Soft Segmentation of Viral Labeled Neurons

Sabri Bolkar

20-08-2018

Master's Thesis
30 ECTS

Supervisor: Prof. Karl Farrow
Co-Supervisor: Prof. Jon Hardeberg

Abstract

Fluorescent labeling of neurons by neurotropic viruses has become an important tool used in neuroscientific research for tracing synaptic connections and analyzing neuroanatomy. After the advent of combinatorial spectral expression by means of viral vectors, the technique now allows simultaneous discrimination of neurons in great quantities. However, manual analysis of viral labeled neurons is time-consuming and poses many challenges for scientists. Designing an automatic and robust processing algorithm has been found elusive especially for neural morphology studies because of complex and stochastic appearance of thin dendritic structures. As the analysis of neural morphology requires extraction of full neural arborization (i.e., branching), monochromatic viral labeling only allows processing of sparse regions that do not contain overlapping neurons, thereby motivating the development of spectral labeling for sparsification of neuron populations.

Unlike previous studies favoring hard segmentation, we propose a fully unsupervised pixel-level soft segmentation algorithm to separate spectral labeled neurites and to enable morphological analysis of overlapping neurons. Firstly, to eliminate biological noise and enhance dendrites, we cast prepossessing step as piecewise-image recovery task, and propose a fast image flattening algorithm based on a weighted ℓ_1-norm total variation where close data points are leveled out and high gradients are preserved. We efficiently minimize the energy function by a modular taut-string based proximal algorithm. To further augment preprocessing, background subtraction and linear contrast enhancement are employed. Inspired by color and object manipulation methods employed in computer graphics, we invert weighted Porter-Duff compositing model by ridge and ℓ_2-norm total variation regularized optimization and extract opacity values corresponding each pixel for each of the expressed fluorescent labels (i.e., pseudo-colors) as soft segments. Finally, we show the effectiveness of our algorithm by qualitative analysis applied to confocal image stack obtained from mice retina and by quantitative measures using synthetically generated images with the corresponding ground truth.

Our algorithm design aims classification of retinal ganglion cells, and it is specifically designed to relax the hard boundaries and to keep dendrites intact as their arborization carry the information for identification of these cells. To reconstruct full branching of neurons in 3d, we propose a post-processing algorithm based on nonlinear morphological operations and convert soft segments to binary masks. This masking step ensures preservation of details and removes non-neural components. The code of the full pipeline is publicly available at github.com/elras/SoftsegmentNeurons.

Acknowledgment

This master thesis was conducted at the Neuro-electronics Research Flanders (research initiative by KU Leuven, IMEC & VIB). The thesis report is submitted to fulfill COSI Erasmus Mundus Joint MSc Degree.

I would like to thank my supervisor Karl Farrow for providing me the opportunity to contribute neurotechnology research. This work would not have been possible without Chen Li's and Katja Reinhard's direct assistance during the course of this master thesis.

I would like to express my gratitude to all other Farrowlab members, Norma Kühn, Marta Pereira, Arnau Sans Dublanc, Daniel De Malmazat, Anna Chrzanowska, Carolina Avelar and Ilayda Alkislar for their great friendships.

I also would like to thank my co-supervisor Jon Hardeberg and NTNU Colour and Visual Computing Lab members for their support.

Contents

List of Figures

List of Tables

1 Introduction

Historically considered as a qualitative technique, advances in computational resources and imaging modalities have made microscopy a powerful quantitative tool for modern biology research. The new research directions such as computational microscopy and bioimage analysis show great promise for many challenges from understanding subcellular mechanisms to large neural networks. Both electron and optical microscopy have become the primary ways to retrieve information. However, unlike electron microscopy, optical microscopy (e.g., widefield, confocal, multiphoton, light-sheet) is able to provide high spatial, spectral and temporal resolution in vivo. Target specific imaging of fluorescence labels enabled by genetic engineering and viral tracing makes optical microscopy even more advantageous and inalienable for many tasks where specificity is of importance. In this project, we particularly focus on the analysis of confocal microscopy images where fluorescent emission is sustained by means of viral tracing. Although this technique creates cell specificity and enables tracing of connected networks by transsynaptic transmission of viral vectors, it is stochastic and inherently generates random spectral labelling of targeted neurons. From image processing point of view, viral labeled neural data poses many challenges that have to be addressed before proceeding to analysis. Furthermore, advances in automatic processing (e.g., denoising, segmentation) has been tampered as biosignatures are generally application-dependent and may not be unveiled by general algorithms. For that reason, segmentation is known to be one of the most challenging tasks in bioimage analysis [2].

Identification of neural cell types, understanding their function and deciphering their connectome are of the main goals of neuroscientific research [3, 4, 5]. But even for the retina, one of the most studied parts of the nervous system, the number of distinct retinal ganglion cell (RGC) types and their projections to corresponding brain areas remain unclear [6]. Considering the importance of ganglion cells as they transmit the sole output of visual signals from the retina, determination of their types and function become an important research question in visual neuroscience. To distinguish distinct RGC types; morphological characteristics (i.e., dendritic branching, arborization), electrical response properties and molecular or genetic markers have been employed in the literature [7, 8, 9, 10, 11, 12]. Among these methods, quantitative analysis of neural anatomy (i.e., neural morphology) is particularly attractive as the development of fluorescent expressing viral vectors and sophisticated microscopes make simultaneous capturing of neurons and image-based analysis of their morphology possible [13, 14]. Augmented with automatic image processing pipelines, imaging of viral labelled neurons create the opportunity to disclose characteristics of large neural populations [15].

Neurotropic viruses have become a standard tool for labeling distinct cell types and unveiling their connectivity sustained by synaptic connections [16]. Specific virus types have natural charac-

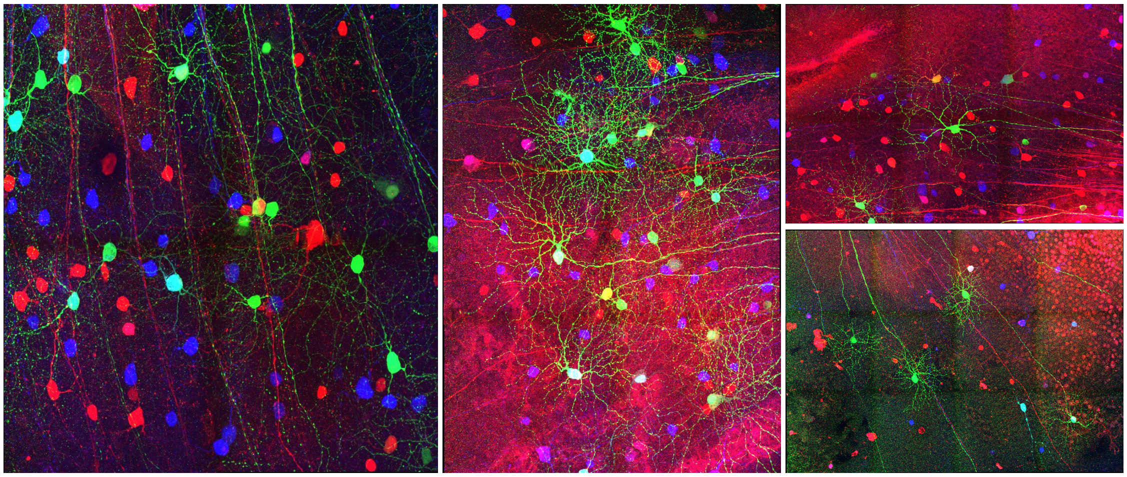


Figure 1: Sample spectral viral labeled retinal ganglion cells (Retinabow). In this four different maximum intensity projection images obtained by confocal laser scanning microscopy, expression and mixing of pseudo-color fluorescent proteins are clearly visible.

teristics that make them ideal neural tracers: They can surpass neuron membrane and interact with cell nucleic acids, they can travel along axons in anterograde or retrograde direction, they can be manipulated to fill the whole cell or to amplify the relevant signal, when necessary eliminating the need for immunochemistry, they allow spectral tracing, and lastly they can cross synapses and tag post-synaptic targets [17]. Therefore, when incorporated with fluorescent proteins (e.g., GFP [18]), they become a powerful tool for neural circuit analysis as they enable image-based observation and analysis of neural components [19]. Yet, manual tracing of captured images requires expert labour, besides automatic processing is heavily complicated by naturally overlapping structure of dendritic trees. One way to overcome this challenge is to label neurons with spectral signature expressing fluorescent proteins (i.e., commonly illustrated by pseudo-colors), thereby utilizing spectral information as the cue for automatic separation of dendrites. Spectral neuron labeling has been previously applied for analyzing neural connectome of mouse hippocampus (brainbow) [20], drosophila (flybow) [21] and zebrafish (zebrabow) [22]. Combinatorial multicolor fluorescent expression can be obtained via variety of strategies. In our case, three distinct fluorescent proteins; mCherry emitting in the red end of visible spectrum, BFG in the blue end, and GCaMP6 in the green band have been employed (i.e, three-channel spectral imaging). Details for our protocol to obtain multi-color expression in retina (retinabow) is explained in the Appendix A. Briefly, the method is based on injection of three different retrograde viral vectors to wide-field neurons in the mouse superior calculus. As wide-field neurons take input from retinal ganglion cells, by allowing viruses to transverse for a certain amount of time (approximately 21 days), retinal ganglion cells can be contaminated by combinatorial mixture of these three viral vectors (Figure 1).

Our goal differs from these works as we desire to analyze neuron morphology instead of con-

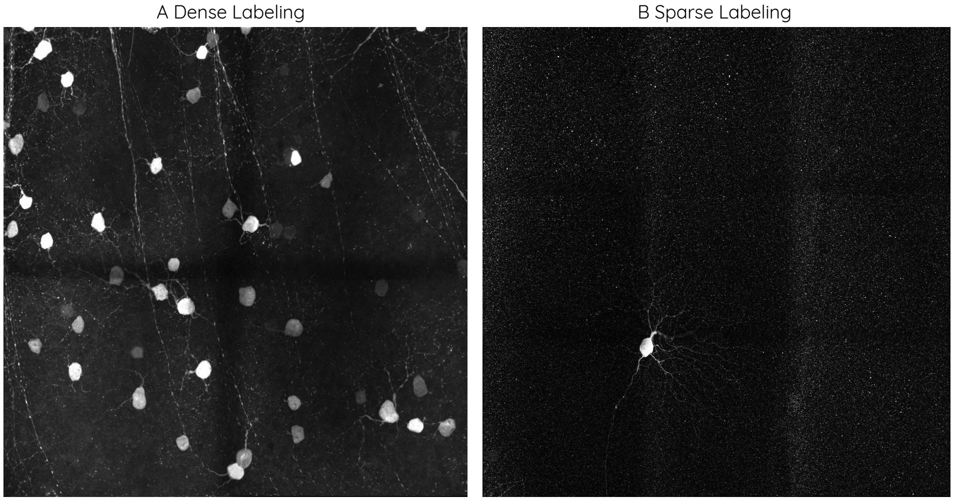


Figure 2: Dense vs sparse viral monochrome labeling: (A) Labeling of densely distributed neurons, (B) labeling of an isolated neuron.

nectomics. To do so, we create fluorescent expression by using three different viral vectors and then sparsify juxtaposed RGC neurites. This thesis focuses on image processing aspects and aims to further contribute image-based cell type identification by developing an unsupervised algorithm to extract soft segments from confocal microscope image stacks that are stained by distinct spectral signatures.

1.1 Problem Statement

The established way of classifying neural cells is done by clustering of features obtained from the distribution of neuron's dendritic tree. This clustering is further supported by additional genetic or molecular markers, therefore giving the opportunity to identify unknown cell types [13]. However, the main problem with this approach is that it can only be applied when neurons are isolated, in other words, their dendrites show minimum overlap with neighbouring neurons' arborization (Figure 2). This problem restricts the applicability of the approach for most of the neurons, especially in the retina, as retinal ganglion cells are distributed densely to enable even sampling across the visual field.

By discovering different fluorescent proteins that emit light in non-overlapping spectral bands, one may exploit traditional viral tracing and genetical tools to multiplex different fluorescent labeling for densely situated neurons. Therefore, the specification can be obtained by segmentation of neurons homogeneously labeled by fluorescence mixture.

Existing literature uses both terms, multicolor and spectral labeling, interchangeably when referring the technique. However in this project, we are interested in the unsupervised fuzzy classification of pixels (e.g., soft clustering, segmentation) according to distinct fluorescence labels, and

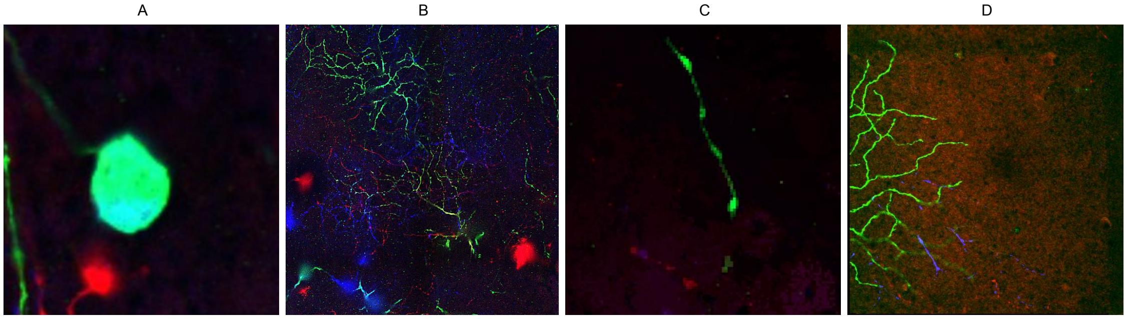


Figure 3: Challenges in processing retinabow. (A) Inhomogenous labelling (both cyan and green hue can be observed), (B) overlapping nature of dendrites, (C) very thin appearance of dendrites (an example of 1 pixel-wide branch), (D) biological background noise (unintended staining of background tissues).

perceptual attributes are not of interest. Therefore, we prefer using the term "fluorescent label" or "spectral label" instead of "multicolor" when discussing the analysis of these images. In this work, pixels are merely tensors that can be represented in any orthogonal coordinate space. Following previous works, our pseudocolor spectral images are illustrated in sRGB color space. These images contain 4 spectral channels where the 4th channel is used for reference purposes to label starburst amacrine cells by ChAT antibodies, and other three channel is used to label and segment retinal ganglion cells.

As stated previously, spectral labeling is stochastic; in other words, spectral mixing cannot be directly controlled and (if happens) it is random. As a work-in-progress protocol, it also exhibits severe challenges further complicating analysis of captured images. As illustrated in Figure 3, the main challenges that must be addressed during the design step are: Heavily overlapping nature of dendrites, viral diffusion inhomogeneity, very thin appearance of dendrites, and background biological noise.

