u^3+3u^2+3u+1)/6 \\
B_3(u) &= u^3/6
\end{aligned} \tag{2.36}$$

where η is the coefficient of the control point $\mathbf{c}$ with displacement $\mathbf{d}_{\mathbf{c}}$ at point $\mathbf{x}$. $i = \lfloor x/n_x \rfloor - 1$, $j = \lfloor y/n_y \rfloor - 1$, and $k = \lfloor z/n_z \rfloor - 1$ are the index of control points. $u = x/n_x - \lfloor x/n_x \rfloor$, $v = y/n_y - \lfloor y/n_y \rfloor$, and $w = z/n_z - \lfloor z/n_z \rfloor$ are spline parameters. B_p represents the pth basis function of the uniform B-spline,

Existing SRS methods can be categorized into two main types based on their optimization techniques: variational-based and Bayesian-based.

Variational-based framework solves the SRS problem in the continuous space [22, 89, 90]. This framework usually decouples the neuron registration and neuron segmentation problems, and alternatively optimize E_{seg} and E_{reg}. The popular optimization technique for this framework is the gradient descent algorithm, where the derivatives are computed by variational calculus. Then, the energy functions E_{seg} and E_{reg} are updated by calculating $dE_{seg}/d\theta$ and $dE_{reg}/d\mathbf{T}$ respectively.

Bayesian-based framework follows the probabilistic framework by estimating the maximum a posteriori probability [85,91,92]. The framework operates in the discrete space to make the problem feasible. The benefit is that probabilistic framework has efficient inference techniques for computing the good approximation of the global optimum. In addition, the probabilistic framework enables the more sophisticated energy functions E_{seg} and E_{reg} since their derivatives are not required.

Our global-to-local approach adopts the Bayesian-based framework with the integration of the structure information (i.e., axonal centerline) into both energy functions E_{seg} and E_{reg}. Existing frameworks use only the appearance features of neurons, which are not suitable for this task because the artificial template can be very different to the input neurons in term of the appearance. However, the artificial template and input neurons share the same structure. Our framework employs the global-to-local approach: first, the global information from the artificial template is used to obtain the part-wise alignment. Then, the local registration is applied to align compartment boundaries.

3 NEURON SEGMENTATION FOR TEMPORAL IMAGE SEQUENCES

In this chapter, we introduce the novel segmentation method [93]. It is a boundary-based approach for extracting the neuron reconstruction. A video presentation of our work with supplemental material is available at `https://youtu.be/MqzPoIBDsps`. We improve the performance of the segmentation process by exploiting the occurrence of common neuron structures over time-lapse sequences. This problem is solved using the co-segmentation principle, which allows us to aggregate structure information across image stacks to improve the segmentation of individual volume, especially in ambiguous regions like branching bifurcations, and out-of-focus regions.

3.1 Segmentation of Neuronal Morphology

Studying the changes in neuronal topology is crucial for understanding the underlying biological processes in the nervous system, such as the functional mechanisms yielded from protein interactions. Another example is the wiring of motoneurons during the developmental stages. A complete anatomical description of larval motor neurons at single-cell resolution during the morphogenesis is required for observing how synaptic connectivities of motor circuits regulate the locomotion behavior of *Drosophila* [94]. Therefore, we apply the segmentation over time-lapse image stacks to model the morphological dynamics across developmental stages of neurons. Time-lapse neuronal stacks allow us to exploit the temporal continuity to improve the segmentation in a similar way to the methods in [24, 95].

Most existing neuron reconstruction methods rely on initialization in the form of either seed points or initial/approximate segmentation; this step usually exploits the notion of *vesselness*. Popular choices of *vesselness* are the Frangi filter [96] and the Optimally Oriented Flux (OOF) [97]. However, these methods do not provide robust-

ness under locally varying contrast between neurite and background, as it appears in our examined dataset. Here we aim to improve the segmentation of the neuron from surroundings over time-lapse image stacks with globally varying contrast.

To address this issue, we solve the neurite segmentation problem by incorporating information from temporally neighboring image stacks (successive instances during development) via the co-segmentation principle, in a seamless fashion. The idea of co-segmentation exploits common information about the appearance and/or shape of target regions throughout a given set (collection of images) [19, 98], instead of using prior knowledge in the form of training samples in a supervised classification manner. Most of co-segmentation approaches focus on dealing with object shape and appearance variations across images, as well as approximating the locations of commonly appearing objects. On the contrary, the goal in neurite co-segmentation is finding the correct branch correspondences, as a simultaneous tree segmentation and registration-like task, across temporally successive volumes.

In our co-segmentation framework, we introduce (a) a neurite model that tackles branch intensity 'gaps' by incorporating local characteristics, such as shape smoothness and continuity (Sec. 3.4), and (b) a tree similarity term that disambiguates neurite from background, especially at branching points, by enforcing coherency over time (temporally successive stacks) (Sec. 3.5). We evaluate our neuron co-segmentation method using the sensory neurons of *Drosophila* larvae dataset, which has spatio-temporally varying contrast background, and compare our results with *vesselness*-based segmentation (Sec. 3.6).

3.2 Related Work

One way to reconstruct neurites is to segment each image stack in the development sequence independently from temporally neighboring volumes. Then, one can register the results using graph-based or image-based registrations to enforce temporal smoothness and model the neuronal dynamics. Most of segmentation techniques

for a single image stack involve detecting vessel-like structures. There are numbers of vesselness filters such as [96, 97, 99–102]. Then, the binary segmentation can be obtained by thresholding the vesselness responses.

The method presented in [24, 95] shows that exploiting temporal continuity in tracing time-lapse neuronal stacks can yield more robust results. The co-segmentation framework can achieve the same goal of incorporating consistency within and across image stacks. Here we discuss three possible solutions to extend these vesselness filters for segmentation over time-lapse image stacks: graph-based registration, image-based registration, and co-segmentation.

3.2.1 Graph-Based Registration

The structure of neurons, i.e., centerline of neurite, that is extracted from the segmentation of the single input stack, can be represented by tree data structures. Hence, matching neuron branches becomes a graph registration problem. Some techniques in [40,41] match graph branch points based on Euclidean or geodesic distances, which are sensitive to changes in the graph appearance like length and size; therefore, those techniques are inappropriate for this task because of neuronal dynamics, which cause the changes in the morphology and appearance across image stacks. To deal with non-linear deformation, authors in [42, 43] proposed a non-rigid graph registration. They divide the registration process into two steps, coarse and fine alignments. In coarse alignment, they find the correspondence between branch and leaf nodes using Gaussian Process Regression (GPR) [103]. Then, they refine the edge alignment using the Hungarian algorithm [104]. Another approach in [44] aligns branches in the hierarchical fashion to model the neuronal dynamics.

Alternatively, the neuron reconstruction can be solved simultaneously over all image stacks in the time-lapse sequence by the method proposed in [24,95]. The method yields more robust results by exploiting temporal continuity and spatial consistency. However, all these methods relies heavily on the initialization step, either as detecting

seed points or extracting tree structures. This initialization process is computed in each image independently and it performs poorly on noisy datasets.

3.2.2 Image-Based Registration

Image-based registration computes the mapping by maximizing image similarity criteria such as normalized cross-correlation and mutual information [105, 106]. For example, the method in [107] finds the mapping that maximizes mutual information between the model and the image. Authors in [88] solved the registration problem by minimizing a combination of the cost associated with the transformation smoothness and image similarity. These methods assume that image stacks contain distinguishable objects; however, the varying signal-to-noise ratio (SNR) in our dataset violates this assumption, which makes this approach inappropriate to our task.

3.2.3 Co-Segmentation

Co-segmentation approach is typically defined as the task of jointly segmenting the similar regions in a given set of images [108]. There are many ways to tackle this problem. For instance, authors in [19] solved the problem using the Markov random field (MRF) framework, which separates foreground from background and maximizes the similarity between foreground features of two images. While, authors in [98] formulated this problem as the hard combinatorial optimization problem. They used spectral clustering to determine similarity between foregrounds and discriminative clustering to separate foreground and background. Some methods also incorporate region matching or registration to improve accuracy. For example, authors in [109] performed co-segmentation at pixel-level as well as region-level. They enforced the consistency between adjacent pixels and regions within the images, as well as the compatibility between pixels and assigned regions. While, authors in [110] applied the feature matching method based on hough transform and inverted hough transform. Then, they enforced the appearance consistency across images and the geometric

coherency of the object over correspondences. Another example is the method presented in [111], which uses the consistent functional maps to represent the relationship among the set of image stacks. The new representation permits efficient optimization and ensures the temporal consistency and the agreement between image cues and segmentation boundary.

Most of these approaches focus on dealing with the variation of objects and approximating their location; however, the main issue in neuron co-segmentation lies in finding the correct registration across volumes and locating the exact position of the *neurites*. So none of the previous methods are suitable for the neuron segmentation task.

3.3 Neuron Co-Segmentation

Let $\mathbf{X} = \{\mathbf{X}^t, \mathbf{X}^{t+1}\}$ be a pair of data volumes of the same neuron at two consecutive time instances; for simplicity, here we consider two volumes, although our framework can be extended to higher number of successive data. We postulate the co-segmentation framework as the MRF model [19], where sites are the voxels $x_i^t \in \mathbf{X}^t$, $x_l^{t+1} \in \mathbf{X}^{t+1}$, and the objective function at time t is,

$$E(\mathbf{X}^t) = (1-\alpha)\left(\left(\sum_{x_i^t \in \mathbf{X^t}} A(x_i^t) + \sum_{\substack{(x_i^t, x_j^t) \in \mathbf{X}^t, \\ i \neq j}} I(x_i^t, x_j^t)\right)\right) - \alpha \sum_{(x_i^t, x_l^{t+1}) \in \mathbf{C}} \left(H(x_i^t, x_l^{t+1}) \right) \quad (3.1)$$

where $\alpha \in [0, 1]$ is a model parameter regulating temporal smoothing, i.e., the continuity assumption across successive stacks (red line correspondences in Fig. 3.1).

Data penalty term A in eq. (3.1) is the cost of labeling voxels as 'neurite' or 'background', independently from the spatio-temporally neighboring voxels. It is defined as the negative logarithm of the product of image and model probabilities,

$$A(x_i^t) = -\ln P_{img}(x_i^t) - \ln P_{model}(x_i^t) \quad (3.2)$$

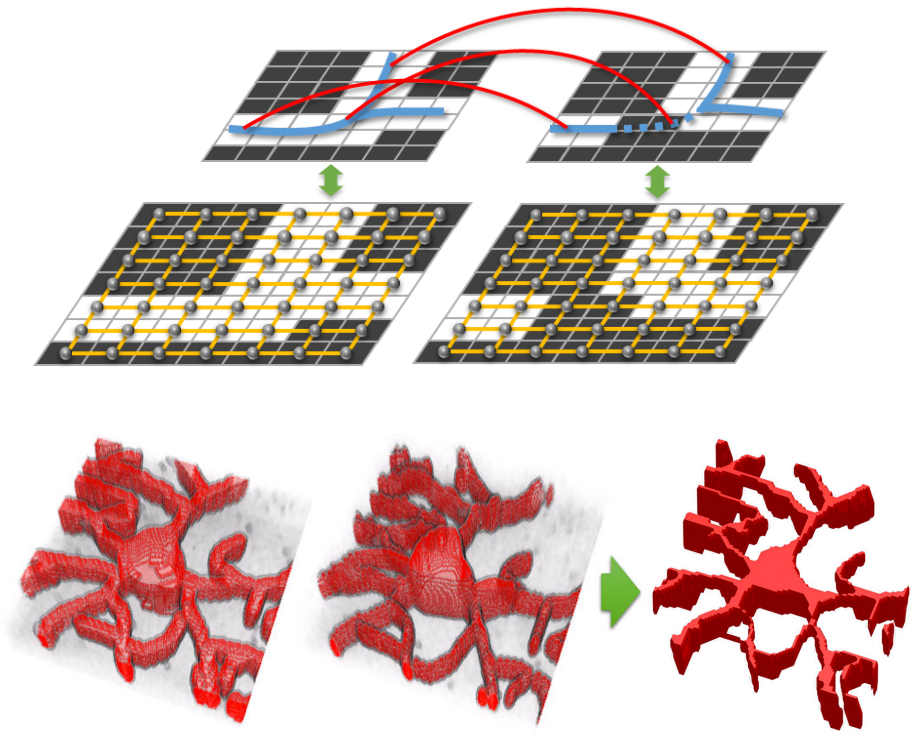


Figure 3.1. Overview of our approach. **Top**: For the two instances, the gray nodes, yellow edges, and red lines represent data, shape smoothness, and tree similarity terms respectively. The blue lines show the local neurite structure, while the dashed blue line illustrates correspondence with local structure in the previous instance. Introducing temporal co-segmentation aims at solving such conflicts in structure that raise from intensity ambiguities. **Bottom**: Volume rendering of the raw intensity of two image stacks at consecutive time steps (left); the ground truth is highlighted in red. The rightmost panel is the segmentation result for the first volume.

The image probability is the posterior distribution from a Gaussian Mixture Model (GMM) [112]: a GMM with K components is applied over the feature space f, which is composed by intensity and vesselness, and is computed by the OOF [97],

$$P_{img}(x_i^t) = \sum_{k=1}^{K} \left(\upsilon_k \frac{1}{\sqrt{2\pi\sigma_k^2}} \exp\left\{ -\left(\frac{[f(x_i^t) - \mu_k]^2}{2\sigma_k^2} \right\} \right) \right. \tag{3.3}$$

where w_k, μ_k, and σ_k are the weight, mean, and covariance of the Gaussian component k. The model probability P_{model} is the probability map generated by the neurite configuration/topology and is detailed in Section 3.4.

Smoothness penalty I in eq. (3.1) measures the cost of assigning different labels to two neighboring voxels within the same stack, and is defined as,

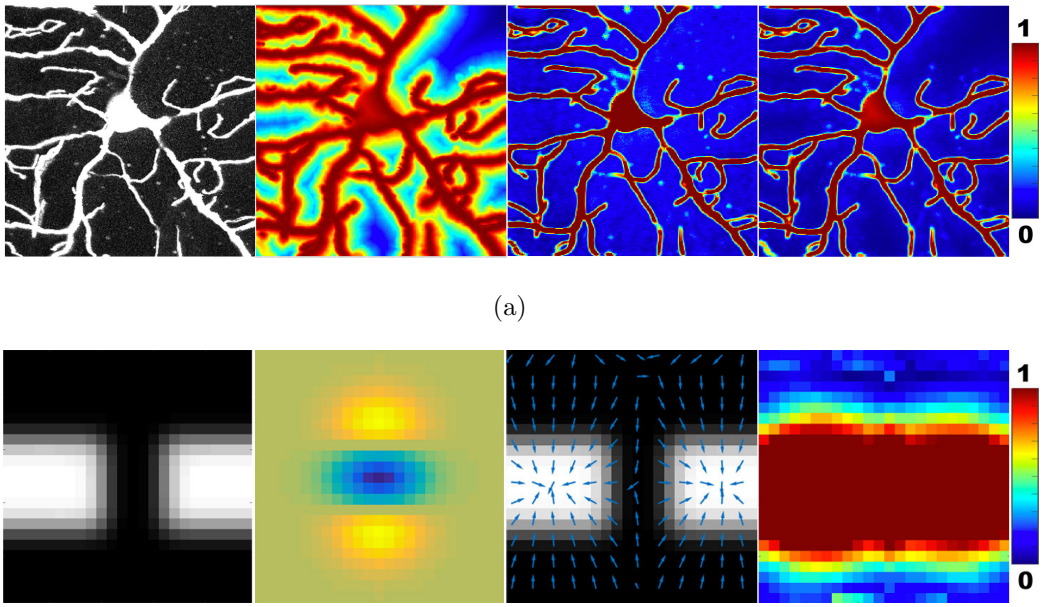
$$I(x_i^t, x_j^t) = \exp\left(-\left(\frac{\|f(x_i^t) - f(x_j^t)\|}{2\langle\|f(x_i^t) - f(x_j^t)\|\rangle}\right)\right) \tag{3.4}$$

where $\langle\cdot\rangle$ denotes the expectation value over all neighboring voxels within the stack [112].

Tree similarity function H in eq. (3.1) encodes the tree structure coherency of the neurite across image stacks, measured over a set $\mathbf{C}$ of temporally matched voxels between t and $t+1$, as detailed in Section 3.5.

3.4 Neurite Model

Our model probability, P_{model} in eq. (3.2), indicates regions that are likely to be part of the neurite, based on an intermediate segmentation outcome, and is similar to the interaction potential in [81]. We observe that neurites have two local characteristics: spatial continuity and local shape smoothness. Therefore, our model incorporates two main factors: proximity and orientation (Fig. 3.2). The proximity term captures nearby branches using the geodesic distance over intensity, which is computed by the fast marching method [60]. This term ensures spatial continuity of neurites, since it has a high value when there is a path towards foreground voxels of high intensity (Fig. 3.2(a)). It also contributes in removing isolated, high intensity voxels, which are usually noise or parts of other neurons in the background. The orientation term detects smooth paths between neurite parts in an intermediate segmentation result: the orientations of two branch segments indicate a smooth neurite path between them when the angles at their proximal, terminal voxels and the angle



(a)

(b)

Figure 3.2. Neurite model. (a) From left to right: original image, proximity term, orientation term, and probability map. (b) The orientation term; from left to right: synthetic image of a neurite gap, ridge filter response, VFC, and the orientation map.

under which one 'faces' the other are in agreement [81]. We calculate this term using the steerable filter in [113] at multiple scales along with the Vector Field Convolution (VFC) [114] (Fig. 3.2(b)). Alternatively one can first extract the centerlines of the segmented branches and then use the approach in [81].

We define the model probability at each voxel x_i^t as,

$$P_{model}(x_i^t) = \underbrace{\exp\big(-D(x_i^t)\big)}_{\text{Proximity}} \cdot \underbrace{R_{vfc}(x_i^t)}_{\text{Orientation}} \tag{3.5}$$

where D is the geodesic (unsigned) distance function from foreground voxels, while R_{vfc} is the ridge filter response along VFC.

3.5 Tree Similarity

The tree similarity term in eq. (3.1) enforces tree structure coherency across time-lapse stacks. Although neurite morphology does not change dramatically between successive time instances, variations in the appearance are commonly observed due to imaging-related issues (e.g., random out-of-focus effects). We enforce tree coherency when there is a conflict between paths across volumes, and specifically when a neurite path exists in one volume but not in the other. We define,

$$H(x_i^t, x_l^{t+1}) = \max\left\{P_{model}(x_i^t), P_{model}(x_l^{t+1})\right\}, \quad \forall (x_i^t, x_l^{t+1}) \in \mathbf{C} \tag{3.6}$$

where $\mathbf{C}$ is the set of all corresponding voxel pairs $x_i^t \in \mathbf{X}^t$ and $x_l^{t+1} \in \mathbf{X}^{t+1}$, such that either one or both of x_i^t, x_l^{t+1} belong to the imaged neurite (we ignore background voxels for efficiency).

Generating $\mathbf{C}$ requires volume registration as well as local branch matching, taking into account the neuronal dynamics. We find a mapping between the two tree

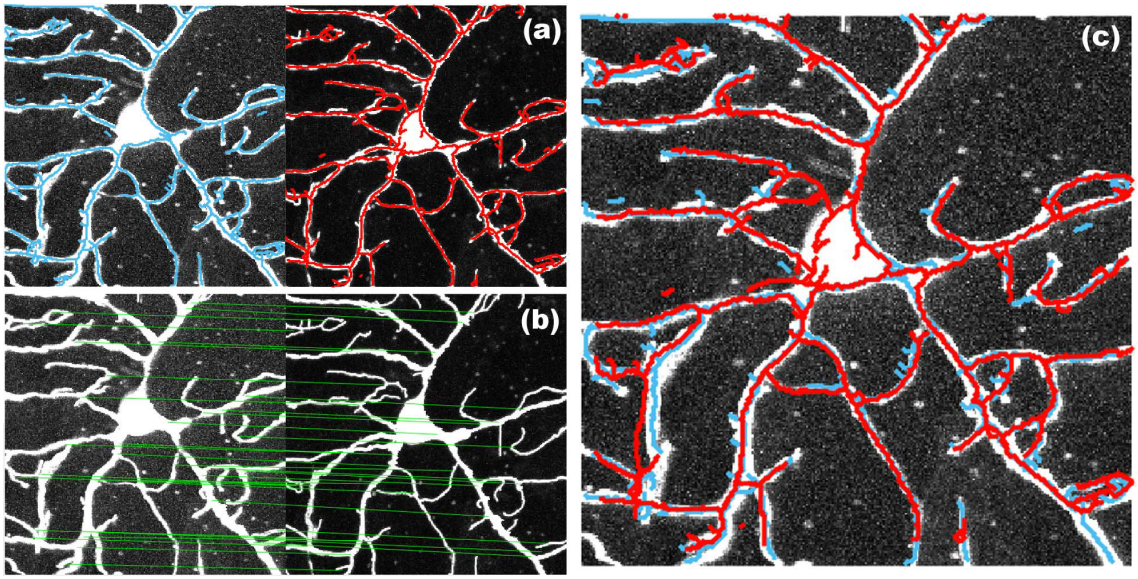


Figure 3.3. Tree Registration of two consecutive image stacks. (a) medial axes, (b) correspondence by thresholding NCC at branch points. and (c) GPR Geometric mapping.

structures, as approximated by the medial axis [59] in an initial/intermediate segmentation. We initialize registration by first aligning somata (neuron bodies existing in the stacks) and then applying a rigid transformation using the finite Iterative Closest Point (ICP) approach [115], assuming there is no significant global change in the neuron configuration (which holds given the imaging time intervals). We calculate sample correspondences through thresholding the Normalized Cross-Correlation (NCC) values between branching points [24]. Then, we apply Gaussian Process Regression (GPR) to find the non-linear mapping between two graphs based on these correspondences [42, 103] (Fig. 3.3), which yields the set $\mathbf{C}$.

3.6 Experiments

We evaluate our method on the larval *Drosophila* sensory neurons dataset, which contains four neuron samples imaged with three time instances each and we apply pair-wise co-segmentation over image stacks with consecutive time steps.

Our method was executed iteratively, where two iterations were enough in all experiments. For initialization, we used randomly thresholded OOF vesselness [97], and in all experiments the image probability was modeled by GMMs with 4 components, fitted by the Expectation-Maximization algorithm. The objective function in eq. (3.1) was solved using graph cuts [53]. For post-processing, we kept only the largest segment based on the assumption that the image stack contains only one neuron and a component represents a neuron.

We compare our results with the popular or state-of-the-art curvilinear structure detection methods such as OOF [97], Frangi filter [96], and the Multiscale Centerline Detection (MCD) [102]. Their segmentation outcomes are produced by thresholding responses with the optimal value for each image stack. The responses of the MCD are computed using the pre-trained model on Vivo2P, which is the dataset of the 3D *in vivo* two-photon images of a rat brain. We use the optimal threshold value for each data volume to show the best possible outcomes of these methods in the ideal

scenarios. Nevertheless, the following evaluation suggests that our method is superior to the competition.

Four evaluation criteria that we adopted in this work include the Dice index $\mathcal{D}(A, B) = 2|A \cap B|/|A| + |B|$, where A denotes the ground truth and B is a segmentation outcome ($D \in [0, 1]$), as well as precision/recall: precision is defined as the number of correctly classified foreground voxels over the total number of voxels predicted as foreground, and recall is defined as the number of correctly classified foreground voxels over the total number of ground truth foreground voxels. In addition, we also validate our results using the mutual information metric that measures the similarity between two segmentations. It is computed by the segmentation evaluation software in [116]. It was first used as a similarity measurement by Viola and Wells [107]. The advantage of the mutual information metric is that it is insensitive to errors that increase the recall, so it prefers the errors from enlarging the segmented volume over the errors from reducing the segmented volume. This characteristic is desired in this case because we want to recover the whole neuron from the image stack, whereas the incorrect background classification is more tolerable than missing a neurite. However, it also takes into account the true negatives, so the score tends to have small value since it is overwhelmed by the background as neurons occupy only a small portion of the data volume [116]. It is defined by,

$$MI(A, B) = H(A) + H(B) - H(A, B) \tag{3.7}$$

where H is the entropy of the ground truth A and the segmentation outcome B,

$$\begin{aligned} H(A) &= -\sum_i P(A^i) \log P(A^i) \\ H(B) &= -\sum_i P(B^i) \log P(B^i) \\ H(A, B) &= -\sum_{ij} P(A^i, B^j) \log P(A^i, B^j) \end{aligned} \tag{3.8}$$

Specifically, in the binary case, the entropy value is given by,

$$\begin{aligned} P(A^1) &= \frac{TP+FN}{n} \qquad & P(A^0) &= \frac{TN+FN}{n} \\ P(B^1) &= \frac{TP+FP}{n} \qquad & P(B^0) &= \frac{TN+FP}{n} \end{aligned} \tag{3.9}$$

where n is the total number of voxels, and A^0, A^1 are the background and foreground of A respectively, and the same goes for B. The joint probabilities are defined by,

$$\begin{aligned} P(A^1, B^1) &= \frac{TP}{n} \qquad & P(A^1, B^0) &= \frac{FN}{n} \\ P(A^0, B^1) &= \frac{FP}{n} \qquad & P(A^0, B^0) &= \frac{TN}{n} \end{aligned} \tag{3.10}$$

Note that we also tested the method in [101], which produced better results than the considered competition, however it took more than 24 hours to segment each $1024 \times 1024 \times 20$ image stack, on a Mac Pro (2×2.66 GHz 6-Core Intel Xeon, 20GB 1333MHz DDR3 ECC); the run time of our method was about 10 minutes per stack.

Figures 3.4 and 3.5 show the Dice index and mutual information results of our method, OOF, Frangi filter, and MCD for 11 different cases. Both metrics illustrate that our method outperforms the three competitions in all cases. The superiority of our results is due to three inherent properties of our method: (a) it successfully handles data with spatially inhomogeneous SNR, (b) it bridges neurite gaps sourcing from imaging inaccuracies, and (c) it exploits the 'best' local intensity and shape information among image stacks in a co-segmentation fashion. Figure 3.6 illustrates the effect of the tree similarity term that helps connect branches together and removes spurious voxels/segments. The limitation of our method is in handling highly cluttered branches that may be either over-simplified (under-segmented) by the neurite model (Sec. 3.4, Fig. 3.2), or mismatched during calculation of the local tree similarity. Therefore, as shown in Table 3.1, our method can yield lower precision, compared to the competition, however the recall is overwhelmingly higher, which renders the overall performance of our method better. Moreover, our method was able to segment ambiguous regions, such as bifurcation points, where the competition failed (Fig. 3.7).

Table 3.1.
Precision and recall of our method and the competitions. For these results, we empirically set the values of α in eq. (3.1): #1–5 : $\alpha = 0.5$, #6–8 : $\alpha = 0.06$, #9–11 : $\alpha = 0.1$.

Sample	Frangi		OOF		MCD		Ours	
Volume	P	R	P	R	P	R	P	R
#1	0.828	0.286	0.597	0.505	0.514	0.579	0.473	0.730
#2	0.935	0.172	0.606	0.529	0.412	0.509	0.558	0.669
#3	0.700	0.153	0.510	0.588	0.375	0.528	0.530	0.586
#4	0.359	0.215	0.277	0.363	0.213	0.385	0.345	0.572
#5	0.834	0.290	0.620	0.548	0.598	0.619	0.770	0.633
#6	0.446	0.424	0.375	0.502	0.389	0.473	0.352	0.759
#7	0.911	0.396	0.749	0.668	0.664	0.664	0.754	0.815
#8	0.817	0.541	0.691	0.669	0.578	0.676	0.547	0.906
#9	0.917	0.165	0.637	0.524	0.564	0.583	0.795	0.580
#10	0.929	0.173	0.640	0.566	0.676	0.586	0.778	0.707
#11	0.904	0.199	0.637	0.499	0.589	0.616	0.745	0.655

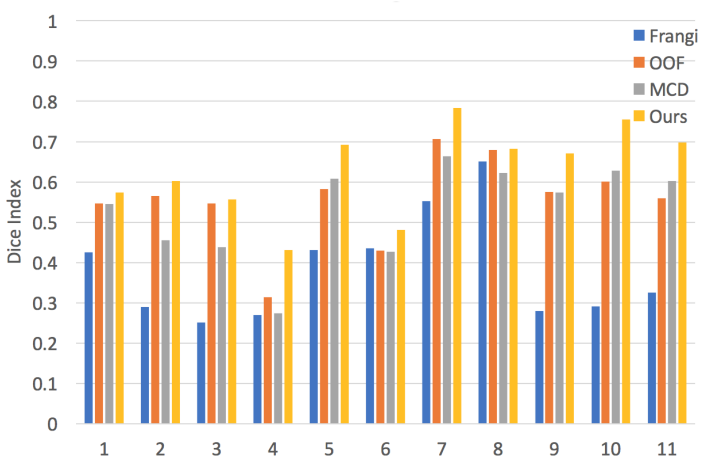


Figure 3.4. Dice index results from our neuron co-segmentation method and the competitions for 11 image stacks.

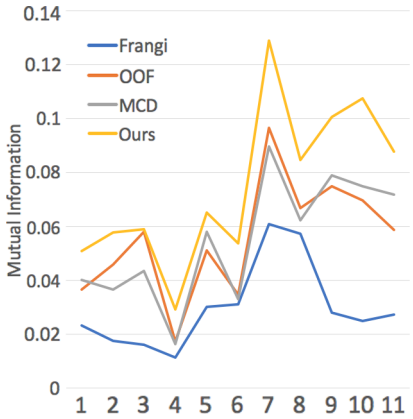


Figure 3.5. Mutual information results from our neuron co-segmentation method and the competitions for 11 image stacks.

Finally, Figure 3.8 shows the Dice index when the value of α in eq. (3.1) varies. For a larger α value, we emphasize the tree similarity across image stacks. Due to tree similarity errors during volume registration, spurious voxels may be introduced in the results. Therefore, we see a rise in recall and a drop in precision when the α value increases. When α is zero, our method degenerates into traditional MRF-based segmentation, where each stack is considered independently.

3.7 Conclusion

We presented a method for neuron co-segmentation over time-lapse image stacks. Our method was motivated by the neurite structure coherency over successive time instances during development. Our framework incorporates spatial and temporal constraints in a seamless fashion, also taking into account local structural dynamics. We evaluate our method using noisy datasets, the *Drosophila* sensory neurons with spatio-temporally varying contrast between neurite and background, and we compare our results with *vesselness*-based segmentation. In the next chapter, we will discuss the centerline-based approach, which extracts the structure by tracing instead of delineating the boundary by segmentation.

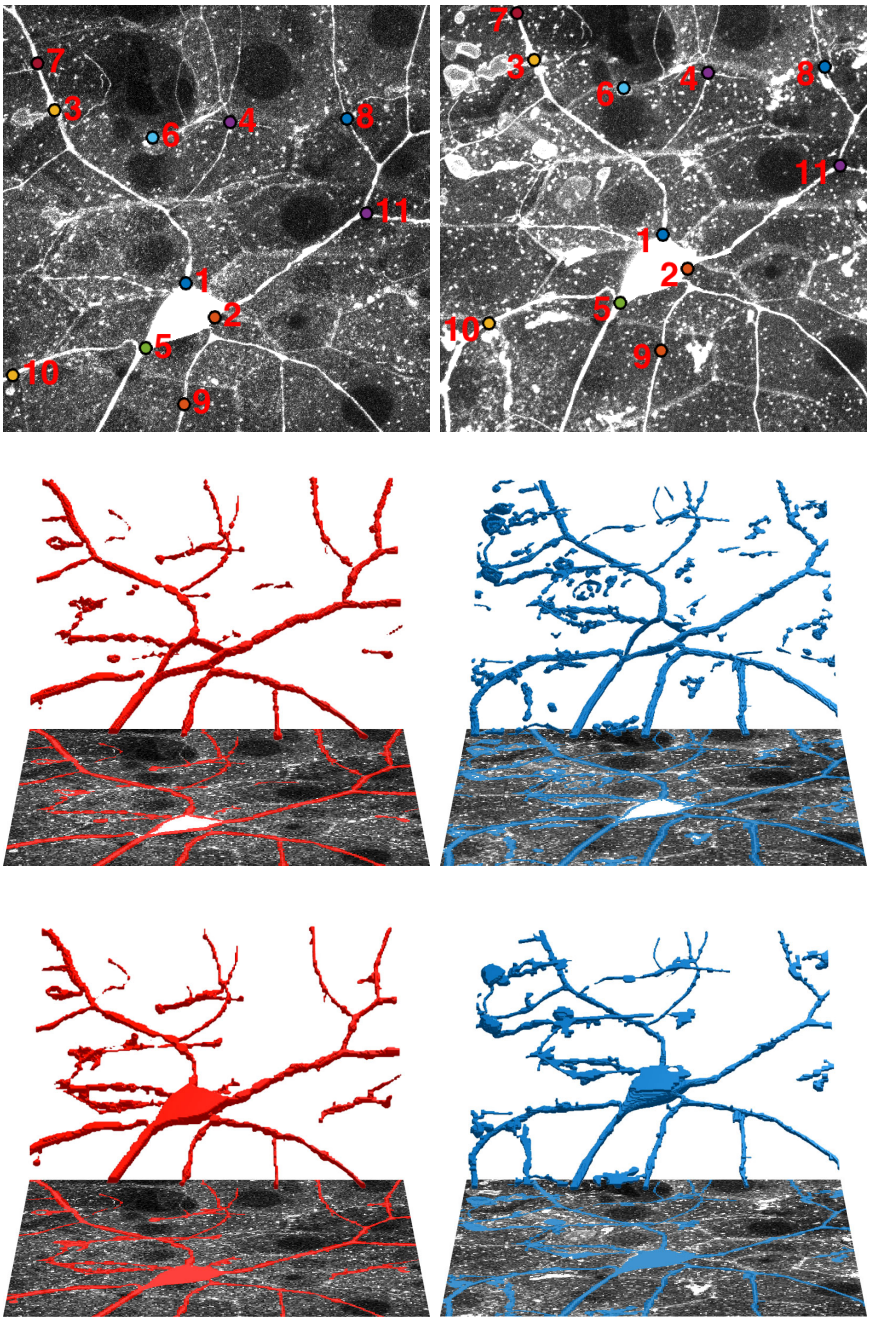


Figure 3.6. The effect of the tree similarity term. The first row shows the collapsed image stacks at two consecutive time steps. The numbered points indicate correspondences between the two tree structures. The second row shows the initial segmentation computed by thresholding vesselness produced by OOF. The third row illustrates the results from our method that removes noise/spurious structures and bridges neurite gaps.