1.2 Research Proposal and Contributions

Dendritic trees contain features used to identify the cell types, hence an algorithm aiming segmentation of neurons should be carefully designed not to spoil any relevant information. However, dendrites are much harder to segment comparing neuron soma and they are more vulnerable to color variations especially caused by background noise (i.e., because of their thin structure). Deterioration caused by the imaging system may adversely affect dendrites more than soma. To solve this problem, it is necessary to implement a deconvolution step during preprocessing. On the other hand, deconvolution is actually not desired as we aim to improve label homogeneity in neurons and deconvolution might unmix observed spectral mixing i.e., by applying deconvolution, diffusion homogeneity of a purple labeled neuron with visible blue and red parts might be reduced. This trade-off makes this segmentation task even more challenging one.

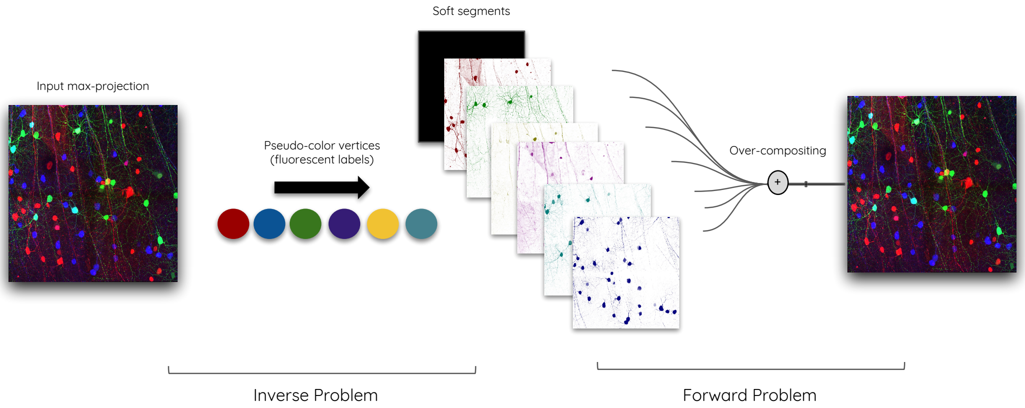


Figure 4: Porter-Duff over-compositing model: Our algorithm accepts user input for number of expected fluorescent labels and automatically computes data convex hull vertices (i.e., color palette). Then it recovers RGBA segments for each label such that when composited by weighted Porter-Duff forward model, the input image is synthesized. This enables us to use weighted Porter-Duff model in a least-squares optimization framework where alpha segments are estimated using a priori regularizers.

To address this challenges, in this work we propose to develop a soft segmentation algorithm that is flexible enough to extract very thin homogeneously labeled neurons. We set the preprocessing as a piecewise image recovery task that is able to balance the trade-off between desired color mixing and extraction of thin structures. For this purpose, we exploit weighted ℓ_1-norm total variation edge-aware flattening where the balance is achieved through weighting of the $TV - \ell_1$ regularization term. Prior to flattening, the images are enhanced by BaSIC background subtraction [23] and linear contrast stretching. Soft segmentation step is carried out by inverting weighted Porter-Duff forward model [24]. Here, the idea is inspired by color manipulation algorithm proposed by Tan et al. [25] where Porter-Duff model is used as the forward model. Inversion is carried by numerical optimization where regularization is designed to achieve optimum soft segments with minimum overlap. As sketched in Figure 4, the algorithm results in alpha layers (i.e pixelwise opacity values) for each of the observed pseudo-color. The proposed method requires no supervision except an user-input indicating number of expected fluorescent expressions (i.e., number of soft segments). Fluorescent images have different statistics than natural reflective images, and we see that previously applied priors and heuristics do not hold for them. For example, representative-based methods to find the dominant color (e.g., k-means, fuzzy c-means Gaussian mixture models) are shown to fail as fluorescent image pixels values are actually distributed on the outer surface of the manifold. In this work, we compute convex polyhedron (i.e., hull) encompassing all colors and find distinct vertices matching user input [25]. This enables automatic discovery of spectral labels even if it only exists in just a single pixel.

We can list the contributions of this work as:

- Our work differs from previous studies in neuron segmentation as they opt for segmenting neurons by using hard boundaries (i.e., only one class attributed to each pixel). To the best of our knowledge, our work also illustrates the first example of soft segmentation applied on 3d multi-channel image stacks.
- We propose the proximal image flattening algorithm that is able to recover piecewise constant version of the input image by using weighted taut-string optimization [26]. The proposed method reaches an operation speed of couple of seconds, and improves the state of the art [27] that operates in a few minutes.
- We modify the standard Porter-Duff equation [24, 25] and propose to invert weighted model where higher inter-class separation is achieved.
- The developed unsupervised post-processing pipeline, which is based on nonlinear morphological operators, results in equivalent performance and speed comparing the state of the art without depending on a trained deep neural network (e.g., [14]).
- Our overall pipeline achieves much better segmentation results than the previous study [1] on viral labeled neuron image stacks.

1.3 Outline of The Thesis

The remainder of the paper is organized as follows: In Chapter 2, we give a detailed overview of previous spectral labeled neuron segmentation methods and numerical optimization which is exploited during development of the algorithms. The proposed pipeline including preprocessing, soft-segmentation and post-processing, that is used to obtain binary masks of neurons, explained in Chapter 3. Chapter 4 gives results and discussion of qualitative analysis applied to real retinabow data and quantitative analysis evaluated on synthetically generated image stacks. Conclusions are drawn in Chapter 5 with proposition of possible future research directions.

1.4 Mathematical Notation

$\Re^n$	n dimensional space of real numbers
Z^n	n dimensional space of integer numbers
I^T	transpose of matrix I
$\nabla f \triangleq \left(\frac{\partial f}{\partial x_1}, .. \frac{\partial f}{\partial x_n}\right)$	the gradient operator on $f : \Re^n \to \Re^n$
$\nabla \mathbf{f} \triangleq \begin{pmatrix} \frac{\partial f_1}{\partial x_1} & .. & \frac{\partial f_1}{\partial x_n} \\ .. & \frac{\partial f_2}{\partial x_2} & .. \\ \frac{\partial f_m}{\partial x_1} & .. & .. \end{pmatrix}$	the Jacobian matrix on $f_i \in \Re^n \mid i \in \{1, m\}$
$\Vert x \Vert_p \triangleq \left(\sum x_i^p\right)^{\frac{1}{p}}$	L_p norm defined on $x : \Re^n \to \Re$
$\Vert x \Vert_\infty \triangleq \underset{i \in 1,n}{Max} \Vert x_i \Vert_1$	L_∞ norm on $x : \Re^n \to \Re$
$\Vert x \Vert_0$	L_0 norm to count number of nonzero elements in $x \in \Re^n$
$\nabla^2 f \triangleq \begin{pmatrix} \frac{\partial^2 f}{\partial x_1^2} & .. & \frac{\partial^2 f}{\partial x_1 x_n} \\ .. & \frac{\partial^2 f}{\partial x_2^2} & .. \\ \frac{\partial^2 f}{\partial x_n x_1} & .. & .. \end{pmatrix}$	the Hessian matrix on $f \in \Re^n$
$prox(x)_{\lambda,f} \triangleq argmin_u(f(u) - \frac{1}{2\lambda}\Vert u - x \Vert_2^2).$	proximal operator of $f(x)$
$TV^{\ell_p}(f) \triangleq \Vert \nabla f \Vert_p$	total variation regularization using L_p norm ($TV - \ell_p$)

2 Background

Image segmentation is one of the core research topics heavily studied in computer vision. The current state of the art in supervised image segmentation (e.g., semantic, instance or panoptic segmentation) is dominated by deep neural networks [28, 29, 30, 31]. On the other hand, in unsupervised segmentation; hierarchical [32, 33], graph [34, 35, 36, 37], clustering [38, 39] and optimization [40, 41] based methods are also proposed in addition to popular deep neural network based approaches aiming unsupervised grouping of pixels [42, 43]. Furthermore, there is a growing interest towards unsupervised low-level processing of images e.g., for contour detection [44, 45, 46, 47] and intrinsic decomposition [27, 48, 49, 50, 51] that targets identification of homogeneous regions of images similar to segmentation. Yet, deep image segmentation still suffers in capturing single-pixel-level details, thereby motivating researchers to employ augmentations such as conditional random fields (CRF) [52] employed in Deeplab [53] (previous leader in PASCAL-VOC segmentation challenge). The inability of convolutional neural networks to adapt pixel-level variations is attributed to their large receptive fields especially present in the final layers. Current research frontier has been pushed to seek alternatives to fully convolutional networks e.g., hypercolumns in [54, 31] which exploits history of each output pixel from previous layers as features to a classifier (e.g., fully connected net). This has been shown to compensate large receptive field of output layers and increase pixel-level details in segmentation tasks.

Soft segmentation has also gained attention during the last decade particularly in the graphics and VFX industry. The main motive driving soft segmentation research is the desire for perceptual color and object manipulation directly in photographs and videos [55, 56, 25, 57, 58, 59]. Similarly image matting research, where opacity values belonging to foreground object is estimated, unveiled sophisticated ideas [60, 61]. Although there have been attempts for fully unsupervised image matting (e.g., [62]), the current state-of-the-art still depends on user-input indicating the location of the foreground object (e.g., trimap, seeds). Also, matting algorithms excel in segmenting a single foreground object, and most of the time not suitable for multiple objects. By allowing more than one class for each pixel, in this work we exploit the flexibility of soft pixel-level decomposition when extracting salient dendritic segments. Our method segments all the neurons belonging to a specific class (i.e., a fluorescent label), and allows multiple layers as opposed image matting where only foreground and background separation is succeeded.

Data-eager nature of deep neural networks and lack of available expert annotated datasets has motivated bioimage computing research community to concentrate more on the development of interactive segmentation frameworks such as Ilastik that has been frequently used for segmentation of transmission electron microscopy stacks [63]. Although there have been previous attempts for automatic segmentation of fluorescent microscopy images, they are mainly based on hard segmentation which is not suitable for segmenting elongated dendritic structures (i.e., failure of hard segmentation to capture fine dendritic details are shown in the results section) or they aim to segment neuron soma or axons, that are larger comparing dendrites, for cytometry and connectomics purposes. In the following sections, previous studies proposing segmentation of spectral labeled fluorescent image stacks and our algorithmic approach, learning by numerical optimization, will be summarized.

2.1 Related Work: Segmentation of Spectral Labeled Neurons

Previous studies on segmentation of spectral labeled neural structures can be divided into three main categories. The focus in all these studies are generally segmentation of neuron soma or axon, and they do not aim morphological analysis of dendrites. These three categories are: Unsupervised segmentation methods designed for segmentation of neuron soma and axons [64, 65, 66, 67, 68, 69, 1, 70], interactive segmentation of brainbow stacks (i.e., spectral labelled hippocampus neurons) [71] and a recent work on data-driven segmentation of brainbow images using a recurrent neural network [72, 73]. In the results section, because of the lack of available codes and relevancy of algorithms, our results will be compared with the results obtained from [1]. Here, methods aiming neural tracing in monochrome images are skipped as they constitute the next step in connectome analysis after sparsification of overlapping neurites.

2.1.1 Unsupervised Segmentation

In [64, 65], Bas et al., propose axon tracing algorithms, first based on piecewise cylindrical model that aims to follow the curvilinear structure of axons starting from a seed point, and a ridge estimation by statistical modelling of the principal curve (i.e., skeleton) [74]. Authors evaluate their methods on synthetically generated axons and a captured brainbow fiber stack with the manually annotated ground truth. Although they present the first processing method for brainbow stacks, their work is limited to tracing of axons and is not generalized to other neural structures. In [67], a simple level set based axon segmentation method is presented. In [68], a Bayesian brainbow segmentation algorithm is proposed that can be applied to extract a single whole-body neuron in a sparse setting. In [69], authors propose a retinal ganglion cell soma segmentation method using active contours and watershed transform. Unlike previous papers, Shao et al., realize the importance of soft mating not to sacrifice highly detailed stratification of dendritic trees [66]. Authors apply morphological operations to smooth homogeneous regions and choose region growing followed by alpha mating to extract overlapping segmented regions. The method is evaluated on a flybow image stack. It should be noted that seeded image segmentation can be interpreted as graph node labeling that also encompasses graph-cut and random-walking as special cases [75]. Methods in this set, compute hard region boundaries via energy minimization and may induce spatial discontinuities while segmenting dendrites (see Chapter 4). In [70], Chen et. al., borrow density functional theory from physical modelling and presents illustrative applications for MRI and brainbow ROI segmentation.

Unlike previous studies, Sumbul et al., propose segmentation of combinatorially expressed spec-

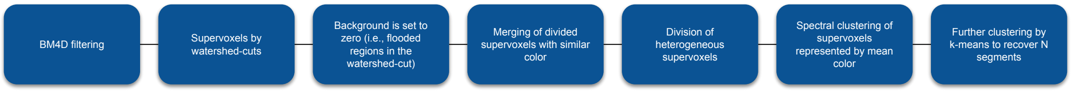

BM4D filtering
Supervoxels by watershed-cuts
Background is set to zero (i.e., flooded regions in the watershed-cut)
Merging of divided supervoxels with similar color
Division of heterogeneous supervoxels
Spectral clustering of supervoxels represented by mean color
Further clustering by k-means to recover N segments

Figure 5: Overview of the algorithm employed in [1]: Authors propose an algorithm based on spectral clustering of supervoxels created by watershed-cuts. Clustering is carried out using the mean color of the supervoxels. At the outset, authors denoise input image stack by patch-based BM4D Gaussian denoising. At the last step, segments are recovered according to user input indicating the number of segments.

tral labeled neurons in a real-world setting. Moreover, they show its applicability on brainbow images captured from mice hippocampus and on a synthetic dataset with corresponding ground truth [1]. Their implementation is publicly available online, which allow us to show the results of the proposed algorithm on our viral labeled retinabow images. The pipeline follows the pattern recognition tradition and consists of several steps applied end-to-end. Firstly authors' denoise image stack by collaborative filtering (i.e., joint transform domain filtering of similar regions found by patch matching). Then, over-segmented version of the stack (i.e., supervoxels) is created by watershed transform; by representing each supervoxel as graph nodes having mean color as the node weight, normalized cuts is carried out to further group supervoxels. To improve segmentation results, authors' split heterogeneous supervoxels and merge if neighbouring voxel color difference is smaller than a predefined threshold. The last step is k-means clustering where a predetermined number of color volumes are quarried from volumes labeled by normalized cuts (Figure 5).

2.1.2 Interactive Segmentation

In [71], Rossien et al., present nTracer, made available as an ImageJ plug-in, that allows interactive tracing of brainbow spectral images (3-4 channels). The idea is to allow the user to specify end-points of neurites then by using a color similarity based flooding technique, close voxels are merged. Similar interactive tracing has been previously applied for monochromatic images in [76].

2.1.3 Supervised Segmentation by Deep Learning

Two recent works employ a data-driven approach and exploit annotated image sets to train deep neural networks for segmentation of spectral labeled neuron image stacks. In [72], a fully convolutional network is trained for segmentation of neuron soma only. Yan et al., [73] on the other hand, trains a recurrent network to attempt segmenting full brainbow branching. To do that they hand-label a large dataset obtained from mice hippocampus.

2.2 Numerical Optimization

In the proposed pipeline, algorithmic components of both preprocessing and soft segmentation is based on mathematical optimization. Therefore, in this section a brief overview of numerical optimization is presented with a special emphasis on the theoretical aspects in addition to the algorithmic details. For classical textbook analysis, Nocedal & Wright [77], Boyd & Vandenberghe [78] and Luenberg et al. [79] can be referred. Here, we only concentrate on continuous deterministic optimization methods as we perform *relaxation* of variables in our image processing and carry out computations with the assumption that pixel value $p_i \in [0,1],\ i \in \{1, N\}$ where N is the number of pixels in the image I. For detailed overview of discrete and stochastic optimization, [80] and [81] can be worth consulting. It should be noted that vector notations (i.e., bold letters or arrow indications) are not used in this section. As we work on N-dimensional variables (images), we assume all variables to be vectors in this context.

Mathematical optimization aims to find optimal values x^* for a set of variables, vectorized as $x \in S$ satisfying constraints C_i such that an objective function $f(x)$ is minimum:

$$x^* \ = \ argmin_{x \in S} f(x) \quad \text{subject to} \begin{cases} C_i^e(x) = 0 \\ C_i^i(x) \leq 0 \end{cases} . \tag{2.1}$$

Here $f(x) : \Re^N \rightarrow \Re$ and $C_i^{\{e,i\}}(x) : \Re^N \rightarrow \Re$ represents the energy function and constraint functions of vector of variables x chosen from a set S. Constraint functions are allowed to be equality or inequality constraints. Working in the discrete domain constitutes solving a NP-hard problem [77], hence we choose to work in the space of real numbers. For our problem, bounded nature of image pixel values $p_i \in [0,1]$ creates a box constraint optimization problem (i.e., 8 bit or 16 bit unsigned integer encoded input image is converted to work with values from the real-valued set $S = [0,1]$). In optimization problems, the objective function can be linear or nonlinear, and for each case methods to obtain a successful optimization result differs. As it will be explained in the methods chapter (Chap. 3), soft segmentation is posed as a nonlinear programming because of the design of the energy function (i.e., here programming denotes optimization, this term has been widely used since the 1940s [77]).

Optimization algorithms are iterative and aims finding $\{\mathbf{x_i}\}_{i=1}^m$ from the initial guess $\mathbf{x_0}$ where it is desired to reach an optimal set of variable values $\{\mathbf{x_i^*}\}_{i=1}^m$ such that $f(x_i^*) < f(x_i)$. There exist no absolute categorization of optimization algorithms in the literature. Also, mostly choice of a method depends on the application at hand. For example, we can classify optimization algorithms according to objective function characteristics (e.g., linear/nonlinear, or zero-order/first-order/second-order according to the availability of gradient and Hessian), according to types of variables considered (e.g., continuous/combinatorial), or according to constraints. In addition to these, there are also cases when the global minimum of an unknown function is calculated (e.g., Bayesian optimization). Energy minimization also formulates the means of learning in graph-based machine learning methods such as Markov random fields (MRF), conditional random fields (CRF) [82] and neural networks [83].

Since complete treatment of optimization algorithms is not feasible, in this section some of the mainstream algorithms related to our optimization problem will be briefly presented. We also give an overview of ℓ_1-norm and total variation (TV) regularized optimization that has become very popular because of its sparsity-enforcing capabilities when solving inverse problems. In the preprocessing step, an ℓ_1 total variation regularized piecewise constant image recovery will be presented. Soft segmentation method, which is inspired by [25], applies L-BFGS-B [84], a version of BFGS (i.e., Broyden–Fletcher–Goldfarb–Shanno) optimization algorithm that allows large-scale minimization of energy functions with simple bounds. BFGS will be presented in the next section as an example of quasi-Newton methods. At this stage, we divide optimization into three main categories for explanation purposes: Optimizing unconstrained smooth functions, constrained smooth and general non-differentiable functions. Although general optimization methods such as proximal algorithms used for non-smooth functions can be also used for smooth functions, very efficient solvers already exist for the latter and they are preferred over the non-smooth solvers.

Optimizing Unconstrained Smooth Functions

Minimization of a smooth function $f : \Re^n \rightarrow \Re$ of real variables $x \in \Re^n$ with no constraints is defined as:

$$\underset{x}{Min} \;\; f(x). \tag{2.2}$$

Since global behaviour of $f(x)$ is unknown generally, optimization aims finding local minimum around a local neighbourhood. But, for the case when $f(x)$ is known to be convex where $f(ax + (1-a)y) \leq af(x)+(1-a)f(y) \mid x \in \Re^n$ and $y \in \Re^n, \; \forall a \in [0, 1]$, one can prove that a local minimum point $\mathbf{z} \in \Re^n$ is also a global minimum:

Proof. To lead contradiction, suppose $z \in \Re$ is a local minimum but not global, then we should be able to find points z^* for a convex function $f : \Re^n \rightarrow \Re$ such that $f(z^*) < f(z)$. Now, consider close neighbourhood of z such that for some small λ, $f(\lambda z^* + (1 - \lambda)z) < f(z)$ is valid due to convexity. Therefore, z actually cannot be a local minimum. □

Convexity plays an important role in optimization research. For example, convex optimization can be used to bound nonconvex problems arising in global optimization [78] in addition to convexification techniques for approximating nonconvex problems [85]. It can be further speculated whether it is necessary to evaluate f for every point $\mathbf{x} \in \Re^n$ to find local minima of $f(x)$. Actually, it is possible to exploit gradient vector ∇f and Hessian matrix $\nabla^2 f$ to deduce the direction to which iterator should move. However, computation of gradient or Hessian might be expensive or not feasible due to noisy data. Therefore, sometimes it can be necessary to approximate some of these operators (e.g., quasi-Newton when Hessian approximated) or change the strategy totally by not using any gradient information at all (e.g., zero order optimization).

As optimization algorithms approach local minima iteratively, direction and step length need to be updated at each iteration. According to the strategies employed to find these two parameters, we can categorize unconstrained optimization methods into two classes: Line search and trust region methods. In this work, we are more interested in line search methods, but trust regions methods have correspondence with line search methods and they require computation of analogues parameters at each iteration. Four main techniques for line search are steepest descent, Newton, conjugate gradient and quasi-Newton methods which also include BFGS algorithm. This categorization is made according to the calculation of search direction where minimization is achieved. Details about these strategies are presented below. For the theoretical derivation of sufficient and necessary conditions for minimizing unconstrained functions, we refer Wright & Nocedal [77].

At this point, It is necessary to state Taylor's theorem as parameters of line search can be derived by using it. Let $f : \Re^n \rightarrow \Re$ be continuous function about a point x_0, Taylor's expansion for the function f can be defined by:

$$f(x) = \sum_n \frac{1}{n!}((x - x_0)\nabla)^n f(x), \tag{2.3}$$

when $f(x)$ is a differentiable function, for some $\nu \in \Re^n$ and $t \in (0, 1)$ theorem reduces to:

$$f(x + \nu) = f(x) + \nabla f(x + t\nu)^T \nu, \tag{2.4}$$

if $f(x)$ twice differentiable, following expression can be derived:

$$f(x + \nu) = f(x) + \nabla f(x)^T \nu + \frac{1}{2}\nu^T \nabla^2 f(x + t\nu)\nu. \tag{2.5}$$

Line search algorithms iterates along a direction p_k with a step length a_k on the surface of $f(x)$ to reach local minima satisfying Wolfe conditions (i.e., sufficient decrease and curvature conditions, see [77]):

$$\underset{a>0}{Min} \;\; f(x_k + a_k p_k), \tag{2.6}$$

thus it is necessary to find optimum direction and step length parameters. Descent methods utilize steepest descent direction, i.e., direction where decrease in f is highest, $-\nabla f$. We can justify this claim by Taylor's theorem written as:

$$f(x_k + ap) = f(x) + ap^T \nabla f_k + \frac{1}{2}a^2 p^T \nabla^2 f(x + tp)p. \tag{2.7}$$

Consider neighbourhood of $f(x)$ at x_k, by using the theorem we can see the fastest change, df/da, for $a \rightarrow 0$ is the solution to the following minimization problem:

$$\underset{p}{Min} \;\; p^T \nabla f_k \qquad \text{s. t. } \|p\| = 1. \tag{2.8}$$

It is straightforward to see that function is minimum when direction p is opposite of the direction ∇f_k. There are many ways to find optimum step length a. A common method is called backtracking where iteratively the best a that achieves sufficient minimization is chosen. Though intuitive, steepest descent may not give desired results and can be very slow. Another way of finding descent direction is by Newton's method. Newton direction is derived from second-order expansion of f by Taylor's series, m_k at a point x_k such that $m_k(p) \sim f(x_k + p)$:

$$m_k(p) = f_k + p^T \nabla f_k + \frac{1}{2}p^T \nabla^2 f_k p. \tag{2.9}$$

Solving for $\underset{p}{Min}\ \{m_k\}$ by setting $dm_k/dp = 0$, one can show that Newton direction p^N at point x_k is:

$$p^N = -(\nabla^2 f_k)^{-1} \nabla f_k. \tag{2.10}$$

The advantage of Newton's method is it exploits second order information in addition to the gradient. Yet, it requires inversion of Hessian matrix $\nabla^2 f$ which may not be available or computational very expensive. To facilitate second order information, yet also to eliminate the need to calculate exact Hessian, quasi-Newton methods are proposed which estimates $\nabla^2 f$ to find search direction.

Quasi-Newton methods aim to approximate Hessian, and let B_k be the approximate following previous iteration, then similar to Newton direction, the quasi-newton direction is written as:

$$p^N = -B_k^{-1} \nabla f_k. \tag{2.11}$$

To ensure approximation behaves like true Hessian, we require *secant condition* derived from Taylor expansion to hold where $s_k = x_k - x_{k-1}$ and $y_k = \nabla_k - \nabla_{k-1}$:

$$B_k s_k = y_k. \tag{2.12}$$

In this part of the section, we briefly focus on BFGS (independently proposed by Broyden, Fletcher, Goldfarb, and Shanno in 1970) quasi-Newton algorithm and its extension L-BFGS (a.k.a limited memory BFGS) [86]. They are one of the most popular unconstrained optimization algorithms utilized. At the soft segmentation step, its enhanced version allowing bounded constrained optimization, L-BFGS-B, will be used to find optimum opacity values for soft segments. Update of BFGS (B_k) and its inverse (B_k^{-1}) is respectively defined by (for the proof see [87, 88, 89, 90]):

$$B_k = B_{k-1} + \frac{y_k y_k^T}{y_k^T s_k} + \frac{B_{k-1} s_k s_k^T B_{k-1}}{s_k^T B_{k-1} s_k}, \tag{2.13}$$

$$B_k^{-1} = \left(I - \frac{s_k y_k^T}{y_k^T s_k} \right) B_{k-1}^{-1} \left(I - \frac{y_k s_k^T}{y_k^T s_k} \right) + \frac{s_k s_k^T}{y_k^T s_k}. \tag{2.14}$$

Pseudocode of quasi-Newton BFGS iterations is provided in Algo. 1. This overall structure is similar to other quasi-Newton line search methods. But as there is still a need to store and process large amount of data, limited-memory versions of these algorithms also appeared that decreases the load in Hessian approximations. L-BFGS is the more efficient version of BFGS where for a chosen $m > 0 \mid m \in Z$, only $m << n$ components of the vector s and y are stored and rest is deleted (Algo. 2). Also, strong Wolfe conditions are set to increase stability [77]. It should be noted that estimating B_k^{-1} at each iteration can be advantageous to skip the computationally expensive matrix inversion step. We also propagate B_k^{-1} instead of B_k during optimization.

Conjugate gradient methods (CG) are an efficient class of minimization algorithms that do not require storage of matrices. In CG, the search direction is defined by following where β is a scalar sustaining conjugate relation between p_k and p_{k-1}.

$$p_k = -\nabla f_k + \beta p_{k-1}. \tag{2.15}$$

Algorithm 1 BFGS algorithm

```
procedure BFGS()
    $B_0^{-1} \leftarrow I$, $k \leftarrow 0$
    while $\|\nabla f_k\| < \epsilon$ do
        $p_k \leftarrow -B_k^{-1} \nabla f_k$
        $x_{k+1} \leftarrow x_k + \nabla p_k a_k$
        $s_k \leftarrow x_{k+1} - x_k$
        $y_k \leftarrow \nabla f_{k+1} - \nabla f_k$
        $B_{k+1}^{-1} \leftarrow$ BFGS update (Eq. 2.14)
        $k \leftarrow k + 1$
    end while
    return $x_{k+1}$
end procedure
```

Algorithm 2 Limited memory BFGS algorithm (m-storage)

```
procedure L-BFGS()
    Choose an $m | n > m > 0$, $k \leftarrow 0$
    Initialize $B_0^{-1}$
    while $\|\nabla f_k\| < \epsilon$ do
        $p_k \leftarrow -B_k^{-1} \nabla f_k$
        $x_{k+1} \leftarrow x_k + \nabla p_k a_k$
        Delete $s_{k-m}$ and $y_{k-m}$ if $k > m$
        Save $s_k \leftarrow x_{k+1} - x_k$
        Save $y_k \leftarrow \nabla f_{k+1} - \nabla f_k$
        Perform BFGS update
        $k \leftarrow k + 1$
    end while
    return $x_{k+1}$
end procedure
```

Optimization with Constraints

Unlike unconstrained optimization, constrained optimization seeks local optima satisfying a set of equality and/or inequality feasibility conditions $c_i(x),\ i \in \{E, I\}$. For problems including only equality constraints, augmented Lagrangian algorithms are known to be the standard way of performing the optimization. However, for the cases with inequality constraints, the problem becomes difficult as it is necessary to check which constraints are active in the constraint set I. Accordingly, carrying out programming with an active subset $S \subset I$ is a smart action such that at optimization stage, S can be taken as equality constraints in the Lagrange multiplier setting. This strategy of solving optimization with inequality constraints is generally employed by algorithms called active-set methods. The more efficient way to optimize this type of problems is by applying nonlinear interior-point algorithms where the optimal solution is obtained by first starting from a point that is far from the feasibility border, and then approaching there (border) iteratively until the solution is obtained

[77]. L-BFGS-B algorithm exploits the active-set strategy when incorporating box-constraints to the L-BFGS formulation given in Algo. 2.

Optimizing Non-differentiable Functions

Minimizing general non-smooth energy functions is not possible by first or second order gradient-based optimization methods. Considering the nonlinear and large-scale nature of visual computing problems that include non-differentiable terms with ℓ_1 norm (e.g., $TV - \ell_1$ denoising, basis pursuit) number of research papers for efficient solvers have multiplied in recent years. One class of algorithms, called Proximal methods [91], paved the way for fast and robust minimization with non-smooth terms. In fact, many of the recent state of the art optimization algorithms can be proven to be the special cases of proximal methods e.g., ADMM (or equivalently known as Douglas-Rachford) [92, 93], split Bregman [94], Pock-Chambolle [95], FISTA [96], half-quadratic splitting [97]. For further reading, we refer to Wotao Yin's lecture notes on proximal operators and algorithms [98].