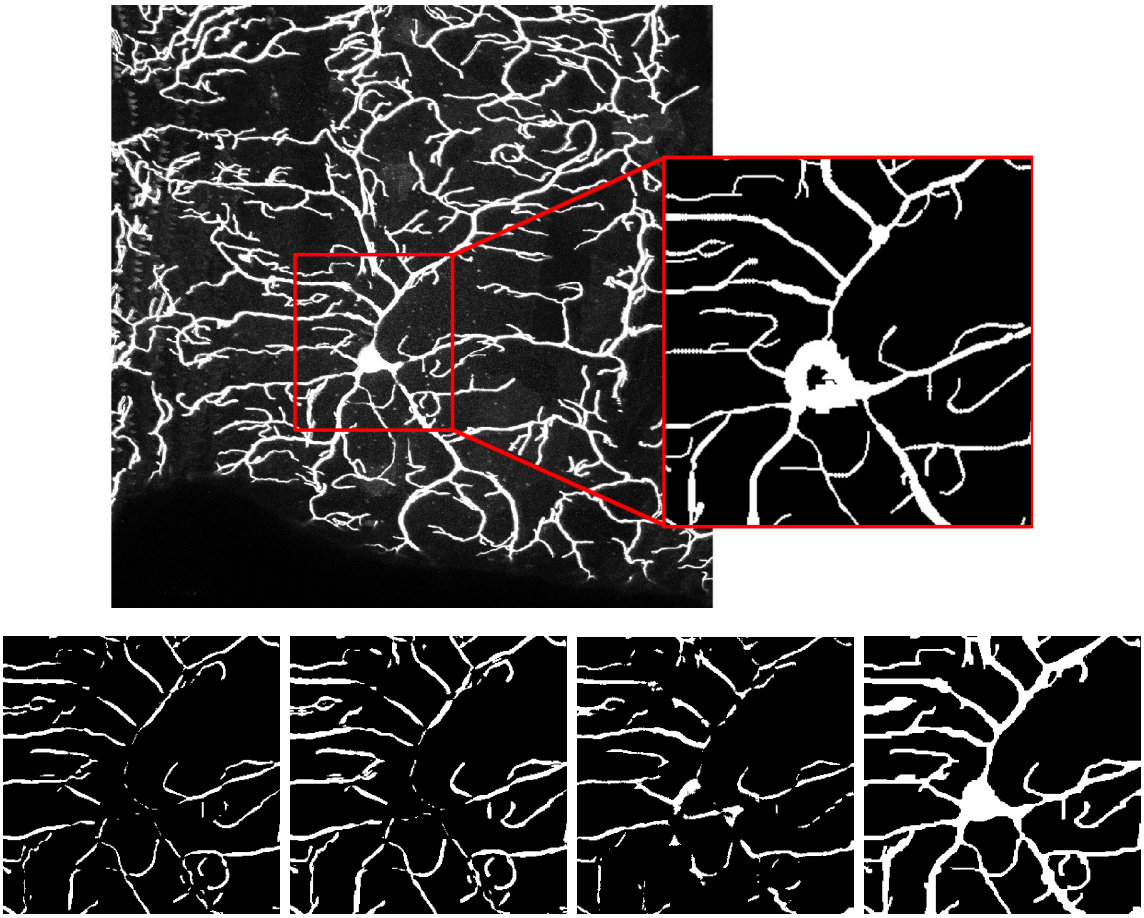


Figure 3.7. Segmenting ambiguous regions. The first row shows the original image and the ground truth (magnification). The second row shows the results of (from left to right) Frangi filter, OOF, MCD, and our method.

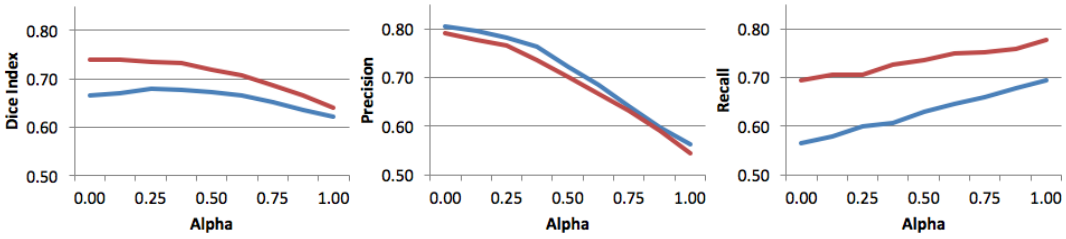


Figure 3.8. Effect of α parameter. Blue and red lines represents scores for two indicative stacks in our dataset, #9 and #10 in Table 3.1

4 NEURITE CENTERLINE DELINEATION

In this chapter, we introduce the novel tracing methods [80, 81] to compute the centerline-based neuron reconstruction. In the previous chapter, we discussed the boundary-based approach, which extracts the neuronal morphology for low-level analysis. On the contrary, the centerline-based approach extracts the topological skeleton, which is more suitable for computing high-level measurements of biological variables. A video presentation of our work with supplemental material is available at `https://youtu.be/sxnIxPcQA7Q`. Extraction of centerlines is the fundamental task critical for quantitative research on the morphological properties that directly influence the neural integration and synaptic input processing [32]. Hence, the automated 3D digital neuron reconstruction system is required to reduce the manual tracing time and storage size. Such a system must be capable of identifying neuron's parts (soma, axon, dendrites), tracing axon and dendrite branches, and storing traces usually as graphs or tree data structures [1]. Our methods based on the population of snakes are designed for data volumes with globally varying signal-to-noise ratio (SNR).

4.1 Tracing of Neuronal Morphology

We assert that the examined images do not provide sufficient contrast between neurite and background, and the SNR varies spatially. Although many tracing methods have been shown to perform sufficiently on specific data [1], one common drawback limits their generalization: they usually perform poorly, if they are examining data with insufficient contrast and low SNR. The issue with most existing active contour-based methods [23, 66] is that they evolve multiple active contour models along the neurite centerline in a sequential manner so they do not take into account the global shape geometry, i.e., smoothness at snake endpoints. We address the problem by the

simultaneous evolution, which enables the pairwise interaction between neighboring snakes to smoothly merge snakes together. Using neighbors to guide the evolution also improves the correctness of tracing, as it compensates for the unreliable intensity.

We tackle this problem using a hybrid framework (Sec. 4.2). Our methods aim to handle image stacks with insufficient contrast and spatially varying SNRs. We developed two tracing methods for a single image stack based on open-snake population with simultaneous evolution. The first method [80] in Section 4.3 formulates the problem as the joint probability between shape and appearance that optimizes both tracing and segmentation. The image stack is split into box sub-volumes to handle the varying contrast. Inside each box, a number of curves deform based on three criteria: local image statistics, local shape smoothness, and a term that enforces pairwise attraction between snakes, given their spatial proximity and shapes.

The second method [81] in Section 4.4 is built upon our first method and findings from method comparisons [80] to construct a mathematically simpler framework that retains the increased accuracy in noisy data, while significantly reducing the computational complexity. Specifically, in a similar way as in [80], this approach simultaneously evolves the population of open-curve active contours within the conditional random field (CRF) framework, which allows us to specify the relations among snakes in a more principled manner and exploit the efficient inference technique.

Section 4.5 illustrates and compares the complexity analysis of our methods. In Section 4.6, we validate our two methods against existing frameworks for neuron reconstruction on three public datasets: wild-type sensory neurons in the larval *Drosophila*, DIADEM, and FlyCircuit datasets.

4.2 Related Work

Neuron reconstruction of the single data volume problem is composed of two main steps — segmentation and tracing. However, these two subproblems are tightly related, so they are usually solved simultaneously. There is a large number of neuron

reconstruction approaches that have been extensively reviewed, e.g., in [1] and [23]. They can be categorized into global, local, and a combination of the two methods.

4.2.1 Global Tracing

Global methods extract skeletons of neurites based on the global signal distribution of the input images [58]. They can be categorized into skeletonization, minimal path, and graph-based methods.

Skeletonization is the process of extracting the binary skeleton image of the neuronal structure. Early methods usually use the thinning algorithm or the medial axis extraction [59] for the centerline detection. Authors in [20] use multi-scale Center Surround Filters (CSF) or multi-scale Laplacian of Gaussian (LoG) for enhancing tubular structure; a binary image is obtained by thresholding the filter responses then applying the voxel-coding algorithm [117]. This type of technique usually suffers from binarization errors, and it is likely to produce spurs and loops.

Minimal path-based methods trace neurites by finding the geodesic path between seed points, usually computed by the fast marching method [60]. For example, authors in [118] use Optimally Oriented Flux (OOF) [97] as steerable filter [113]. They formulate the metric function based on the filter responses and use the fast marching method for path tracing. The main issues with such methods are that they rely heavily on the seed point selection, and extracted paths may not be in the center of the neurites. To avoid these issues, some techniques [118, 119] require manual selection of seed points, whereas some others obtain sample points by either spatially distributing over the image domain [120], or selecting by a classifier [121]. Nonetheless, these methods have a potential for semi-automatic systems, but not for the complete automation [23]. So the minimal path is used for initialization or with the post-processing step to mitigate its drawbacks. The method in [122] uses minimal path to obtain the initial trace, then the active learning is applied on branching morphology to correct the neuronal topology. While All-Path-Pruning (APP and APP2)

algorithms [123, 124] create the over-reconstruction of the neuron using minimal path between the seed points and every voxel, then the initial result is pruned to obtain the topology.

Graph-based methods pose the neurite tracing problem as a graph problem, considering the inherent tree structure of neuron topology. For instance, the method in [125] detects anchor points using steerable filter responses [113] and formulates the graph using those points as nodes; thereby the trace can be obtained by finding the k-minimum spanning tree (k-MST). The performance of these methods relies on the graph formulation and the optimization technique. Most graph based methods adopt the geodesic path to formulate edges in the graph. However, this is usually not robust to spatially varying noise. To overcome the limitation of the geodesic path in intensity inhomogeneity, authors in [61, 62] convert edges into the Histogram of Gradient Deviation (HGD) descriptors, instead of deriving the feature vectors from geodesic paths. Then, they adopt the Gradient Boosted Decision Tree classifier to determine the probability of the path being neurite, and find the optimal neuronal topology using the Quadratic Mixed Integer Program (Q-MIP).

4.2.2 Local Tracing

Local methods start tracing from seed points located automatically or manually, and iteratively trace the neurite centerline using local signal features. They can be categorized into sequential tracing and deformable model-based methods.

Sequential tracing involves iterative tracing in the vicinity from the seed points until the neuronal morphology is recovered. Authors in [58] fit a cylinder model representing a neuron fiber to the seed points. Then, they advance along the direction of a neuron fiber and repeat the process until the entire neuron is traced. The method in [63] extracts the centerline using the principal curve: starting from a seed point, it iteratively computes the mean-shift to determine the new principal curve direction, and moves along that direction until the whole branch is detected. These methods

are most effective when neurites meet the modeling assumption. They usually suffer from discontinuities, i.e., gaps and holes, and their performance depends on the preprocessing step, i.e., roughly separating neurite from background.

Deformable models, which are widely used in medical image segmentation, detect the object's boundary based on the shape of the boundary curve and local image data. They can be categorized into two types based on the curve representation: parametric [64–66] and geometric [67–69] deformable models. Both categories have been used in solving neurite tracing [23, 126, 127].

Parametric deformable model or active contour models use parametric curves to represent the region of interest (ROI). It evolves towards desired features by minimizing the energy function, regulated by internal and external forces. The internal force discourages the curve from stretching and bending, while the external force moves the curve towards the desired features like edges. There is a large variety of external forces proposed by [128–130]. The most popular techniques for solving the optimization problem include variational calculus [64], dynamic programming [71], and greedy algorithms [72]. Variational calculus operates in continuous space, while dynamic programming and greedy algorithms work in discrete space over pixels/voxels. Although greedy algorithms are usually inferior to dynamic programming in terms of accuracy, they are far more efficient.

Geometric deformable models or implicit models use level set function to represent the contour of the ROI in a higher dimension [68], which allows a more direct mathematical description. The contour is evolved using the gradient descent process by solving the partial difference equation (PDE) of the level set function [69]. For example, authors in [127] evolve the curve based on the Tubularity Flow Field (TuFF), which utilizes the local tubularity of the neurites to guide the curve evolution along the centerline and towards the boundary of the tubular structures. They also introduce local attraction force to mitigate the discontinuities in neuronal topology. The advantage of geometric deformable models is that the model of the contour has the same dimensionality as the data. This property allows us to handle topology varia-

tion naturally and use a more direct mathematical description of a deformable model. However, geometric active contours usually have difficulty extracting the boundaries of objects having open or broken edges because of its flexibility in handling the topology [130].

4.3 Joint Probability between Shape and Appearance Formulation

In this approach, we deal with high intensity variation in the background by splitting the image stack into box sub-volumes, where the foreground and background

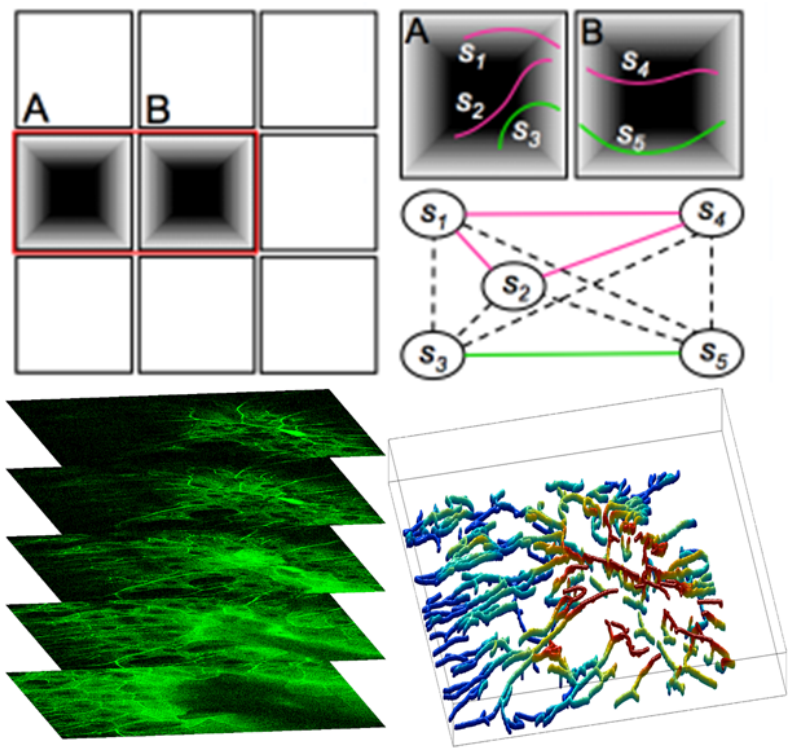


Figure 4.1. First approach. A stack is divided into box sub-volumes (top left). In each box, we evolve open-curve snakes based on local image features, shape smoothness, and snake interactions, regulated by proximity and shapes (top right): (s_1, s_2, s_4) are close to each other and have *compatible* shapes; (s_3, s_5) do not appear to belong to the same branch, and therefore do not interact with (s_1, s_2, s_4). Under noise, we can successfully reconstruct the neuron in 3D (bottom).

intensities can be sufficiently classified. Within each sub-volume, we started from initial seeds, detected by classification, and we evolved different open-curve snakes based on three criteria: (i) local intensity, (ii) individual curve shape smoothness, and (iii) interactions between neighboring snakes (Sec. 4.3.1), based on their proximity and local shapes, including snakes in neighboring sub-volumes (Fig. 4.1). We formulated the solution as a maximum a posteriori probability estimation between shape and appearance, where the objective is a joint probability of the snake population configuration (positions and shapes) and the labels of the voxels (neurite against background). The optimization of the joint probability is explained in Section 4.3.2.

Let the data volume be $\mathbf{V} \in \mathbb{R}^3$, partitioned into non-overlapping box sub-volumes. Also let V^i and V^j, $i \neq j$, denote two adjacent boxes, considering locally 27-box neighborhoods. Within each box sub-volume V^i, we detected seed points probabilistically, as we describe below; these seeds are the initialization for a set of snakes that we evolve locally.

Let S^i be the set of snakes in sub-volume V^i, $S^i = \{s_n\}_{n=1}^{N}$, and $\mathbf{S} = \bigcup_{k=1}^{K} s_k$ denote the set of all snakes in the 27-box neighborhood ($K \geq N$). There can be attraction between any two curves in $S^i \bigcup S^j$, $i \neq j$, given (a) that V^i and V^j are adjacent sub-volumes, (b) their close spatial proximity and their *compatible* shape configurations (Fig. 4.1).

On the other hand, every voxel $v \in \mathbf{V}$ can be classified as neurite or background, i.e., $y(v) = \{+1, -1\}$, where y is the binary variable of the voxel label. Let the set of voxel labels in sub-volume V^i be denoted by $Y^i = \{y(v)\}_{v \in V^i}$, and $\mathbf{Y} = \bigcup_{i=1}^{27} Y^i$ be all the labels in the box-neighborhood. Each Y^i is calculated independently from the neighboring sub-volumes, to account for local feature variations.

Given the above notations, we formulate our objective locally as a joint optimization of the snake configuration and voxel labels,

$$\langle \widehat{\mathbf{S}}, \widehat{\mathbf{Y}} \rangle = \arg\max_{\langle \mathbf{S}, \mathbf{Y} \rangle} P(\mathbf{S}, \mathbf{Y} | \mathbf{X}) \tag{4.1}$$

where $\mathbf{X} = \bigcup_{i=1}^{27} X^i$ denotes the features calculated from the data, and the objective joint conditional probability is,

$$\begin{aligned} P(\mathbf{S}, \mathbf{Y}|\mathbf{X}) &\propto P(\mathbf{X}|\mathbf{Y}) \cdot P(\mathbf{Y}|\mathbf{S}) \cdot P(\mathbf{S}) \\ &\propto P(\mathbf{X}) \cdot P(\mathbf{Y}|\mathbf{X}) \cdot P(\mathbf{Y}|\mathbf{S}) \cdot P(\mathbf{S}) \end{aligned} \tag{4.2}$$

The data prior $P(\mathbf{X})$ is expressed locally as a Gaussian distribution of tubularity. Here, as data features, we use indicatively the Frangi filter responses [96]; however, one may use other *vesselness*/tubularity features. Thus, we define,

$$P(x^i(v)) = \frac{1}{\sqrt{2\pi\sigma_0^2}} \exp\left\{-\frac{[x^i(v)]^2}{\sigma_0^2}\right\} \tag{4.3}$$

where $x^i(v)$ is the Frangi filter response at the location $v \in V^i$. Again, to calculate such features, we consider each sub-volume independently, as separate data.

The factor $P(\mathbf{Y}|\mathbf{X})$ is essentially the data term that drives the snake evolution in the vicinity. It is calculated independently within each box sub-volume, as $P(Y^i|X^i)$; therefore, there is no probability smoothness imposed between neighboring boxes, with respect to spatial distribution. The reason for such choice is to consider each box as separate data (stack), and calculate probabilities based only on local intensity (Frangi response) statistics. To calculate such probability fields, we used a Discriminative Random Field (DRF) [50], where we solved inference with the graph cuts algorithm [50], due to its accuracy and efficiency. In our implementation, we used the positively classified voxels with high tubularity as initial seeds that determine the snake population.

The term $P(\mathbf{Y}|\mathbf{S})$ encodes the likelihood of voxels being part of the neurite, given the configuration of the snakes. If d_k is the (unsigned) distance function for each snake $s_k \in \mathbf{S}$,

$$\begin{aligned} P\big(y(v)|s_k\big) &= \exp\{-d_k(v)\} \\ P\big(y(v)|\mathbf{S}\big) &= \max\left\{P\big(y(v)|s_1\big), ..., P\big(y(v)|s_K\big)\right\} \end{aligned} \tag{4.4}$$

Finally, $P(\mathbf{S})$ corresponds to the local smoothness of the models and their pairwise attractions,

$$P(S^i) = \exp\left\{-E_s(S^i) - E_a(S^i)\right. \quad , \tag{4.5}$$

The smoothness energy for the population S^i in V^i is,

$$E_s(S^i) = \sum_{n=1}^{N} \left\{ \int_0^1 \left(\left[\alpha(t) \ \nabla s_n \ ^2 + \beta(t) \ \nabla^2 s_n \ ^2 \right] dt \right) \right\}, \tag{4.6}$$

with t encoding the normalized arc-length of each snake, and (α, β) being the elasticity and rigidity weights; we consider $\alpha = \beta = 1$ for all snakes in the entire volume.

4.3.1 Attraction Model

The effect of attraction energy E_a in eq. (4.5) is shown in Fig. 4.1. Each point at a (normalized) location t along a snake, contributes to an attraction field, i.e., an energy that drives the extension of all snakes towards certain directions.

Let us consider a single model $s_n \in S^i$ and how it affects its surroundings. Using its distance function d_n, we define,

$$D_n(v) = \exp\left\{-\gamma d_n(v)\right\}, \quad 0 \leq \gamma \leq 1, \tag{4.7}$$

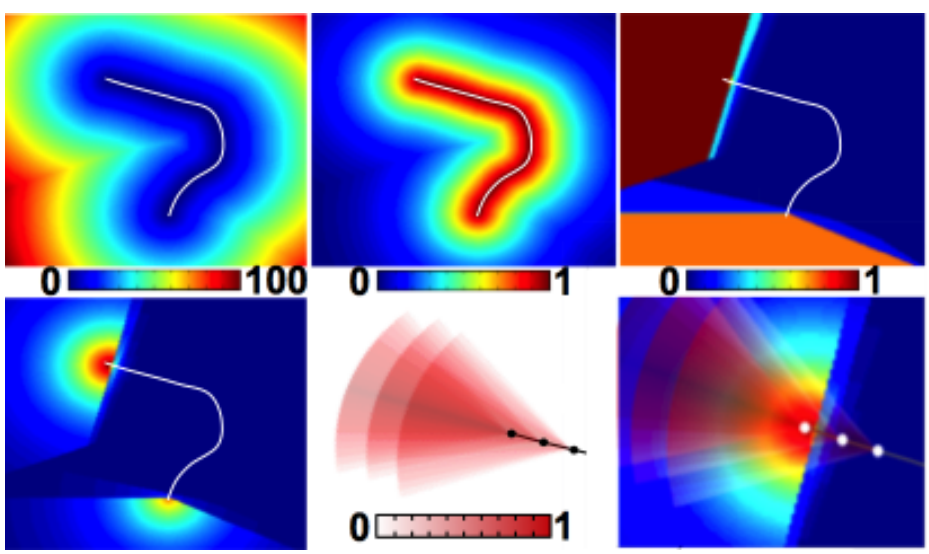


Figure 4.2. Illustration of how a single snake participates in the attraction energy. Top (from left to right): distance map d_n, D_n in eq. (4.7), and F_n in eq. (4.9). Bottom (from left to right): D_nF_n, G_n in eq. (4.11), and $D_nF_nG_n$. In eq. (4.12) we use $(1 - D_n)(1 - F_n)(1 - G_n)$ for energy minimization.

which implies that the further a location v from s_n, the lower the degree of influence at v from that model. We used γ to control the slope of the exponential: higher γ values produce narrower zones of influence around the curve (Fig. 4.2).

We also consider that only the end-parts of the snake can create attraction; therefore, we define the functional that assigns weights to curve points as,

$$f(t) = e^{-\delta t} + e^{-\delta(1-t)}, \quad t \in [0,1], \tag{4.8}$$

which imposes higher attraction by points at the two curve ends, and practically no influence by intermediate points. We calculate the map (Fig. 4.2),

$$F_n(v) = \left\{ f(t_0) \quad |v(t_0) - v| = d_n(v) \right\}, \tag{4.9}$$

where t_0 is the location along the normalized arc length of the curve, with cartesian coordinates $v(t_0)$, closest to the voxel v.

We also determine an angle of influence θ, with respect to the direction of the curve's first derivative. Every location v can be affected by any point t on the snake based on,

$$g_n(v,t) = \begin{cases} \cos\left(\angle \nabla s_n|_t\right) - \cos\left(\angle \nabla s_n|_t + \theta\right), & \theta \in [-\vartheta, +\vartheta] \\ 0, & \text{otherwise} \end{cases}, \tag{4.10}$$

where $\angle \nabla s_n|_t$ is the tangential orientation at point t on the curve, and θ is the angle between $\overline{v(t)v}$ and $\overline{\nabla s_n|_t}$. In all our experiments, we considered $\vartheta = \pi/3$. We define (Fig. 4.2),

$$G_n(v) = \prod_{t \in [0,1]} g_n(v,t), \tag{4.11}$$

Using the definitions of eqs. (4.7), (4.9), and (4.11), since $0 \leq D_n(v)$, $F_n(v)$, $G_n(v) \leq 1$, we formulate the attraction map of snake s_n,

$$A_n(v) = (1 - D_n(v))(1 - F_n(v))(1 - G_n(v)) \tag{4.12}$$

where low values indicate high influence. Therefore, for any $s_n \in S^i$ and $s_k \in \mathbf{S}$, $s_n \neq s_k$,

$$\mathcal{A}_{nk}(v) = \max\left\{ A_n(v), A_k(v) \right. \quad , \quad \forall v \in \bigcup_{i=1}^{27} V^i \tag{4.13}$$

which essentially encodes that two models can be attracted to each other only when the highest value among the two attraction maps at location v is low enough. The attraction energy of every curve s_n in V^i sources from all snakes in the box-neighborhood as,

$$e_a(s_n) = \int_0^1 \left[\sum_{s_k \in \mathbf{S}} \mathcal{A}_{nk}(v(t)) \right] dt, \quad s_k \neq s_n \tag{4.14}$$

and for the population $S^i = \{s_n\}_{n=1}^N$,

$$E_a(S^i) = \sum_{n=1}^{N} e_a(s_n) \tag{4.15}$$

Notice that all snakes in the box sub-volume neighborhood participate in this energy of the set S^i, given their pairwise proximity imposed by the terms D (eq. (4.7)) and F (eq. (4.9)).

4.3.2 Optimization

Using the definitions in eqs. (4.6) and (4.15), we calculate the prior $P(S^i)$ in eq. (4.5). Then, we solved the objective in eqs. (4.1) and (4.2) iteratively in a narrow band around each open curve. In eq. (4.2), the terms $P(\mathbf{Y}|\mathbf{X})$ and $P(\mathbf{Y}|\mathbf{S})$ are calculated within each box sub-volume separately, while $P(\mathbf{S})$ involves shape and location information from snakes across all neighboring sub-volumes. Figure 4.3 illustrates in

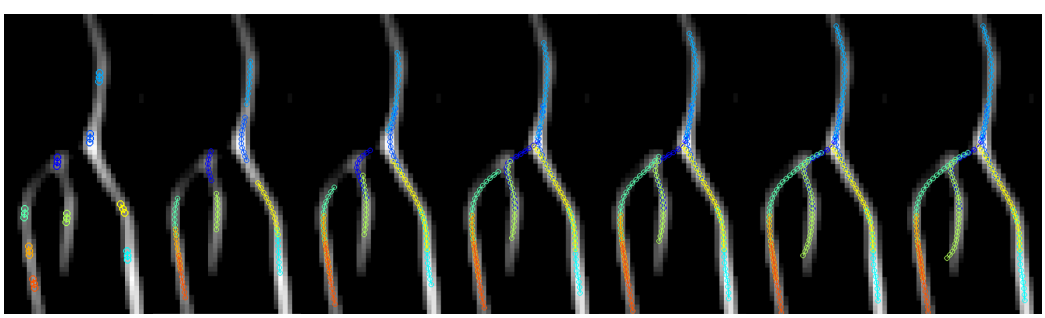


Figure 4.3. Evolution instances of different snakes (in colors). From left to right: iteration ♯5, 10, 15, 20, 25, 30 and 40 (termination). The images have been enhanced for illustration.

2D the evolution of nine snakes, corresponding to the initial seeds detected by the DRF-based classification. For simplicity, we considered two adjacent box-volumes with virtually no background noise.

4.4 CRF Formulation

The second approach was built upon the first method in Section 4.3. We increased robustness under spatially varying contrast, and at the same time reduced the computational complexity by a simpler mathematical framework, compared to the state-of-the-art method in [80]. Here we describe how we model the snake population within the CRF framework and how we solve inference in a more efficient way (Sec. 4.4.3). The implementation details is also given in Section 4.4.4.

Let $\mathbf{X} \in \mathbb{R}^3$ be the data volume, and $\mathbf{S} = \{s_i\}_{i=1}^N$ be the configurations (shapes) of the population of N open curve snakes. Our objective function is formulated as the conditional distribution over the set $\mathbf{S}$, given the data volume $\mathbf{X}$ (Fig. 4.4),

$$\begin{aligned} P(\mathbf{S}|\mathbf{X}) &= \frac{1}{Z} \exp\left\{ -\left(\sum_{s_i \in \mathbf{S}} E(s_i) \right) \right\} \\ &= \frac{1}{Z} \exp\left\{ -\left(\sum_{s_i \in \mathbf{S}} \left(\mathbf{A}(s_i, \mathbf{X}) - \sum_{s_i \in \mathbf{S}} \sum_{s_j \in N^i} \mathbf{I}(s_i, s_j, \mathbf{X}) \right) \right\} \right(\\ &\quad \left(\right. \end{aligned} \tag{4.16}$$

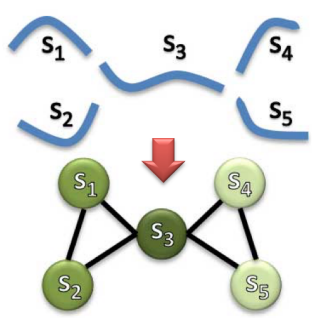


Figure 4.4. Second approach. The snake population $(s_1, s_2, s_3, s_4, s_5)$ is formulated using the CRF framework, where snakes are nodes and snake configurations are states. Edges exist between neighboring snakes.

where Z is the partition function, $E(s_i)$ is the individual snake energy, and N^i is the set of neighboring snakes of s_i. $\mathbf{A}$ and $\mathbf{I}$ are the association (Sec. 4.4.1) and interaction (Sec. 4.4.2) potentials respectively [50].