Proximal operator enables minimization of convex closed functions $f : \Re^N \to \Re \cup \{+\infty\}$ at the *proximity* of x and is defined as the mapping $\Re^N \to \Re^N$:

$$prox_{\lambda f}(x) \triangleq argmin_u(f(u) - \frac{1}{2\lambda}\|u - x\|_2^2). \tag{2.16}$$

As an example, we can derive gradient descent as a special case of proximal update. Consider a function $f(x)$ and its linear Taylor approximation at point x $f_l(x) = f(x) + \nabla^T f(x)(u - x)$. By using proximal operator of $f(x)$ at the vicinity of x, we can write,

$$prox_{\lambda f_l}(x) = argmin_u\left((f_l(x) + \nabla^T f_l(x)(u - x)) - \frac{1}{2\lambda}\|u - x\|_2^2\right), \tag{2.17}$$

As the first-order optimality condition, taking derivative with respect to u where minimum is achieved and $u = prox_{\lambda f_l}(x)$ results in following expression (see [98, 91]):

$$\nabla^T f(x)_l + \frac{1}{\lambda}(prox_{\lambda f_l}(x) - x) = 0, \tag{2.18}$$

here local minimization problem ($prox_{\lambda f_l}(x)$) reduces to gradient descent optimization where λ indicates the step length:

$$prox_{\lambda f_l}(x) = x - \lambda \nabla^T f_l(x)). \tag{2.19}$$

Similar to above described descent derivation, Hessian approximating quasi-Newton algorithms (e.g., Levenberg-Marquardt method) and closed form proximity operators of L_p norms from ℓ_∞ to ℓ_1 (e.g., soft-thresholding or shrink operator in split-Bregman [94]) can also be derived from proximal methods.

Trick of the Trade: L_1 Norm Regularization and Total Variation

Before describing ever-growing interest in ℓ_1 norm based optimization, it is necessary to introduce the concepts of sparse signals and sparse recovery. A sparse signal is a signal with few non-zero elements in a transform domain. This can be explained intuitively considering a natural image where regions with uniform color variation are present. Therefore, by changing representation basis such as by taking wavelet transform, an image can be transformed to a signal with a few non-zero elements which particularly capture salient details (e.g., edges). The simple notion behind sparse signals and recovery of these basis functions that gives sparse representation has far-reaching applications that enable reconstruction with few measurements (i.e., below Nyquist rate). The reason this works very well is the nature of information and the fact that only a portion of the captured/processed signal is sufficient for desired applications.

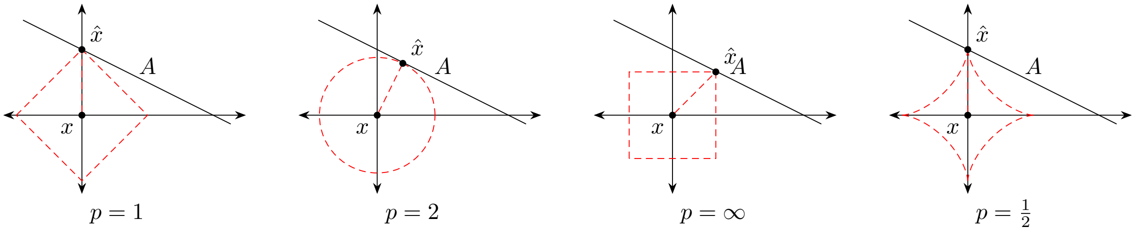
Sparse recovery in numerical optimization can be carried out by ℓ_0 minimization (i.e., minimizing number of nonzero elements). But, this gives a non-convex optimization problem and the solution is known to be NP-hard [99]. Instead, consider minimization of convex ℓ_1 norm in a linear basis pursuit problem which unfolds to:

$$\underset{x\Re^n}{Min}\ \|x\|_1,\ \ s.t.\ Ax = B, \tag{2.20}$$

we can relax the above constrained optimization problem by writing it in the form of least-squares and constraints as penalties:

$$\underset{x\in\Re^n}{Min}\ \|x\|_1 + \lambda\|Ax - b\|_2. \tag{2.21}$$

Now consider below diagram where ℓ_p constrains are visualized for the problem described above. It is striking to realize that $p = 1$ results in the only convex relaxation of ℓ_0 by intersecting orthogonal axis at sparse locations i.e., $x_i = (0, k),\ x \in \Re^2,\ k \in \Re$.



During the last decades, sparsity-inducing capabilities of ℓ_1 norm is exploited in different fields as: Compressed sensing in imaging [100], regularized regression (LASSO with least squares or as LAD-LASSO) [101], Dantzig selector [102] in statistical analysis and total variation (TV) denoising in image processing [103]. At this point, we can also address the high dimensionality of problems consisting of processing of images. Although commonly used by visual computing community, $\|.\|_2$ based optimization actually do not do good for most of the computer vision tasks. Dimension of the solution space for these problems is so high that (e.g., up to 10^8) euclidean distance may not simply capture the desired measure between data points.

Proposed in [103], total variation (or as ROF, Rudin-Osher-Fatemi, regularization in some sources) is a powerful regularization scheme employed in visual computing. It is defined by $TV(x) \triangleq \|\nabla x\|$ where in classical formulation norm is chosen to be ℓ_1. For image processing applications, we can define isotropic ($TV - \ell_2$) and anisotropic ($TV - \ell_1$) total variation formulations by:

$$TV_i(x) \triangleq \sqrt{\sum_{i,j}(x_{i+1,j} - x_{i,j})^2 + (x_{i,j+1} - x_{i,j})^2}, \tag{2.22a}$$

$$TV_a(x) \triangleq \sum_{i,j}\|x_{i+1,j} - x_{i,j})\| + \|x_{i,j+1} - x_{i,j})\|. \tag{2.22b}$$

Availability of methods allowing fast gradient computation makes TV regularization attractive for image processing applications that give outstanding results. TV regularization enforce solutions to have sparse gradient components, thus emphasizing salient features such as edges. Most commonly used data fidelity term used in $TV(x)$ optimization is the least-squares term (i.e., ℓ_2 fidelity) and defined by:

$$\underset{x}{Min} \;\; \|x - y\|_2^2 + TV(x)^{\ell_1}. \tag{2.23}$$

Although the current state of the art in several low-level computer vision problems is achieved by $TV - \ell_1$ norm based optimization. One can also speculate whether they are of the optimal way of solving inverse problems in visual computing. Total variation restricts solution space to only bounded variation images, yet one might fall a pitfall by claiming that natural images are of bounded variation type. In fact, there are hints showing natural images are not actually bounded variation, and it has been shown that total variation of images rockets to infinity as resolution increases [104].

2.3 Confocal Fluorescence Microscopy

Confocal laser scanning microscopy is one the most commonly used imaging techniques in neuroscience. The idea of using a pinhole to block out-of-focus fluorescent light dates back to the 1950's. At that time, this trick has been shown to improve image resolution and allow 3d imaging as the focus-plane could be actually controlled to be able to scan the whole specimen. The main problem was the low amount of light received by the detector. On the other hand, modern confocal microscopes with their embedded laser stimulation and digital preprocessing algorithms made confocal imaging the *de facto* way to capture high-resolution fluorescent signals. In a general confocal imaging setting, laser excitation, focused by a pinhole, is reflected on a specimen using a dichroic mirror. At the point stimulated by the laser, fluorophores emit fluorescence in a longer wavelength than the excitation wavelength. The emitted light passes through the same dichroic mirror and is focused on an image sensor (e.g., CCD, CMOS, sCMOS). The key trick in confocal imaging is the use of pinholes at both excitation to localize laser excitation and at the detected side to eliminate out-of-focus light (Figure 6). In most of the confocal microscopes, band-pass filters are used before the detector in order to block stray light of unintended wavelengths.

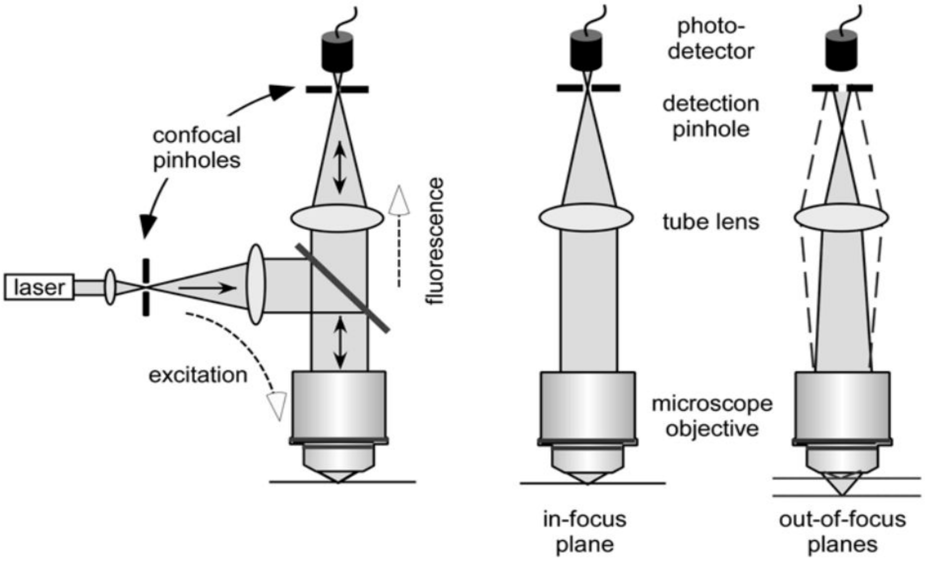


Figure 6: Confocal Fluorescence Microscopy: Illustration of out-of-focus light path blocking. It allows much higher resolution comparing widefield microscopy (image source: [105]).

During the last decade, microscopy imaging had gone beyond the diffraction limit and reached imaging resolution close to nanometer scale (e.g., <50 nm). Some examples of super-resolution microscopy methods are STED, PALM, STORM and SIM. For instance the pioneering work, stimulated emission depletion (STED) microscopy, exploits two laser excitation to selectively deactivate fluorescent emission, hence it is able to resolve point-sources much closer than previously possible. In this work, we employed Zeiss LSM 710 confocal microscopy, and image acquisition was not performed as a part of the thesis work. Viral injections and scanned image stacks were made available by my collaborators Chen Li and Katja Reinhard. Overview of how spectral viral labeling obtained and imaging carried out is briefly explained in A.

3 Proposed Method

The overlapping position of dendrites, when combined with high biological noise, imposes a challenging segmentation task. Although the soft segmentation idea is put forward to optimize extraction of thin structures, preprocessing should be properly designed to balance the trade-off between two of our aims; increasing viral labeling homogeneity (e.g., by means of convolution) and preserving salient structures (e.g., edges). Also, soft segmentation should be designed to add extra robustness to the pipeline so that whole neuron labeled with a single pseudo-color can be segmented without dividing the neuron itself. In this section, our proposed processing pipeline to address these problems is presented. An overview of the pipeline is illustrated in Figure 7. In the below sections, algorithms constituting preprocessing, soft segmentation and post-processing where soft segments are converted to binary layers will be explained in detail.

3.1 Preprocessing

Preprocessing consists of three main steps, and each one is designed to decrease the severity of deterioration observed in fluorescent signal captured by confocal microscopy (Figure 3). As the first step background subtraction is carried out to eliminate undesired biological staining (image D in Figure 3), contrast enhancement is applied to emphasize fluorescent signal of neural components comparing the relatively darker background, and image is sparsified by transforming image into its piecewise constant version to increase homogeneity in the regions belonging to neuron. This is obtained by minimizing a handcrafted energy function that preserves dendritic trees while flattening it. In addition to the three-step preprocessing, it is advised to apply 3d median filtering if the image acquired has high-level impulse noise (e.g., salt & pepper noise). This denoising step will be further discussed in post-processing.

3.1.1 Background Subtraction

Background fluorescent is a common problem in fluorescent confocal imaging. Nonspecific sample staining and autofluorescence are two of the main reasons causing background fluorescent noise [106], and severity of this noise can halt quantitative bioimage analysis and may give inaccurate segmentation results. Spectral viral tracing is an ongoing project and the protocol to optimize neuron-specific fluorescent staining has not been fully established yet. Therefore, it is necessary to augment the soft segmentation idea with a proper removal of undesired fluorescent components. In our pipeline, we utilized ImageJ plug-in of BaSIC background subtraction framework proposed by Peng et al., [23] as it offers an efficient way to remove undesired components. The algorithm is augmented with flatfield, darkfield and baseline correction (i.e., average intensity variation along depth). The algorithm sets the objective of fluorescent signal extraction from noisy multichannel

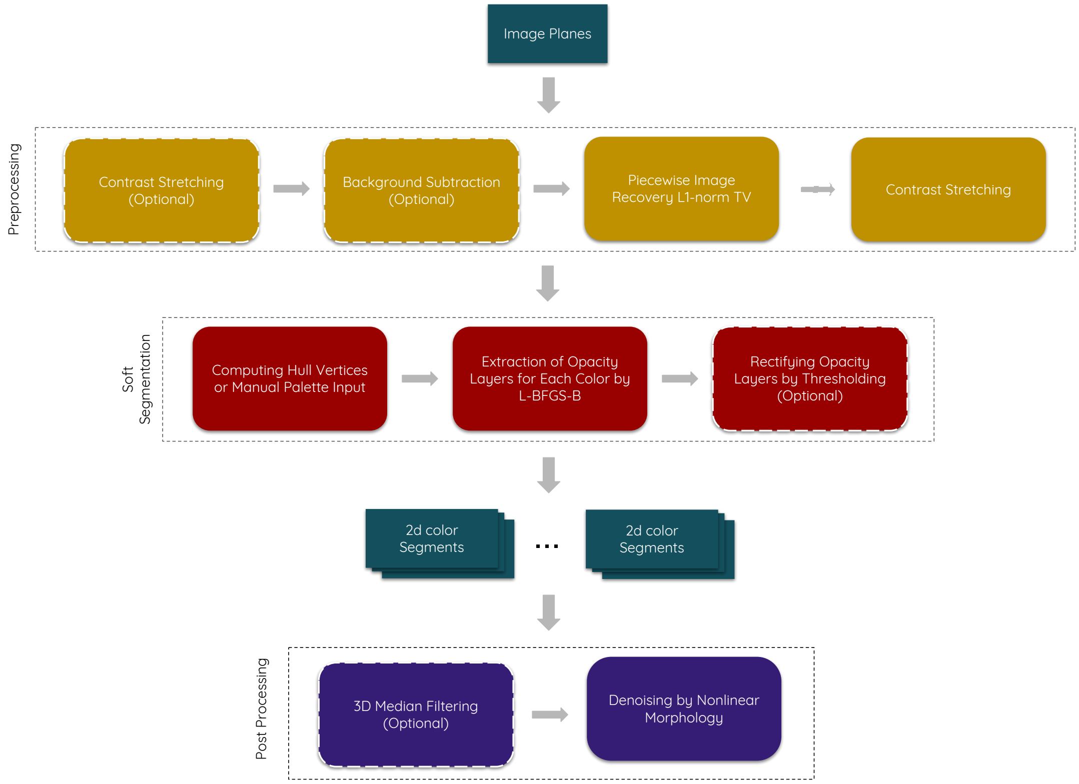


Image Planes
Preprocessing
Contrast Stretching (Optional)
Background Subtraction (Optional)
Piecewise Image Recovery L1-norm TV
Contrast Stretching
Soft Segmentation
Computing Hull Vertices or Manual Palette Input
Extraction of Opacity Layers for Each Color by L-BFGS-B
Rectifying Opacity Layers by Thresholding (Optional)
2d color Segments
...
2d color Segments
Post Processing
3D Median Filtering (Optional)
Denoising by Nonlinear Morphology

Figure 7: Schematic illustrating proposed method. Preprocessing and soft segmentation are carefully designed to preserve information present in tiny and complex dendritic trees

measurements as an ℓ_1-norm based optimization problem, and carries out the solution by linearized augmented Lagrangian method for low-rank representations [107]. Below, the idea behind BaSIC will be briefly explained.

In the original work, authors sort the input image intensity values in ascending order and uses this vectorized version instead. Their key (also dangerous) assumption is that for a large image stack, distribution of foreground fluorescent objects can be assumed to be independent of pixel location. Since we target segmenting densely labeled stacks, the assumption has minimum adverse effects on our processing pipeline. Then, authors model measured intensity values of a sorted image stack $I \in \Re^{m,d}$ at pixel i of the kth plane as follows:

$$I_{i,k} = B_k S_{i,k} + R_{i,k}, \tag{3.1}$$

where $B_k,\ k \in \{1,d\}$ is the constant baseline signal, $S_{i,k},\ i \in \{1,m\},\ k \in \{1,d\}$ denotes flatfield multiplicative shading and $R_{i,k},\ i \in \{1,m\},\ k \in \{1,d\}$ is the residual. The model is further augmented by inclusion of additive dark field component that models dark noise at pixel i:

$$I_{i,k} = B_k S_{i,k} + D_i + R_{i,k}, \tag{3.2}$$

with this model, the task of background subtraction reduces estimation of B, S and D unknowns. Note that splitting D_i into two could be useful here to combine multiplicative scalar (D^z) and additive (D^r) residual components together to form a compact solution:

$$I_{i,k} = (B_k + D_i^z) S_{i,k} + D_i^z (1 - S_{i,k}) + D_i^r + R_{i,k}, \tag{3.3}$$

the sum $(B_k + D_i^z)$ is low-rank, and the problem requires an efficient solver (i.e., solving a low-rank representation is known to be computationally costly). In this problem, authors opt for an augmented Lagrangian based solver [107]. In the solution, they elucidate sparse structure of images at transform domain and pose the estimation as an ℓ_1-norm regularized constrained optimization problem of the form:

$$\underset{B,S,D}{Min} \; \|R^w\|_1 + \lambda_s \|F(S)\|_1 + \lambda_d \|F(D^r)\|_1 + \lambda_d \|D^r\|_1, \tag{3.4}$$

$$\text{subject to } \; I_{i,k} = B_k S_{i,k} + D_i + R_{i,k},$$

where $F(.)$ denotes the Fourier transform and R^w is the element-wise weighted version of R, the weighting done iteratively by setting initial weight to zero and using estimated parameter values at the previous iteration as:

$$R_{i,k}^w \leftarrow \frac{R_{i,k}}{\sqrt{\sum_j (R_{i,k}/B_{i,k})^2 + \epsilon}}. \tag{3.5}$$

The parameter $\epsilon > 0$ has been placed to increase robustness of the solution. The developed algorithm further splits the optimization function into two to decrease computational time. For the details of implementation aspects, we refer supplementary document provided by [23]. Using the package, we further drop darkfield (D) computation and estimate only baseline drift B and shading components S. This choice results in faster computation without losing denoising quality. Also, we have observed that the effect of darkfield quite negligible in our viral labeled images.

3.1.2 Contrast Enhancement

Spoiled by background fluorescent, neuron-specific fluorescent signals appear with low contrast than what is expected in a clean confocal image. Although background subtraction achieves an improvement in signal-to-noise ratio, a contrast enhancement step is also applied in our preprocessing pipeline. Here, we take a heuristic approach and apply linear contrast stretching. We allow some level of saturation in both dark and light pixels, thus amplifying the fluorescent emitted from neural structures. For a pixel $x_i, \; i \in \{1, N\}$ in an image I where N is the number of pixels, $\{p_u, p_l\}$ being the upper and lower percentile of which saturation allowed and n is the bit depth, the contrast stretched pixel value x_i^* is obtained by:

$$x_i^* = (x_i - p_l) \, \frac{2^n}{p_u - p_l}. \tag{3.6}$$

Allowing a certain level of saturation and using percentile rather than maximum and minimum intensity values is an important factor for making the algorithm robust to outliers. Although the order of contrast enhancement is not important, it is observed that applying it before and after background subtraction in an iterative way may improve fluorescent signal quality for some images. That is the main reason why preliminary contrast enhancement is left as an optional step. In our pipeline, we choose lower and upper percentile as 2% and 98% respectively.

3.1.3 Obtaining Piecewise Constant Image

Edge-aware image operations are of the hot topics in low-level computer vision research due to its applicability to wide range of problems from image manipulation (i.e., by means of albedo recovery) to image analysis (e.g., segmentation of desired homogeneous regions) [108, 109, 110, 111]. Natural images have sparse characteristics in a transform domain, and they generally contain set of regions with uniform color and shading variations. However, they also include textural details that play an important role in identification of their unique characteristics. For our problem, we desire to increase pseudo-color homogeneity (i.e., viral diffusion homogeneity) in labeled neurons so that information loss by partitioning of neurons can be avoided. Yet, we also desire to preserve thin dendritic structures and discourage any blurring operation. This poses a challenging task as filtering based approaches, though operate fast, are experimented to be not sensitive enough for such problems as they further blur the input image. Therefore, we take an alternative approach and develop an efficient image flattening algorithm by proximal optimization. We exploit inherent sparsity by means of weighted ℓ_1-norm total variation regularization where pixel-wise weights are designed to plummet intensity variation of homogeneous regions, yet amplify pseudo-color changes (e.g., dendrites). Our proximal image flattening algorithm exploits the strength of non-smooth convex optimization to reach blazing processing speed (i.e., up to sub-second processing speed for a single plane).

We formulate our problem of recovering piecewise flat image $I^f \in \Re^{nxm}$ as minimization of a weighted anisotropic TV function from an input image $I^{in} \in \Re^{nxm}$:

$$\underset{I^f}{Min}\left(\sum_i^n\sum_j^m w_{i,j}\|I^f_{i,j} - I^{in}_{i-1,j}\|_1 + w_{i,j}\|I^f_{i,j} - I^{in}_{i,j-1}\|_1\right), \tag{3.7}$$

$$\text{such that } AI^f = I^{in},$$

We relax the constrained optimization by turning it to minimization of penalized objective function where optimal solution is sought at the proximity of I^{in}:

$$\underset{I^f}{Min}\left(\sum_i^n\sum_j^m \frac{1}{2}\|(I^f_{i,j} - I^{in}_{i,j})\|_2^2 + w_{i,j}\|I^f_{i,j} - I^{in}_{i-1,j}\|_1 + w_{i,j}\|I^f_{i,j} - I^{in}_{i,j-1}\|_1\right). \tag{3.8}$$

As discussed in the Section 2.2, proximal algorithms offer an efficient solution for optimizing functions with nonsmooth regularization and smooth convex data fidelity term. Here, we opt for 1-d taut-string based [112] modular optimization algorithm implemented by Barbero and Sra [26].

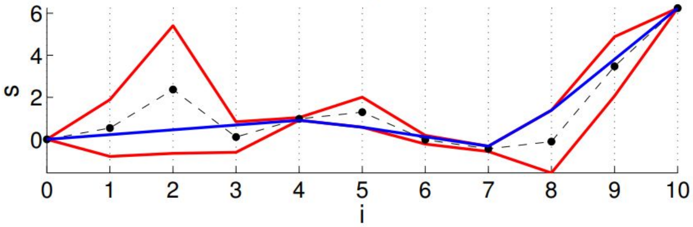


Figure 8: Illustration of taut-string solution for the weighted $TV - \ell_1$ problem: Blue indicates the shortest path (minimizer) of the problem constrained by weights w_i shown by red (image source [26]).

Weighting is the most important factor as it will enable expected flattening of homogeneous regions, but also preserve salient details. Inspired by [27], we assign exponentially high weights for a pixel whose color difference between neighbouring pixels is low, and low weights if that pixel has high color gradient. This enables over-smoothing of pixels with similar colors to compensate assigned high weighting term and preserves pixels with distinct colors by pushing piecewise points even farther:

$$w_{i,j} = exp\Bigg(- \gamma \, \big(\|L^f_{i,j} - L^{in}_{i-1,j}\|^2_2 + \|L^f_{i,j} - L^{in}_{i,j-1}\|^2_2\big)\Bigg), \tag{3.9}$$

where L represents an image in CIELab color space and $\gamma > 0$ denotes scalar weighting term heuristically chosen to be in the range $\{5, 25\}$ after some experimentation. This hyperparameter can be modified according to the image at hand. For example, setting $\gamma = 24$ for confocal stacks and $\gamma = 8$ for natural images was found to be enough to obtain a satisfactory flattening effect. Comparing previous work [27] that is based on split-Bregman iteration, our method improves processing speed up to 50 times while obtaining comparable flattening effect. The reason is the simple geometric interpretation provided by taut-string minimization. This formulation of total variation optimization allows us to simplify our problem and obtain very efficient solvers (O(n)) surpassing state of the art. At this point, weighted taut-string $TV - \ell_1$ minimization algorithm will be briefly explained.

Weighted $TV - \ell_1$ problem with a least-squares data fidelity term can be defined by:

$$\underset{x}{Min} \; \frac{1}{2}\|x - y\|^2_2 + \sum_{i=1}^{n-1} w_i \|x_{i+1} - x_i\|_1, \tag{3.10}$$

where pixelwise weighting scalar $w_i > 0$. In the implementation, dual of this expression is preferred as it allows simpler formulation of the optimization function (see [26] for derivation),

$$\underset{u}{Min} \; \frac{1}{2}\|D^T u\|^2_2 - u^T D y \;\text{ such that } \|u_i\|_1 \leq w_i \; \forall i \mid 0 < i < n, \tag{3.11}$$

where D is the total variation inducing matrix with all zero elements except $d_{i,i} = -1$ and

$d_{i,i+1} = 1 \mid 0 < i < n$. The function is further transformed by setting $u_0, u_n \leftarrow 0$,

$$\underset{u}{Min} \sum_{i=1}^{n} (y_i - u_i + u_{i-1})^2 \ \text{ such that } \|u_i\|_1 \leq w_i, \ \ \forall i \mid 0 < i < n, \tag{3.12}$$

by performing change of variables $s = r - u$ where $r \triangleq \sum_i y_i$ we obtain,

$$\underset{s}{Min} \sum_{i=1}^{n} (s_i - s_{i-1})^2 \ \text{ such that } \|s_i - r_i\|_1 \leq w_i, \ \ \forall i \mid 0 < i < n, \ \ s_0 = 0, \ s_n = r_n. \tag{3.13}$$

Theoretically Eq. 3.13 and the following taut-string formulation can be shown to have the same minima (see [113] for the proof):

$$\underset{s}{Min} \sum_{i=1}^{n} \sqrt{1 + (s_i - s_{i-1})^2} \ \text{ such that } \|s_i - r_i\|_1 \leq w_i, \ \ \forall i \mid 0 < i < n, \ \ s_0 = 0, \ s_n = r_n. \tag{3.14}$$

Taut-string interpretation of the minimized objective in Eq. 3.14 is not intuitive, but it is actually the euclidean length of a piecewise constant line through the points (i, s_i). As a result, this formulation aims finding the minimum length of a string stretching between s_0 and s_n. In Figure 8, visualization of the shortest path (taut-string solution) is shown with blue color whereas red indicates weights constraining the objective function and dashed line as cumulative sum of the input signal (r). At this stage, determination of the shortest line can be performed by applying linear geometric tricks. We refer [26] for details of the geometric implementation. In the results Section 4, effects of our piecewise image recovery on confocal fluorescent stack and natural images will be shown.

3.2 Soft Segmentation

Inspired by Tan et al. [25], soft segmentation is posed as an optimization task where opacity values for each pixel and for each pseudo-color label are estimated assuming Porter-Duff alpha compositing model as the forward model. Optimization task is formulated as ridge (i.e, ℓ_2-norm) and $TV - \ell_2$ regularized least-squares, and it is carried out using L-BFGS-B algorithm [84]. We exploit a fluorescent image *prior* and compute convex polyhedron containing all pseudo-colors, then vertices of this polyhedron progressively contracted to find vertices that correspond to distinct pseudo-colors observed in the image stack. This step requires user input to provide the number of desired (or expected) vertices. The reason why the convex hull based approach is preferred over density-based is due to the prior information we have about the distribution and characteristics of fluorescent data. Unlike images of natural scenes or objects that are dominated by reflectance, stacks obtained by fluorescent imaging posses several challenges from clustering analysis point of view. In the latter, pixels corresponding fluorescent emitting objects are mainly concentrated outer part of the manifold close to the bounding surface (i.e., background noise and biological noise has lower intensity values and is located in the inner regions of the space formed by pixel intensity values). This formation creates non-convex clusters and complicates the unsupervised segmentation task. Furthermore,

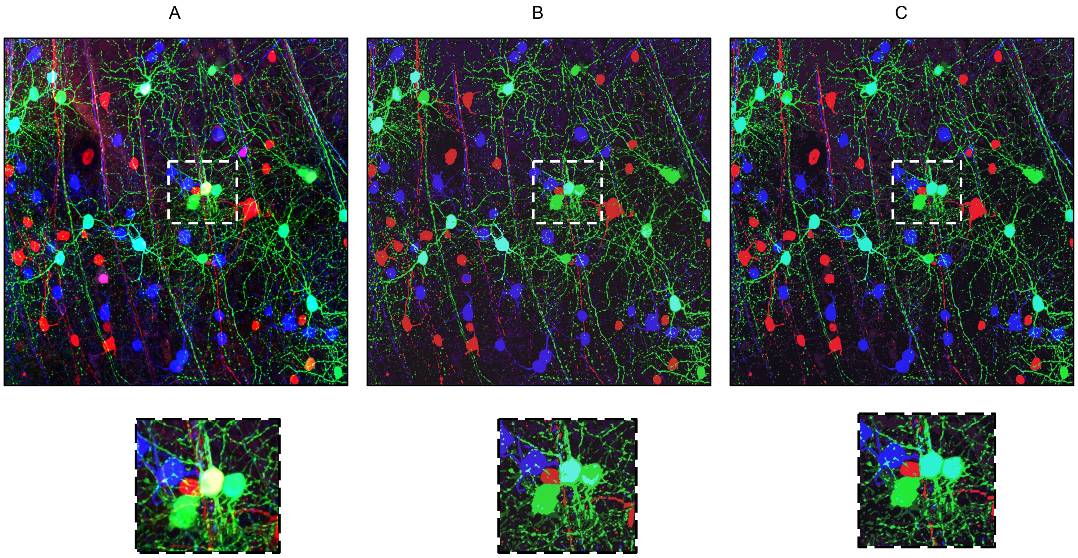


Figure 9: Clustering distinct fluorescence labels: (A) original image, (B) k-means clustering results for k=7, (C) fuzzy c-means clustering results for 7 clusters. Bottom: Close-up views of yellow neuron, for both k-means and fuzzy c-means, it is unidentified due to its low density comparing other pseudo-colors.

as we aim to reconstruct neurons with all possible distinct fluorescent labels, *density bias problem* caused by the unbalanced density of different pseudo-colors becomes another big issue that has to be addressed [39]. As illustrated in Figure 9, the neuron labelled with yellow fluorescent color (A) is not identified by k-means (B) and fuzzy c-means clustering (C). In below sections, we will first describe how distinct pseudo-colors are determined and explain modified Porter-Duff model that is utilized as the forward model in our optimization function. Then we present the details of algorithm employed to obtain soft segments.