4.4.1 Association Potential

Association potential $\mathbf{A}(s_i, \mathbf{X})$ measures the likelihood of the snake s_i fitting the neurite centerline so it is basically the energy function of active contour model [64]. It is given in terms of the snake shape and appearance energies,

$$\mathbf{A}(s_i, \mathbf{X}) = E_{shape}(s_i) + E_{app}(s_i), \tag{4.17}$$

The shape term E_{shape} models the local shape smoothness through the first and second order derivatives of snake curve [80], similar to eq. (4.6),

$$E_{shape}(s_i) = \int_0^1 \left(\left[\alpha(t) \ \nabla s_i(t) \ ^2 + \beta(t) \ \nabla^2 s_i(t) \ ^2 \right] dt, \right. \tag{4.18}$$

where t encodes the normalized arc-length of the snake curve, and (α, β) are the elasticity and rigidity weights. In this work, discrete snakes were adopted, specifically, the snake curve $s_i(t)$ is represented by a sequence of points, where t becomes the index of snake point. ∇s_i is approximated by the deviation of the distance between two adjacent points from the average to encourage even spacing between snake points [72],

$$\nabla s_i(t) \ = \ |\vec{u}_t| - \langle |\vec{u}_t| \rangle \tag{4.19}$$

where $\langle \cdot \rangle$ is the expected value and $\vec{u}_t = s_i(t) - s_i(t-1)$ is the vector between adjacent snakes points at position t of snake s_i [72]. $\alpha(t) = 0$ at endpoints to allow snakes to stretch along the centerline [23]. The curvature is approximated by,

$$\nabla^2 s_i(t) \ = \ \frac{\vec{u}_t}{|\vec{u}_t|} - \frac{\vec{u}_{t+1}}{|\vec{u}_{t+1}|} \ ^2 \tag{4.20}$$

Other approximations of curvature can be used and they are reviewed in [72].

The appearance term E_{app} deforms snakes towards the centerline based on local image features. It is a function of tubularity that we compute here using the OOF [97]; however, other tubularity measures can also be used. Regions with low tubularity response, or already occupied by other snakes, penalize snake energy; this prevents snakes from evolving in the background or overlapping with neighboring snakes,

$$\begin{aligned} E_{app}(s_i) &= \prod_t \left(\frac{1}{1 + f\big(s_i(t), \mathbf{X}\big)} \right) \\ f(s_i(t), \mathbf{X}) &= \begin{cases} 0 & \text{, if } s_i(t) \text{ is occupied} \\ f_{vessel}\Big(\mathbf{X}\big(s_i(t)\big)\Big) & \text{, otherwise} \end{cases} \end{aligned} \tag{4.21}$$

where f_{vessel} is the *vesselness*/tubularity response.

4.4.2 Interaction Potential

$\mathbf{I}(s_i, s_j, \mathbf{X})$ measures how snakes s_i and s_j interact given the data volume $\mathbf{X}$ based on the compatibility between snakes. The objective is to merge neighboring snakes that (i) 'align' towards each other, (ii) are close to each other, with distance regulating their interaction, and (iii) have sufficient length, considering that short curves can be produced by spurious background features. The compatibility is influenced by 4 factors: orientation, endpoint, distance, and curve length. It is similar to the attraction model in Section 4.3.1 (Fig. 4.2).

If $\theta(t)$ encodes the angles between the evolving directions of s_i along t, $\nabla s_i(t)$, and the direction towards another snakes $s_j \in N^i$ in the vicinity, $\overline{s_i(t)s_j}$, we define,

$$G(t) = \max(\cos\theta(t), \cos\vartheta) \tag{4.22}$$

where ϑ is the maximum angle that s_i is allowed to bend towards s_j; in all our experiments, we considered $\vartheta = \pi/3$. Similarly to our work in [80], attraction between snakes is created only at curve endpoints; therefore, we assign weights along each curve like in eq. (4.8), where δ is a positive constant,

$$F(t) = e^{-\delta t} + e^{-\delta(1-t)}, t \in [0, 1] \tag{4.23}$$

The interaction attenuates as the distance increases; therefore, we define,

$$D(t) = \overline{s_i(t)s_j}^{\,2} d(t) \tag{4.24}$$

where $d(t)$ accounts for the angle $\theta(t)$ described above in eq. (4.22),

$$d(t) = \frac{1}{1 + e^{c(\cos\theta(t) - \cos\vartheta)}} \tag{4.25}$$

where the constant c regulates the exponentially decreasing slope as $\theta(t)$ becomes smaller than ϑ (we assumed angles in the range $[0,\pi/2]$ from $\nabla s_i(t)$ and here we fix $c = 10$).

From these definitions, the interaction potential in eq. (4.16) is,

$$\mathbf{I}(s_i, s_j, \mathbf{X}) = (|s_i||s_j|)^{-\max_t\left\{\frac{G(t)F(t)}{D(t)}\right\}} \tag{4.26}$$

where $|s_i|$ and $|s_j|$ are snake curve's geodesic lengths of s_i and s_j respectively, used to ignore short curves corresponding to spurious high tubularity regions in the background.

4.4.3 Inference

Using the above formulation to evolve the snake population and capture the neurite centerline, we pose the problem as a multiclass CRF model, where snakes are the sites. The main challenge is that the evolving curves yield a practically infinite number of CRF configurations or site states, owing to the infinite number of possible snake curve configurations. Therefore, inference methods like the loopy belief propagation and graph cuts algorithms whose complexity depends on the number of classes are infeasible. On the other hand, the Highest Confidence First (HCF) framework [52, 131] allows us to optimize multiple sites in parallel without the dependency on the number of states.

Highest Confidence First for Snake Population

According to HCF, site *stability* [52] determines the order in which sites are optimized. Iteratively, HCF chooses to optimize the sites of the CRF whose stabilities are local minima in their neighborhood; these sites are then fixed and their neighbors can be influenced by them (i.e., change their stability) according to the CRF formulation in eq. (4.16). Note that snakes are considered neighbors if there are within a fixed number of voxels from each other (here it is 20 voxels), as in [58].

While traversing the sites, priority is given to snakes whose curves are short and have the least amount of neighbors in order to avoid evolving snakes in sequence (Fig. 4.5). A low number of neighbors means that these snakes affect and are affected by a lower number of snakes, and therefore the computation of the interaction term is faster. On the other hand, when a site/snake becomes stable, it means that its energy is locally minimized, its curve is sufficiently long, and it has a higher number of neighbors that it can affect. Low energy can be seen as certainty of being on the neurite centerline. Thus, we defined *stability* as,

$$stability(s_i) = \frac{|s_i| \cdot |N^i|}{E(s_i)} \tag{4.27}$$

where $|s_i|$ is the length of snake s_i and $E(s_i)$ is from eq. (4.16),

$$E(s_i) = \mathbf{A}(s_i, \mathbf{X}) - \sum_{s_j \in N^i} \left(\mathbf{I}(s_i, s_j, \mathbf{X}) \right. \tag{4.28}$$

where N^i and $|N^i|$ denote the set of neighbors of s_i and its size respectively.

Greedy Algorithm for Individual Snake

After HCF selects sites or snakes with locally minimal stability, each selected snake can be optimized independently when its neighbors are fixed. There are multiple methods to find the optimal configuration of the individual snake like dynamic programming [71] and variational calculus [64], but greedy algorithm is preferred because it gives comparable results and it is more efficient [72]. It works as follows:

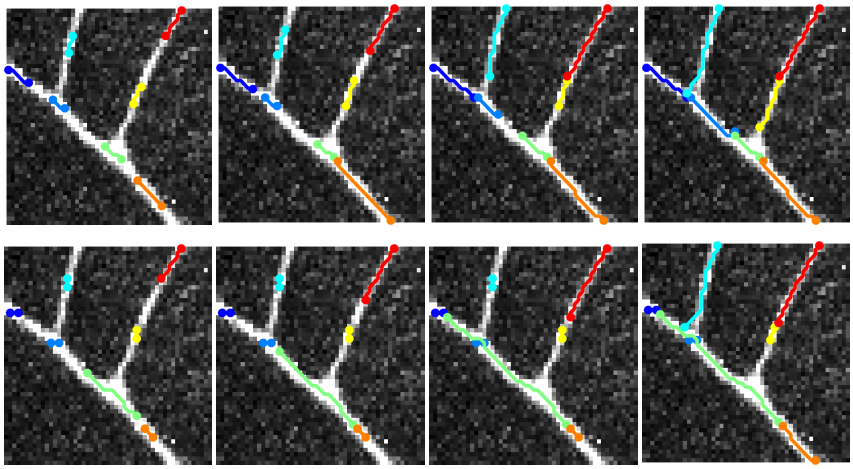


Figure 4.5. The population of seven snakes (in different colors) evolves based on the stability function given in eq. (4.27) (top) compared to $stability(s_i) = E(s_i)$ (bottom). From left to right, the evolution of seven snakes at iterations #5, #10, #25, #40 respectively. Our stability function evolves all snakes simultaneously, while the latter evolves snakes in sequence.

first, it computes the snake energy when the snake point is at the original location and adjacent voxels using eqs. (4.17) and (4.26). Then, the snake point moves to the adjacent voxel with the optimal energy, or remains at the same voxel if it is the optimum. These steps are repeated for every snake point from one endpoint to the other end.

4.4.4 Implementation

The overall algorithm for optimizing our model in eq. (4.16) consists of the following steps.

(i) Image preprocessing is required to deal with blurred boundaries and intensity inhomogeneity of the neurite. First, Curvelet transformation [132] and LoG [20] are applied to enhance the linear structure. Then, the OOF [97] is applied to calculate the *vesselness*/tubularity.

(ii) Initial seed points are automatically selected according to three criteria: (a) they should fall inside the neurite volume, (b) they should not overlap each other, and (c) there exists at least one seed point on every branch of the neurite (between intersection/branching points). In our methods, ridge points [66], or local maxima of tubularity along any axis that have OOF responses higher than a user-defined threshold, are selected as candidate seeds. Although these points are usually inside the neurite volume, they may be spatially adjacent or too close to each other; therefore, the local maxima among the candidate seeds are selected as the initialization of the snake population. Finally, each snake is initially expanded along the eigenvector with the smallest eigenvalue, as computed by eigen-decomposition of the tubularity Hessian matrix [96].

(iii) Inference of the model in eq. (4.16) is calculated by finding sites that satisfy criteria described in Section 4.4.3. Then greedy snake optimization is applied to these sites to update curve configurations.

(iv) The stability of sites is updated using eqs. (4.27) and (4.28). This method usually gets trapped in local minima if we only consider the lowest energy configuration of each snake/site in each iteration, due to local image inhomogeneities. Thus, sites converge when their stability does not increase for a few consecutive iterations, to avoid premature convergence.

(v) Finally, after the HCF convergences, we obtain a set of snakes that capture individual parts of the neurite. To merge all snakes into a single trace, we apply the minimum spanning tree (MST) technique.

4.5 Complexity

Our Joint Probability (JP) method [80] in Section 4.3 considers the probabilistic model, where for a population of snakes $\mathbf{S}$, a prior $P(\mathbf{S})$ requires the most computational cost. The prior probability incorporates shape and appearance terms of

each snake along with pairwise interactions in the set $\mathbf{S}$. Shape and appearance are jointly optimized using the variational calculus, with complexity $\mathcal{O}(n)$ [64], given n points along each snake. For N snakes in $\mathbf{S}$, this yields $\mathcal{O}(Nn^2)$. Moreover, for each snake, the fast marching method [60] is applied to find the shortest geodesic distance to neighboring snakes in order to compute pairwise interactions. The complexity of the fast marching method is $\mathcal{O}(A)$ [133, 134], where A is the number of grid points. If H is the average number of neighboring snakes, the computation of the interaction term takes $\mathcal{O}(NnHA)$ overall, since we increase the length of N snakes that costs $\mathcal{O}(n)$. Therefore, the total complexity of our first method is $\mathcal{O}(Nn^2 + NnHA) = \mathcal{O}(Nn(n + HA))$.

In our Conditional Random Field (CRF) method in Section 4.4, deforming a snake using the greedy algorithm takes $\mathcal{O}(nm)$ per site (snake), where m is the number of possible moves of each point [72]. The HCF algorithm visits sites in $\mathcal{O}(V)$, where V is the number of visits. For every visit, we need to go through all (H, on average) neighbors to update stability and check for collision. Therefore, updating the stability of HCF takes $\mathcal{O}(VH)$. Moreover, maintaining stability in sorted order takes $\mathcal{O}(\log N)$ per visit, which can be ignored as insignificant. Thus, the total complexity is $\mathcal{O}(Vnm + VH)$; however, the number of visits V is bounded by Nn ($V = \mathcal{O}(Nn)$) because our method visits every site, and each snake takes n iterations before it converges. Therefore, the total complexity of the second method becomes $\mathcal{O}(Vnm + VH) = \mathcal{O}(Nn^2m + NnH) = \mathcal{O}(Nn(n + H))$, since m is considered a constant ($m = 27$ for 3D data) [72], which yields significantly less computation compared to our JP method in [80].

4.6 Results

In this section, we compared the results of our two methods: JP (Sec. 4.3) [80] and CRF (Sec. 4.4) [81] methods, with the APP2 algorithm [124] and all five finalist algorithms from the DIADEM challenge [28], namely the k-MST based method [125],

the Farsight toolbox [23], the Neuron Circuit Tracer (NCT) [20], the Principal Curve Tracing (PCT) [63], and the neuTube [58, 135]. All algorithms were tested on three different sources: (a) 10 stacks from two datasets of the DIADEM challenge, which are available online at the challenge's website [28]; (b) 10 image stacks from FlyCircuit [136], which is the public database of neurons in the *Drosophila* brain; (c) 11 image stacks of sensory neurons in the wild-type larval *Drosophila* for the study of dendritic arborization patterns over the four instars of development.

The accuracy of our approach and the competition was quantified by five measurements, the DIADEM metric [137], precision, recall, F1 score, and mean absolute error (MAE). DIADEM metric was selected because it measures the correctness of the position of branch points and leaf nodes, as well as the trace topology by comparing the path length of branches [137]. The metric is designed to substitute the expert qualitative measurement and measure the editing time, which are the two most important quantities for evaluating automatic neuron reconstruction. The advantage of DIADEM metric is that it produces the similarity value ranged from 0 to 1, rather than a distance value produced by other metrics like the tree edit distance (TED) [138]. Another issue with the TED is that it ignores node locations. The mistake arises when there is a growth in one branch while another branch shrinks. Then, the TED would not detect the total length difference. To make matters worse, if these two branches originate from the same bifurcation, then the TED would consider them correct.

Because DIADEM metric considers both location and topology, the poor quality in one of the aspects can heavily reduce the score. Therefore, we also validate our work with the F1 score, which emphasizes only on the correctness of the neurite location. Precision and recall are computed by measuring the length of incorrect traces as false positives (FP), and missing traces as false negatives (FN). The length of correct traces is counted as true positives (TP): a trace is considered correct if it is within a small distance (here it is 4 voxels) from the closest point of ground truth. From these definitions, precision (P) $= \frac{TP}{TP+FP}$, recall (R) $= \frac{TP}{TP+FN}$, and F1 $= \frac{2PR}{P+R}$.

While the F1 score measures the correctness of the centerline position, the MAE measures its quality based on the distance by which the predicted trace deviates from the ground truth. It is defined by the mean between (a) the average distance of predicted nodes to the closest ground truth nodes and (b) the average distance of ground truth nodes to the closest predicted nodes. The MAE in voxels is given by,

$$MAE(\mathbf{P}, \mathbf{Q}) = \frac{1}{n}\sum_{i=1}^{n} \min_{j} |p_i - q_j| + \frac{1}{m}\sum_{i=1}^{m} \min_{k} |q_i - p_k| \tag{4.29}$$

where $\mathbf{P} = \{p_1, ..., p_n\}$ are points of the ground truth and $\mathbf{Q} = \{q_1, ..., q_m\}$ are points of the predicted trace [127].

4.6.1 DIADEM Challenge

All algorithms are validated on the *Olfactory Projection Fibers* (OP) and *Cerebellar Climbing Fibers* (CF) datasets. In these datasets, it is not always obvious how much better our approach performs compared to the other six examined methods. In Figure 4.6, we illustrate three cases from the OP dataset where our CRF method and the competitions perform overall equally well. This is mainly due to the fact that in such data, there is very little background variation, and we do not fully exploit our method's focus on local ambiguities. Our observation is reinforced by the quantitative results in Table 4.1 as well as the MAE score in Figure 4.7, which indicates that trace results produced by our methods and other frameworks have the comparable quality.

Some methods like NCT and Farsight have difficulty detecting neurites from parts of images with poor SNR, resulting in fragmented traces. Although NCT can recover most of the traces, they are in fragments so it produced a low DIADEM score because the metric considers only one tree structure of the trace. Our CRF method addressed this problem using the interaction potential (Sec. 4.4.2) to attract snakes together, which in turn helps remove gaps and improve neurite detection in inhomogeneous regions. As a result, our CRF method produced higher F1 score (Table 4.2).

Another issue is the images with dense dendrite branches like OP9 in Figure 4.6. Our approach has the limitation in recovering the topology from cluttered branches,

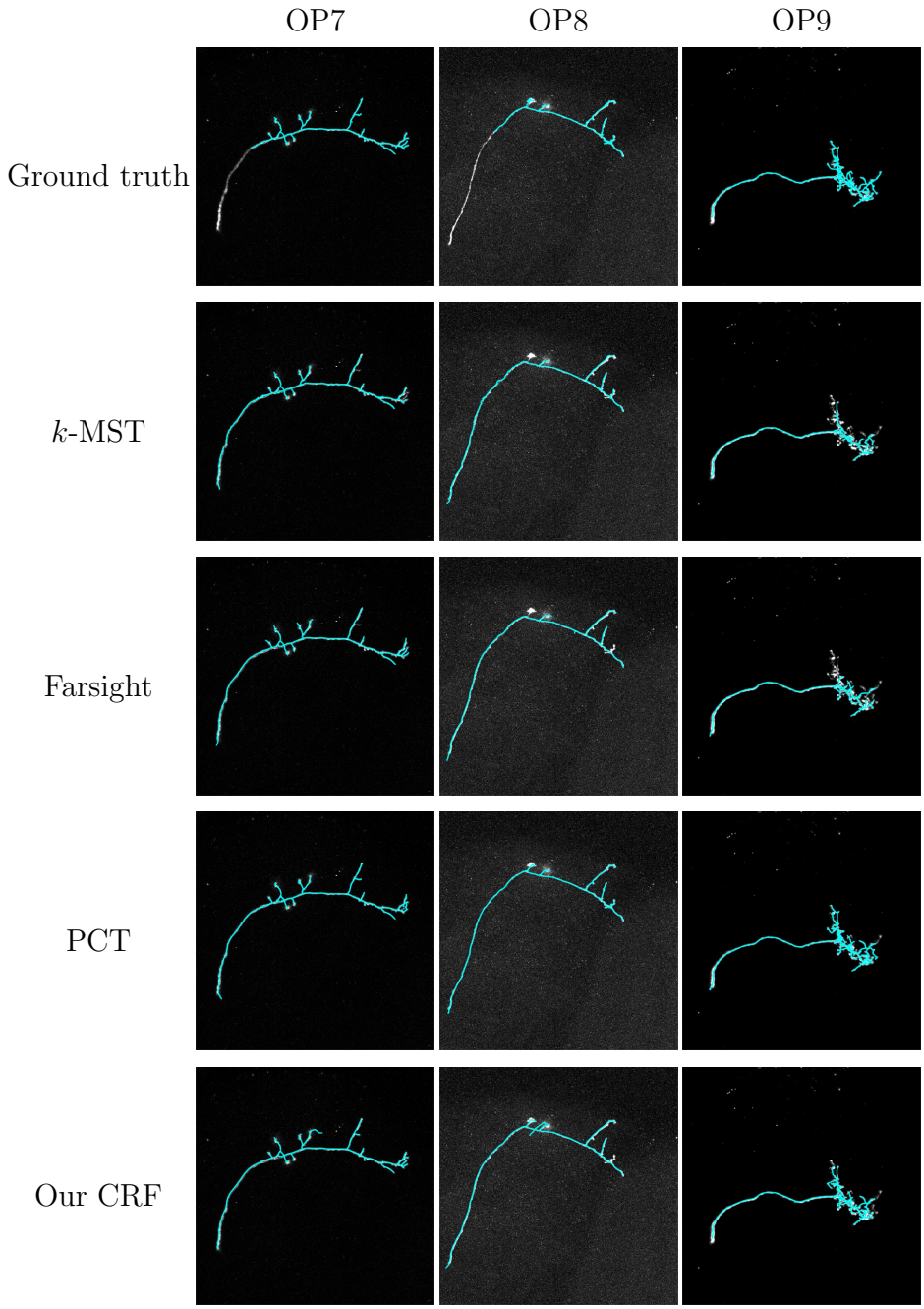


Figure 4.6. Tracing results (in cyan color) on the DIADEM *Olfactory Projection Fibers* dataset. From top to bottom: the ground truth, the results of k-MST based method [125], Farsight [23], PCT [63], and our CRF method [81] respectively. Columns correspond to different image stacks (OP7, OP8, and OP9).

Table 4.1.
DIADEM score on *Olfactory Projection Fibers* dataset of our methods and existing frameworks per sample stack; the best scores are shown in bold.

	Size (voxel)	Our CRF	Our JP	k-MST	Farsight	NCT	PCT	neuTube	APP2
OP1	$512 \times 512 \times 60$	0.852	0.815	0.841	0.878	0.127	0.814	0.224	**0.900**
OP3	$512 \times 512 \times 62$	0.131	0.278	**0.674**	0.009	0.067	0.342	0.439	0.000
OP4	$512 \times 512 \times 67$	**0.774**	0.686	0.641	0.295	0.005	0.679	0.319	0.761
OP5	$512 \times 512 \times 76$	0.533	0.313	0.000	0.016	0.000	**0.714**	0.481	0.412
OP6	$512 \times 512 \times 101$	0.876	0.735	0.464	0.669	0.296	0.858	0.121	**0.887**
OP7	$512 \times 512 \times 71$	**0.924**	0.884	0.703	0.575	0.195	0.640	0.714	0.773
OP8	$512 \times 512 \times 85$	**0.824**	0.742	0.518	0.321	0.321	0.719	0.208	0.373
OP9	$512 \times 512 \times 92$	0.768	0.613	0.782	0.289	0.267	0.617	0.340	**0.786**

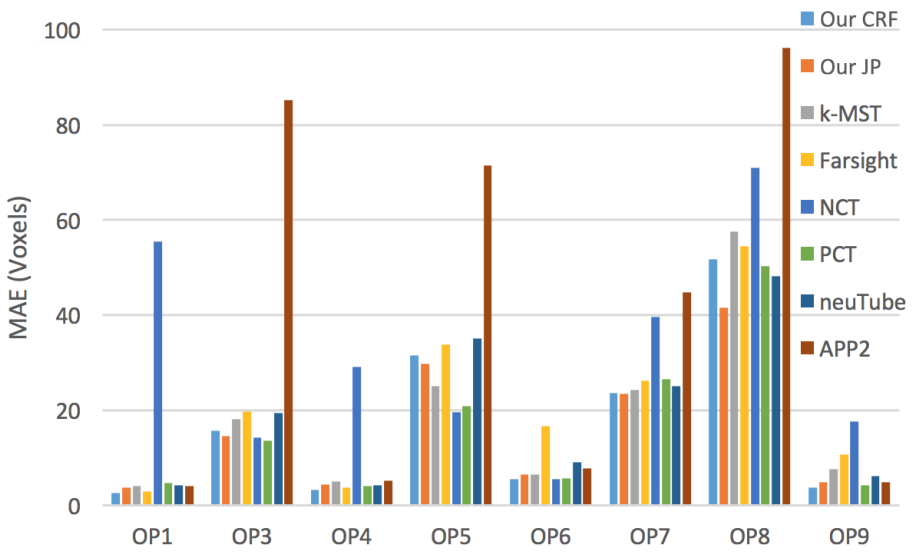


Figure 4.7. MAE score on OP dataset of our methods and existing frameworks.

Table 4.2.
F1 score on *Olfactory Projection Fibers* dataset of our methods and existing frameworks per sample stack; the best scores are shown in bold.

	Our CRF	Our JP	k-MST	Farsight	NCT	PCT	neuTube	APP2
OP1	**0.875**	0.680	0.818	0.840	0.399	0.700	0.753	0.680
OP3	**0.526**	0.245	0.429	0.466	0.495	0.132	0.438	0.197
OP4	**0.796**	0.576	0.624	0.745	0.439	0.606	0.742	0.506
OP5	**0.167**	0.029	0.072	0.004	0.106	0.019	0.098	0.069
OP6	0.194	**0.271**	0.137	0.000	0.226	0.261	0.184	0.239
OP7	**0.586**	0.537	0.389	0.521	0.534	0.428	0.445	0.399
OP8	0.626	0.308	0.376	0.639	**0.658**	0.251	0.572	0.270
OP9	**0.673**	0.458	0.504	0.524	0.420	0.408	0.482	0.457

which results in a lower DIADEM score. Although the APP2 algorithm can trace dense branches well, it is sensitive to noise so its score fluctuates. The noise sensitivity of the APP2 algorithm is reflected by the MAE score as shown in Figure 4.7.

Figure 4.8 shows the qualitative result on one of the image stacks from the CF dataset that our methods and others produced similar quality results. Figure 4.8(e)

Table 4.3.
DIADEM score on *Cerebellar Climbing Fibers* dataset of our methods and existing frameworks per sample stack.

	Size (voxel)	Our CRF	Our JP	k-MST	Farsight	PCT	neuTube	APP2
CF1	$6120 \times 4343 \times 34$	0.241	0.146	0.383	0.243	0.144	0.000	0.421
CF2	$5100 \times 3102 \times 28$	0.173	0.266	0.107	0.101	0.170	0.096	0.243

Table 4.4.
F1 score on *Cerebellar Climbing Fibers* dataset of our methods and existing frameworks per sample stack.

	Our CRF	Our JP	k-MST	Farsight	PCT	neuTube	APP2
CF1	0.430	0.443	0.295	0.242	0.274	0.062	0.079
CF2	0.328	0.340	0.271	0.242	0.171	0.113	0.074

Table 4.5.
Precision and recall on *Cerebellar Climbing Fibers* dataset of our CRF method and three other existing frameworks per sample stack.

Image	Our CRF		k-MST		Farsight		APP2	
Stack	P	R	P	R	P	R	P	R
CF1	0.255	0.241	0.108	0.120	0.106	0.089	0.014	0.022
CF2	0.218	0.260	0.128	0.170	0.142	0.136	0.016	0.031

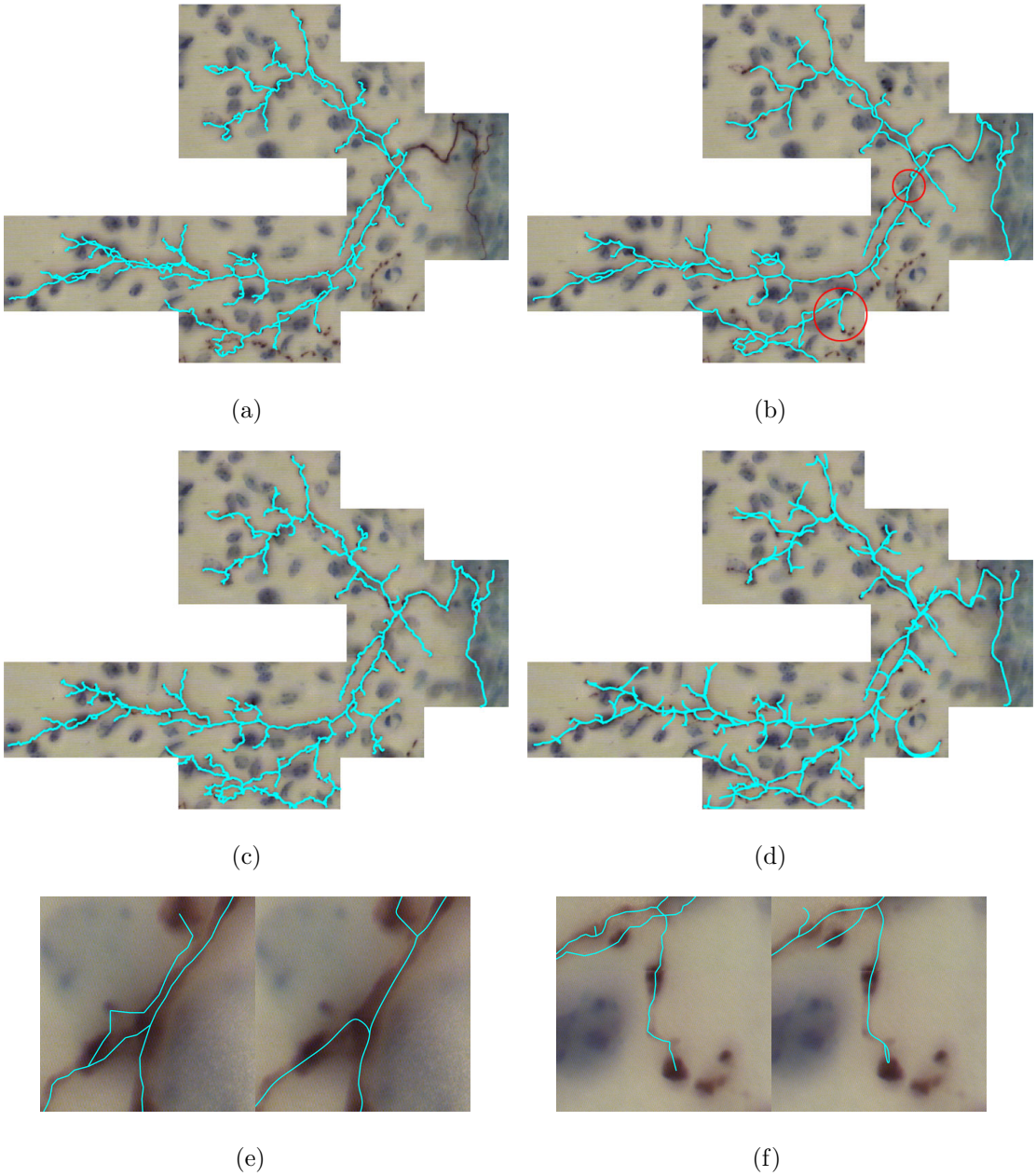


Figure 4.8. Tracing results (in cyan color) on the CF1 image stack of the DIADEM *Cerebellar Climbing Fibers* dataset. The first row shows (a) the ground truth and (b) the result of our CRF method. The second row shows traces obtained from (c) k-MST based method [125] and (d) APP2 algorithm [124]. The third row shows the ground truth and our CRF results respectively at magnified regions circled in (b). From left to right, (e) the example of parallel trace that our method failed to detect and (f) an example of noise similar to neurite that was correctly traced.

shows the main issue of our method that we encountered in this dataset when two or more adjacent dendrites elongate in the direction parallel to each other. Our method failed to recognize two separate neurites and merged them into one neurite, resulting in the over-simplified topology. Hence, our DIADEM score is lower than the APP2 algorithm as shown in Table 4.3.

Nevertheless, our methods produce a much higher F1 score as well as precision and recall compared to others, especially the APP2 algorithm, as shown in Tables 4.4 and 4.5. The reason is that there are segments from other neurons that appear in the image stack. Some of these undesired segments locate close to the neuron we are tracing; and the gaps between these segments and the neuron are so small that other methods consider them as the intensity inhomogeneity, and incorrectly merge these gaps. Our methods manage to distinguish the gaps and intensity inhomogeneity by incorporating the global shape geometry through the interaction potential in Section 4.3.1 and 4.4.2 as shown in Figure 4.8(f).

4.6.2 FlyCircuit

We evaluate our CRF method on 10 image stacks from FlyCircuit dataset. Most data volumes in this dataset have a constant background similar to the DIADEM dataset in the previous section. Hence, the main source of error comes from the topology recovery and neurite detection over cluttered dendrite branches.

Here we emphasize the significance of the interaction among the population of open-snake. The results of two sample volumes of our CRF method with and without the interaction potential in Section 4.4.2 are shown in Figure 4.9. In eq. (4.16), when $\mathbf{I}(s_i, s_j, \mathbf{X}) = 0$, our method evolves snakes independently based solely on the local image feature. So it is no difference to the conventional frameworks that evolve snakes in the sequential manner and ignore the global shape information. As a consequence, snakes failed to trace over locally ambiguous regions, e.g., bifurcation points and branches with inhomogeneity intensity; therefore, gaps occur in the result traces

and the neuron fragmentation issue arises. On the other hand, our results with the interaction potential do not have any of these issues because the global shape geometry of neurons is taken into account to resolve the ambiguities.

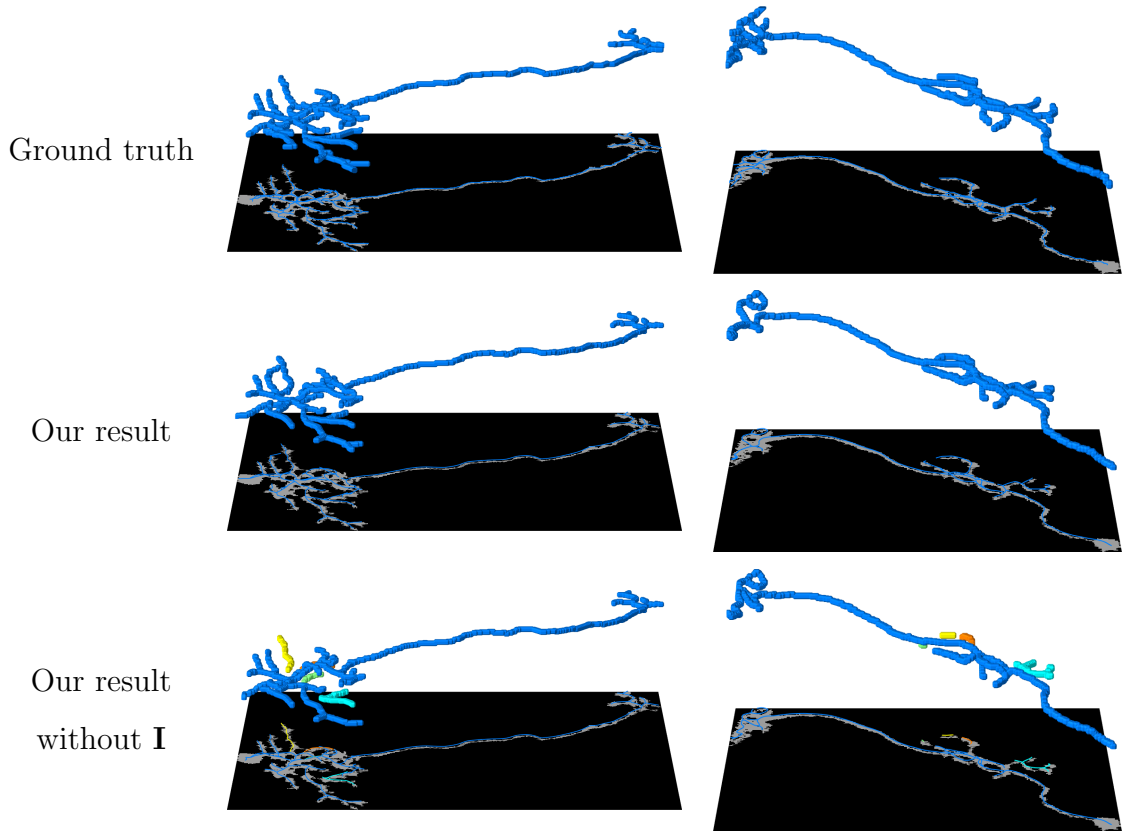


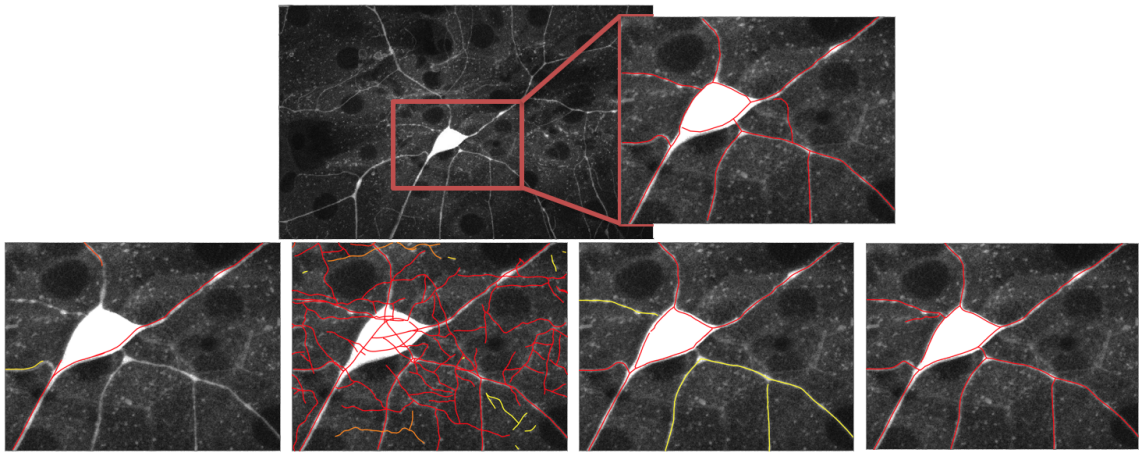
Figure 4.9. The significance of the interaction potential. From top to bottom, the ground truth, our tracing results with and without the interaction potential, i.e., $\mathbf{I}(s_i, s_j, \mathbf{X}) = 0$, respectively. Columns correspond to two sample volumes from FlyCircuit dataset. The blue colored lines represent the 3D traces. Their 2D projection overlays the maximum intensity projection of the image stack. Other colors show neurite fragments that failed to merge with the neuron.

4.6.3 Larval *Drosophila* Sensory Neurons

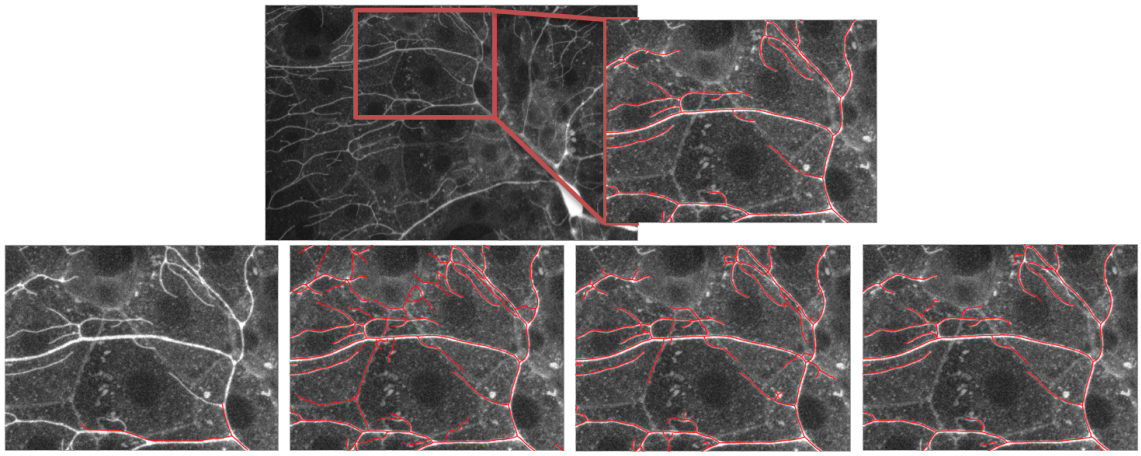
All algorithms were tested on our datasets depicting sensory neurons in the wild-type larval *Drosophila*. These volumes are around 1024 × 1024 × 20 in size and representative of spatially inhomogeneous signal-to-noise ratios. Indicatively, on a MacBook Pro (2.7 GHz Intel Core i7, 16GB 1600 MHz DDR3), our second method [81] in Section 4.4 ran on average 20 minutes per image stack (Matlab v.2014a), compared to our first method [80] in Section 4.3 that needs on average 2 hours per stack. Each of the methods [124], [125], [23], [20], and [63] ran on average 10-60 minutes per stack.

In Figure 4.10, we illustrate tracing results of our approaches, compared with the two existing methods. The two cases shown in (a) and (b) are typical examples of where our approaches provide increased robustness, in the presence of varying contrast and noise. According to the DIADEM score in Table 4.6, our CRF method performs better in most of the examined volumes. The quality of our results in term of distance from the predicted traces to the ground truth is also superior overall, as shown by the MAE score in Figure 4.11. However, the F1 score of our method is comparable to our JP method [80] as shown in Table 4.7. Note that our definition of the F1 score for this application only considers the correctness of the trace localization. The DIADEM metric also accounts for nodes having matching paths from matching ancestor nodes, as well as path length errors [137]. These comparisons, along with the complexity analysis in Section 4.5 and the average execution times we reported above, justify our claim that our CRF approach improves both accuracy in topological recovery over noisy data and the trade-off between accuracy and efficiency compared to our JP method.

Figure 4.12 shows the qualitative analysis of two sample volumes that our methods produce lower DIADEM score than existing methods, namely the k-MST based method [125] and Farsight [23]. The k-MST based method can accurately trace main branches but it fails to trace small neurites. The DIADEM score assigns weight to trace nodes based on the size of the subtrees, and it takes into account the correctness



(a)



(b)

Figure 4.10. Qualitative comparison between the ground truth traces, the results of our methods, and the competition in two sample volumes of *Drosophila* sensory neurons. Top rows in (a) and (b) (from left to right): collapsed image stack of the neuron, and magnified region where the ground truths (in red) from manual tracing are shown. Bottom rows in (a) and (b), the collapsed 3D traces superimposed on the collapsed stacks for illustration purposes (from left to right): NCT [20], Farsight [23], and our JP and CRF methods [80,81] respectively. Different colors correspond to different neurons.

of both topology recovery and trace localization. Due to the fact that the poor trace quality of nodes close to the root can overwhelm the DIADEM score, our results produce a lower score than the k-MST based method in a few cases. On the other hand, the Farsight method evolves snakes independently and sequentially so the discontinuity may occur in the trace. There are a few image volumes without discontinuities so the Farsight method performs better than ours on this occasion. Nonetheless, our methods outperform the competition in trace localization as shown by the average precision and recall in Figure 4.13.

Finally, we must note that we have identified one main problem that causes inaccuracies in our methods, which is also common for most existing approaches: perceived, even after pre-processing, gaps in the neurite that bring up the issue of the trade-off between 'bridging' gaps and falsely tracing different neurites in the vicinity. Intuitively, according to our methods, when more emphasis is given to the attraction between snakes, it is more likely for such gaps to be bridged. In single neuron imaging, this is not a problem and our methods yield great accuracy. However, when more than one neurons are present in the examined volume, a gap-width threshold must be manually imposed.

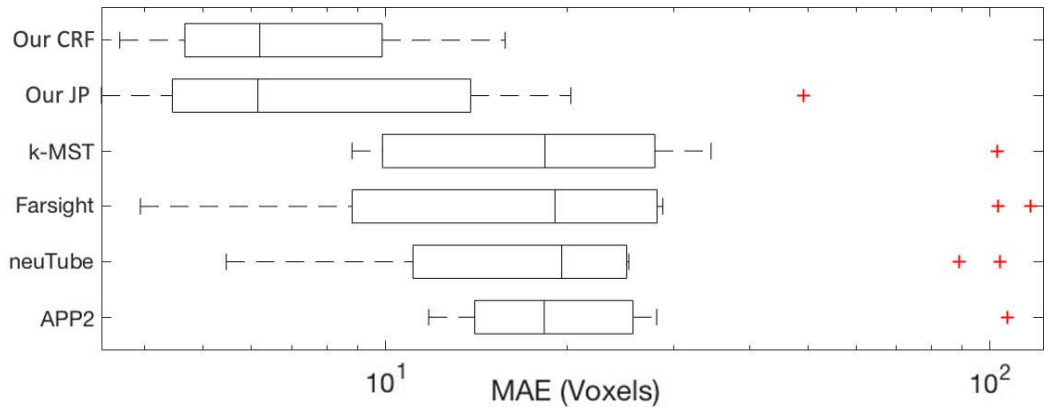


Figure 4.11. The box plot of the MAE score in log scale of our methods and existing frameworks.

Table 4.6.
DIADEM score on larval *Drosophila* sensory neurons dataset of our methods and existing frameworks per sample stack; the best scores are shown in bold.

	Size (voxel)	Our CRF	Our JP	k-MST	Farsight	NCT	neuTube	APP2
#1	$1024 \times 1024 \times 23$	**0.669**	0.669	0.420	0.284	0.029	0.151	0.109
#2	$1024 \times 1024 \times 13$	**0.617**	0.496	0.389	0.153	0.003	0.298	0.162
#3	$1024 \times 1024 \times 19$	0.286	0.232	**0.393**	0.088	0.034	0.048	0.046
#4	$1024 \times 1024 \times 15$	**0.615**	0.275	0.288	0.012	0.185	0.006	0.150
#5	$1024 \times 1024 \times 24$	**0.537**	0.312	0.355	0.417	0.107	0.052	0.453
#6	$1024 \times 1024 \times 11$	0.581	0.539	**0.684**	0.096	0.460	0.369	0.630
#7	$1024 \times 1024 \times 16$	**0.804**	0.572	0.596	0.582	0.314	0.703	0.724
#8	$1024 \times 1024 \times 16$	0.811	**0.813**	0.541	0.513	0.508	0.582	0.776
#9	$1024 \times 1024 \times 14$	**0.606**	0.565	0.404	0.573	0.074	0.395	0.306
#10	$1024 \times 1024 \times 15$	0.704	0.627	0.330	**0.784**	0.272	0.233	0.428
#11	$1024 \times 1024 \times 16$	**0.708**	0.526	0.349	0.704	0.222	0.044	0.491

Table 4.7.
F1 score on larval *Drosophila* sensory neurons dataset of our methods and existing frameworks per sample stack; the best scores are shown in bold.

	Our CRF	Our JP	k-MST	Farsight	NCT	neuTube	APP2
#1	**0.901**	0.864	0.852	0.686	0.157	0.683	0.463
#2	**0.811**	0.615	0.727	0.604	0.097	0.658	0.516
#3	0.737	0.739	**0.829**	0.470	0.199	0.470	0.426
#4	**0.837**	0.794	0.529	0.319	0.548	0.231	0.399
#5	**0.916**	0.871	0.698	0.728	0.163	0.319	0.520
#6	0.909	**0.921**	0.768	0.543	0.319	0.797	0.719
#7	0.926	**0.944**	0.823	0.905	0.111	0.848	0.730
#8	0.923	**0.939**	0.830	0.869	0.198	0.879	0.737
#9	0.903	**0.919**	0.695	0.805	0.222	0.663	0.732
#10	0.921	**0.935**	0.665	0.931	0.316	0.597	0.742
#11	0.944	0.884	0.639	**0.951**	0.266	0.432	0.752

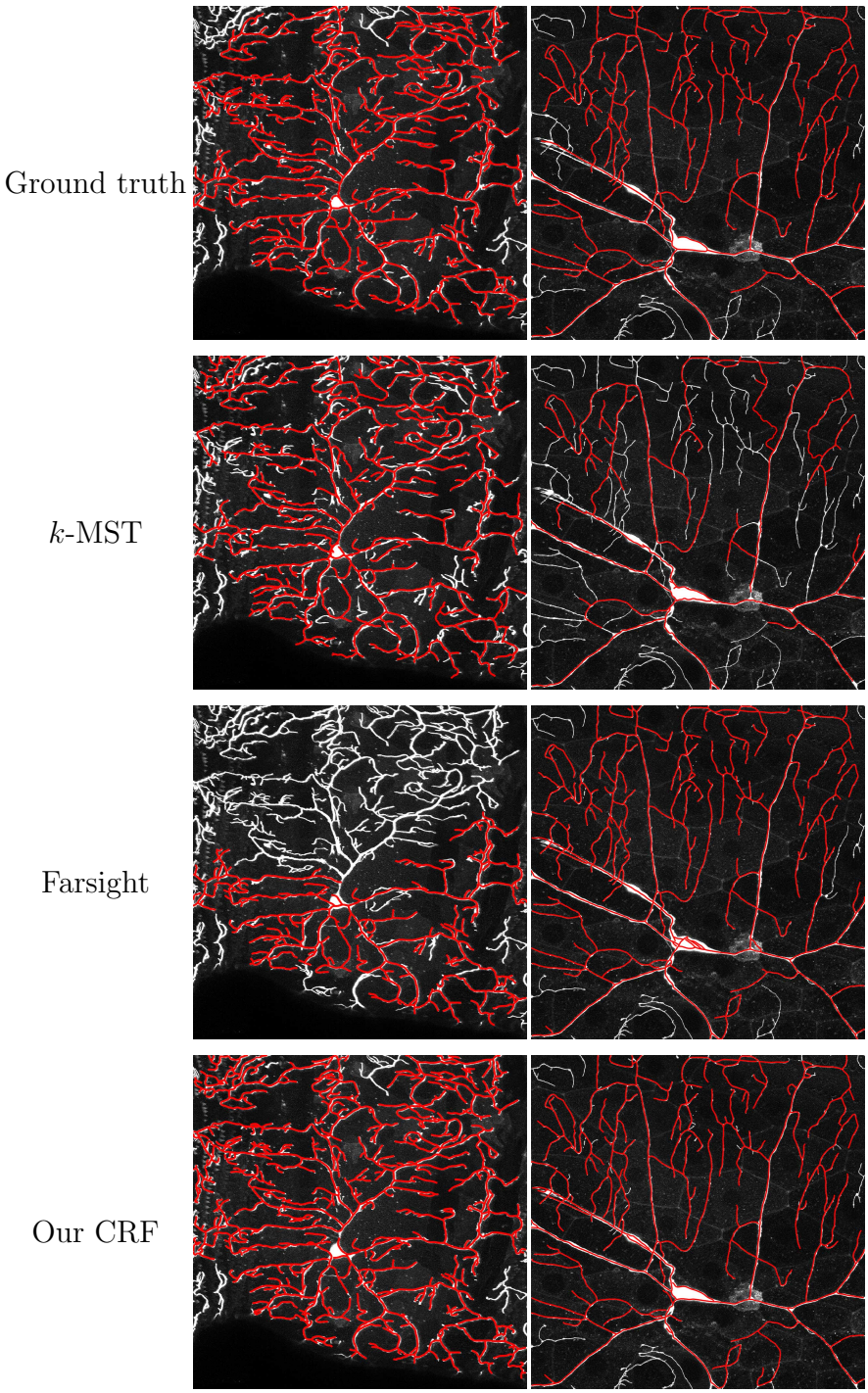


Figure 4.12. Tracing results (in red color) on the sensory neurons of larval *Drosophila* dataset. Columns correspond to different stacks (#6 and #10). From top to bottom: the ground truth, the results of k-MST based method [125], Farsight [23], and our CRF method [81] respectively.

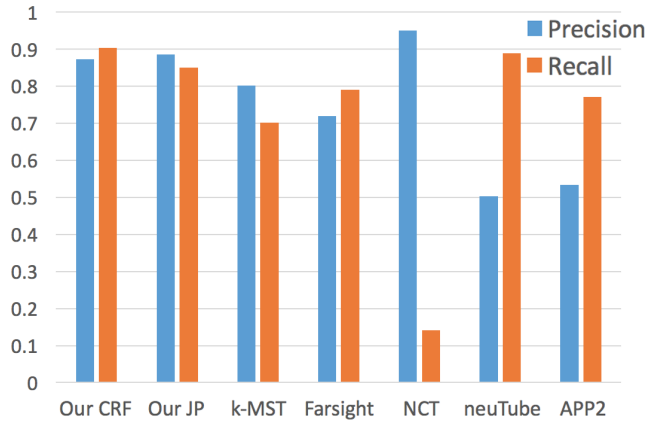


Figure 4.13. Average precision and recall on larval *Drosophila* sensory neurons dataset of our methods and existing frameworks.

4.7 Conclusion

We presented the joint probability between shape and appearance and the conditional random field frameworks. They are for evolving populations of snakes and capturing neurite centerlines in three dimensions. The implementation of the simultaneous evolution allows the incorporation of the global shape geometry of neurons to address the noise and fragmentation problems due to spatially varying contrast. Our CRF method considers each snake as a site, and spatially neighboring sites interact with each other, which results in snake collisions and finally merging. We showed theoretically and experimentally how we improve efficiency and accuracy compared to our JP method in [80], while maintaining increased accuracy compared to other existing approaches. We validate our results using publicly available datasets, namely the DIADEM challenge and FlyCircuit datasets, as well as sensory neurons in the wild-type larval *Drosophila*, whose volumes illustrate spatially varying signal-to-noise ratios.

5 NEURON TRACKING IN TIME-LAPSE CALCIUM IMAGES

In this chapter, we examine temporal morphology changes to gain understanding of how neuronal mechanisms are related to the morphology dynamics. Topology changes are captured over the appropriate (domain-specific) time intervals and recorded along with other neuronal properties to reveal their association that controls the neuronal mechanisms. Here, our case study is about how the neuronal activity and morphology dynamics respond to motion stimuli during larval locomotion. The morphology analysis of single data volume, i.e., neuron tracing and segmentation, is insufficient. Therefore, we develop the novel neurite tracking method [139] for time-lapse calcium images to capture the morphology dynamics of neurons. A video presentation of our work with supplemental material is available at `https://youtu.be/N4TLjLFP8-M`. Our method follows the local-to-global approach to handle severe deformations of neurons and local intensity ambiguities from calcium images.

5.1 Tracking Morphology Dynamics over Time-Lapse Sequence

Neuronal topology and morphology dynamics control how neurons respond to stimuli [34]. Nonetheless, it remains unknown exactly how these neurons are activated during the responses. To investigate the underlying molecular mechanisms, we need the reconstruction of the morphology dynamics, which involves neuron tracking problem. Hence, neuron image sequences are used instead of static images. Moreover, the changes in the intrinsic properties of neurons must be gathered as well over the suitable time intervals depending on the properties of interest. Then, these two features could be analyzed together to establish the patterns of the system. In this case study, we explore the correlation between morphology dynamics and neuronal activity in order to understand the locomotion behaviors of *Drosophila* larvae.

Sensory neurons beneath the epidermis of larval *Drosophila* [140] provide important proprioceptive feedback to the brain. In the absence of sensory signals from these proprioceptors, larval locomotion is uncoordinated and slow. Although the signals of these sensory neurons are important to proper locomotion [141], it is unknown exactly what these signals are. For instance, what is the relationship between neuronal firing and muscle contractions in a segment? Are sensory neurons activated by stretching during muscle relaxation? Or are the sensory neurons activated during muscle contraction?

To uncover the relationship between the neural activities and the locomotion mechanism of the *Drosophila* larvae, the imaging process has been developed to visualize the calcium responses in the proprioceptive neurons during larval locomotion. Calcium imaging reveals spatio-temporal information of activities in neurons at the single-cell level. Due to the large amount of image sequences required for learning the link between locomotion and sensory neuron activity, an automated neurite tracking system is needed. However, a unique characteristic of calcium images is that they produce low responses when there is little or no neuronal activity, which causes parts of dendrites to become ambiguous or even invisible. In addition, larval locomotion produces movement of the dendrites, which results in severe deformations, rendering dendrite tracking rather challenging.

Here we consider the time-lapse sequences over a short period of time so there is no sudden changes to neuronal topology, apart from the displacement of neurons. To address these problems, our method represents the neuron trace by the generalized pictorial structure. Neuron traces are modeled following the Markov random field (MRF) framework, where pre-determined neurite parts (branches) are the sites, adjacency between parts is captured by edges, and global configuration also includes relative part orientation. Using hard constraints on topology allows us to tackle intensity ambiguities and abrupt deformations during locomotion.

Section 5.3 describes how we integrate 5 factors that drive our tracking system: a) image feature, b) global dynamics, c) displacement smoothness, d) repellent term, and

e) line potential. These factors take into account appearance, global motion, local deformations, part-wise structure, and local shape constraints respectively. Each factor addresses the issue that occurs in neuron tracking. No frameworks in our knowledge have ever incorporate this number of factors before. They adopted only the partial of factors, so our method outperforms others because they did not handle all issues like ours. Section 5.4 explains the implementation details of our pictorial structure that make the problem feasible. In Section 5.5, our method is compared against the state-of-the-art optical flow estimation technique and the automated tracing method applied to each frame; all methods are validated with our calcium image dataset.

5.2 Related Work

In addition to existing problems in neuron reconstruction, another main issue of neuron tracking in time-lapse sequences is the mapping function estimation between consecutive image stacks. Existing tracking approaches usually fail to detect and track neurites under spatially inhomogeneous intensity ambiguities, commonly observed in calcium images, and under non-smooth movement patterns. There are a number of techniques proposed for recovering the mapping function. Here we reviewed three possible solutions: tracking-by-detection, articulated neuron, and optical flow.

5.2.1 Tracking-by-Detection

The simplest way to track neuron is to apply the neuron detection technique to every frame. There are a number of methods for neurite tracing in each frame independently, such as neuron reconstruction methods discussed in Chapter 4. Then, trace registration, e.g., [44], can be applied to improve temporal smoothness and obtain local trajectories.

Tracing and registration problems could be solved simultaneously using methods similar to [24, 95]. The method traces neuron at multiple time instances simultaneously by enforcing consistency within and across image stacks, yielding more robust

results. However, these methods rely on seed point detection for initialization, which are not robust to disappearing neurite parts and local deformations as appeared in calcium images, resulting in error accumulation over time.

5.2.2 Articulated Neuron

In neuronal movement analysis, the centerlines of neuron's axon and dendrite can be represented by tree data structures. With the isomorphism property, neurons can be tracked by fitting a fixed model. We can model the neuron's dendritic tree using the kinematic structure of articulated models in a similar way to body pose tracking and recognition [142, 143]. Authors represented the human body by the loose-limbed model, which is a variant of the pictorial structure [82, 83] with elastic connections between adjacent parts. Tracking problem can be formulated as a graphical model, where body parts are sites and their configuration contains the position and orientation of parts. Edges encode the position and angle relationships between adjacent parts. Other variations of kinematic skeleton-based methods are reviewed in [144].

The articulated neuron can also employ a part-wise (global) shape model, which incorporates the domain knowledge with respect to the neuron's tree structure and motion, to provide the desired robustness. Furthermore, the assumption about the neuron structure also limits the number of possible solutions and makes the problem feasible. Another benefit of using the model like the articulated body is that only regions of interest are detected and tracked, along with the recovery of its pose estimation or the neuronal morphology in an efficient manner.

An issue we face in the neuron tracking problem, with respect to existing articulated models, is the neurite's high degrees of freedom (large configuration search space), which is due to a potentially large number of branching points that define the considered joints. Second, the degrees of freedom vary among different neurites, as opposed to a pre-determined fixed structure (e.g., human body). Another issue

is the unreliable image cues from calcium images, unlike the issues in human-body tracking, which are self-occlusion and appearance variation from clothing.

5.2.3 Optical Flow

A large number of optical flow-based methods were reviewed in [84]. Optical flow produces dense motion fields over the entire image domain. Two classic methods for optical flow estimation are Lucas-Kanade [145] based on the the least-squares approach, and Horn-Schunck [146] following the variational approach. These methods are local methods because they take a small vicinity into account for updating the optical flow. Their main issue is the large displacement. The standard technique for coping with the issue is to adopt the coarse-to-fine scheme [147, 148]. Its drawback is that small and/or rapidly moving objects are smoothed in the coarse level and disappear into the background.

Recently, new methods combine the feature matching to handle the very large displacement problem [149]. Due to the recent breakthrough in feature matching using the deep learning technique [150, 151], new optical flow methods are able to handle occlusions and large displacement motions by incorporating the feature matching technique using the Convolutional Neural Network (CNN) [152]. To handle the local minimum problem, some methods solve the optical flow estimation in the discrete space, which allows them to employ the efficient approaches for finding good approximated global optimum [21].

Most optical flow methods follow the temporal intensity smoothness constraint [84], which assumes that pixel intensity remains constant locally during displacement. This assumption does not hold in calcium images whose responses change with the activities of neurons. While some methods, e.g., [84], employ successful data-specific modeling that limits their generalization. Moreover, the deforming tissue during locomotion introduces noise in the background, which occupies most of the image area, yielding misleading neurite movement despite any spatial regularization/smoothing.

5.3 MRF-Based Neurite Tracking

Let the centerline of a neurite at time $t \in [0, \mathrm{T}]$ be represented by the pictorial structure model [82, 83], $\mathcal{G}^t = \{\mathbf{X}^t, \mathcal{E}_R\}$, where $\mathbf{X}^t = \{\mathbf{x}_i^t\}$ is the set of the tree structure vertices, $i = 1, ..., n$, depicting Cartesian coordinates, and $\mathcal{E}_R = \{e_{ij}^R\}$ is the set of edges between all pairwise adjacent vertices (i, j), $i \neq j$ (Fig. 5.1). Let $K_n = \{\mathcal{V}, \mathcal{E}\}$ be the complete graph of the set of vertices $\mathcal{V}$ with n vertices and edge set $\mathcal{E} = \{e_{ij}\}$, and $\mathcal{E}_G = \{e_{ij}^G\}$ be the set of repulsive edges [153], which are not edges of the tree structure, $\mathcal{E}_G = \mathcal{E} \setminus \mathcal{E}_R$. Also, let $\mathbf{V}^t = \{\mathbf{v}_i^t\}$ be a set of the mapping vectors that transform $\mathbf{X}^{t-1}$ to $\mathbf{X}^t$: $\mathbf{v}_i^t = \mathbf{x}_i^t - \mathbf{x}_i^{t-1}$.

We formulate the objective function as the conditional distribution over the neurite shape $\mathbf{S}^t = \{\mathbf{X}^t, \mathcal{E}\}$ for each frame, given the image sequence $\mathbf{I} = \{\mathbf{I}^t\}$ and previous neurite shape $\mathbf{S}^{t-1}$,

$$\begin{aligned} \widehat{\mathbf{S}}^t &= \arg\max_{\mathbf{S}^t} P(\mathbf{S}^t | \mathbf{I}, \mathbf{S}^{t-1}) \\ &= \arg\max_{\mathbf{X}^t} P(\mathbf{X}^t | \mathbf{I}, \mathbf{S}^{t-1}) \end{aligned} \tag{5.1}$$

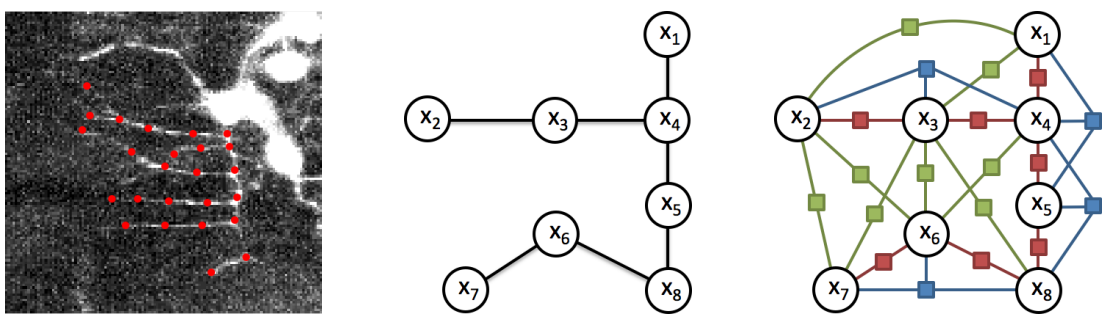


Figure 5.1. Graphical model principle for quantifying the topological structure among neighboring branches: pictorial structure (middle) of the considered neurites (left; dots show control points/nodes) and the corresponding graphical model (right). Red cliques $\mathcal{E}_R$ form the tree structure. Green cliques $\mathcal{E}_G$ represent the repulsive edges, while blue cliques correspond to the line potential. Some of the green and blue cliques are not shown for visualization purposes.

We assume that the topology of the neurite is fixed, i.e., $\mathcal{G}^t$ is isomorphic for $\forall t \in [0, T]$, and there are local deformations. Thus, our objective optimization corresponds to displacing the centerline locations $\mathbf{X}^t$. It can be solved using a second-order multi-label MRF, where sites are nodes $\mathbf{x}_i^t \in \mathbf{X}^t$ and their configuration is every possible coordinate.

$$P(\mathbf{X}^t|\mathbf{I}, \mathbf{S}^{t-1}) \propto \exp\left\{-\sum_{\mathbf{x}_i^t \in \mathbf{X}^t} E(\mathbf{x}_i^t, \mathbf{I}, \mathbf{S}^{t-1})\right\}$$

$$E(\mathbf{x}_i^t, \mathbf{I}, \mathbf{S}^{t-1}) = -\log U(\mathbf{x}_i^t, \mathbf{I}, \mathbf{S}^{t-1}) + \sum_{(\mathbf{x}_i^t, \mathbf{x}_j^t):\, e_{ij}=1} W(\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{S}^{t-1}) + \sum_{\substack{(\mathbf{x}_i^t, \mathbf{x}_p^t, \mathbf{x}_q^t): \\ \{e_{ip}^R, e_{iq}^R\}=1,\ p \neq q}} H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t), \tag{5.2}$$

In the above notations, we consider binary representation of the adjacency matrix of the graph $\mathbf{S}^t$. The term U is the unary potential (Sec. 5.3.1), W is the pairwise smoothness (Sec. 5.3.2), and H is a local line-type smoothness (Sec. 5.3.3), where we also show that the careful choice of the function H allows us to decompose the second-order term into a pairwise function without adding an auxiliary node; therefore, the graphical model can be described by the graph $\mathbf{S}^t$.