3.2.1 Determining Fluorescent Labels: Convex Polyhedron Vertices

The vertices of the polyhedron encapsulating image pseudo-colors in sRGB space, in other words distinct fluorescent labels, are determined by progressive hulls, a mesh simplification technique where mesh edges are contracted to vertices by linear optimization such that new vertex adds minimal volume to the hull at the previous iteration [114, 115]. In our pipeline, we employ the implementation of [25], this formulation of the problem by iterative contraction and merging results in robust and fast simplification of the hull. In addition to its efficiency, vertex-based computation of pseudo color labels ensure we will not miss any fluorescent label. We experimented an extreme case where there was only one pixel present in the in the image for a randomly chosen label. Since the vertices encapsulate all existing colors, a single pixel was enough to discover that fluorescence

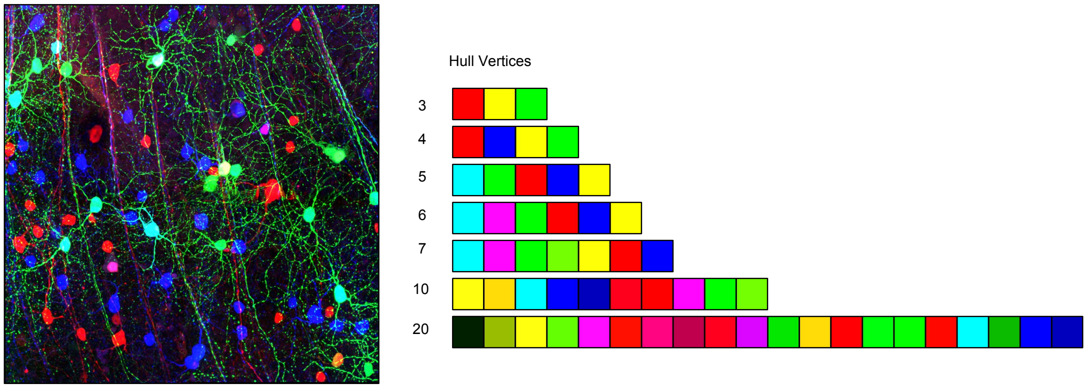


Figure 10: Example results of automatic palette calculation: Pseudo-color labels computed according to user-chosen number from convex hull vertices of the same image. It retrieves vertices of the convex polyhedron that encompass the all data instances.

label. Therefore, we can claim that vertex-based automatic palette computation outperforms mainstream clustering methods such as k-means and fuzzy c-means four fluorescent image processing, and theoretically all distinct pseudo-colors are guaranteed to be revealed. To increase processing speed, data in maximum, minimum and mean projection of the stack is used to find polyhedron vertices. It has been seen that this approach speeds up processing up to ten times. An example pseudo-color palette calculation for different vertex numbers can be seen in Figure 10.

3.2.2 Weighted Porter-Duff Model

Porter-Duff equations are the *de facto* compositing models used in computer graphics. In the original paper [24], author's propose a set of operations based on Lie algebra [116] that can handle compositing of RGBA images satisfying realistic appearance. Here, fourth channel A encodes opacity information (i.e., contribution of that color to the appearance of that pixel). Let $a_i \in [0,1]$ and $c_i \in [0,1]$ be the alpha and sRGB color values of a pixel in ith image, $i \in 1, n$ where n is the number of images to be composited. Assuming the base image used in compositing is opaque with color c_0 (i.e., for our fluorescent images, we assume $c_0 = [0,0,0]^T$), color values of the resulting final image can be obtained by Porter-Duff over-compositing as:

$$c* = c_n + \sum_{i=1}^{n} \left[(c_{i-1} - c_i) \prod_{j=1}^{n} (1 - a_j) \right]. \tag{3.15}$$

Final opacity values of all pixels become unity as we assumed the black background alpha layer to be pure opaque at the outset. In this formulation, our task reduces to the estimation of alpha layers a_i for each discovered pseudo-color i by inverting Eq. 3.15. Let us call each of the combined image chunks (i.e., alpha layers) as soft segments. As stated in [25], for $i > 4$ the problem becomes under-determined and requires an optimization framework to obtain a solution. Before proceeding to solution, we further modify the model to encourage sparsity and penalize overlaps between soft segments by adding an extra weighting term w_i:

$$c* = c_n + \sum_{i=1}^{n} \left[(c_{i-1} - c_i) \prod_{j=1}^{n} (1 - w_j a_j) \right], \tag{3.16}$$

$$\text{such that, } \; w_j = exp\left(- \frac{\sum_{k}^{n-1} \| c_j^{lab} - c_{k=1}^{lab} \|_2^2}{2\sigma} \right),$$

where σ is hyperparameter heuristically set to 2. Our weighting term w_j brings novelty to original Porter-Duff model used by Tan et al., [25] by discouraging overlaps between segments (Note: c_i^{lab} indicates CIELab transformed pseudo-color vertice values). Considering color vertices as objects and ℓ_2 color difference between vertices as pairwise distances, weighting can be considered to mimic gravitation-like force. Affected by this, pixels of any pseudo-color label are pulled and/or pushed in a nonlinear fashion depending on the overall force imposed on them. Consequently, this gravitation-like force creates relatively more separated clusters comparing the original Porter-Duff forward model.

3.2.3 Obtaining Soft Segments by Optimization

Energy function is designed to estimate spatially smooth soft segments without large gradients, and segments with minimum inter-class overlaps. Let E_s be the function to be minimized, optimization for soft segments $a_i : a_i \in [0, 1], i \in \{1, n\}$ is formulated as:

$$\underset{a_i}{Min} \left(\lambda^{ls} E_{ls}(c_i, a_i) + \lambda^{ridge} E_{ridge}(a_i) + \lambda^{\ell 2-TV} E_{\ell 2-TV}^2 (a_i) \right), \tag{3.17}$$

where λ^* indicates scalar weight factors, and data fidelity term for least squares Porter-Duff model is denoted by E_{ls}; E_{ridge} and $E_{TV-\ell 2}^2$ are ridge regularization and total variation term penalizing large spatial gradients. These energy terms are defined by:

$$E_{ls} \triangleq \frac{1}{3} \left\| c_{in} - c_n + \sum_{i=1}^{n} \left[(c_{i-1} - c_i) \prod_{j=1}^{n} (1 - w_j a_j) \right] \right\|_2^2 \tag{3.18a}$$

$$E_{\ell 2-TV}^2 \triangleq \frac{1}{n} \sum_{i=1}^{n} (\nabla a_i)^2 \tag{3.18b}$$

$$E_{ridge} \triangleq \frac{1}{n} \sum_{i=1}^{n} (a_i)^2 \tag{3.18c}$$

Since the problem is ill-posed and there exist infinitely possible solutions if no restrictions are set, we require regularizations complementing least-squares data fidelity term. In the energy function, ridge and squared ℓ_2-norm total variation regularization are chosen as the regularizers. Although we had several attempts for ℓ_1-norm based regularization, solving the formulated large-scale nonlinear and non-smooth energy function turned out to be cumbersome and unstable. Thereby motivating

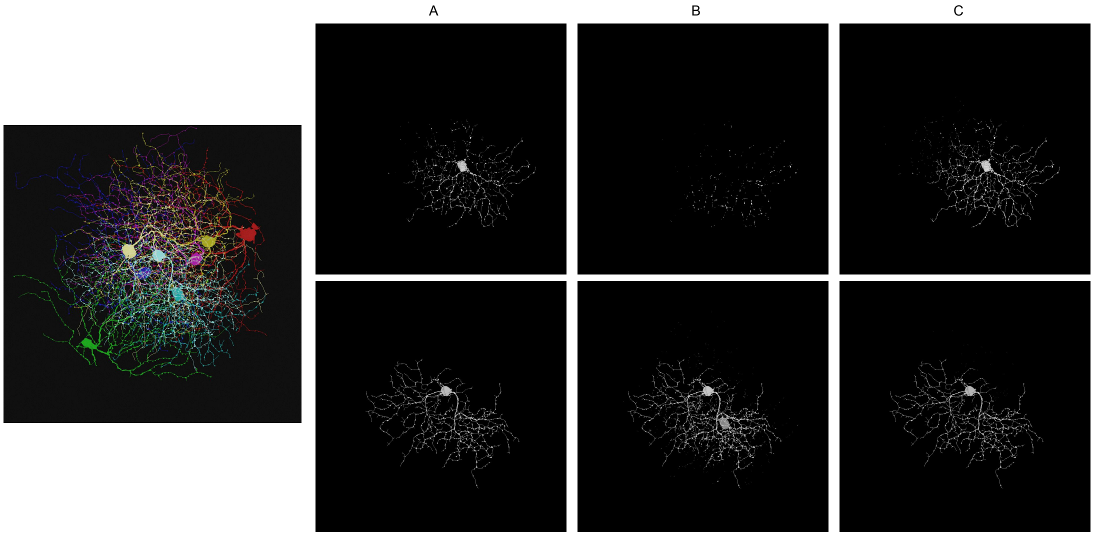


Figure 11: Effects of hyperparameters on two soft segments recovered from the synthetic image shown on left: (A) weights $\lambda^{ls} : 40,\ \lambda^{ridge} : 2,\ \lambda^{TV^{\ell_2}} : 20$, (B) $\lambda^{ls} : 40,\ \lambda^{ridge} : 10,\ \lambda^{TV^{\ell_2}} : 20$, (C) $\lambda^{ls} : 40,\ \lambda^{ridge} : 2,\ \lambda^{TV^{\ell_2}} : 1$. It is clearly visible that increasing weight of ridge regularization results in unbalanced segments, on the other hand, decreasing $\lambda^{TV^{\ell_2}}$ encourage higher gradients. Computation times also varies according the choice of weights. In the case (B), processing speed is twice faster, and in (C) twice slower.

us to employ convex and smooth regularization schemes. Ridge (i.e., Tıkhonov) is a widely used regularizer discouraging over-fitting in the optimization step. Since the posed task is unsupervised, ridge regularization outputs inherently bounded segments and discourages overlaps between them. On the other hand, squared $TV - \ell 2$ term penalizes high gradients and creates smooth segments. It also enables natural inpainting and assists merging of divided segments. We should note that scalar weighting (λ^*) hyperparameters play a very important role in optimization and machine learning tasks. To ensure successful recovery of soft segments, we reserve several z-stacks as the validation set for manual tuning of λ^* values. After our experiments, we found that, $\lambda^{ls} = 40$, $\lambda^{ridge} = 2$ and $\lambda^{TV-\ell 2} = 20$ values gave satisfactory results. An example figure showing the effects of weighting hyperparameters can be viewed in Figure 11.

Due to relaxation of pixel values, the solution can be posed as a bound constrained nonlinear optimization problem where pixel values are restricted to [0, 1] range. As in Tan et al.'s [25] implementation, active set based L-BFGS-B (i.e., limited-memory and bound-constrained BFGS) is employed in our optimization task [84]. Similar to other quasi-Newton algorithms, L-BFGS-B does not require the Hessian matrix, but rather an approximation B_k is used and updated in each iteration. Previously explained L-BFGS [117] for large-scale optimization is central in L-BFGS-B as well. L-BFGS-B is an extended version of original work to allow optimization of variables x^n with bound constraints of type $l^n \leq x^n \leq r^n$. Original implementation written in FORTRAN is provided by the authors [118], moreover several python wrappers can be also found in scientific computing libraries such as Scipy.

Pseudo-code of BFGS and L-BFGS quasi-Newton approximation were explained in Section 2.2. At this point, we can briefly provide the details of L-BFGS-B algorithm that exploits BFGS iteration to approximate Hessian matrix. It should be noted that applications of L-BFGS-B are not limited to constrained optimization, and it can be used to minimize constrained problems without worrying about the active set of variables. Referring the quasi-Newton update obtained by Taylor expansion at Eq. 2.5, function $f(x)$ around the neighbourhood of x_k can be modeled by:

$$m_k(x) = f_k + (\nabla^T f_k)(x - x_k) + \frac{1}{2}(x - x_k)^T B_k (x - x_k), \tag{3.19}$$

where B_k denoted limited-memory Hessian approximation of BFGS. Here L-BFGS-B algorithm steps can be given as:

1. Predefined tolerance condition is controlled. Let $proj(.)$ denote gradient projection defined as:

   $$proj(x, l, u) \triangleq \begin{cases} l_i, \ x_i < l_i \\ u_i, \ x_i > u_i \ , \\ x_i, \ o.w. \end{cases} \tag{3.20}$$

   then iteration is stopped if:

   $$\|proj(x_k - \nabla f_k, l, u) - x_k\|_\infty < Tol, \tag{3.21}$$

   where Tol indicates the predefined tolerance value given by user input. In our case, we found Tol in the range of $[10^{-4}, 10^{-5}]$ sufficient to reach the desired outcome.
2. Cauchy point is computed as described in [84]. CP is used to determine active and free variables among x that restrict search space.
3. Subspace minimization and strong-Wolfe line search: m_k is minimized in an unconstrained fashion for the determined free variables. Search direction is computed using conjugate gradient and step length is determined using backtracking to satisfy the strong-Wolfe condition (see [77]).
4. Limited memory BFGS Hessian approximation (B_{k+1}) is computed by storing first m variables, then the algorithm continues to the next iteration.

After determination of pseudo-color vertices, segments for each vertex are estimated sequentially. In other words, we recover segments from each 2d color plane in z-stack one by one. This choice is due to computational restrictions. Considering the size of the viral labeled images used in our experiments, minimizing approximately 10^8 variables took its toll and caused memory errors. Since the vertices are globally determined, this actually brings several advantages from practical point of view. Firstly, it is possible to check the results while the computation is in progress. Secondly, it gives the possibility to segment only from specific planes chosen by the user. Since pseudo-color palette is computed globally, proposed plane-wise segmentation does not skip any fluorescence label.