5.3.1 Unary Potential

The unary potential determines the centerline position based on image features and global model dynamics,

$$U\left(\mathbf{x}_i^t, \mathbf{I}, \mathbf{S}^{t-1}\right) = U_{img}\left(\mathbf{x}_i^t, \mathbf{I}\right) + U_{model}\left(\mathbf{x}_i^t, \mathbf{I}, \mathbf{S}^{t-1}\right) \tag{5.3}$$

Image feature term U_{img} in eq. (5.2) assigns node positions based on the responses of Frangi filter [96] for branches, and of another Hessian-based function for branching points (Fig. 5.2). Frangi filter is a *vesselness*-filter designed to detect only the line-like structure and it produces low responses over branch points; hence, a different function is required for detecting bifurcation points. In 2D, the branch point appears like a blob so we design a function similar to Frangi filter that has high value

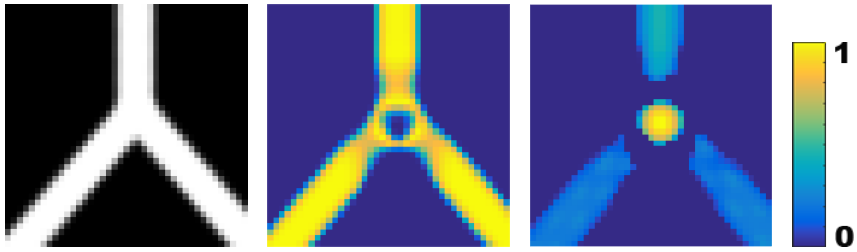


Figure 5.2. The synthetic neurite at a branching point (left) with the corresponding Frangi filter (middle) and the Hessian-based function (right) responses from eq. (5.4). Frangi filter yields lower response at the branching point while the Hessian-based function yields high response at the branching point and low values elsewhere.

when both eigenvalues of Hessian matrix have high magnitude (the sign indicates the brightness/darkness).

$$U_{img}\left(\mathbf{x}_i^t, \mathbf{I}\right) = \begin{cases} \left(\exp\left(-\frac{R_B^2}{2\beta^2}\right)\left(1-\exp\left(-\frac{S^2}{2c^2}\right)\right)\right) & \text{, if } deg(\mathbf{x}_i^t) \leq 2 \\ \left(\left(1-\exp\left(-\frac{R_A}{2\beta^2}\right)\right)\left(1-\exp\left(-\frac{S^2}{2c^2}\right)\right)\right) & \text{, otherwise} \end{cases} \tag{5.4}$$

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 branches; otherwise, they are branching points. $R_B = \lambda_1/\lambda_2$ indicates *vesselness* using the Hessian matrix eigenvalues (λ_1, λ_2) at $\mathbf{x}_i^t$, where λ_k is the eigenvalue with the kth smallest magnitude. $R_A = \sqrt{min(\lambda_1, 0) \cdot min(\lambda_2, 0)}$ measures the similarity with a blob (both eigenvalues are negative at branching points for images with dark background). $S = \lambda_1^2 + \lambda_2^2$ distinguishes the neurite from the background, while β and c are parameters controlling the sensitivity to (R_A, R_B) and S respectively [96].

Global dynamics term U_{model} in eq. (5.2) adjusts the position of each node $\mathbf{x}_i^t$ based on the global motion determined by the soma displacement. In calcium images, it is often very challenging to detect branches due to aforementioned problems; therefore, image feature alone is insufficient. Nonetheless, the soma is always visible. It can be detected using the circular Hough transform [154,155], and tracked using the FullFlow approach [21]. The dynamics model solves this problem by introducing the initial

guess of the centerline location through the global motion model to complement the image feature. Our global motion model at time t is denoted by $\mathbf{Y}^t = \{\mathbf{y}_i^t\}$, which is obtained by shifting the previous trace $\mathbf{S}^{t-1}$ so that the soma is at the current location, $\mathbf{y}_i^t = T_{t-1,t}(\mathbf{x}_i^{t-1})$ and $\mathbf{y}_i^0 = \mathbf{x}_i^0$, where $T_{t-1,t}(\mathbf{x}_i^{t-1})$ is the displacement operator for soma's position from $t-1$ to t applied to $\mathbf{x}_i^{t-1}$. Also, when locomotion (deformation and translation) is small and periodic, we can replace the previous trace $\mathbf{S}^{t-1}$ with the initial centerline configuration $\mathbf{S}^0$. With the above global motion model, the global dynamics term is formulated by,

$$U_{model}\left(\mathbf{x}_i^t, \mathbf{I}, \mathbf{S}^{t-1}\right) = \exp\left\{-\alpha_d \quad \mathbf{x}_i^t - \mathbf{y}_i^t \right. \tag{5.5}$$

where $\|\cdot\|$ is the Euclidean distance, α_d regulates the sensitivity to global motion.

5.3.2 Pairwise Potential

The pairwise smoothness term W in eq. (5.2) assigns node positions such that tracking is part-wise smooth over the edge set $\mathcal{E}_R = \{e_{ij}^R\}$, and the new neurite location maintains its global shape by including the repulsive [153] edges $\mathcal{E}_G = \{e_{ij}^G\}$,

$$W\left(\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{S}^{t-1}\right) = \begin{cases} \alpha_p \cdot \|\mathbf{v}_i^t - \mathbf{v}_j^t\| & \text{, if } e_{ij}^R = 1 \quad (5.6a) \\ \left[\left\langle \mathbf{x}_i^t - \mathbf{x}_j^t \quad \leq \quad \mathbf{x}_k^t - \mathbf{x}_l^t \right\rangle_{e_{kl}^R=1}\right] & \text{, if } e_{ij}^G = 1 \ \wedge \ br(\mathbf{x}_i^t) \neq br(\mathbf{x}_j^t) \quad (5.6b) \\ 0 & \text{, otherwise} \quad (5.6c) \end{cases}$$

where $\alpha_p = \left(2 \quad \mathbf{v}_i^t - \mathbf{v}_j^t\right)^{-1}$ scales small and large vector differences to a proper range [156], $[\cdot]$ is the binary indicator function, $\langle\cdot\rangle$ represents the expectation value, and $br(\mathbf{x}_i^t)$ denotes the branch where node $\mathbf{x}_i^t$ is in the tree structure. Again, in the above notation we consider binary representation of graph adjacencies, i.e., $\{e_{ij}^R, e_{ij}^G\}_{i \neq j} \in \{0, 1\}$.

The first case in eq. (5.6a) enforces the displacement smoothness between mapping vectors of adjacent nodes. It is defined as the difference between mapping vectors of

nodes $\mathbf{x}_i^t$ and $\mathbf{x}_j^t$ when $e_{ij}^R = 1$. It encourages adjacent nodes to have a similar mapping vector.

While the second case in eq. (5.6b) constrains the morphology such that dendrites repel each other to prevent undesired intersections. Repulsive edges e_{ij}^G are added between nodes from different branches [153]. Repellent term gives higher energy when dendrites overlap. A pair of dendrites are overlapped if the distance between them is less than the average length between adjacent nodes.

5.3.3 Line Potential

Local line-type potential H in eq. (5.2) ensures local branch shape smoothness using the internal energy of the active contour model [64]. Considering the neighbors $\mathbf{x}_p^t$, $\mathbf{x}_q^t$ of $\mathbf{x}_i^t$, i.e., $\left[\{\mathbf{x}_p^t, \mathbf{x}_q^t\} : \{e_{ip}^R, e_{iq}^R\} = 1\right]$, the line potential can be expressed in terms of curvature,

$$H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t) = \left(\mathbf{x}_p^t - \mathbf{x}_i^t\right)^2 + \left(\mathbf{x}_p^t - 2\mathbf{x}_i^t + \mathbf{x}_q^t\right)^2 \tag{5.7}$$

Computing curvature requires the second-order potential function. Unlike [157], here we avoid adding auxiliary nodes by using the triangle inequality, where the clique $\{\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t\}$ of three nodes is broken down into three pairwise cliques — $\{\mathbf{x}_p^t, \mathbf{x}_i^t\}$, $\{\mathbf{x}_i^t, \mathbf{x}_q^t\}$, and $\{\mathbf{x}_p^t, \mathbf{x}_q^t\}$,

$$H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t) = \|\mathbf{x}_p^t - \mathbf{x}_i^t\|^2 + \|\mathbf{x}_i^t - \mathbf{x}_q^t\|^2 - \frac{\|\mathbf{x}_p^t - \mathbf{x}_q^t\|^2}{4} \tag{5.8}$$

The optimal curvature value is obtained when $\mathbf{x}_p^t$, $\mathbf{x}_i^t$, and $\mathbf{x}_q^t$ form 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 our simplification in eq. (5.8) has the same optimum as the potential in eq. (5.7) (Fig. 5.3). The individual quadratic terms also enforce short distances between nodes.

The last term is divided by 4 to ensures that it is always $H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t) \geq 0$. According to the triangle inequality, we get,

$$\|\mathbf{x}_p^t - \mathbf{x}_i^t\| + \|\mathbf{x}_i^t - \mathbf{x}_q^t\| \geq \|\mathbf{x}_p^t - \mathbf{x}_q^t\|$$

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

Figure 5.3. Our curvature simplification. Although our decomposition is different from the curvature definition, they share the same optimum when nodes form a straight line, $\overline{\mathbf{x}_p^t\mathbf{x}_q^t} = \overline{\mathbf{x}_p^t\mathbf{x}_i^t} + \overline{\mathbf{x}_i^t\mathbf{x}_q^t}$ (left), and discourage 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).

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

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

5.4 Implementation

Algorithm 1 shows the summary of our method, given $\mathbf{I}$ and $\mathbf{S}^0$, where the Cartesian coordinates of the centerline location $\mathbf{X}^t$ is discretized for the feasibility of the problem. In line 8, the energy function of eq. (5.2) is non-submodular, and therefore the MRF is optimized using the α-expansion [53] with quadratic pseudo-boolean optimization [158,159], also known as *fusion moves* [54]. For efficiency, the search space is reduced by limiting the number of candidate positions and pruning the redundant edges from $\mathbf{S}^t$.

Specifically, to limit candidate positions, we consider the surrounding regions (within 20 pixels in our experiments) around the starting location $\widetilde{\mathbf{X}}^t$, which are the possible node locations based on the previous trace and the global displacement. The starting location of each node at time t during the optimization is computed by applying the displacement operator that shifts soma's location in the previous instance to the current position, $\widetilde{\mathbf{x}}_i^t = T_{t-1,t}(\mathbf{x}_i^{t-1})$.

Redundant edges are repulsive edges linking nodes that are unlikely to overlap; therefore, removing those edges has no effect to the model. $\mathbf{S}^t$ is pruned such that there is at most one edge between a node and a set of nodes on a different branch; we choose the edge determining the shortest distance. For example, in Fig. 5.1, node $\mathbf{x}_6$ would be connected (green cliques) to all other nodes, considering a fully connected graph. Instead, we simplify the graph accordingly to this schema by maintaining only the clique between $\mathbf{x}_6$ and $\mathbf{x}_3$ due to their proximity. This way, we limit the number of repulsive edges at each node to the maximum number of branches in the fixed-sized surrounding area, which is some constant values. Thus, we reduce the number of repulsive edges $|\mathcal{E}_G| = \mathcal{O}(N)$, and the total number of edges in our model is also $\mathcal{O}(N)$.

Algorithm 1: Neurite tracking

Input: $\mathbf{I}$, $\mathbf{S}^0$

Result: $\mathbf{S}$

for $t = 1$ *to* T **do**

  Compute $T_{t-1,t}$ using FullFlow [21]

  `// Soma is detected using the Hough transform [154, 155]`

  Update global motion model in eq. (5.5)

  `// For severe deformation,` $\mathbf{y}_i^t = T_{t-1,t}(\mathbf{x}_i^{t-1})$ `and` $\mathbf{x}_i^{t-1} \in \mathbf{S}^{t-1}$.

  `// For small and periodic deformation,` $\mathbf{y}_i^t = T_{0,t}(\mathbf{x}_i^0)$ `and` $\mathbf{x}_i^0 \in \mathbf{S}^0$.

  Initialize starting location $\widetilde{\mathbf{X}}^t = T_{t-1,t}(\mathbf{X}^{t-1})$

  Compute $U\left(\mathbf{x}_i^t, \mathbf{I}, \mathbf{S}^{t-1}\right)$ using eq. (5.3)

  Compute $W\left(\mathbf{x}_i^t, \mathbf{x}_j^t, \mathbf{S}^{t-1}\right)$ using eq. (5.6)

  Compute $H(\mathbf{x}_p^t, \mathbf{x}_i^t, \mathbf{x}_q^t)$ using eq. (5.8)

  $\mathbf{S}^t \leftarrow \arg\max_{\mathbf{X}^t} P(\mathbf{X}^t | \mathbf{I}, \mathbf{S}^{t-1})$

  `// Optimized by` ***`fusion moves`***

end

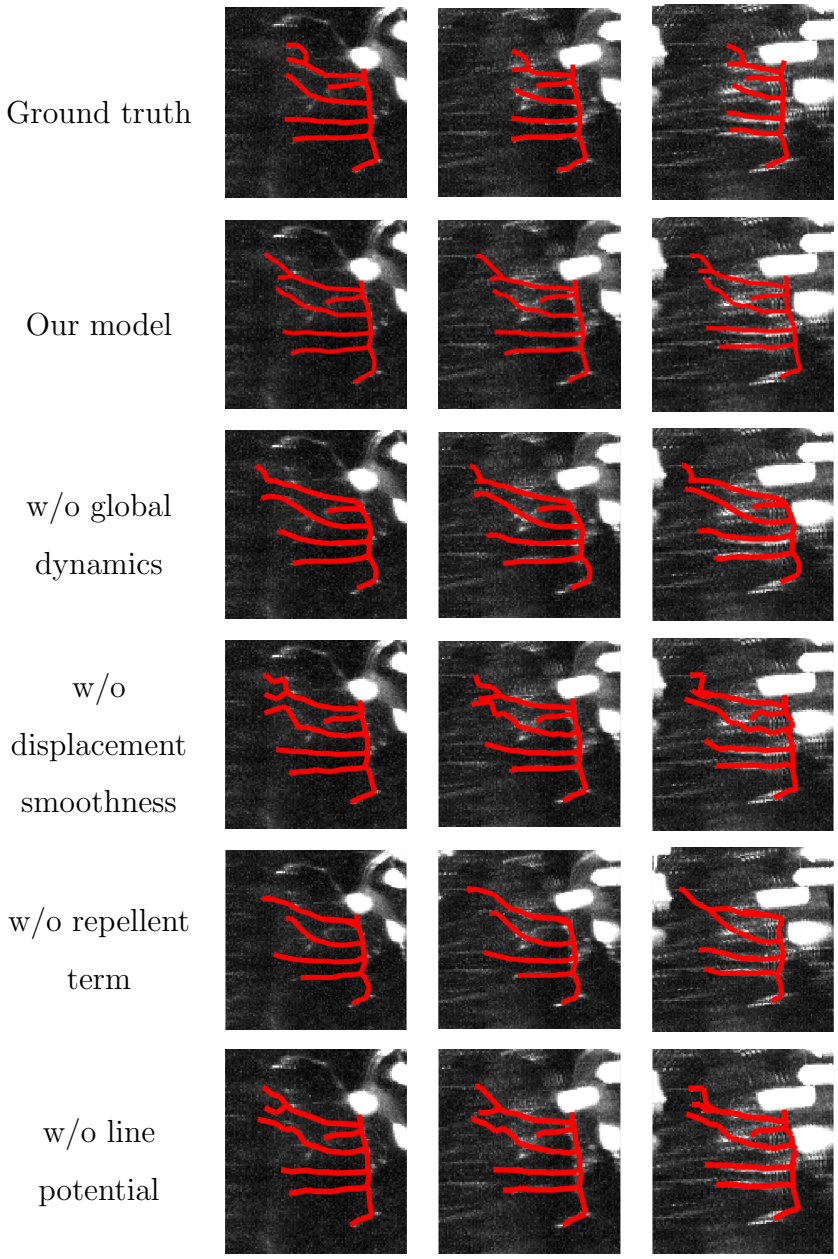


Figure 5.4. Tracking results (superimposed in red) for 3 successive frames (columns), depicting the importance of individual terms in our model. From top to bottom: ground truth, our integrated model, our model without the global dynamics term of eq. (5.5), our model without the local displacement smoothness (first case of eq. (5.6a)), our model without the repellent term (second case in eq. (5.6b)), and our model without the line potential of eq. (5.8). Note that here we study dendrite arborization neurons of class I type, which have the simplest arbor structure.

5.5 Results

Larvae expressing the genetically-encoded calcium sensor GCaMP6.0F in the proprioceptors were observed in four-dimensional confocal microscopy imagery. This is possible because the neurons are found directly beneath the cuticle and epidermis, which are optically transparent. Analysis of the neurons in these volumes presents specific challenges that are not present in typical calcium time series, since the neurons of interest are moving in 3D. However, the z-axis has low resolution due to the device limitation. Hence, we used the 2D maximum intensity projection images for tracking, albeit the availability of image stacks.

In this dataset, each sequence captured neuronal displacements and deformations during larval locomotion, with an average duration of about 10 frames per cycle. To obtain the ground truth, we traced the neurites of interest manually in frames, where they could be observed due to sufficient calcium signal, using neuTube [135]. For quantifying accuracy, we used five metrics: Spatial Distance (SD), Substantial SD (SSD), the percentage of sample points used in SSD (%SSD), false negative rate (FNR), and false positive rate (FPR). Specifically, let a ground truth trace be $\mathbf{S}_A$ and a calculated trace be $\mathbf{S}_B$. Then, SD($\mathbf{S}_A$,$\mathbf{S}_B$) is defined as the mean between (a) the average Euclidean distance of nodes in $\mathbf{S}_A$ to the closest nodes in $\mathbf{S}_B$ and (b) the average Euclidean distance of nodes in $\mathbf{S}_B$ to the closest nodes in $\mathbf{S}_A$. (Note: all nodes are one pixel apart.) Using only SD, mismatches that are due to local deformations cannot be captured, when the majority of the nodes between ground truth and calculated traces coincide. Therefore, we also used SSD($\mathbf{S}_A$,$\mathbf{S}_B$), which quantifies the SD error for only the mismatched nodes of $\mathbf{S}_A$ and $\mathbf{S}_B$ (nodes that are at least 2 pixels apart), and %SSD that indicates the ratio of mismatched nodes between $\mathbf{S}_A$ and $\mathbf{S}_B$, as detailed in [160]. In addition, we also computed FNR and FPR to measure geometric errors. These criteria identify what kinds of errors occurred: missing or extra branches. They are defined as follow,

$$FNR = M(\mathbf{S}_A, \mathbf{S}_B)$$
$$FPR = M(\mathbf{S}_B, \mathbf{S}_A) \tag{5.9}$$

where $M(\mathbf{S}_1, \mathbf{S}_2)$ is the metric that estimates the fraction of $\mathbf{S}_1$ omitted from $\mathbf{S}_2$. It is defined by,

$$M(\mathbf{S}_1, \mathbf{S}_2) = \frac{1}{n} \sum_{\mathbf{x} \in \mathbf{X}_1}^{n} \left(1 - \exp\left\{ \frac{d(\mathbf{x}, \mathbf{X}_2)^2}{2\sigma^2} \right\} \right) \tag{5.10}$$

where $d(\mathbf{x}, \mathbf{X}_2)$ is the distance from point $\mathbf{x}$ to the closest point in $\mathbf{X}_2$, and σ is the sensitivity parameter. Eq. (5.10) is equivalent to applying the Gaussian filter to $\mathbf{S}_2$. The Gaussian filter is required because it is extremely unlikely that $\mathbf{S}_1$ and $\mathbf{S}_2$ will precisely overlap. Therefore, σ indicates the tolerance of the distance error. The geometric errors are computed by the software called 'NetMets' [161].

We compare our method with the optical flow-based tracking, 'FullFlow' [21] and with the automated tracing feature in neuTube [58, 135] applied to each image stack independently. Our experiments were carried out on a Mac Pro (2×2.66 GHz 6-Core Intel Xeon, 20GB 1333MHz DDR3 ECC). The run time of our method was on average one second per frame, per neuron (for cases where multiple neurons were present in the data), with frame dimensions of 512×192 pixels.

Figure 5.4 shows the importance of each factor in our model, or a lack thereof. The global dynamics term is required to compensate for intensity ambiguities at the neurite region and enforces the initialization for the MRF optimization. Otherwise, nodes move towards pixels with high intensity regardless of neuron's shape, especially the dendrites close to adjacent neurons. Displacement similarity between adjacent nodes prevents local movement ambiguities and results in locally smooth shape mapping between time instances. The repellent term prohibits intersections between branches. Finally, the line potential yields local branch smoothness.

We evaluate our method and the competition on sequences with low and high signal-to-noise ratios (SNR), as shown in Figure 5.5. The box plot of SD scores in the first column shows that the results of our method and the optical flow-based tracking are comparable when SNR is high; however, when SNR is low, our method yields

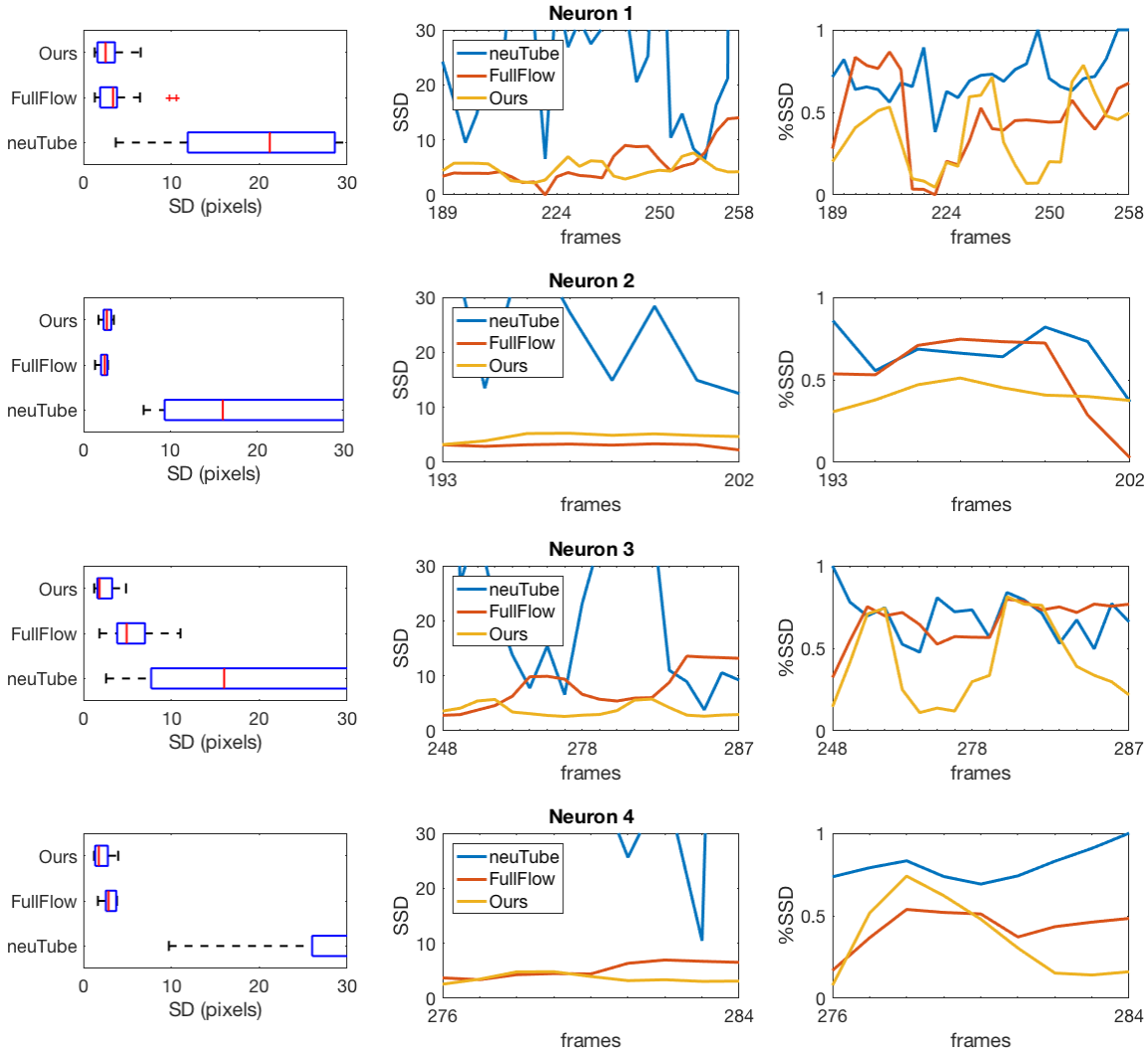


Figure 5.5. Similarity (error) scores — SD, SSD, and %SSD, of neuTube [58], FullFlow [21], and our method. Rows illustrate the tracking results over different neurons when SNR is high (top two rows) and low (bottom two rows). SD is displayed in the box plot (left), while SSD and %SSD are displayed for every frame (middle and right, respectively). The results correspond to indicative segments from longer sequences.

better accuracy. Whereas, the tracing method produces poor results because it fails to detect neurons when they are totally or partially inactive. The second and third columns illustrate the SSD and %SSD scores per frame (x-axis), respectively. Error peaks in our method (primarily visible in %SSD) occur when local neurites deform significantly, due to locomotion (and its effect on the 3D-to-2D projection), as well as blurring effects under significant movement. Yet, the error magnitudes are low — less than 5 pixels on average (SSD).

The results above are also supported by the qualitative observations in Figure 5.6. The optical flow-based method fails to recover the motion field when SNR is low so it yields poorly tracked centerline. While the tracing method is unable to extract the neuron in any case due to noisy data, it misses neurite branches and detects adjacent neurons instead. Our method, on the other hand, integrates global and local shape with motion information and intensity to produce robust results.

In Figure 5.7, we show the overall performance of our method compared to the competition. Results agree across metrics that our tracking method outperforms others. Here we show the boxplot of SD, %SSD, FPR, and FNR over the whole dataset. Our data contain the ground truth of subsequences during the motion. We do not evaluate when neurons are stationary to avoid bias over the inactive periods since they occupy most of the sequence. These criteria point to the same suggestion that our method is superior as our method produces lower errors as shown by all four metrics. In addition, the errors our method produced tend to be FPR rather than FNR, which is more desirable if the total level of errors (i.e., the sum of FPR and FNR) is the same because over-segmentation is less severe than missing parts of neurons for this task. Tracing neuron in each frame independently ignores the time dependency, resulting in poor performance. On the other hand, the optical flow-based method performs quite well; however, the shortcoming is that errors occur when there is a significant movement, especially during the contraction, where neurons appear blurry owing to the out-of-focus effect. Then, errors continue to propagate throughout the whole sequence.

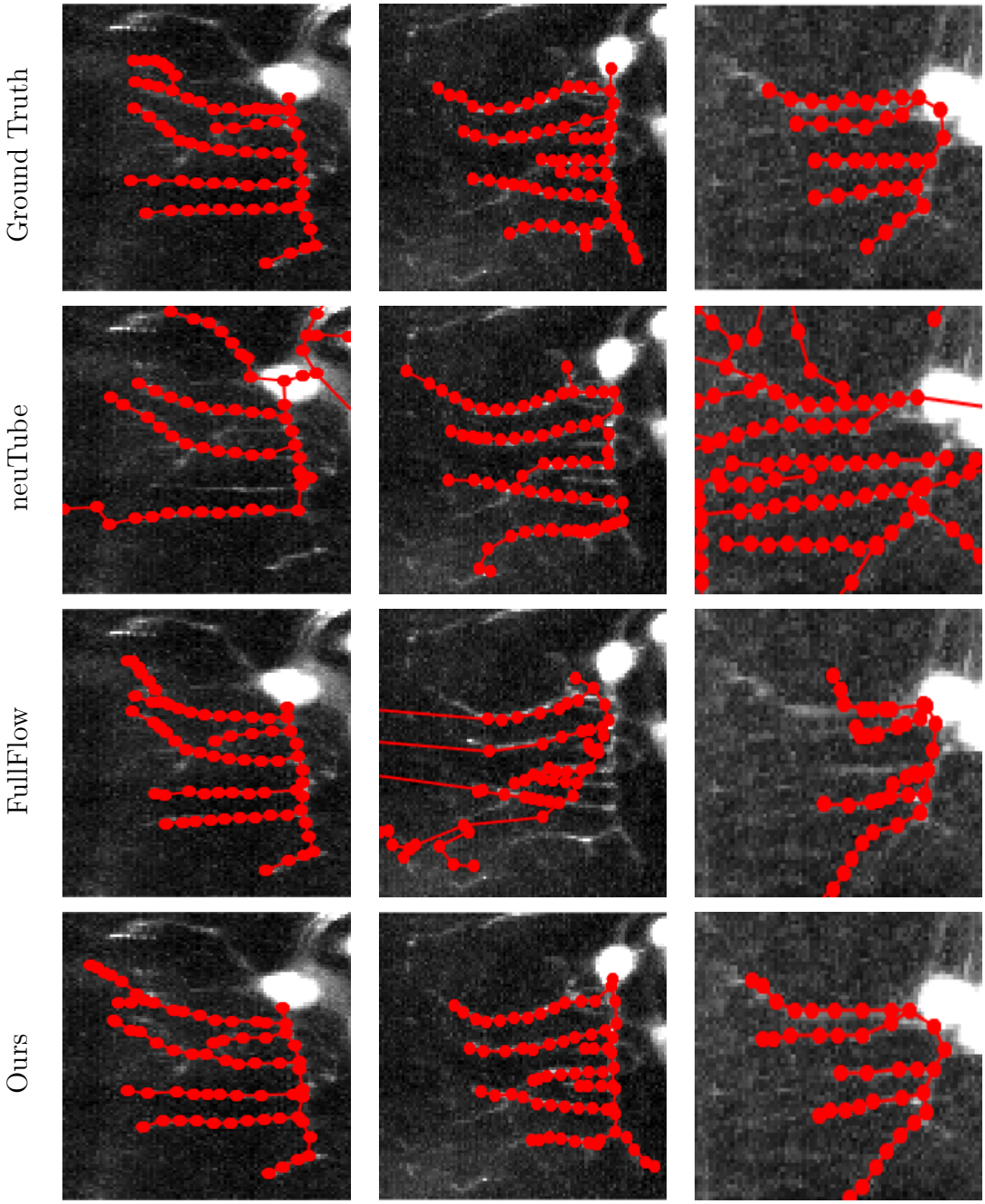


Figure 5.6. Qualitative comparisons (from top to bottom): ground truth, neuTube, FullFlow, and our method. Columns correspond to tracking in different frames when SNR is high (first column) and low (last two columns).

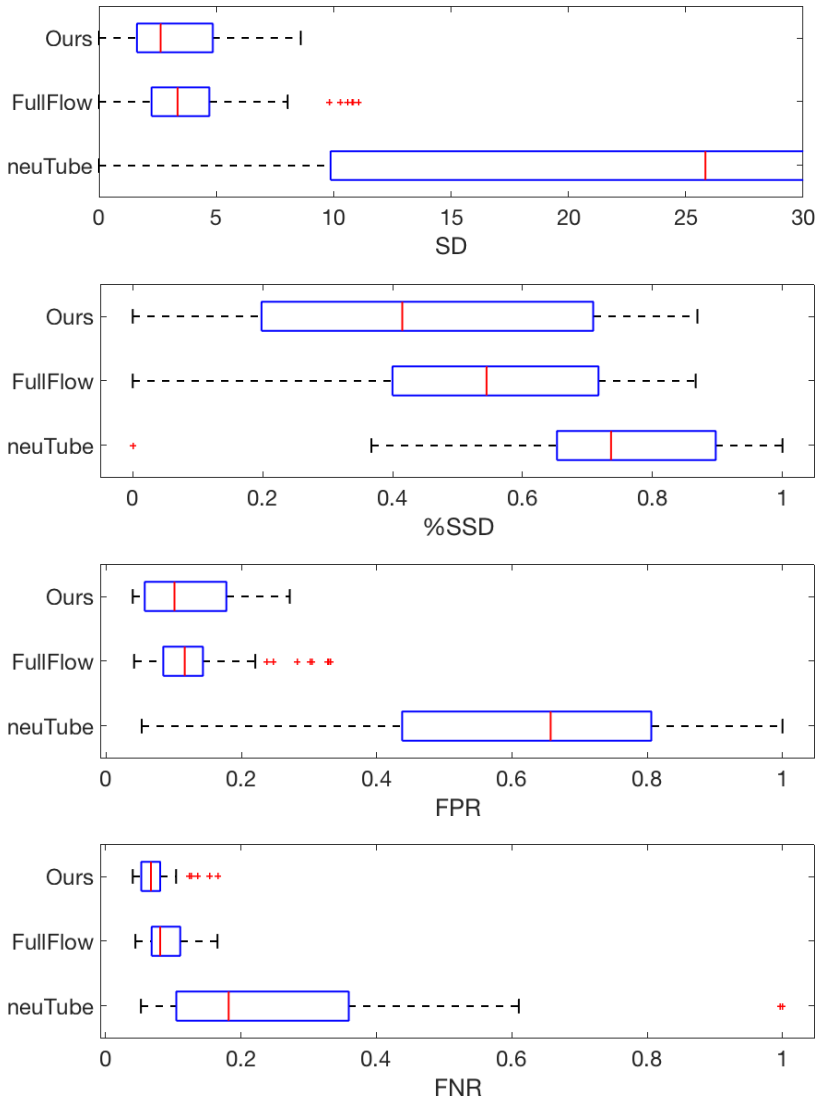


Figure 5.7. Box plots of SD, %SSD, FPR, and FNR respectively over the whole sequence during the motion.

5.6 Conclusion

We introduce a neurite tracking approach that allows for automated, *in vivo*, single-cell analysis from calcium image sequences. Our method incorporates local image appearance, global and local shape characteristics, and global motion. We formulate the objective with a non-submodular, second-order, multi-label MRF, considering the neurite as an articulated body. We evaluate our method using noisy calcium image sequences of *Drosophila* sensory neurons, and we compare our results with the optical flow-based tracking and automated tracing methods.

6 PART-WISE SEGMENTATION WITH ARTIFICIAL TEMPLATES

Neuron mechanisms cannot be explained based on a single aspect such as structure information obtained from neuron reconstruction. Hence, other aspects such as functionality must be considered as well. Integration of multi-modality images involves the domain knowledge-based part-wise segmentation problem to provide a meaningful registration and compartmentalization. We solve this problem using our novel simultaneous registration and segmentation method [162], where the registration is needed to ensure that segmentations comply with the artificial template encoding the domain knowledge. A video presentation of our work with supplemental material is available at `https://youtu.be/sDdOicyIq6I`. Our main contributions are developing the first method that uses the totally artificial template drawn by experts, and introducing the structure information, which is resolution-agnostic unlike the appearance information, in the global-to-local design, resulting in increased partition accuracy.

6.1 Domain Knowledge-Based Part-Wise Segmentation

Different modalities of single-neuron imaging can provide information for analyzing morphology at different levels of detail. For example, Mosaic Analysis with a Repressible Cell Marker (MARCM) [163] can be used for classifying global morphology [4], whereas GFP-labeled neurons under a confocal laser scanning microscope can yield branching details for local dendritic complexity analysis [80, 81]. In addition, different modalities also give different perspectives. For instance, nano-scale images from FRET (Förster Resonance Energy Transfer) microscopy provide the concentration map of protein-protein interactions on neurons, in contrast to micro-scale images from confocal microscopy. Integrating information across modalities would create the platform for analyzing multiple aspects of neurons altogether and give valuable

insights into the relationship between the structure and functionality of neurons. Furthermore, such scheme would provide the necessary tools to study how specific genes affect neuronal function and morphology by comparing wild-types and knockouts, as some mutations result in the drastic changes in the appearance of neurons.

The integration of multi-modality images is not a trivial task. It involves aligning different neurons in different resolutions and finding correspondences across neuron volumes for meaningful comparison. Volume registration techniques cannot be applied directly because the alignment must be executed based on some context that provides the biological meaning for further analysis. Instead, we segment image stacks according to the generated prototypes with pre-determined compartmentalization, which acts as a standardized representation of neurons. Then, the part-wise registration is applied to align neurons by compartments; this provides the biological interpretability. In this chapter, we develop the novel global-to-local method for part-wise segmentation of image stacks across different modalities with respect to the artificially created template that enables the comparison among neurons with meaningful interpretation.

Independently from spatial resolutions, generating morphology prototypes can provide top-down solutions for segmenting neuron volumes and partitioning them into pre-determined compartments (segmentation-by-registration). Instead of building (training) such prototypes using computational approaches, it is often required that domain expertise directly drives the objective: using microscopic observations and domain knowledge, we can design artificial (instead of data-driven) neuron prototypes with compartments annotated based on some context (e.g., topology with respect to the natural environment, shape, and/or function). The data-driven templates are undesirable and infeasible in this problem because atlas generation methods [85–87] require training data that captures a substantial amount of the variation of the dataset, which is not true for the available data. Moreover, in biology or life science, experts usually describe the neuronal morphology template qualitatively based on their observations. Here, we use such a generated prototype with 10 predetermined compartments to segment and partition single-neuron volumes (Fig. 6.1).

Our approach is motivated by the application of mapping protein interactions [164] into a 'standardized' (prototype) aCC motorneuron model, for studying causality between proteome and phenome at different development stages of the *Drosophila* embryo.

Section 6.3 introduces our method, which is based on the simultaneous registration and segmentation (SRS) principle using Markov random fields (MRFs) for fitting an artificial 10-compartment neuron prototype to input neuron volumes. Since the reference image is an artificial 'hypothetical' neuron model, the intensity in the reference is not available; therefore, the registration process is guided by the segmentation of the target volume as well as the pre-determined neuronal structure. Section 6.4 presents our global-to-local approach. We first employ the global transformation based on the neuron-specific 'standardized' coordinate system, namely the axonal centerline, to initialize the SRS process, since it can provide a unifying reference across all neurons of any type and at any stage of development. Then, we apply the local registration using MRF-based free-form deformation (FFD) to deform compartments towards their perimeters. In Section 6.5, we validate our results against an existing SRS method using aCC motorneuron image stacks from larval *Drosophila*, at multiple developmental instances and different spatial resolutions.

6.2 Related Work

One possible solution for matching artificially created neuron templates with input volumes is to perform the tasks of part-wise segmentation (e.g., multiple object geometric deformable model [76]) and registration (e.g., mutual information-based technique [105]) sequentially. However, in our context, these two tasks depend on each other and therefore coupling them into a single process improves the accuracy [22, 25, 89, 91]. SRS methods can be categorized into two types — variational-based and Bayesian-based methods.

6.2.1 Variational-Based Methods

The variational-based methods usually represent objects using level sets so the problem can be solved using the gradient descent algorithm. These methods interleave segmentation and registration iteratively, which are optimized locally using the variational calculus. They operate in the continuous space, which usually yields to getting trapped at local minima. Moreover, they only support certain functions that are differentiable with respect to the shape and transformation parameters.

The variational-based SRS was introduced by Yezzi et al. in 2001. The piecewise-constant geometric active contour model [68] and the linear transformation were used [89]. Authors segment the target image by the transformed segmentation of the reference image, optimized by the gradient descent algorithm. Derivatives are computed with respect to objects' boundary and the geometric transformation.

This method was extended to non-rigid transformation in [22, 165–167]. Introducing the non-rigid registration creates the ill-posed problem. To avoid this issue, authors in [165] penalize the segmentation's area to regularize the object's boundary curve, and employ the linear diffusion [168] to regularize the mapping vector field. While, some methods adopt the more sophisticated regularization to provide a unique solution and handle the large distance deformations. Authors in [22, 167] use the nonlinear-elasticity-based regularization based on the Saint Venant-Kirchhoff model. They assume that the segmentation of the reference image is given; and the partition of the target image is the transformation of the reference segmentation. Thus, only the vector field is needed to be optimized. The complex regularization creates the non-differentiable objective function, which is simplified by adding an auxiliary variable to replace the non-linear term and decouple the segmentation and registration tasks.

Using the segmentation and its transformation creates a tight dependency between registration and segmentation tasks. Hence, authors in [166] decouple these two tasks by introducing the auxiliary term that measures the compatibility between the two

solutions using the distance between two boundary curves in the level set space. The introduction of this new term plays a crucial role in separating the segmentation and registration tasks so they can be optimized independently in an alternate manner.

6.2.2 Bayesian-Based Methods

The second type of SRS techniques adopts the Bayesian framework [85, 87, 90–92, 169–171]. This type of method formulates the solution as a maximum a posteriori (MAP) probability estimation, using MRF or Expectation Maximization (EM) frameworks. They usually operate in the discrete space. Despite the loss of precision compared to the variational-based methods, the benefit is that they can use efficient techniques for finding a good approximation of the global optimum. Also, they can accommodate more complex functions, since they do not require the computation of derivatives.

In the early stage, the transformation and segmentation were formulated as two MAP functions and estimated alternately [90]. Authors compute a hidden Markov random vector field that assigns the probability of pixels belonging to regions, instead of the hard segmentation. While, the non-linear transformation is represented by the cubic B-spline FFD [88].

Then, the segmentation and registration tasks are formulated as a single MAP estimation. For example, authors in [87] estimates MAP using the Iterative Conditional Mode (ICM) strategy, which alternately optimizes the segmentation and registration components in the objective function. While, methods in [85, 91, 169] use the EM algorithm. The optimization alternates between the soft segmentation of magnetic resonance images in the E-step and updating mapping parameters as well as object intensity distribution in the M-step.

Later on, some methods extend the Bayesian framework assumption further by employing the graphical model, which simultaneously optimizes both segmentation and registration tasks. Authors in [92, 170, 171] assume that the segmentation of the

reference image is given; and the objective function is defined as a function of transformation, represented by the cubic B-spline FFD [88]. They then apply the MRF over the control points, where states are their displacement and object labels. To improve the performance, the novel optimization method, Fast-PD [55], was introduced. The advantage of this method is the optimality of the solution but its drawback is the limited precision based on the spacing of control points.

This problem can be alleviated using the hierarchical cubic B-spline FFD. The method in [25] is based on three factors: segmentation, registration, and coherence. Segmentation and registration are solved separately, while the coherence term ensures the consistency between these two tasks by measuring the dissimilarity between the segmentation of the atlas and the transformed segmentation of input images. The intensity distribution of compartments are modeled by Gaussian Mixture Model (GMM). The inference of MRF is computed using the graph cuts algorithm [53].

Previous SRS methods are not suitable for this problem because their similarity criteria is based solely on appearance. Since the reference image is the artificial template without image intensity and target images are acquired from multiple modalities, the appearance-based matching criteria are ineffective. We employed the graphical model because it provides a good approximated global solution and it can accommodate sophisticated objective functions, which can incorporate resolution-agnostic attributes like structure information.

6.3 Simultaneous Registration and Segmentation

Given a target image stack $\mathbf{I}\colon \Omega \to \mathbb{R}$, where Ω is the volume domain, and $L_{\mathbf{J}}$ an employed, artificially created template containing N compartments, $O_0, O_1, ...O_N$; to facilitate notation, we also consider the background as a compartment, O_0. Each compartment contains points $\mathbf{x} \in \Omega$ such that points from all compartments cover the whole image domain without overlapping, $\bigcup_{i=0}^{N} O_i = \Omega$ and $O_i \cap O_j = \emptyset$, $\forall i \neq j$. Each compartment O_i is represented implicitly [76] via a signed distance function

$\phi_i \colon \Omega \to \mathbb{R}$, $O_i = \{\mathbf{x} \mid \phi_i(\mathbf{x}) > 0\}$, $\forall \mathbf{x} \in \Omega$, while its boundary is defined by $\partial O_i = \{\mathbf{x} \mid \phi_i(\mathbf{x}) = 0\}$. Therefore, the artificial template including the background can be defined as $L_{\mathbf{J}}(\mathbf{x}) = \{i \mid \phi_i(\mathbf{x}) > 0, i = 0, ..., N\}$.

We aim at finding the geometric transformation $\mathbf{T}$ that best fits the template $L_{\mathbf{J}}$ to the target volume $\mathbf{I}$ globally and locally, namely $L_{\mathbf{I}} = \mathbf{T}(L_{\mathbf{J}}(\mathbf{x})), \forall \mathbf{x} \in \Omega$ (Fig. 6.1). We formulate the objective function as the conditional distribution of the transformation $\mathbf{T}$ over the entire domain Ω,

$$P(\mathbf{T}|\mathbf{I}, L_{\mathbf{J}}) \propto \exp\left\{ -\sum_{\mathbf{x}\in\Omega} U(\mathbf{x}) - \sum_{\mathbf{x}\in\Omega} \sum_{\mathbf{y}\in\mathcal{N}_{\mathbf{x}}} W(\mathbf{x}, \mathbf{y}) \right\} \tag{6.1}$$

where U and W are the unary (Sec. 6.3.1) and pairwise (Sec. 6.3.2) potentials respectively. $\mathcal{N}_{\mathbf{x}}$ is the set of spatial neighbors of $\mathbf{x}$. The geometric transformation solution is derived as $\widehat{\mathbf{T}} = \arg\max_{\mathbf{T}} P(\mathbf{T}|\mathbf{I}, L_{\mathbf{J}})$.

6.3.1 Unary Potential

U in eq. (6.1) registers compartments based on shape, appearance, and relative location within the neuron with respect to the axonal centerline. The shape com-

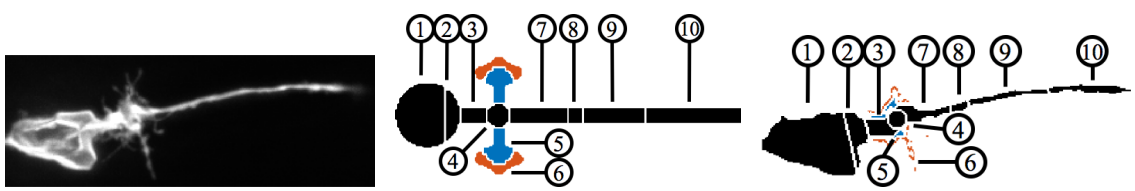


Figure 6.1. Target neuron volume $\mathbf{I}$ (collapsed, left), an artificially created template $L_{\mathbf{J}}$ with 10 compartments (middle), and the desired segmentation outcome $L_{\mathbf{I}}$ (right). Here, the target is an aCC motorneuron in the *Drosophila* embryo, at the first instar (right at the dendrite initiation), labeled genetically with plasma membrane-targeted eGFP. The (numbered) pre-determined compartments in the reference sketch are: cell body (#1), axon hillock (#2), proximal neurite (#3), dendrite base (#4), dendrite shaft (#5), dendrite tip (#6), distal neurite (#7), neurite junction (#8), nerve exit (#9), and axon (#10).

ponent ensures the smoothness of the transformed compartments by applying the regularization. Here the intensity image of the template does not exist, so the appearance component is the similarity function that measures the compatibility of the segmentation outcome $L_{\mathbf{I}}$ (the transformed template) to the target image $\mathbf{I}$. While, the structure of neurons is taken into account by matching centerlines in the target image and template.

$$U(\mathbf{x}) = U_{shp}(\mathbf{x}) + U_{img}(\mathbf{x}) + U_{cl}(\mathbf{x}) \tag{6.2}$$

Shape energy term U_{shp} imposes the shape regularization of the transformed compartments based on their local boundary smoothness and area [68],

$$U_{shp}(\mathbf{x}) = \sum_{i=1}^{N} \left\{ \alpha_i \|\nabla \phi_i(\mathbf{x})\| \partial \phi_i(\mathbf{x}) + \beta_i H\big(\phi_i(\mathbf{x})\big) \right\} \tag{6.3}$$

where α_i and β_i are the corresponding regularization parameters for compartment O_i, $\partial \phi_i$ is an indicator function of a narrow band around the compartment boundary, and H is the Heaviside function, which is defined by,

$$H(z) = \begin{cases} 1, & \text{if } z \geq 0 \\ 0, & \text{if } z < 0 \end{cases} \tag{6.4}$$

Image energy term U_{img} penalizes intensity dissimilarity between corresponding compartments of the target image $\mathbf{I}$ and a solution $L_{\mathbf{I}}$, i.e., it biases a transformation towards producing compartments with piecewise-constant intensity,

$$U_{img}(\mathbf{x}) = \sum_{i=0}^{N} \left\{ \left(\mathbf{I}(\mathbf{x}) - \mu^i_{L_{\mathbf{I}}}\right)^2 H\big(\phi_i(\mathbf{x})\big) \right\} \tag{6.5}$$

where $\mathbf{I}(\mathbf{x})$ denotes the intensity of voxel $\mathbf{x}$ in the target image and $\mu^i_{L_{\mathbf{I}}}$ denotes the average intensity of compartment i in $L_{\mathbf{I}}$,

$$\mu^i_{L_{\mathbf{I}}} = \frac{\int_{\Omega} \mathbf{I}(\mathbf{T}(\mathbf{y})) H(\mathbf{T}(\phi_i(\mathbf{y}))) d\mathbf{y}}{\int_{\Omega} H(\mathbf{T}(\phi_i(\mathbf{y}))) d\mathbf{y}} \tag{6.6}$$

Figure 6.2. Axonal centerline (in red) in the target image $\mathbf{I}$ (left) and a segmentation outcome $L_{\mathbf{I}}$ (middle), along with its corresponding unsigned distance function $\varphi_{\mathbf{T}}$ (right).

Centerline location term U_{cl} matches the axonal centerlines of the target image and the template. Though the appearance of neurons may change dramatically, its structure captured by the axonal centerlines usually exhibit slight changes. Centerlines are defined by the unsigned distance function over the volume domain instead of point distributions or splines, so the centerline location term can be easily integrated into the framework.

$$U_{cl}(\mathbf{x}) = \Big(\varphi_{\mathbf{T}}(\mathbf{x}) \exp\Big\{ - \varphi_{\mathbf{I}}(\mathbf{x}) \Big)^2 \tag{6.7}$$

where $\varphi_{\mathbf{T}}$ and $\varphi_{\mathbf{I}}$ are the unsigned distance functions of the axonal centerlines in $L_{\mathbf{I}}$ and $\mathbf{I}$ respectively (Fig. 6.2).

6.3.2 Pairwise Potential

W in eq. (6.1) enforces spatially smooth transformation; in this work, we use the diffusion regularization, which is defined as [172],

$$W(\mathbf{x}, \mathbf{y}) = \|\mathbf{d}_{\mathbf{x}} - \mathbf{d}_{\mathbf{y}}\| \tag{6.8}$$

where $\mathbf{d}_{\mathbf{x}}$ and $\mathbf{d}_{\mathbf{y}}$ denote the displacement vectors of $\mathbf{x} \in \Omega$ and $\mathbf{y} \in \mathcal{N}_{\mathbf{x}}$ respectively.

6.4 Transformations

Given that every input target image $\mathbf{I}$ is inherently in Cartesian while the employed morphology template in a 'standardized', with respect to some context-specific reference, coordinate system, the desired transformation is [88],

$$\mathbf{T} = \mathbf{T}_{local} \circ \mathbf{T}_{global} \tag{6.9}$$

where $\mathbf{T}_{global}$ transforms the template from 'standardized' to Cartesian coordinates, while $\mathbf{T}_{local}$ finds the best fitting solution $L_{\mathbf{I}}$ to the data $\mathbf{I}$. The 'standardized' coordinate system is preferred because it describes template compartments with respect to the axonal centerline and it is resolution-agnostic, unlike Cartesian coordinates that describe spatial relationships between compartments for a specific data acquisition modality.

Our method is implemented as in Algorithm 2. We assume that there exists only one neuron in data $\mathbf{I}$; for images with multiple neurons, a user-defined mask is applied to filter out neurons in the background.

Algorithm 2: Implementation of our MRF-based SRS

Input: $\mathbf{I}, L_{\mathbf{J}}$

Result: $\mathbf{T}$

Compute centerline spline S from $\mathbf{I}$; `// use shortest geodesic distance`

Compute global transformation $\mathbf{T}_{global}$

$L_{\mathbf{I}} = \mathbf{T}_{global}(L_{\mathbf{J}})$;

$\mathbf{T} = \mathbf{T}_{global}$; `// initialization as` $\mathbf{T}_{global}$

for *iter=1;* $\nabla E > \epsilon$; *iter++* **do**

  `// E = energy of MRF (see eq.` (6.1)`)`

  Update $\mu^i_{L_{\mathbf{I}}}$, $i = 0, ..., N$

  `// here we consider N=10 compartments`

  `// plus background (also see eq.` (6.5)`)`

  Solve MRF for $\mathbf{T}_{local}$ using graph cuts [53]

  $L_{\mathbf{I}} = \mathbf{T}_{local}(L_{\mathbf{I}})$;

  $\mathbf{T} = \mathbf{T}_{local} \circ \mathbf{T}$;

end

6.4.1 Global Transformation

To compute $\mathbf{T}_{global}$, the axonal centerline in the target image $\mathbf{I}$ is required: we define it as the shortest geodesic path between soma and the axon's endpoint. Specifically, the soma $\mathbf{x}_{soma}$ is detected using the Hough transform [154] in the intensity domain; alternatively, one can use, e.g., tensor voting [99]. An approximation of axon is derived using high values of *vesselness*, as computed by the Hessian-based Frangi filter [96]. The axon's endpoint $\mathbf{x}^*$ is defined as the farthest voxel from the soma along the axon's orientation $\mathbf{u}$ (unit vector), with the smallest geodesic distance,

$$\mathbf{x}^* = \arg\max_{\mathbf{x}} \left((\mathbf{x} - \mathbf{x}_{soma}) \cdot \mathbf{u} - \min_{P_{\mathbf{x}}} \int_{p \in P_{\mathbf{x}}} \frac{1}{\mathbf{I}(p)} dp \right) \tag{6.10}$$

where $P_{\mathbf{x}}$ is the shortest geodesic path, which is the axonal centerline, computed by the fast marching method [60]; alternatively, one can use neurite-specific tracing, such as [80, 81].

Given a spline S representing the axonal centerline in the input volume, any point $\mathbf{x} = (x, y, z)$ in Cartesian coordinates can be represented with respect to S and its normalized arc-length indexing parameter. Specifically, let $r(\mathbf{x})$ be the minimum Euclidean distance of $\mathbf{x}$ from S, and $\{A(\mathbf{x}), a(\mathbf{x})\}$ be the azimuth and altitude of $\mathbf{x}$ from its closest point $t(\mathbf{x}) \in S$ respectively, with respect to the spline's normal, $\mathbf{u}_\perp$, and tangential, $\mathbf{u}_\parallel$, vectors on the xy-plane at $t(\mathbf{x})$ (Fig. 6.3),

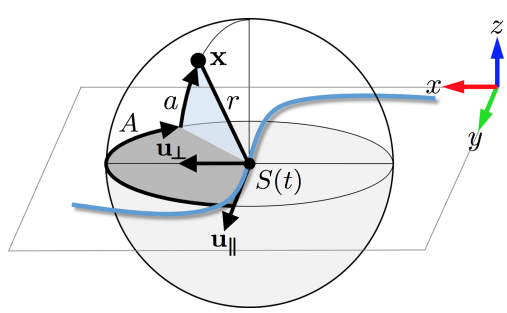


Figure 6.3. Location parameters for resolution-agnostic (standardized) representation: the axonal centerline (in blue) can be used as reference to represent the relative location of every point.

$$
\begin{aligned}
t(\mathbf{x}) &= \arg\min_{l} |\mathbf{x} - S(l)| \\
r(\mathbf{x}) &= |\mathbf{x} - S(t(\mathbf{x}))| \\
A(\mathbf{x}) &= \arctan \frac{\mathbf{u}_{\perp}(t(\mathbf{x})) \bullet [\mathbf{x} - S_{xy}(t(\mathbf{x}))]}{\mathbf{u}_{\parallel}(t(\mathbf{x})) \bullet [\mathbf{x} - S_{xy}(t(\mathbf{x}))]} \\
a(\mathbf{x}) &= \arcsin \frac{z - S_z(t(\mathbf{x}))}{r(\mathbf{x})}
\end{aligned}
\tag{6.11}
$$

where $S_{xy}(t(\mathbf{x}))$ is the projection of $S(t(\mathbf{x}))$ on the xy-plane, $S_z(t(\mathbf{x}))$ is the z-coordinate of $S(t(\mathbf{x}))$ (Fig. 6.3), and '$\bullet$' denotes inner product.

We transform the resolution-agnostic prototype from its Cartesian coordinates $\mathbf{x}$, as (Fig. 6.4),

$$
\begin{aligned}
L_{\mathbf{I}}(\mathbf{x}) &= \mathbf{T}_{global}[L_{\mathbf{J}}(\mathbf{x})] = L_{\mathbf{J}}(\tilde{\mathbf{x}}) \\
\tilde{\mathbf{x}} &= S^{\mathbf{J}}(t(\mathbf{x})) + r(\mathbf{x}) \cdot R_z(A(\mathbf{x})) \cdot R_{\mathbf{u}^{\mathbf{J}}_{\perp}(t(\mathbf{x}))}(a(\mathbf{x})) \cdot \mathbf{u}^{\mathbf{J}}_{\parallel}(t(\mathbf{x}))
\end{aligned}
\tag{6.12}
$$

where $S^{\mathbf{J}}$, $\mathbf{u}^{\mathbf{J}}_{\perp}$, and $\mathbf{u}^{\mathbf{J}}_{\parallel}$ are the axonal centerline spline, normal vectors, and tangential vectors in the prototype $L_{\mathbf{J}}$ respectively, while $R_{\mathbf{u}^{\mathbf{J}}_{\perp}}$ and R_z are the rotation matrices about the normal vector $\mathbf{u}^{\mathbf{J}}_{\perp}$ and z-axis respectively. Since any point can be

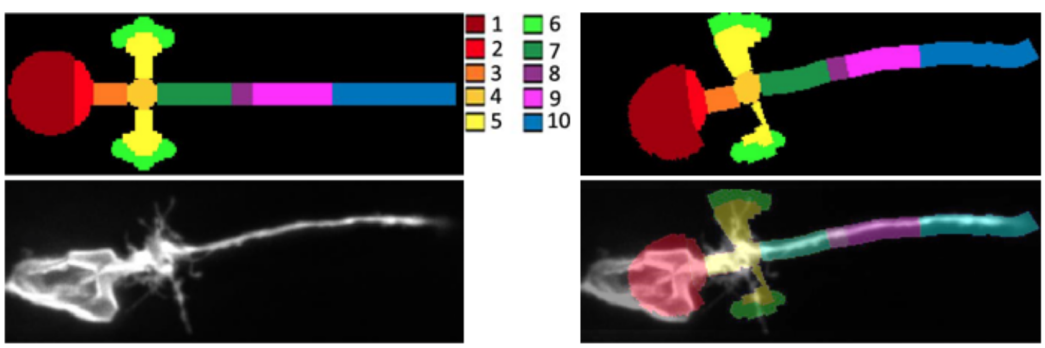


Figure 6.4. Global alignment of the artificially created morphology prototype (top left) with the target volume (bottom left); color-labels correspond to the ten pre-determined, context-specific compartments (also see Fig. 6.1). The global transformation of the prototype (top right) provides a good initialization for the entire volume and each compartment individually (bottom right: transformed prototype superimposed on the target).

represented by either Cartesian coordinates or with respect to the reference axonal centerline, the distance functions ϕ_i, $i = 0, ..., N$, in the original volume domain can be transformed based on the axon reference, and *vice versa* (Fig. 6.4).

6.4.2 Local Transformation

$\mathbf{T}_{local}$ can be formulated using a MRF, based on [173]; however, this framework requires costly computations due to a large number of parameters. We can reduce the number of parameters by using the uniform cubic B-spline FFD. Specifically, let $\mathbf{T}_{local}(\mathbf{x})$ be the 3D FFD at voxel location $\mathbf{x}$, and $\mathbf{C}$ be the set of the grid control points. Then, the transformation at $\mathbf{x}$ is defined as,

$$\mathbf{T}_{local}(\mathbf{x}) = \sum_{\mathbf{c}\in\mathbf{C}} \eta(\mathbf{x}, \mathbf{c})\mathbf{d}_{\mathbf{c}} \tag{6.13}$$

where η is the coefficient of control point $\mathbf{c}$ at location $\mathbf{x}$ with displacement $\mathbf{d}_{\mathbf{c}}$. The MRF is formulated over FFD, thus the objective function in eq. (6.1) is redefined over the set of grid control points $\mathbf{C}$,

$$\begin{aligned} P(\mathbf{T}|\mathbf{I}, L_{\mathbf{J}}) &\propto \exp\left\{ -\sum_{\mathbf{c}\in\mathbf{C}} U(\mathbf{c}) - \sum_{\mathbf{c}\in\mathbf{C}} \sum_{\mathbf{b}\in\mathcal{N}_{\mathbf{c}}} W(\mathbf{c}, \mathbf{b}) \right\} \\ U(\mathbf{c}) &= \iint_{\mathbf{x}\in\Omega} \hat{\eta}(\mathbf{x}, \mathbf{c}_i) U(\mathbf{x}) d\mathbf{x} \\ W(\mathbf{c}, \mathbf{b}) &= \|\mathbf{d}_{\mathbf{c}} - \mathbf{d}_{\mathbf{b}}\| \end{aligned} \tag{6.14}$$

where $\mathcal{N}_{\mathbf{c}}$ is the set of neighbors of $\mathbf{c}$ in the grid and $\hat{\eta}(\mathbf{x}, \mathbf{c})$ is the influence of voxel $\mathbf{x}$ to the control point $\mathbf{c}$,

$$\hat{\eta}(\mathbf{x}, \mathbf{c}) = \frac{\eta(\mathbf{x}, \mathbf{c})}{\int_{\mathbf{y}\in\Omega} \eta(\mathbf{y}, \mathbf{c})} \tag{6.15}$$

To guarantee diffeomorphism of the transformation, we limit the maximum displacement to less than half of the control point spacing [174,175] and adopt the FFDs in [176] to compensate for small deformations.

6.5 Experiments

We validate our method using 45 aCC motorneuron image stacks from *Drosophila* larvae[1], wild-types and mutants, in two different resolutions, and at different development instances. We evaluate our results using four metrics: Dice index, Hausdorff distance, mean distance, and mutual information. The Dice index is $\mathcal{D}(\mathbf{s}, G) = 2|\mathbf{s} \cap G|/(|\mathbf{s}| + |G|)$ [177], where $\mathbf{s}$ is a segmentation outcome and G denotes the ground truth ($\mathcal{D} \in [0, 1]$). The Hausdorff distance is the maximum distance from a set of points to the nearest point in another set. It measures the proximity of outliers between two compartments, which is defined by,

$$\begin{aligned} HD(\mathbf{s}, G) &= \max\{hd(\mathbf{s}, G), hd(G, \mathbf{s})\} \\ hd(\mathbf{s}, G) &= \max_{\mathbf{x}\in\mathbf{s}} \min_{\mathbf{y}\in G} |\mathbf{x} - \mathbf{y}| \end{aligned} \tag{6.16}$$

Unlike the Hausdorff distance, the mean distance is the average distance from a set of points to the nearest point in another set. It measures the average proximity between two compartments,

$$MD(\mathbf{s}, G) = \frac{1}{2|\mathbf{s}|} \sum_{\mathbf{x}\in\mathbf{s}} \min_{\mathbf{y}\in G} |\mathbf{x} - \mathbf{y}| + \frac{1}{2|G|} \sum_{\mathbf{y}\in G} \min_{\mathbf{x}\in\mathbf{s}} |\mathbf{y} - \mathbf{x}| \tag{6.17}$$

The last metric is the mutual information, which measures the similarity between compartments, $MI(G, \mathbf{s}) = H(G) + H(\mathbf{s}) - H(G, \mathbf{s})$, as described in Section 3.6.

For the multi-class problem, we evaluate the segmentation of each compartment separately, and calculate the average index over all compartments. In this work, our artificially created template contains 10 compartments. The ground truth was generated by manual part-wise segmentation. We used the FFD with adaptive spacing between 5 and 80 voxels. The largest image stack we used was $1024 \times 1024 \times 10$, so the maximum number of control points was $208 \times 208 \times 5$.

We evaluate the segmentation of: (a) the entire volume, what we denote as **binary** (neuron/background); (b) soma, axon, and dendrite, **SAD** (three compartments plus

[1]The samples were labeled genetically with plasma membrane-targeted eGFP, and imaged with a conventional fluorescent microscope with $250 \times 250 \times 500nm$ spatial resolution.

background), where we merged compartments #1 and #2 for the soma, #5 and #6 for the dendrite, and the rest for the axon (Fig. 6.1); and (c) all ten-compartments, denoted as **All**. Different segmentation scenarios emphasize the difficulty of this problem, as well as the effectiveness of our method.

To show the significance of describing the volume topology with respect to the axonal centerline, we tested our method against two scenarios (Table 6.1). In the first scenario we replaced our global transformation with a rigid transformation computed by the finite iterative closest point approach [115]. In the second scenario, we removed the centerline term U_{cl} of eq. (6.7) from the energy function. Finally, we also compare our method against sequential segmentation and registration, as well as existing SRS framework in [25] (Table 6.1).