3.3 Post Processing

After recovery of soft segments, one might require binary masks of the actual neuron depending on the application at hand. For the case of image-based identification of neurons from its dendritic arborization, it is desirable to have a denoised 3d reconstruction where pixels not belonging to neurons are masked out of the stack. Unlike previous works employing a trained deep network to augment nonlinear morphological processing [14], here we develop an end-to-end framework using only nonlinear mathematical morphology with only a few hyperparameters that need tuning. This enables fast processing of the whole stack in a few seconds.

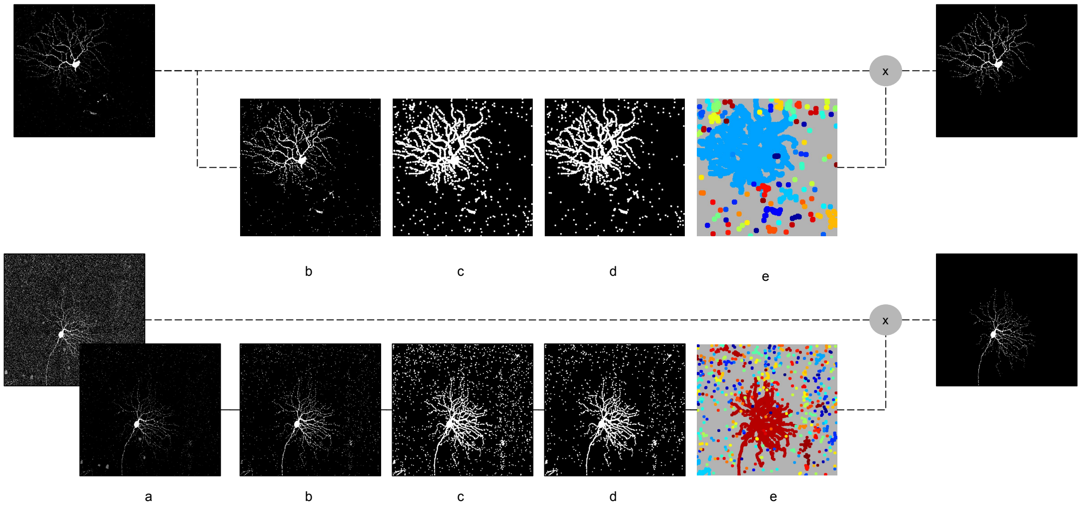


Figure 12: Step by step post processing algorithm: (a) optional median filtering, (b) dilation, (c) applying distance transform and combining connected components closer than a predefined distance threshold, (d) preserving N largest components and removing smaller ones, (e) relaxing distance transform threshold and merging components to create the large mask. Lastly, the largest connected component is used as a mask to segment hard boundaries.

As the first step, a 3d median filter is applied to remove impulse noise. This step is optional and the user is asked to decide whether to filter the stack or not. We observed that median filtering actually adversely affects images that are not spoiled by impulsive noise. Therefore this decision is set as a hyperparameter in our algorithm. We also observed filtering applied to a local neighbourhood $hxhxh$, $h \in \{1, 3\}$ gives enough satisfactory results. As the second step, each voxel is dilated using a $3x3x3$ structural element, and then using distance transform connected components closer to 3 voxels are merged. At this point, it is necessary to remove relatively small objects, thus largest N

Parameter Names	Explanation
w_fidelity_lsl2	weight of L2-norm least-squares data fidelity term
w_ridge	weight of ridge (Tikhonov) regularization
w_tvl2	weight of L2-norm total variation regularization
threshold_opacity	clips opacity values below the threshold to 0 before saving
stack_path	directory containing input stack
output_path	directory to save results
automatic_color_vertices	set 1 to compute pseudo-color palette, 0 to manually indicate
number_soft_segments	number of output segments
manual_vertices	if automatic computation set 0, indicate pseudo-colors
drop_color	ignore one of the vertices when computing soft segments
FAST	disable weighting term in the optimization function
SAVE_COLOR	also saves colored versions of soft segments: 1 or 0
start_plane	compute segments from substack starting from, set -1 to disable
end_plane	compute segments from substack until, set -1 to disable
level_contrast_enhancement	within 1, 10, choose -1 not to enhance contrast
level_flattening	choose a value $>$ 1, you may get assertion error if too high
iterations_flattening	flatten several times (e.g., 2-4)

Table 1: List of parameters that can be directly changed from the json file

connected component is preserved and others are simply set to zero. This N is the second parameter that needs user input. Later in our experimentation, we found out that N in the range $\{500, 800\}$ performs equivalently well, without causing much distortion. Since our aim is to create a mask, we relax our distance transform threshold this time and combine connected components smaller than 8 voxels. This over-dilation will enable preservation of the complete dendritic tree without causing any division. As the last step, we take the largest connected component obtained at the previous step and apply it as a mask onto the original stack to recover the full branching of the neuron. It is striking to realize that applying nonlinear mathematical operations to obtain a mask actually results in more accurate and robust reconstruction than applying the pipeline directly on the image stack. Step by step illustration of the process is also illustrated in Figure 12.

3.4 Implementation

During implementation, we exploit advantage of our Debian based operating system (Ubuntu 16.04) and create a Conda python environment that includes the whole soft segmentation pipeline except BaSIC background subtraction and post-processing Matlab script. We use original BaSIC ImageJ plug-in that gives fast background subtraction (i.e., approximately 5 min for an image stack of size 150x1000x1000x3). Post-processing is implemented in Matlab by using mathematical morphology functions in the image processing toolbox. To fully exploit its speed advantage, we recommend using a Matlab version higher than R2016a (i.e., processing takes approximately 1 min for an image stack). The software is intended to be user-friendly and open source so that it can be explored freely by neuroscientists. Full code including the external ImageJ BaSIC plugin is publicly available at github.com/elras/SoftsegmentNeurons and github.com/farrowlab/Neuron-Softsegmentation. For easy modification of parameters, we decided to use JSON data interchange format. Since parameters are parsed from "params.json", it may not be even necessary to modify the code itself. In the Table 1, full list of of parameters included in the json file and corresponding explanations are provided.

4 Results and Discussion

In this chapter, experiments designed to evaluate the proposed method, the outcome of these experiments, and discussion of the results are presented. Three types of experiments are carried out. Firstly, quantitative analysis on two synthetically generated image stacks with the corresponding ground truth is computed, and our method is compared with the state of the art work on brainbow volume segmentation [1]. Then, we move to qualitative experiments and apply our method to real image datasets in addition to the synthetic stacks. In this section, full reconstruction of neuronal branching from several confocal microscopy stacks hindered by severe noise is presented. Also, several partitioning-based clustering algorithms, k-means, fuzzy c-means and meanshift is compared against ours using a highly overlapping 2d plane (i.e., the maximum projection of a stack is used as the input image). Complementing the analysis of soft segmentation, we also show results of each step in the pipeline to illustrate effects of prepossessing. Before going through the experiments, below we provide mathematical definitions of similarity metrics employed to measure the accuracy of compared algorithms.

4.1 Metrics

Quantitative evaluation is performed using three standard metrics employed in image segmentation: Probabilistic Rand index (PRI) [119], variation of information (VI) [120] and Dice coefficient ($DICE$). Details of the evaluation metrics are given below.

Probabilistic Rand Index

Probabilistic Rand index measures consistency between two segmentation. Let l_i and l'_i two set of label assignments from the sets S and $S'$for a pixel $i \in \{1, N\}$, the index can be defined by:

$$PRI(S,S') \triangleq \frac{1}{\binom{N}{2}} \sum_{i,j \ | \ i \neq j} \Big[I(l_i = l_j) I(l'_i = l'_j) + I(l_i \neq l_j) I(l'_i \neq l'_j) \Big], \tag{4.1}$$

where $I(.)$ is the identity function [119]. Higher the PR value is the closer the segmentation to the ground truth.

Variation of Information

Variation of information is an information theoretic dissimilarity metric where the smaller value indicates better segmentation. The idea is based on computing conditional entropy of one segmentation given the other one. Let $H(S)$ denote entropy of the label set S with corresponding probability distribution function P_S, VI can be defined by [120]:

$$VI(S,S') \triangleq H(S|S') + H(S'|S), \tag{4.2a}$$

$$H(S|S') \triangleq -\sum_{L} P_{S|S'} log(P_{S|S'}), \tag{4.2b}$$

where L is the set of distinct pixel labels, or equivalently using joint entropy:

$$VI(S,S') \triangleq H(S) + H(S') - 2H(S,S'), \tag{4.3a}$$

$$H(S,S') \triangleq -\sum_{L} P_{S,S'} log(P_{S,S'}), \tag{4.3b}$$

Sorensen-Dice Coefficient

Sorensen-Dice coefficient, or simply Dice coefficient, is a similarity metric measuring the level of match between computed and provided ground truth labels. It is also known as F1 score in statistical analysis and defined by:

$$Dice \triangleq \frac{2TP}{2TP + FP + FN}, \tag{4.4}$$

where TP, FP and FN is used to denote true-positive, false-positive and false-negative labeling respectively.

4.2 Quantitative Experiment Results

Metrics		Evaluation for Exp-1 (Our Method)					
	S1	S2	S3	S4	S5	S6	Mean	Std
PRI	0.9926	0.9910	0.9799	0.9978	0.9960	0.9582	0.98	0.01
VI	0.0355	0.041	0.0819	0.0122	0.0207	0.1490	0.056	0.05
DICE	0.9981	0.9977	0.9529	0.9949	0.9995	0.9990	0.99	0.01
Metrics		Evaluation for Exp-1 (Sumbul et al.)						
	S1	S2	S3	S4	S5	S6	Mean	Std
PRI	0.7597	0.7726	0.7406	0.7526	0.7471	0.7285	0.75	0.01
VI	0.1968	0.3971	0.4423	0.3492	0.2573	0.7608	0.4	0.2
DICE	0.3561	0.3511	0.3682	0.3287	0.3351	0.3511	0.34	0.01

Table 2: Evaluation results for 6-neuron segmentation task computed using PRI, VI and DICE metrics. Our method is shown to clearly outperform the algorithm presented in [1].

To evaluate the performance of our method, two synthetic image stacks are generated with 6 and 8 neurons with distinct fluorescent labels. Each has dimensions of 200x1024x1024x3, and is rendered using sRGB color space. Images are created using sparse viral labeled monochrome images that are manually traced from previous confocal stacks. Traced neurons are heuristically colorized (i.e., by using 6 and 8 distinct pseudo-colors) and converted to RGBA where intensity values are

Metrics	Evaluation for Exp-2 (Our Method)									
	S1	S2	S3	S4	S5	S6	S7	S8	Mean	Std
PRI	0.9855	0.9565	0.9935	0.9995	0.9911	0.9831	0.9794	0.9999	0.98	0.01
VI	0.0630	0.0565	0.0342	0.1256	0.0414	0.0771	0.0899	0.0002	0.06	0.03
DICE	0.9963	0.9970	0.9984	0.9926	0.9978	0.9958	0.9948	0.9999	0.99	0.002
Metrics	Evaluation for Exp-2 (Sumbul et al.)									
	S1	S2	S3	S4	S5	S6	S7	S8	Mean	Std
PRI	0.7613	0.8119	0.7328	0.7701	0.7531	0.7598	0.7663	0.7843	0.76	0.02
VI	0.1969	0.3348	0.4100	0.2748	0.3165	0.2902	0.3024	0.2775	0.3	0.06
DICE	0.3376	0.3073	0.3464	0.2987	0.3279	0.3705	0.3287	0.2906	0.32	0.02

Table 3: Evaluation results for 8-neuron segmentation task computed using PRI, VI and DICE metrics. Similar to the experiment with 6 neurons, our method successfully extracts segments and results are quite close to the ground truth locations.

randomly assigned for each layer from a range of values. Subsequently, they are merged using Porter-Duff model where fixed opacity value 0.8 is assigned to each segment. To simulate realistic imaging conditions, we also added zero-mean Gaussian noise whose standard deviation value is chosen after visual inspection. Segmentation results can be viewed in Table 2 for 6-neuron case and in Table 3 for 8-neuron case. We also visually illustrate maximum projection of results in Figure 13 for 6 segments and Figure 14 for 8 segments. In these two figures; GT, S and O indicate the maximum projection of ground truth, Sumbul et al.'s [1] and our segmentation respectively. Columns correspond to recovered neurons with distinct spectral labels. We show that our soft formulation of the segmentation task clearly outperforms the algorithm proposed in [1] for both 6 and 8 neuron cases.

4.3 Qualitative Experiment Results

In this part, enhancing effects of background subtraction and image flattening are illustrated on captured viral labelled neuron image stacks. We also apply classical clustering algorithms, k-means, fuzzy c-means and meanshift on the maximum z-projection of synthetically generated image stack with 8 neurons. This creates a quite challenging task with highly occluded segments. Our aim is to show the capabilities of TV and ridge regularization, and illustrate the proposed method actually performs better than traditional clustering algorithms on the formulated 2d image segmentation task. At the end of this section, results for full 3d reconstruction of confocal neural trees labelled with pseudo-green, orange, cyan, magenta and yellow colors are also presented. From these, green, orange and cyan neurons are successfully reconstructed, but results for the remaining neurons are found to be less satisfactory. Binary masks for all these neurons are obtained by the designed post-processing pipeline.

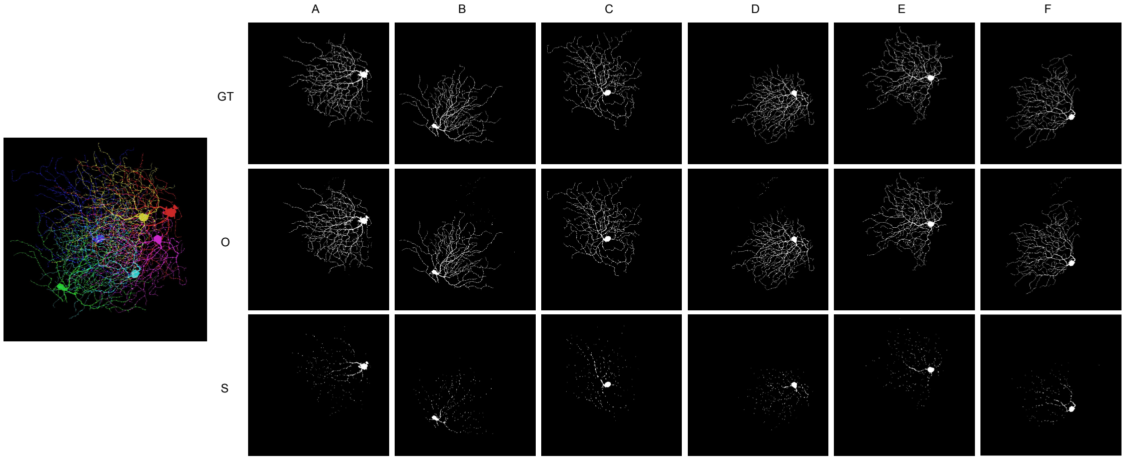


Figure 13: Soft segmentation results for the experiment with 6 neurons: On the left max-projection of the generated stack, (GT) max projection of ground truth, (O) max projection of results of our method, (S) segmentation results of [1] for each neuron.