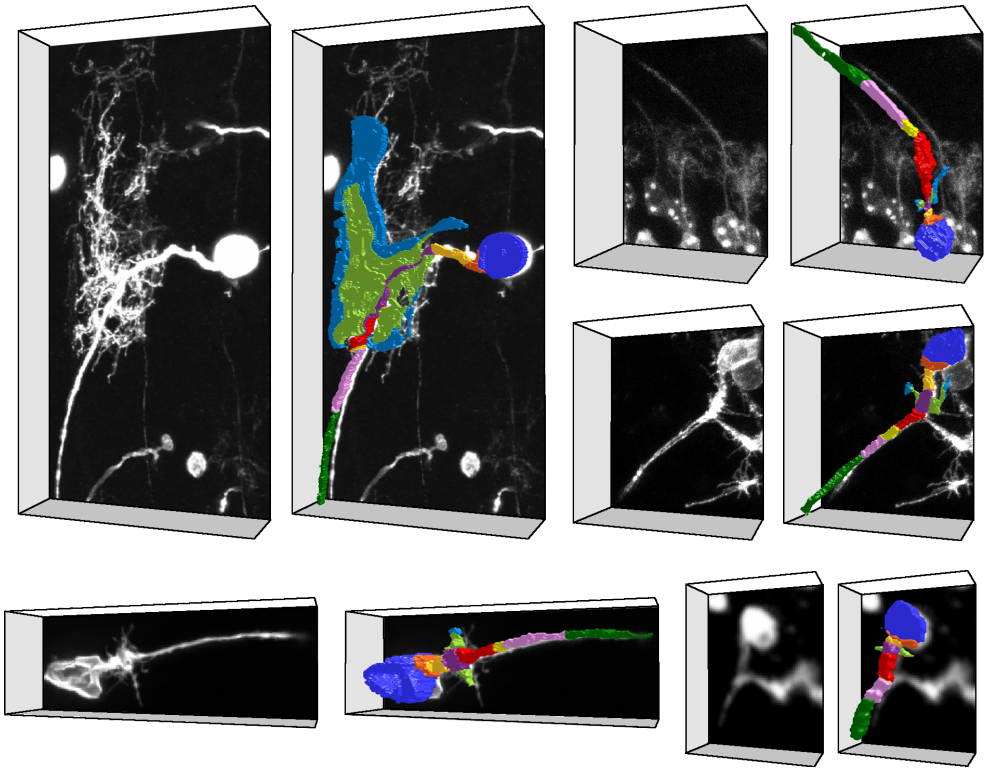


Figure 6.5. Results for three aCC neurons. For each volume, the maximal projection is displayed along with the multiple compartment (in different colors) segmentation.

Table 6.1.
Mean Dice index of our method, existing SRS method [25], and three baselines: (i) '$\mathbf{T}_{rigid}$' replaces $\mathbf{T}_{global}$ with a rigid transformation, (ii) 'No U_{cl}' ignores the centerline location term from the objective function, and (iii) 'Sequential' solves segmentation and registration sequentially. We compute the mean Dice for three segmentation tasks: binary, SAD, and all ten-compartments (see text).

	Ours	$\mathbf{T}_{rigid}$	No U_{cl}	Sequential	SRS [25]
Binary	**0.6869**	0.6657	0.6496	0.5789	0.4623
SAD	**0.5034**	0.4881	0.4673	0.4113	0.3025
All	**0.2748**	0.2650	0.2515	0.2103	0.1537

Table 6.2.
Mutual information of our method and three baselines, as well as the competition [25]. We compute the average mutual information for binary, SAD, and all ten-compartments segmentations (see text).

	Ours	$\mathbf{T}_{rigid}$	No U_{cl}	Sequential	SRS [25]
Binary	**0.1394**	0.1329	0.1265	0.1116	0.0726
SAD	**0.0608**	0.0565	0.0548	0.0518	NaN
All	**0.0152**	0.0136	0.0134	0.0131	NaN

Table 6.3.
Mean Hausdorff distance (in voxels) of our method and three baselines, as well as the competition [25]. We compute the mean Hausdorff distance for binary, SAD, and all ten-compartments segmentations (see text).

	Ours	$\mathbf{T}_{rigid}$	No U_{cl}	Sequential	SRS [25]
Binary	**41.541**	43.107	56.909	53.765	66.776
SAD	**56.475**	56.865	58.979	60.973	∞
All	**53.809**	54.270	54.980	58.005	∞

Table 6.4.
Average mean distance (in voxels) of our method and three baselines, as well as the competition [25]. We compute the average mean distance for binary, SAD, and all ten-compartments segmentations (see text).

	Ours	$\mathbf{T}_{rigid}$	No U_{cl}	Sequential	SRS [25]
Binary	**2.492**	2.689	4.052	4.192	8.187
SAD	**6.493**	6.544	7.448	7.612	∞
All	**13.530**	13.767	14.668	16.3521	∞

Our results in Tables 6.1 and 6.2 indicate improved segmentation performance, which is mainly due to considering structure information in both $\mathbf{T}_{global}$ and $\mathbf{T}_{local}$ (Fig. 6.5). The global transformation exploits the neuronal morphology directly by describing the volume with respect to the axonal centerline. While local transformation integrates the centerline location term into the objective function so that the MRF framework takes into account both compartment shapes and volume structure. Removing any of these components, as shown in the first two scenarios ('$\mathbf{T}_{rigid}$' and 'No U_{cl}' in Table 6.1), reduces accuracy.

Our method also improves the quality of results in term of distance to ground truth as shown in Tables 6.3 and 6.4. The state-of-the-art SRS framework in [25] performs poorly on our dataset because of the drastic changes in appearance, the sole attribute on which this method relies. It fails to detect some compartments so its mutual information score is not-a-number (NaN) and its distance scores are infinity for SAD and all ten-compartments segmentation tasks.

The main source of inaccuracies, shown in figure 6.6, is the numerically 'ambiguous' compartment annotation in the artificially created prototype. Specifically, our method does not consider inter-compartment relationships (e.g., adjacency) or relative locations within the *Drosophila* Ventral Nerve Cord (VNC). Some of the compartments are defined based on the neuron location in its natural environment, and therefore intensity and/or shape are not sufficient for their segmentation: e.g., see

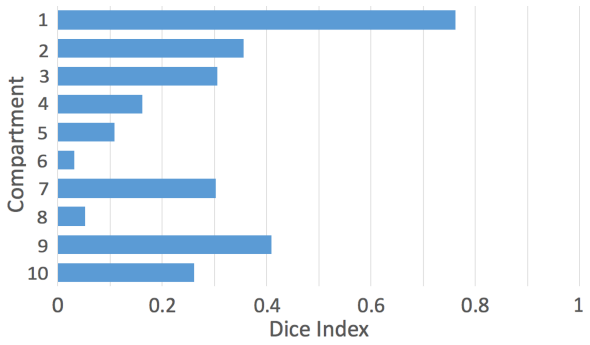


Figure 6.6. Average Dice index by compartment of our results.

compartment #2 in relation to soma, as well as compartments #3–10 along the axon whose context is determined by the surroundings in the animal (Figs. 6.1 and 6.4). Also, due to spatial resolution limitations, the dendritic compartments #5 and #6 are difficult to distinguish.

6.6 Conclusion

We presented the novel SRS method following the global-to-local approach for matching artificially created, multi-compartment neuron morphology templates with target volumes. Our method is the first to use totally artificial templates drawn by experts. This template acts as the standardized representation of neurons allowing the integration of information across imaging modalities, and the meaningful comparison between developmental stages and sample types. We employed the global transformation to align the template/prototype with the data, and the local transformation based on the MRF realization of FFDs. Both transformations use the axonal centerline as a reference, which increases accuracy in compartment-wise segmentation. We validate our results using aCC motorneurons in the *Drosophila* embryo.

7 SUMMARY

In this dissertation, we study the morphology analysis at the single-cell resolution *in vivo*. A number of model-and-appearance-based methods are developed to solve four fundamental tasks essential for morphology analysis. In Section 7.1, we highlight the advantages of our methods and our contributions in computer vision on these four tasks, i.e., segmentation, tracing, tracking, and part-wise segmentation with artificial templates. These tasks solve the morphology quantification problem in different scenarios from static images to image sequences, as well as multi-modal microscopy images. Then, our fully automated morphology quantification methods are employed in neuroscience studies to illustrate the significance of our contributions in biology. Section 7.2 discusses three example applications of our methods that assist in advancing the field of neuroscience: neuron recognition and retrieval, visualization of neuronal activity, and protein-protein interaction visualization. The availability of these applications provide powerful tools for computationally quantifying the morphological attributes of interest, comprehensively visualizing specific characteristics of neurons, and conducting sophisticated statistical analysis. Furthermore, our automated morphology analysis helps create the standard to remove the bias among human operators, and reduce the human intervention time.

7.1 Research Contributions

Our contributions are in both computer science and biology disciplines. In computer vision, we develop computational methods for four main tasks in the automated morphology analysis: segmentation, tracing, tracking, and part-wise segmentation with artificial templates. Our model-and-appearance-based methods solve these tasks by incorporating the appearance information with the sophisticated models that en-

code both local shape and global structure of neurons. These methods have applications in neuroscience as they enable new techniques for quantifying the neuronal morphology in different scenarios and new visualizations for relationships between properties of neurons. Furthermore, we also develop the new 'bottom-up' analysis technique that analyzes the influence of the anatomical aspect of neurons at the single-cell resolution. This allows neuroscientists to investigate a nervous system at the most fundamental level.

In Chapter 3, we describe our novel segmentation method, the boundary-based neuron reconstruction that extracts neurons as binary images. Most existing neuron reconstruction techniques require some kind of segmentation for the initialization process. Poor performance in segmentation resulting in failure to detect neurons can undermine the whole process. To improve the initialization, our method exploits the time-lapse data volumes to improve the segmentation of an individual image stack. We integrate the structural information across the sequence through the co-segmentation principle in the seamless manner. The integration of neuron's structure alleviates out-of-focus effects, and helps resolve ambiguous regions such as bifurcation points.

Chapter 4 presents two novel neurite tracing methods for image stacks with spatially varying noise. Neuron tracing is the centerline-based neuron reconstruction that depicts the neuronal topology as a tree data structure, unlike binary image that outlines the boundary. The tree representation of neurons is suitable for statistical analysis of neuronal morphology [1]. Our two methods are based on the population of open-curve snakes, that evolves simultaneously. The evolution of snakes is driven by local data features, shape smoothness, and pairwise interaction with nearby snakes. The cooperation among snakes helps resolve tracing over regions with local intensity ambiguities.

The neuron reconstruction of the single static image stacks is unable to capture the dynamics of nervous systems. Chapter 5 introduces our novel tracking method for modeling the morphology dynamics over noisy calcium image sequence, which is

sensitive to neuronal activities. Our MRF-based neurite tracking method follows the local-to-global approach designed for modeling the locomotive behavior of *Drosophila* larvae. We track neurons using the local features, which are constrained by our generalized pictorial structure. The constraint helps handle severe deformations and local intensity ambiguities, especially among inactive neurons that appear invisible in calcium images.

Finally, the morphology acquired by confocal laser scanning microscopy is not the only attribute that influences the neuron mechanisms. There are other influential attributes such as neuronal functionality that can be obtained by other imaging system like FRET microscopy. Therefore, the analysis must be conducted across multi-modality imaging to obtain the whole picture of how nervous systems operate. This task can be formulated as the part-wise segmentation problem with artificial templates. Chapter 6 presents the novel simultaneous registration and segmentation technique for solving this problem in the global-to-local manner. Our method is the first to use the generated 'hypothetical' prototype encoding the domain knowledge, which provides the biologically comprehensible comparison across modalities. The global transformation aligns parts of neurons at the morphology level, while the local transformation deforms parts towards their boundary. The advantage of incorporating the structure (i.e., axonal centerline) is that the morphology changes slightly across modalities compared to the appearance.

7.2 Applications

Our modal-and-appearance-based methods are employed to assist neuroscientists on the three following studies: modeling the structure and dynamics of neuronal circuits, finding the patterns of the morphology dynamics, and modeling the connection between structural and functional development. Specifically, we apply our morphology analysis techniques on three applications essential for the studies men-

tioned above: neuron recognition and retrieval, visualization of neuronal activity, and protein-protein interaction visualization.

1. **Neuron recognition and retrieval.** Neuron subtypes can be used for patterning the evolution of nervous systems because they influence the synaptic connectivity in motoneuron circuits. This property combining with the fact that neuron subtypes can be distinguished based on their morphology [31] makes neuron subtypes an important component for modeling the structure and dynamics of neuronal circuits. Existing works in [30–33] proposed the novel neuron subtype classifiers based on the variation of the CRF over the morphology features, which could be computed using our tracing methods. Our tracing software (available at `http://neurovision.cs.iupui.edu`) is able to extract the morphology and compute the global features as well as dendritic arborization features, such as branching point locations, branch lengths, principal directions of branches, etc. These features encode the branching details that could be used to improve the classification of neuron subtypes (Fig. 7.1).

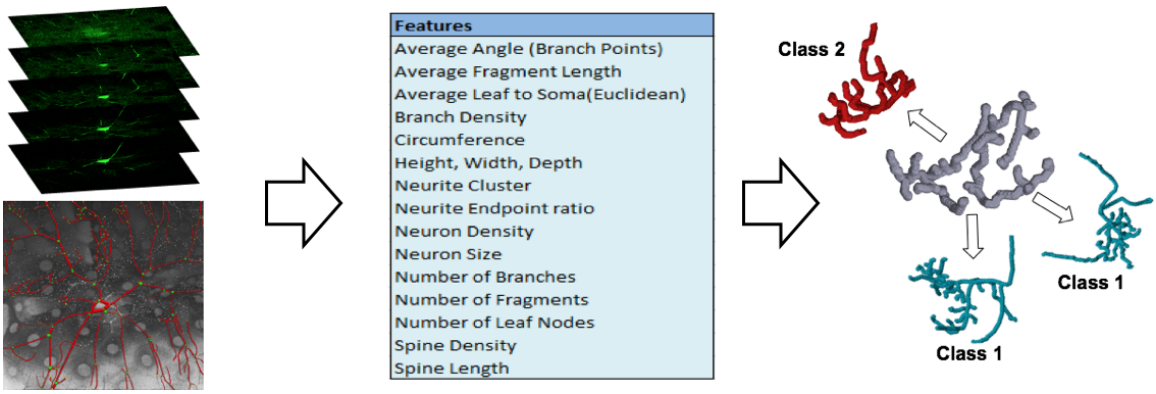


Figure 7.1. Neuron recognition process. In the first step, the traces of neurons are extracted from input image stacks using our method. Second, various morphological attributes are quantified through our application. Last, supervised learning techniques like hidden CRF [31] could be applied on these features to categorize neuron subtypes.

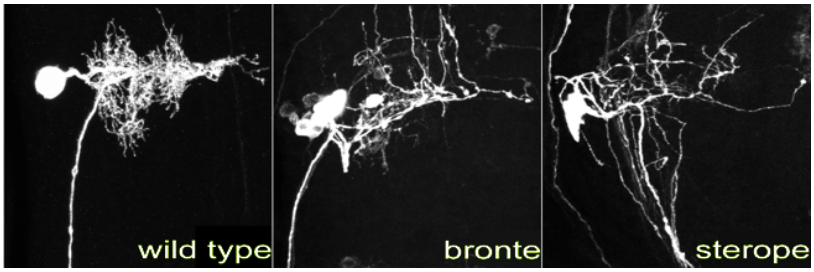


Figure 7.2. Two mutations of motor neurons. Dendritic morphology of motor neuron (left). Bronte mutants show reduced dendritic branching and dendrite guidance defects (middle), while Sterope mutants only show reduced branching (right).

In addition, the neuron reconstruction also has a use in the categorization of single neuron morphology into either wild types or mutants for the study of neuron degeneration. The introduction of mutation usually causes the defect and reduction of dendritic branching (Fig. 7.2). Tracking the change of neuronal morphology over the development and comparing between wild types and mutants would help identify genes that cause the specific motor neuron degenerations.

Neuron recognition also has an application in morphological retrieval, which is an effective way to navigate through the databases of neurons. The method in [178] encodes the traces with the novel binary coding. Then, it applies the hash function on binary codes to index and efficiently retrieve neurons with similar topology [178]. The morphology quantification produced by our framework could improve the encoding process by providing the interpretable hash codes.

2. **Visualization of neuronal activity.** To study the relationship between neuronal activity and morphology dynamics, the new visualization technique is required for exploring the structure of neurons and signal strength of calcium responses altogether in an intuitive manner. To examine how dendritic deformation relates to neuronal signals, we track proprioceptors, i.e., ddaD and ddaE

neurons, in the calcium image sequence during the movement of *Drosphila* embryo. The recording in Fig. 7.3 shows that the morphology deformations of ddaD and ddaE are distinct, as well as the calcium responses. Although the changes in calcium responses can be collected easily, understanding how the responses related to the phases of the muscle contraction cycle is not trivial. The comparison cannot be simply carried out against time because each of the samples moves differently and the samples also vary in size.

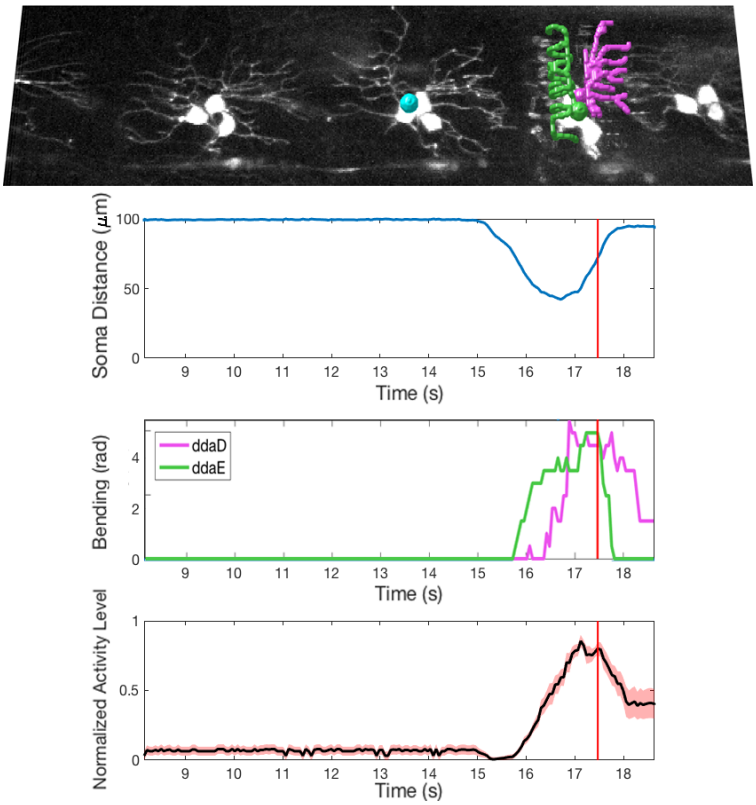


Figure 7.3. Neuron activity visualization over a contraction cycle. From top to bottom, the calcium image overlaid by two reconstructed neurons of ddaD (magenta) and ddaE (green), soma distance, bending parameter, and normalized neuronal activity level of the ddaE neuron. Red lines in the plots indicate the time instance of the image. Soma distance measures the distance between soma (green and cyan spheres). Bending parameter measures the fold angle of dendrites. It shows the difference in morphology changes of two neurons. The normalized neuronal activity level is plotted along with shaded error bar.

To overcome this issue, we use the position of the soma in adjacent segments as fiduciary landmarks to indicate the phase of the segmental contraction cycle. In a cycle, a muscle pull adjacent neurons closer together followed by the relaxation, which put the adjacent neurons back to their original distance. To measure the distance, neurons are tracked in GFP image sequence as well. In our novel scheme (Fig. 7.3), the neuronal activity and morphology deformation are measured by the strength of calcium responses and the bending parameters of traces respectively. The segmental contraction cycle is used as the reference for finding correspondences because the locomotion occurs in cycle with the same pattern. Thus, this representation is reproducible and consistent. This new visualization technique displays the quantified neuronal activity and morphology dynamics to assist neuroscientists in providing statistical analysis and detecting patterns of changes between neuronal activity and morphology dynamics.

3. **Protein-protein interaction visualization.** Another possible application of our methods is the single-neuron computational analysis platform for advancing molecular and cellular biology of neurons. The aim is to investigate the link between the nanometer-scale protein network and the micrometer-scale cellular complexity. This task involves processing hundreds of multi-modal microscopic images of same-type neurons in *Drosophila*, namely aCC neurons, to recognize the normal development patterns of morphology and observe the network of co-expressed proteins. Therefore, this platform must facilitate multi-scale analysis. The relationship between these two vastly different scales can be quantified automatically using our method; however, the new visualization technique is required to provides an intuitive navigation for exploring and modeling the correlation between neuron growth and protein interactions.

   Our part-wise neuron segmentation method with artificial templates divides the neuron volume according to the prototype aCC and provides the mapping

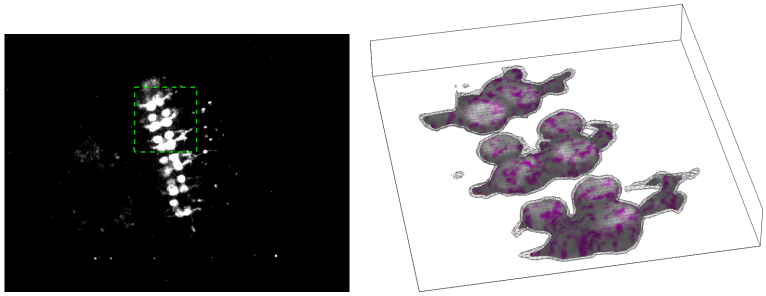


Figure 7.4. Visualization of the protein-protein interaction with respect to the topology. The magnified region (green box) of the image stack (left) is combined with the protein concentration map through the prototype aCC to provides the 3D visualization (right). The color encodes the level of protein interaction, where purple indicates high concentration and gray indicates low level. The wire-frames show the boundary of neurons.

function. Then, a two-way mapping between protein interactions and an aCC volume can be computed using the prototype as the reference.

Given the registration across multiple modalities, the connection between the protein interactions and the neuronal morphology can be quantified and described in the same coordinate system. As a result, the comparison becomes a trivial task and the connection between phenome and proteome can be established. Then, we render these two quantities together (Fig. 7.4). We could apply machine learning models to examine causality between every interacting pair of co-expressed proteins and local morphology complexity.

The core idea in this dissertation is to introduce the automation in neuroscience field using technological developments and research in computer vision. This idea would enable the rapid advance and might revolutionize how research is conducted in this field. At the same time, neuroscience problems also give rise to new kinds of problems extending the horizon of computer vision research. We believe that our work contributes to the invention of the automated morphology analysis framework for multiple microscopy imaging modalities, and in consequence takes us one step closer to understand how a brain works.

REFERENCES

REFERENCES

[1] E. Meijering. Neuron tracing in perspective. *Cytometry Part A*, 77A(7):693–704, 2010.

[2] E. Niedermeyer and F. H. L. da Silva. *Electroencephalography: Basic Principles, Clinical Applications, and Related Fields*. Lippincott Williams & Wilkins, 2005.

[3] P. A. Rinck. *Magnetic Resonance in Medicine: The Basic Textbook of the European Magnetic Resonance Forum*. Wiley-Blackwell, 2001.

[4] X. Chang, M. D. Kim, R. Stephens, T. Qu, A. Chiba, and G. Tsechpenakis. Part-based motor neuron recognition in the Drosophila ventral nerve cord. *NeuroImage*, 90:33–42, 2014.

[5] G. Tsechpenakis, R. E. Gamage, M. D. Kim, and A. Chiba. Motor neuron morphology estimation for its classification in the Drosophila brain. In *IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 7755–7758, 2011.

[6] G. A. Ascoli, D. E. Donohue, and M. Halavi. Neuromorpho.org: A central resource for neuronal morphologies. *J. Neurosci.*, 27(35):9247–9251, 2007.

[7] A. Chatr-aryamontri, B.-J. Breitkreutz, R. Oughtred, L. Boucher, S. Heinicke, D. Chen, C. Stark, A. Breitkreutz, N. Kolas, L. O'Donnell, T. Reguly, J. Nixon, L. Ramage, A. Winter, A. Sellam, C. Chang, J. Hirschman, C. Theesfeld, J. Rust, M. S. Livstone, K. Dolinski, and M. Tyers. The biogrid interaction database: 2015 update. *Nucleic Acids Research*, 43(D1):D470–D478, 2015.

[8] B. Pakkenberg, D. Pelvig, L. Marner, M. J. Bundgaard, H. J. G. Gundersen, J. R. Nyengaard, and L. Regeur. Aging and the human neocortex. *Experimental Gerontology*, 38(1):95–99, 2003.

[9] F. A. C. Azevedo, L. R. B. Carvalho, L. T. Grinberg, J. M. Farfel, R. E. L. Ferretti, R. E. P. Leite, W. J. Filho, R. Lent, and S. Herculano-Houzel. Equal numbers of neuronal and nonneuronal cells make the human brain an isometrically scaled-up primate brain. *J. Comp. Neurol.*, 513(5):532–541, 2009.

[10] J. G. White, E. Southgate, J. N. Thomson, and S. Brenner. The structure of the ventral nerve cord of *Caenorhabditis elegans*. *Phil. Trans. R. Soc. B: Biol. Sci.*, 275(938):327–348, 1976.

[11] K. Eichler, F. Li, A. Litwin-Kumar, Y. Park, I. Andrade, C. M. Schneider-Mizell, T. Saumweber, A. Huser, C. Eschbach, B. Gerber, R. D. Fetter, J. W. Truman, C. E. Priebe, L. F. Abbott, A. S. Thum, M. Zlatic, and A. Cardona. The complete connectome of a learning and memory centre in an insect brain. *Nature*, 548:175–182, 2017.

[12] H. Lagercrantz, M. A. Hanson, L. R. Ment, and D. M. Peebles. *The Newborn Brain: Neuroscience and Clinical Applications*. Cambridge University Press, 2010.

[13] G. Roth and U. Dicke. Evolution of the brain and intelligence. *Trends in Cognitive Sciences*, 9(5):250–257, 2005.

[14] G. A. Ascoli. *Computational Neuroanatomy: Principles and Methods*. Humana Press, 2002.

[15] R. Scorcioni, S. Polavaram, and G. A. Ascoli. L-measure: A web-accessible tool for the analysis, comparison and search of digital reconstructions of neuronal morphologies. *Nat. Protocols*, 3(5):866–876, 2008.

[16] D. A. Sholl. Dendritic organization in the neurons of the visual and motor cortices of the cat. *J. Anat.*, 87(Pt 4):387–406.1, 1953.

[17] L. Costa, K. Zawadzki, M. Miazaki, M. Viana, and S. Taraskin. Unveiling the neuromorphological space. *Frontiers in Comput. Neurosci.*, 4:150, 2010.

[18] L. McGarry, A. Packer, E. Fino, V. Nikolenko, T. Sippy, and R. Yuste. Quantitative classification of somatostatin-positive neocortical interneurons identifies three interneuron subtypes. *Frontiers in Neural Circuits*, 4:12, 2010.

[19] D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. *ICCV*, pages 269–276, 2009.

[20] P. Chothani, V. Mehta, and A. Stepanyants. Automated tracing of neurites from light microscopy stacks of images. *Neuroinformatics*, 9(2-3):263–278, 2011.

[21] Q. Chen and V. Koltun. Full Flow: Optical flow estimation by global optimization over regular grids. *CVPR*, pages 4706–4714, 2016.

[22] C. Le Guyader and L. A. Vese. *A Combined Segmentation and Registration Framework with a Nonlinear Elasticity Smoother*, pages 600–611. Springer, 2009.

[23] Y. Wang, A. Narayanaswamy, C.-L. Tsai, and B. Roysam. A broadly applicable 3-D neuron tracing method based on open-curve snake. *Neuroinformatics*, 9(2-3):193–217, 2011.

[24] P. Glowacki, M. A. Pinheiro, E. Türetken, R. Sznitman, D. Lebrecht, J. Kybic, A. Holtmaat, and P. Fua. Reconstructing evolving tree structures in time lapse sequences. *CVPR*, pages 3035–3042, 2014.

[25] T. Gass, G. Székely, and O. Goksel. Simultaneous segmentation and multiresolution nonrigid atlas registration. *IEEE Trans. Image Process.*, 23(7):2931–2943, 2014.

[26] D. Purves, D. Fitzpatrick, L. C. Katz, A. S. Lamantia, J. O. McNamara, S. M. Williams, and G. J. Augustine. *Neuroscience*. Sinauer Associates, 2001.

[27] R. A. Bradshaw and E. A. Dennis. *Handbook of Cell Signaling*. Elsevier/Academic Press, 2010.

[28] K. M. Brown, G. Barrionuevo, A. J. Canty, V. De Paola, J. A. Hirsch, G. S. X. E. Jefferis, J. Lu, M. Snippe, I. Sugihara, and G. A. Ascoli. The DIADEM data sets: Representative light microscopy images of neuronal morphology to advance automation of digital reconstructions. *Neuroinformatics*, 9(2-3):143–157, 2011.

[29] H. Peng, E. Meijering, and G. A. Ascoli. From DIADEM to BigNeuron. *Neuroinformatics*, 13(3):259–260, 2015.

[30] X. Chang, M. D. Kim, A. Chiba, and G. Tsechpenakis. Patterning motor neurons in the Drosophila ventral nerve cord using latent state conditional random fields. *ISBI*, pages 864–867, 2012.

[31] X. Chang, M. D. Kim, R. Stephens, T. Qu, S. Gulyanon, A. Chiba, and G. Tsechpenakis. Neuron recognition with hidden neural network random fields. *ISBI*, pages 266–269, 2014.

[32] X. Chang, M. D. Kim, A. Chiba, and G. Tsechpenakis. Motor neuron recognition in the Drosophila ventral nerve cord. *ISBI*, pages 1488–1491, 2013.

[33] G. Tsechpenakis, P. Mukherjee, M. D. Kim, and A. Chiba. Three-dimensional motor neuron morphology estimation in the Drosophila ventral nerve cord. *IEEE Transactions on Biomedical Engineering (BME)*, 59(5):1253–1263, 2012.

[34] G.-S. Yi, J. Wang, B. Deng, and X.-L. Wei. Morphology controls how hippocampal CA1 pyramidal neuron responds to uniform electric fields: A biophysical modeling study. *Scientific Reports*, 7(1):3210, 2017.

[35] H. K. Inagaki, Y. Jung, E. D. Hoopfer, A. M. Wong, N. Mishra, J. Y. Lin, R. Y. Tsien, and D. J. Anderson. Optogenetic control of Drosophila using a red-shifted channelrhodopsin reveals experience-dependent influences on courtship. *Nature Methods*, 11(3):325–332, 2014.

[36] T.-W. Chen, T. J. Wardill, Y. Sun, S. R. Pulver, S. L. Renninger, A. Baohan, E. R. Schreiter, R. A. Kerr, M. B. Orger, V. Jayaraman, L. L. Looger, K. Svoboda, and D. S. Kim. Ultrasensitive fluorescent proteins for imaging neuronal activity. *Nature*, 499(7458):295–300, 2013.

[37] C. M. Schneider-Mizell, S. Gerhard, M. Longair, T. Kazimiers, F. Li, M. F. Zwart, A. Champion, F. M. Midgley, R. D. Fetter, S. Saalfeld, and A. Cardona. Quantitative neuroanatomy for connectomics in Drosophila. *eLife*, 5:e12059, 2016.

[38] H. Kohsaka, S. Okusawa, Y. Itakura, A. Fushiki, and A. Nose. Development of larval motor circuits in Drosophila. *Develop. Growth Differ.*, 54(3):408–419, 2012.

[39] X. Yuan, J. T. Trachtenberg, S. M. Potter, and B. Roysam. MDL constrained 3-D grayscale skeletonization algorithm for automated extraction of dendrites and spines from fluorescence confocal images. *Neuroinformatics*, 7(4):213–232, 2009.

[40] K. Deng, J. Tian, J. Zheng, X. Zhang, X. Dai, and M. Xu. Retinal fundus image registration via vascular structure graph matching. *International Journal of Biomedical Imaging (IJBI)*, 2010:14:1–14:13, 2010.

[41] D. Smeets, P. Bruyninckx, J. Keustermans, D. Vandermeulen, and P. Suetens. Robust matching of 3D lung vessel trees. *MICCAI workshop on pulmonary image analysis*, 2:61–70, 2010.

[42] E. Serradell, P. Glowacki, J. Kybic, F. Moreno-Noguer, and P. Fua. Robust non-rigid registration of 2D and 3D graphs. In *CVPR*, pages 996–1003, 2012.

[43] E. Serradell, M. A. Pinheiro, R. Sznitman, J. Kybic, F. Moreno-Noguer, and P. Fua. Non-rigid graph registration using active testing search. *TPAMI*, 37(3):625–638, 2015.

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

[45] J.-C. Yoo and T. H. Han. Fast normalized cross-correlation. *Circuits, Systems and Signal Processing*, 28(6):819, 2009.

[46] S. Vicente, C. Rother, and V. Kolmogorov. Object cosegmentation. *CVPR*, pages 2217–2224, 2011.

[47] H. Zhu, J. Lu, J. Cai, J. Zheng, and N. M. Thalmann. Multiple foreground recognition and cosegmentation: An object-oriented CRF model with robust higher-order potentials. In *IEEE Winter Conference on Applications of Computer Vision (WACV)*, pages 485–492, 2014.

[48] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, 2006.

[49] J. M. Hammersley and P. Clifford. Markov field on finite graphs and lattices. Unpublished manuscript, 1971.

[50] S. Kumar and M. Hebert. Discriminative random fields. *International Journal of Computer Vision (IJCV)*, 68(2):179–201, 2006.

[51] J. Besag. On the statistical analysis of dirty pictures. *J. Appl. Stat.*, 48(3):259–279, 1986.

[52] M. J. Swain. Parameter learning for markov random fields with highest confidence first estimation. Technical report, University of Rochester (NY US), 1990.

[53] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *TPAMI*, 23:2001, 2001.

[54] V. Lempitsky, C. Rother, S. Roth, and A. Blake. Fusion moves for markov random field optimization. *TPAMI*, 32(8):1392–1405, 2010.

[55] N. Komodakis, G. Tziritas, and N. Paragios. Performance vs computational efficiency for optimizing single and dynamic MRFs: Setting the state of the art with primal-dual strategies. *Computer Vision and Image Understanding (CVIU)*, 112(1):14–29, 2008.

[56] C. Sutton and A. McCallum. An introduction to conditional random fields. *Found. Trends Mach. Learn.*, 4(4):267–373, 2012.

[57] J. D. Lafferty, A. McCallum, and F. C. N. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. *International Conference on Machine Learning (ICML)*, pages 282–289, 2001.

[58] T. Zhao, J. Xie, F. Amat, N. G. Clack, P. Ahammad, H. Peng, F. Long, and E. W. Myers. Automated reconstruction of neuronal morphology based on local geometrical and global structural models. *Neuroinformatics*, 9(2-3):247–261, 2011.

[59] T.-C. Lee, R. L. Kashyap, and C.-N. Chu. Building skeleton models via 3-D medial surface/axis thinning algorithms. *CVGIP: Graph. Models Image Process.*, 56(6):462–478, 1994.

[60] J. A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences (PNAS)*, 93(4):1591–1595, 1996.

[61] E. Türetken, F. Benmansour, and P. Fua. Automated reconstruction of tree structures using path classifiers and mixed integer programming. *CVPR*, 2012.

[62] E. Türetken, F. Benmansour, B. Andres, P. Glowacki, H. Pfister, and P. Fua. Reconstructing curvilinear networks using path classifiers and integer programming. *TPAMI*, PP(99):1–1, 2016.

[63] E. Bas and D. Erdogmus. Principal curves as skeletons of tubular objects - Locally characterizing the structures of axons. *Neuroinformatics*, 9(2-3):181–191, 2011.

[64] M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *International Journal of Computer Vision (IJCV)*, 1(4):321–331, 1988.

[65] C. Xu and J. L. Prince. Snakes, shapes, and gradient vector flow. *IEEE Trans. Image Process.*, 7(3):359–369, 1998.

[66] T. Xu, H. Li, T. Shen, N. Ojkic, D. Vavylonis, and X. Huang. Extraction and analysis of actin networks based on open active contour models. In *ISBI*, pages 1334–1340, 2011.

[67] V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *International Journal of Computer Vision (IJCV)*, 22(1):61–79, 1997.

[68] T. F. Chan and L. A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2):266–277, 2001.

[69] C. Li, C. Xu, C. Gui, and M. D. Fox. Level set evolution without re-initialization: A new variational formulation. *CVPR*, 1:430–436, 2005.

[70] J. Ivins and J. Porrill. Everthing you always wanted to know about snakes. *Artificial Intelligence Vision Research Unit (AIVRU) Technical Memo 86*, 2000.

[71] A. A. Amini, T. E. Weymouth, and R. C. Jain. Using dynamic programming for solving variational problems in vision. *TPAMI*, 12(9):855–867, 1990.

[72] D. J. Williams and M. Shah. A fast algorithm for active contours and curvature estimation. *CVGIP: Image Underst.*, 55(1):14–26, 1992.

[73] S. Osher and J. A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *J. Comput. Phys.*, 79(1):12–49, 1988.

[74] D. Mumford and J. Shah. Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42(5):577–685, 1989.

[75] P. Getreuer. Chan-vese segmentation. *Image Processing on Line (IPOL)*, 2:214–224, 2012.

[76] J. A. Bogovic, J. L. Prince, and P.-L. Bazin. A multiple object geometric deformable model for image segmentation. *Computer Vision and Image Understanding (CVIU)*, 117(2):145–157, 2013.

[77] G. Kühne, J. Weickert, M. Beier, and W. Effelsberg. *Fast Implicit Active Contour Models*, pages 133–140. Springer, 2002.

[78] Y. Zhang, B. J. Matuszewski, L. K. Shark, and C. J. Moore. Medical image segmentation using new hybrid level-set method. *MediVis*, pages 71–76, 2008.

[79] E. Bae and X.-C. Tai. *Efficient Global Minimization for the Multiphase Chan-Vese Model of Image Segmentation*, pages 28–41. Springer, 2009.

[80] S. Gulyanon, N. Sharifai, S. Bleykhman, E. Kelly, M. D. Kim, A. Chiba, and G. Tsechpenakis. Three-dimensional neurite tracing under globally varying contrast. *ISBI*, pages 875–879, 2015.

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

[82] M. A. Fischler and R. A. Elschlager. The representation and matching of pictorial structures. *IEEE Trans. Comput.*, C-22(1):67–92, 1973.

[83] P. F. Felzenszwalb and D. P. Huttenlocher. Pictorial structures for object recognition. *International Journal of Computer Vision (IJCV)*, 61(1):55–79, 2005.

[84] D. Fortun, P. Bouthemy, and C. Kervrann. Optical flow modeling and computation: A survey. *Computer Vision and Image Understanding (CVIU)*, 134:1–21, 2015.

[85] A. Gooya, K. M. Pohl, M. Bilello, L. Cirillo, G. Biros, E. R. Melhem, and C. Davatzikos. GLISTR: Glioma image segmentation and registration. *IEEE Trans. Med. Imag.*, 31(10):1941–1954, 2012.

[86] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models-their training and application. *Computer Vision and Image Understanding (CVIU)*, 61(1):38–59, 1995.

[87] C. Lu, S. Chelikani, X. Papademetris, J. P. Knisely, M. F. Milosevic, Z. Chen, D. A. Jaffray, L. H. Staib, and J. S. Duncan. An integrated approach to segmentation and nonrigid registration for application in image-guided pelvic radiotherapy. *Med. Image Anal.*, 15(5):772–785, 2011.

[88] D. Rueckert, L. I. Sonoda, C. Hayes, D. L. G. Hill, M. O. Leach, and D. J. Hawkes. Nonrigid registration using free-form deformations: Application to breast MR images. *IEEE Trans. Med. Imag.*, 18(8):712–721, 1999.

[89] A. Yezzi, L. Zollei, and T. Kapur. A variational framework for joint segmentation and registration. In *IEEE Workshop on Mathematical Methods in Biomedical Image Analysis (MMBIA)*, pages 44–51, 2001.

[90] C. Xiaohua, M. Brady, and D. Rueckert. Simultaneous segmentation and registration for medical image. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 663–670, 2004.

[91] K. M. Pohl, J. Fisher, W. E. L. Grimson, R. Kikinis, and W. M. Wells. A Bayesian model for joint segmentation and registration. *NeuroImage*, 31(1):228–239, 2006.

[92] S. Parisot, W. Wells III, S. Chemouny, H. Duffau, and N. Paragios. Concurrent tumor segmentation and registration with uncertainty-based sparse non-uniform graphs. *Med. Image Anal.*, 18(4):647–659, 2014.

[93] S. Gulyanon, N. Sharifai, M. D. Kim, A. Chiba, and G. Tsechpenakis. Neurite reconstruction from time-lapse sequences using co-segmentation. *ISBI*, pages 410–414, 2017.

[94] M. D. Kim, Y. Wen, and Y.-N. Jan. Patterning and organization of motor neuron dendrites in the Drosophila larva. *Developmental Biology*, 336(2):213–221, 2009.

[95] P. Glowacki, M. A. Pinheiro, A. Mosinska, E. Türetken, D. Lebrecht, A. Holtmaat, R. Sznitman, J. Kybic, and P. Fua. Reconstructing evolving tree structures in time lapse sequences by enforcing time-consistency. *TPAMI*, PP(99):1–1, 2017.

[96] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever. Multiscale vessel enhancement filtering. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 130–137, 1998.

[97] M. W. K. Law and A. C. S. Chung. Three dimensional curvilinear structure detection using optimally oriented flux. In *European Conference on Computer Vision (ECCV)*, pages 368–382, 2008.

[98] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. *CVPR*, pages 1943–1950, 2010.

[99] C.-K. Tang and G. Medioni. Inference of integrated surface, curve and junction descriptions from sparse 3D data. *TPAMI*, 20(11):1206–1223, 1998.

[100] A. Sironi, V. Lepetit, and P. Fua. Multiscale centerline detection by learning a scale-space distance transform. *CVPR*, pages 2697–2704, 2014.

[101] S. Mukherjee and S. T. Acton. Oriented filters for vessel contrast enhancement with local directional evidence. *ISBI*, pages 503–506, 2015.

[102] A. Sironi, E. Türetken, V. Lepetit, and P. Fua. Multiscale centerline detection. *TPAMI*, 38(7):1327–1341, 2016.

[103] X. Yu, J.-W. Tian, and J. Liu. Transformation model estimation for point matching via gaussian processes. In *World Congress on Engineering (WCE)*, 2007.

[104] J. Munkres. Algorithms for the assignment and transportation problems. *SIAM J. Appl. Math.*, 1957.

[105] J. P. W. Pluim, J. B. A. Maintz, and M. A. Viergever. Mutual-information-based registration of medical images: A survey. *IEEE Trans. Med. Imag.*, 22(8):986–1004, 2003.

[106] B. Zitová and J. Flusser. Image registration methods: A survey. *Image and Vision Computing*, 21(11):977–1000, 2003.

[107] P. Viola and W. M. Wells III. Alignment by maximization of mutual information. *International Journal of Computer Vision (IJCV)*, 24(2):137–154, 1997.

[108] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - Incorporating a global constraint into MRFs. *CVPR*, 1:993–1000, 2006.

[109] J. C. Rubio, J. Serrat, A. López, and N. Paragios. Unsupervised co-segmentation through region matching. *CVPR*, pages 749–756, 2012.

[110] H. Y. Chen, Y. Y. Lin, and B. Y. Chen. Co-segmentation guided hough transform for robust feature matching. *TPAMI*, 37(12):2388–2401, 2015.

[111] F. Wang, Q. Huang, and L. J. Guibas. Image co-segmentation via consistent functional maps. In *ICCV*, pages 849–856, 2013.

[112] C. Rother, V. Kolmogorov, and A. Blake. "GrabCut": Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graph.*, 23(3):309–314, 2004.

[113] W. T. Freeman and E. H. Adelson. The design and use of steerable filters. *TPAMI*, 13(9):891–906, 1991.

[114] B. Li and S. T. Acton. Vector field convolution for image segmentation using snakes. *IEEE International Conference on Image Processing (ICIP)*, pages 1637–1640, 2006.

[115] D.-J. Kroon. *Segmentation of the mandibular canal in cone-beam CT data*. PhD thesis, University of Twente, Enschede, the Netherlands, December 2011.

[116] A. A. Taha and A. Hanbury. Metrics for evaluating 3D medical image segmentation: Analysis, selection, and tool. *BMC Med. Imaging*, 15(1):29, 2015.

[117] Z. Vasilkoski and A. Stepanyants. Detection of the optimal neuron traces in confocal microscopy images. *J. Neurosci. Methods*, 178(1):197–204, 2009.

[118] F. Benmansour and L. D. Cohen. Tubular structure segmentation based on minimal path method and anisotropic enhancement. *International Journal of Computer Vision (IJCV)*, 92(2):192–210, 2011.

[119] E. Meijering, M. Jacob, J.-C. F. Sarria, and M. Unser. A novel approach to neurite tracing in fluorescence microscopy images. *Signal and Image Processing (SIP)*, pages 491–495, 2003.

[120] Y. Rouchdy and L. D. Cohen. Geodesic voting for the automatic extraction of tree structures. methods and applications. *Computer Vision and Image Understanding (CVIU)*, 117(10):1453–1467, 2013.

[121] H. Chen, H. Xiao, T. Liu, and H. Peng. SmartTracing: Self-learning-based neuron reconstruction. *Brain Informatics*, 2(3):135–144, 2015.

[122] R. Gala, J. Chapeton, J. Jitesh, C. Bhavsar, and A. Stepanyants. Active learning of neuron morphology for accurate automated tracing of neurites. *Frontiers in Neuroanatomy*, 8:37, 2014.

[123] H. Peng, F. Long, and G. Myers. Automatic 3D neuron tracing using all-path pruning. *Bioinformatics*, 27(13):i239–i247, 2011.

[124] H. Xiao and H. Peng. APP2: Automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics*, 29(11):1448–1454, 2013.

[125] E. Türetken, G. González, C. Blum, and P. Fua. Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. *Neuroinformatics*, 9(2–3):279–302, 2011.

[126] Y. Wang, A. Narayanaswamy, and B. Roysam. Novel 4-D open-curve active contour and curve completion approach for automated tree structure extraction. *CVPR*, pages 1105–1112, 2011.

[127] S. Mukherjee, B. Condron, and S. T. Acton. Tubularity flow field - A technique for automatic neuron segmentation. *IEEE Trans. Image Process.*, 24(1):374–389, 2015.

[128] X. Xie and M. Mirmehdi. MAC: Magnetostatic active contour model. *TPAMI*, 30(4):632–646, 2008.

[129] B. Li and S. T. Acton. Active contour external force using vector field convolution for image segmentation. *IEEE Trans. Image Process.*, 16(8):2096–2106, 2007.

[130] A. K. Mishra, P. W. Fieguth, and D. A. Clausi. Decoupled active contour (DAC) for boundary detection. *TPAMI*, 33(2):310–324, 2011.

[131] P. B. Chou, P. R. Cooper, M. J. Swain, C. M. Brown, and L. E. Wixson. Probabilistic network inference for cooperative high and low level vision. In R. Challappa and A. Jain, editors, *Markov Random Fields: Theory and Application, Boston*, pages 211–243. Academic Press, 1993.

[132] E. Candès, L. Demanet, D. Donoho, and L. Ying. Fast discrete curvelet transforms. *Multiscale Modeling and Simulation*, 5(3):861–899, 2006.

[133] J. N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. In *Proc. IEEE Conf. Decis. Control*, volume 2, pages 1368–1373, 1994.

[134] L. Yatziv, A. Bartesaghi, and G. Sapiro. O(N) implementation of the fast marching algorithm. *J. Comput. Phys.*, 212(2):393–399, 2006.

[135] 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.

[136] A.-S. Chiang, C.-Y. Lin, C.-C. Chuang, H.-M. Chang, C.-H. Hsieh, C.-W. Yeh, C.-T. Shih, J.-J. Wu, G.-T. Wang, Y.-C. Chen, C.-C. Wu, G.-Y. Chen, Y.-T. Ching, P.-C. Lee, C.-Y. Lin, H.-H. Lin, C.-C. Wu, H.-W. Hsu, Y.-A. Huang, J.-Y. Chen, H.-J. Chiang, C.-F. Lu, R.-F. Ni, C.-Y. Yeh, and J.-K. Hwang. Three-dimensional reconstruction of brain-wide wiring networks in Drosophila at single-cell resolution. *Current Biology*, 21(1):1–11, 2011.

[137] T. A. Gillette, K. M. Brown, and G. A. Ascoli. The DIADEM metric: Comparing multiple reconstructions of the same neuron. *Neuroinformatics*, 9(2–3):233–245, 2011.

[138] K. Zhang. A constrained edit distance between unordered labeled trees. *Algorithmica*, 15(3):205–222, 1996.

[139] S. Gulyanon, L. He, W. D. Tracey Jr., and G. Tsechpenakis. Neurite tracking in time-lapse calcium images using MRF-modeled pictorial structures. *ISBI*, 2018.

[140] W. B. Grueber, L. Y. Jan, and Y. N. Jan. Tiling of the Drosophila epidermis by multidendritic sensory neurons. *Development*, 129(12):2867–2878, 2002.

[141] C. L. Hughes and J. B. Thomas. A sensory feedback circuit coordinates muscle activity in Drosophila. *Mol. Cell. Neurosci.*, 35(2):383–396, 2007.

[142] L. Sigal, S. Bhatia, S. Roth, M. J. Black, and M. Isard. Tracking loose-limbed people. *CVPR*, 1:I–421–I–428, 2004.

[143] T. X. Han and T. S. Huang. *Articulated Body Tracking Using Dynamic Belief Propagation*, pages 26–35. Springer, 2005.

[144] T. B. Moeslund, A. Hilton, and V. Krüger. A survey of advances in vision-based human motion capture and analysis. *Computer Vision and Image Understanding (CVIU)*, 104(2):90–126, 2006.

[145] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *International Joint Conferences on Artificial Intelligence (IJCAI)*, pages 674–679, 1981.

[146] B. K. P. Horn and B. G. Schunck. Determining optical flow. Technical report, Massachusetts Institute of Technology, 1980.

[147] D. Sun, S. Roth, and M. J. Black. A quantitative analysis of current practices in optical flow estimation and the principles behind them. *International Journal of Computer Vision (IJCV)*, 106(2):115–137, 2014.

[148] Y. Hu, R. Song, and Y. Li. Efficient coarse-to-fine patch match for large displacement optical flow. *CVPR*, pages 5704–5712, 2016.

[149] J. Revaud, P. Weinzaepfel, Z. Harchaoui, and C. Schmid. EpicFlow: Edge-preserving interpolation of correspondences for optical flow. *CVPR*, pages 1164–1172, 2015.

[150] E. Simo-Serra, E. Trulls, L. Ferraz, I. Kokkinos, P. Fua, and F. Moreno-Noguer. Discriminative learning of deep convolutional feature point descriptors. In *ICCV*, pages 118–126, 2015.

[151] S. Zagoruyko and N. Komodakis. Learning to compare image patches via convolutional neural networks. *CVPR*, pages 4353–4361, 2015.

[152] C. Bailer, K. Varanasi, and D. Stricker. CNN-based patch matching for optical flow with thresholded hinge loss. *CVPR*, 2017.

[153] V. Ferrari, M. Marin-Jimenez, and A. Zisserman. Pose search: Retrieving people using their pose. *CVPR*, pages 1–8, 2009.

[154] R. O. Duda and P. E. Hart. Use of the hough transformation to detect lines and curves in pictures. *Commun. ACM*, 15(1):11–15, 1972.

[155] T. J. Atherton and D. J. Kerbyson. Size invariant circle detection. *Image and Vision Computing*, 17(11):795–803, 1999.

[156] Y. Y. Boykov and M. P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in N-D images. *ICCV*, 1:105–112, 2001.

[157] O. Woodford, P. Torr, I. Reid, and A. Fitzgibbon. Global stereo reconstruction under second-order smoothness priors. *TPAMI*, 31(12):2115–2128, 2009.

[158] E. Boros and P. L. Hammer. Pseudo-Boolean optimization. *Discrete Appl. Math.*, 123(1-3):155–225, 2002.

[159] V. Kolmogorov and C. Rother. Minimizing nonsubmodular functions with graph cuts - A review. *TPAMI*, 29(7):1274–1279, 2007.

[160] 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.*, 28(4):348–353, 2010.

[161] D. Mayerich, C. Bjornsson, J. Taylor, and B. Roysam. NetMets: Software for quantifying and visualizing errors in biological network segmentation. *BMC Bioinformatics*, 13(8):S7, 2012.

[162] S. Gulyanon, N. Sharifai, M. D. Kim, A. Chiba, and G. Tsechpenakis. Part-wise neuron segmentation using artificial templates. *ISBI*, 2018.

[163] T. Lee and L. Luo. Mosaic analysis with a repressible cell marker for studies of gene function in neuronal morphogenesis. *Neuron*, 22(3):451–461, 1999.

[164] N. Sharifai, H. Samarajeewa, D. Kamiyama, T.-C. Deng, M. Boulina, and A. Chiba. Imaging dynamic molecular signaling by the Cdc42 GTPase within the developing CNS. *Public Library of Science (PLoS) One*, 9(2):1–9, 2014.

[165] G. Unal and G. Slabaugh. Coupled PDEs for non-rigid registration and segmentation. *CVPR*, 1:168–175, 2005.

[166] F. Wang and B. C. Vemuri. Simultaneous registration and segmentation of anatomical structures from brain MRI. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 17–25, 2005.

[167] T. Lin, C. Le Guyader, I. Dinov, P. Thompson, A. Toga, and L. Vese. Gene expression data to mouse atlas registration using a nonlinear elasticity smoother and landmark points constraints. *J. Sci. Comput.*, 50(3):586–609, 2012.

[168] B. Fischer and J. Modersitzki. A unified approach to fast image registration and a new curvature based registration technique. *Linear Algebra Appl.*, 380:107–124, 2004.

[169] A. Gooya, K. M. Pohl, M. Bilello, G. Biros, and C. Davatzikos. Joint segmentation and deformable registration of brain scans guided by a tumor growth model. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, 14(Pt 2):532–540, 2011.

[170] S. Parisot, H. Duffau, S. Chemouny, and N. Paragios. Joint tumor segmentation and dense deformable registration of brain MR images. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 651–658, 2012.

[171] S. Parisot, W. Wells, S. Chemouny, H. Duffau, and N. Paragios. Uncertainty-driven efficiently-sampled sparse graphical models for concurrent tumor segmentation and atlas registration. *ICCV*, pages 641–648, 2013.

[172] P. R. Andresen and M. Nielsen. Non-rigid registration by geometry-constrained diffusion. *Med. Image Anal.*, 5(2):81–88, 2001.

[173] B. Glocker, N. Komodakis, G. Tziritas, N. Navab, and N. Paragios. Dense image registration through MRFs and efficient linear programming. *Med. Image Anal.*, 12(6):731–741, 2008.

[174] D. Rueckert, P. Aljabar, R. A. Heckemann, J. V. Hajnal, and A. Hammers. Diffeomorphic registration using B-splines. *Medical Image Computing and Computer-Assisted Intervention (MICCAI)*, pages 702–709, 2006.

[175] Y. Choi and S. Lee. Injectivity conditions of 2D and 3D uniform cubic B-spline functions. *Graphical Models*, 62(6):411–427, 2000.

[176] M. Hagenlocker and K. Fujimura. CFFD: A tool for designing flexible shapes. *The Visual Computer*, 14(5):271–287, 1998.

[177] L. R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[178] Z. Li, F. Shen, R. Fang, S. Conjeti, A. Katouzian, and S. Zhang. Maximum inner product search for morphological retrieval of large-scale neuron data. *ISBI*, pages 602–606, 2016.

[179] R. Cannon, D. A. Turner, G. K. Pyapali, and H. V. Wheal. An on-line archive of reconstructed hippocampal neurons. *J. Neurosci. Methods*, 84:49–54, 1998.

[180] H. Peng, M. Hawrylycz, J. Roskams, S. Hill, N. Spruston, E. Meijering, and G. A. Ascoli. BigNeuron: Large-scale 3D neuron reconstruction from optical microscopy images. *Neuron*, 87(2):252–256, 2015.

[181] C. Buranachai, D. Kamiyama, A. Chiba, B. D. Williams, and R. M. Clegg. Rapid frequency-domain FLIM spinning disk confocal microscope: Lifetime resolution, image improvement and wavelet analysis. *J. Fluoresc.*, 18(5):929–942, 2008.

APPENDICES

APPENDIX A: SWC FORMAT

Digitally reconstructed neurons used in this work are stored as the tree data structure following the SWC format [179], which is non-proprietary. The tree data structure is represented by the 3D vector-based reconstruction. Files may begin with headers, which start with #. Tree structure parameters are organized into 7 columns, where each row represents one trace point. From left to right, these columns are: the unique identity value of trace point, the structure type identifier, xyz-coordinates, radius, and the identity value of parent point (i.e., the trace point that comes before and connects to the current trace point). An example SWC file with 10 points is provided below[1]:

```
#Example header text here
1 2 882 797 19 9 -1
2 2 882 797 19 9 1
3 2 875 821 19 10 2
4 2 852 849 19 21 3
5 2 842 827 18 12 4
6 2 835 816 18 7 5
7 2 827 807 18 7 6
8 2 814 797 18 4 3
9 2 803 785 18 4 8
10 2 785 763 18 4 9
```

The first row is the example header text, which is not a part of the neuron trace. While, the second row represents the trace point which has the identity = 1, type = 2 (i.e., axon), $x = 882$, $y = 797$, $z = 19$, radius = 9, and no parent's trace point

[1]More details in `http://diademchallenge.org/faq.html`

because its parent identity number = -1. Hence, the second row is the root point of the reconstruction. The fourth row has the identity = 3, type = 2 (i.e., axon), x = 875, y = 821, z = 19, radius = 10, and trace point 2 is its parent.

The first, second, and last columns are integers, while other columns are real values whose units correspond to the reconstructions process, e.g., pixels or micometers. The identity number in the first column must be ordered and always increase by one, whereas the parent identity number in the last column does not have such restrictions but it must be less than its identity number in the first column of the same row. If multiple points have the same parent trace point, then that trace point is a bifurcation point.

The value for the second column is the structure type identifier, which encodes the neuronal compartments. It depends on the standard that the dataset adopted. For DIADEM datasets, 1 = cell body, 2 = axon, and 3 = dendrite. For the standardized SWC format[2]: 0 = undefined, 1 = soma or cell body, 2 = axon, 3 = (basal) dendrite, 4 = apical dendrite, 5+ = custom.

[2]More details in `http://www.neuromorpho.org/myfaq.jsp`

APPENDIX B: DATASETS

In this dissertation, we validate our methods using both public and private datasets. The public datasets that we used are DIADEM challenge[1], FlyCircuit[2], and sensory neurons in the larval *Drosophila* dataset[3]. The private datasets that we used are calcium images of ddaD and ddaE neurons, and aCC motorneuron images.

B.1 DIADEM Challenge Datasets

DIADEM challenge datasets are available online on the DIADEM challenge's website. They compose of six datasets: *Cerebellar Climbing Fibers*, *Hippocampal CA3 Interneuron*, *Neocortical Layer 1 Axons*, *Neuromuscular Projection Fibers*, *Olfactory Projection Fibers*, and *Visual Cortical Layer 6 Neuron*. Here we evaluate against only two datasets — the *Olfactory Projection Fibers* (OP) and the *Cerebellar Climbing Fibers* (CF) — because they are the only two datasets with single neuron per image stack.

OP dataset are images of Olfactory Bulb region of *Drosophila* captured by 2-channel confocal microscopy. Green Fluorescent Protein (GFP) was used for labeling axons. This dataset contains 9 grayscale image stacks.

CF dataset contains images of Cerebellar Cortex region of rat taken by transmitted light bright-field microscopy. Biotinylated Dextran Amine (Anterograde) was used for labeling neuronal arbors. The image slice is stored as an RGB image. The dataset contains 3 image stacks.

We evaluate our methods on only 8 stacks from the *Olfactory Projection Fibers* (OP) dataset, and 2 stacks from the *Cerebellar Climbing Fibers* (CF) dataset [28].

[1] `http://diademchallenge.org/`
[2] `www.flycircuit.tw`
[3] `http://neurovision.cs.iupui.edu/`

We left out some image stacks because they either contain more than one neuron or they are unstitched.

B.2 FlyCircuit

FlyCircuit database is one of the largest collection of neuronal images at single-cell resolution. It is a public database for online archiving, cell type inventory, browsing, searching, analysis and 3D visualization of individual neurons in the *Drosophila* brain. Sample image stacks were acquired by a Zeiss LSM 510 confocal microscope with a 40× C-Apochromat water-immersion objective lens (numerical aperture value 1.2, working distance 220 μm). The voxel size ratio is $0.32 \times 0.32 \times 1$ μm and data volumes are around $1024 \times 1024 \times 130$ voxels in size. Images from FlyCircuit were obtained from the NCHC (National Center for High-performance Computing) and NTHU (National Tsing Hua University), Hsinchu, Taiwan [136]. The manual reconstruction was produced by the cooperation of 6 or 7 annotators [29, 180]. We validate our methods on 10 image stacks[4] from this dataset that also appear in the BigNeuron project [29].

B.3 Sensory Neurons in the Larval *Drosophila*

The fruit fly samples were prepared and imaged by Dr. Akira Chiba and his team [80]. The sample preparation process can be divided into five steps:

1. **Fly propagation.** Adult flies (ppk-GAL4 mCD8::GFP) were kept in vials and stored at room temperature. Vials contained fly feed composed of cornstarch, agar, molasses, and yeast.
2. **Genetic crosses.** 3-4 males and females were placed in mating cages to facilitate larval collection. Cages were loaded with grape agar plates containing two drops of live active yeast on the center of the agar plate. Agar plates were

[4] `https://www.dropbox.com/sh/c7tttjv0vicgdo5/AADma_T9New3uGcIA6ofEpWZa?dl=0`

swapped at appropriate times to ensure correct age of larval development (72 hours for 1st day 3rd instar). Larvae were checked for typical morphological features to ensure correct age. The cages and plates were incubated at $25°C$.

3. **Mounting.** Larvae were mounted one per slide ($75 \times 25 \times 1$ mm) in Halocarbon 1000N oil to match the refractive index of microscope objective oil-immersion fluid. Coverslips (22×22 mm) were secured using putty in order to apply appropriate pressure without popping the larva and to prevent larva movement while imaging.
4. **Anatomy.** Larvae for all experiments were 48-72 hours old (2nd-3rd instar). The neurons used in imaging were on the distal left side along the larvae's dorsal end, within hemisegments T2, T3, A1, and A2.
5. **Microscopy.** Image stacks (.5 micron sections) of Class IV da (dendritic arborization) sensory neurons were detected with an inverse confocal laser scanning microscope (Zeiss LSM 780) using a Plan-Neofluar 40x/1.30 Oil M27 objective (Zeiss) and the ZEN 2010 software. The system utilized an argon laser line (Ar-Laser Multiline 458/488/514 nm; Zeiss) for GFP excitation (488 nm) at 5% intensity, beam splitter MBS 488, and a pinhole size of 1 airy unit.

This dataset contains 11 image stacks of the sensory neurons in the wild-type larval *Drosophila*. It is for the study of dendritic arborization patterns over the four instars of development. These volumes are around $1024 \times 1024 \times 20$ in size and their background has spatially inhomogeneous signal-to-noise ratios.

B.4 Calcium Images

Calcium images allow us to visualize the neuronal activities by measuring calcium responses produced by cellular signaling. The dataset contains image sequences during the locomotion of the *Drosophila* larvae. In calcium image acquisition process, larvae expressing the genetically encoded calcium sensor GCaMP6.0F in the proprioceptors, i.e., ddaD and ddaE, were observed in three dimensional confocal microscopy

volumes over time. This is possible because the neurons are found directly beneath the cuticle and epidermis, which are optically transparent. Data analysis of neurons in these volumes presents specific challenges that are not present in typical calcium time series because the neurons of interest are moving in 4D.

Our dataset contains a number of sequence volumes with varying number of frames from a few hundreds to nearly a thousand frames. The entire sequence contains multiple neurons (usually around 6-7 neurons), where each frame contains a few neurons. The ground truths are manually traced only during the larval locomotion because there are only insignificant changes in morphology when neurons are stationary. In addition, inactive neurons occupy most of the sequence so stationary periods should be ignored to remove the bias from recurring frames/traces.

B.5 aCC Motorneuron Images

This dataset contains images of aCC (anterior corner cell) neurons in multiple spatial resolutions at multiple developmental stages. High resolution neuron volumes are for studying morphological complexity, while lower resolution neuron volumes are for detecting protein interaction localization/annotation within the *Drosophila* Ventral Nerve Cord (VNC).

Protein interaction imaging was acquired using a custom-assembled frequency-domain upright Fluorescence Lifetime Imaging Microscope (FLIM) system [181]. Images were taken in a focal plane, where the embryo's Central Nervous System (CNS) possessed maximal neuropil width. FLIM used in this study combines the micrometer spatial resolution of fluorescence imaging with nanosecond temporal resolution of fluorescence lifetime. We measure the lifetime and concentration of protein interactions utilizing Förster Resonance Energy Transfer (FRET). FRET leads to both energy loss from fluorescence donor molecule and corresponding gain by fluorescence acceptor molecule during the protein interaction. Then, FLIM measures the protein

interaction through the change in the donors fluorescence lifetime within each image pixel. More details of image acquisition are described in [164].

The ground truths of 45 aCC motorneuron image stacks of larval *Drosophila* are available. They contain both wild-type and knockout samples, in two spatial resolutions, and at different development instances. Each neuron volume is manually segmented into ten neuronal compartments based on the biological context.

VITA

VITA

Sarun Gulyanon received his Doctor of Philosophy and Master of Science degrees in computer science from Purdue University, and the Bachelor of Science degree in computer science with First Class Honours from the University of Edinburgh. His current research interests are computer vision, biomedical imaging, and machine learning, more specifically on fluorescence microscopy image processing and visualization.