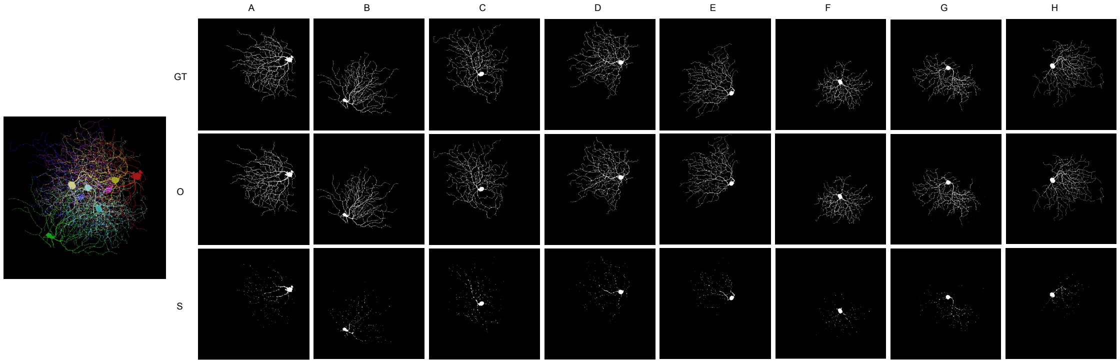


Figure 14: Soft segmentation results for the experiment with 8 neurons: On the left max-projection of the generate stack, (GT) max projection of ground truth, (O) max projection of results of our method, (S) segmentation results of [1] for each neuron.

4.3.1 Background Subtraction

In figure 15, maximum projections of red and blue channels of a stack are shown before (A) and after background subtraction (C) applied. In column (D), the computed mean x/y plane flatfield intensities, and baseline signal along the z-direction are plotted. Since intermediate spectral channel (i.e., green channel) has less deterioration due to noise, it was not even necessary to apply background subtraction. Column (B) shows results after processing of traditional rolling-ball algorithm

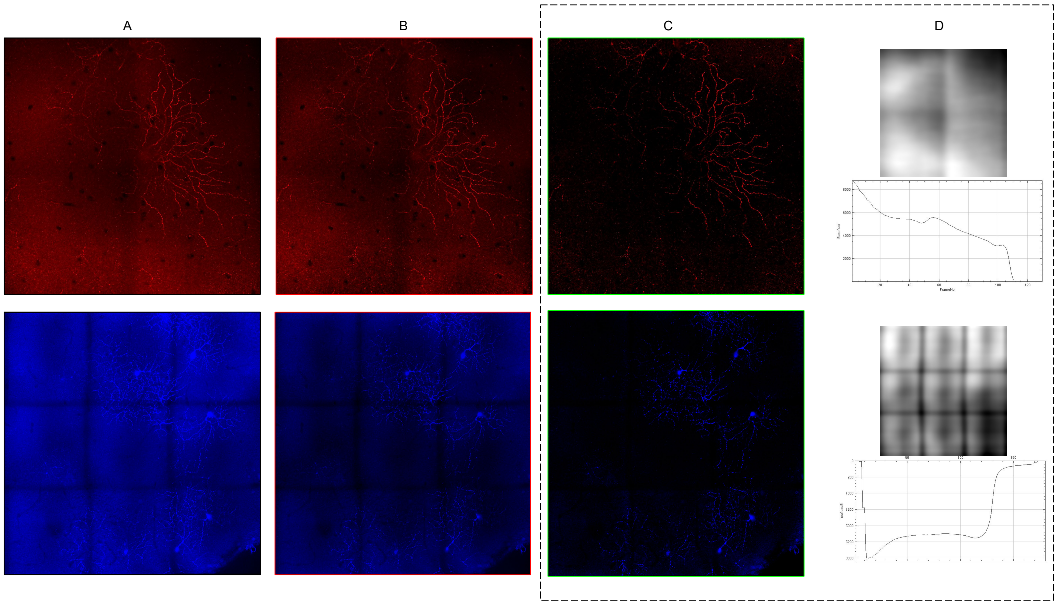


Figure 15: Effect of background subtraction on a sample image stack: (A) An image plane from the input stack for the first and last spectral channels, (B) results of sliding paraboloid ImageJ plugin [121], (C) corresponding channels after BaSIC background subtraction, (D) computed average flat field intensities for x/y plane and baseline signal along z direction.

(implemented in ImageJ) [121]. As can be seen, It is relatively less robust to noise and background removal is not as effective as BaSIC.

4.3.2 Piecewise Constant Image Recovery

Although background subtraction decreases bio-induced noises such as autofluorescence and unintended staining of glia, it does not offer any solution for non-uniform diffusion of viral vectors into neurons. As explained previously, additional edge-aware flattening is carried out and this task is set as piecewise image recovery by weighted L1-norm total variation optimization. In Figure 16, the effect of our piecewise image flattening is shown on a sample image plane. Figure 17 shows plots for the intensity profile along the yellow line for the blue channel of the image plane shown in Figure 16. Lastly in Figure 18, we show results of our flattening on natural images retrieved from Berkeley segmentation benchmark [122]. The preservation of salient details and cartoonization of homogeneous regions is strikingly exposed in these figures.

4.3.3 Comparison with Image Clustering Techniques

Previously, we compared our algorithm with [1] using two synthetically generated image stacks containing 200 color image planes. Here we increase the difficulty and compare our method with k-

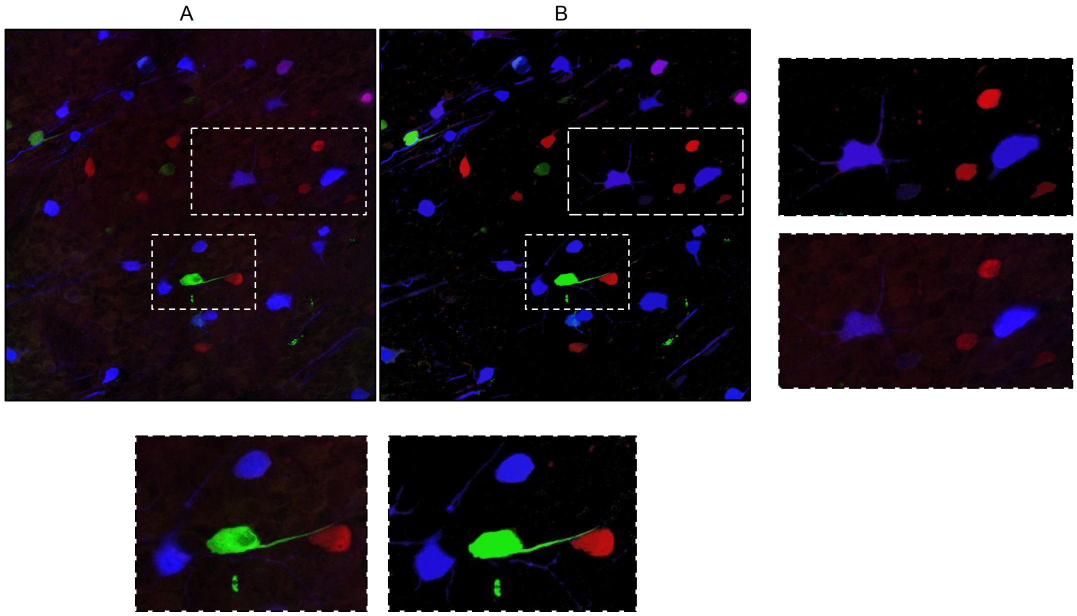


Figure 16: Flattening a viral labeled image plane before (A) and after (B) processing. Two example windows are highlighted to show details of flattening.

means, fuzzy c-means and meashift using an extrame image, the maximum projection of the image stack with 8 neurons. Therefore, we turn our task into segmentation of a densely overlapping 2d image plane. In Figure 19, results can be inspected for each algorithm where columns represent distinct neurons: (A) segmentation results for k-means, (B) fuzzy c-means, (C) meanshift and (D) for our method. As clearly shown, only our method is able to segment all distinct pseudo-color labeled neurons.

4.3.4 Full Neuron Reconstruction

An automatic post-processing algorithm is also carried out to create masks form soft segments. In this section, successful post-processing of three neurons labelled with green, orange and cyan (Figure 20) are shown. In figure 21, we also illustrate segmentation of magenta and yellow neurons. Their full branching is not recovered due to the severe deterioration in staining.

4.3.5 Computation time

Although real-time segmentation and reconstruction are not aimed, improving speed constraints of previous works was a central idea in the project. Therefore, in each step, we avoided heavy computations and employed vectorized algebraic data operations without loops. In the preprocessing step, rather than applying computational expensive non-local collaborative filtering approaches

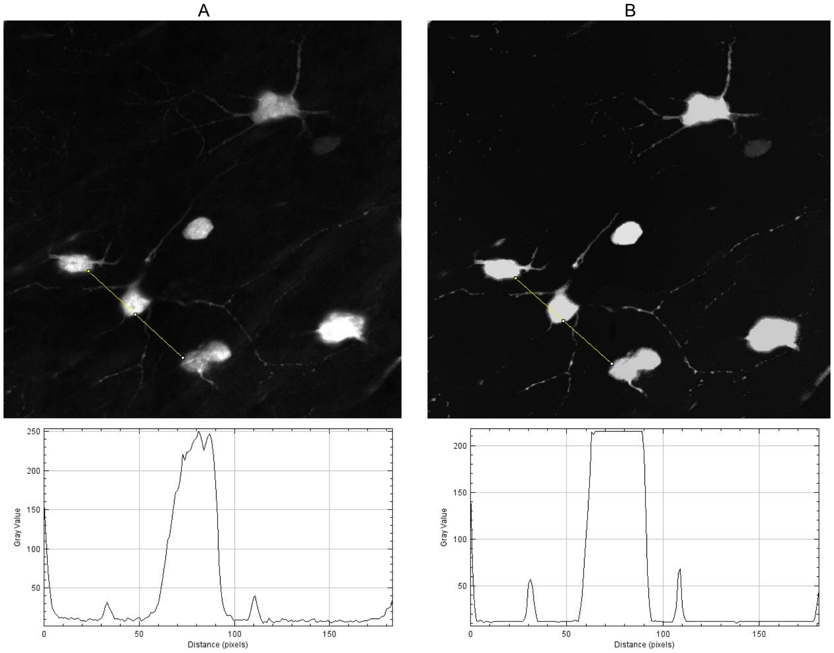


Figure 17: Illustration of flattened profile: (A) a local region clipped from blue channel, (B) resulting piecewise-flat intensity profile for the line indicated by yellow color.

that depend on patch-matching, we employed an efficient proximal algorithm when solving ℓ_1-norm optimization. In Table 4, the approximate time taken by each step in the pipeline is tabulated and compared with the previous work. We use a real image stack with dimensions 150x960x960x3 when carrying out the test.

Steps	Appr. Time (min)		
	O	S1	S2
Preprocessing	10	360	10
Segmentation	500	2400	200
Post-processing	2.5	-	-
Total	512.5	2760	210

Table 4: Comparison of approximate processing times for (O) our algorithm, (S1) algorithm in [1] and (S) results of [1] after replacing the preprocessing by our preprocessing. As a result, our preprocessing increases processing speed of [1] by more than 10 times without negatively effecting its results.

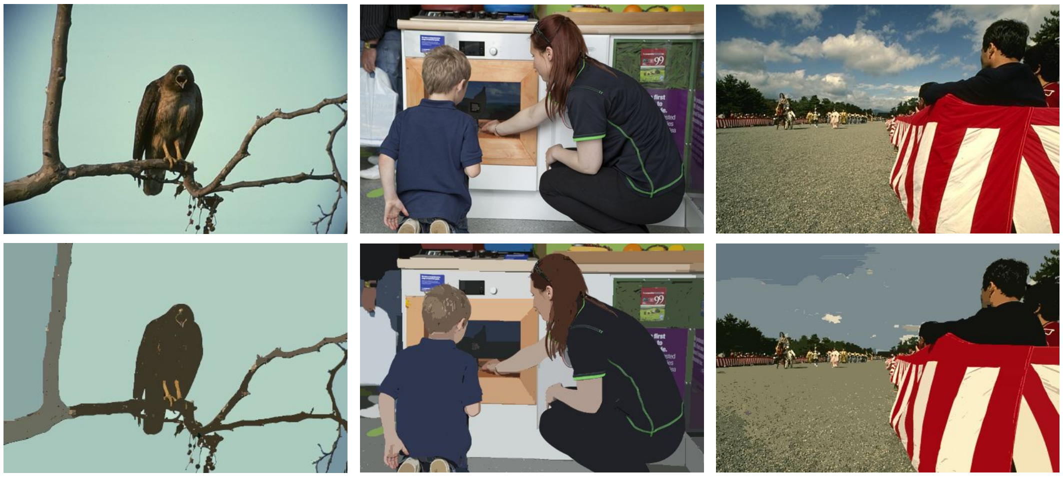


Figure 18: Flattening applied on Berkeley segmentation benchmark [122]: Results for three sample images are shown for illustrative purposes.

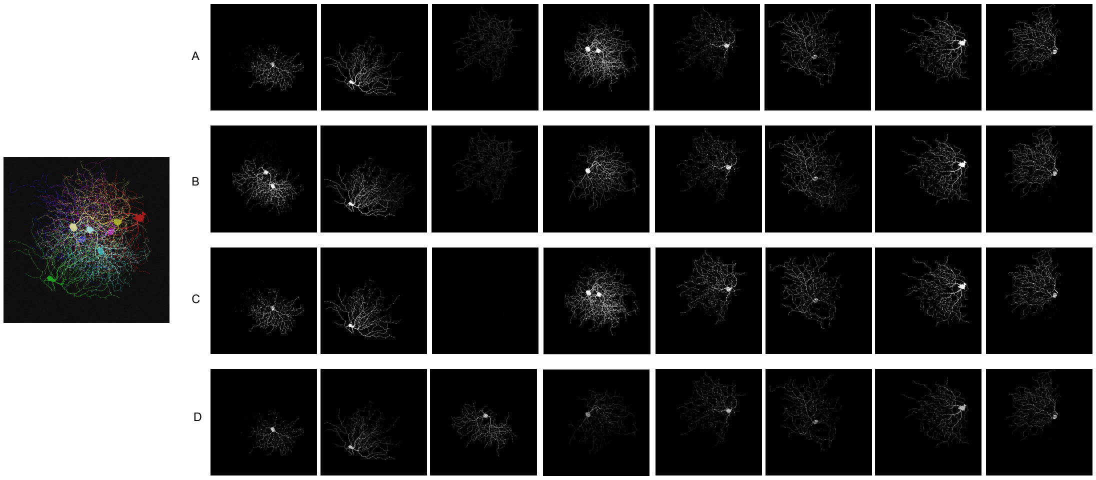


Figure 19: Qualitative comparison with 2d image segmentation algorithms: Results of (A) k-means, (B) fuzzy c-means, (C) meanshift, and (D) our algorithm. Our algorithm is able to separate the highly overlapping imge with 8 labels.

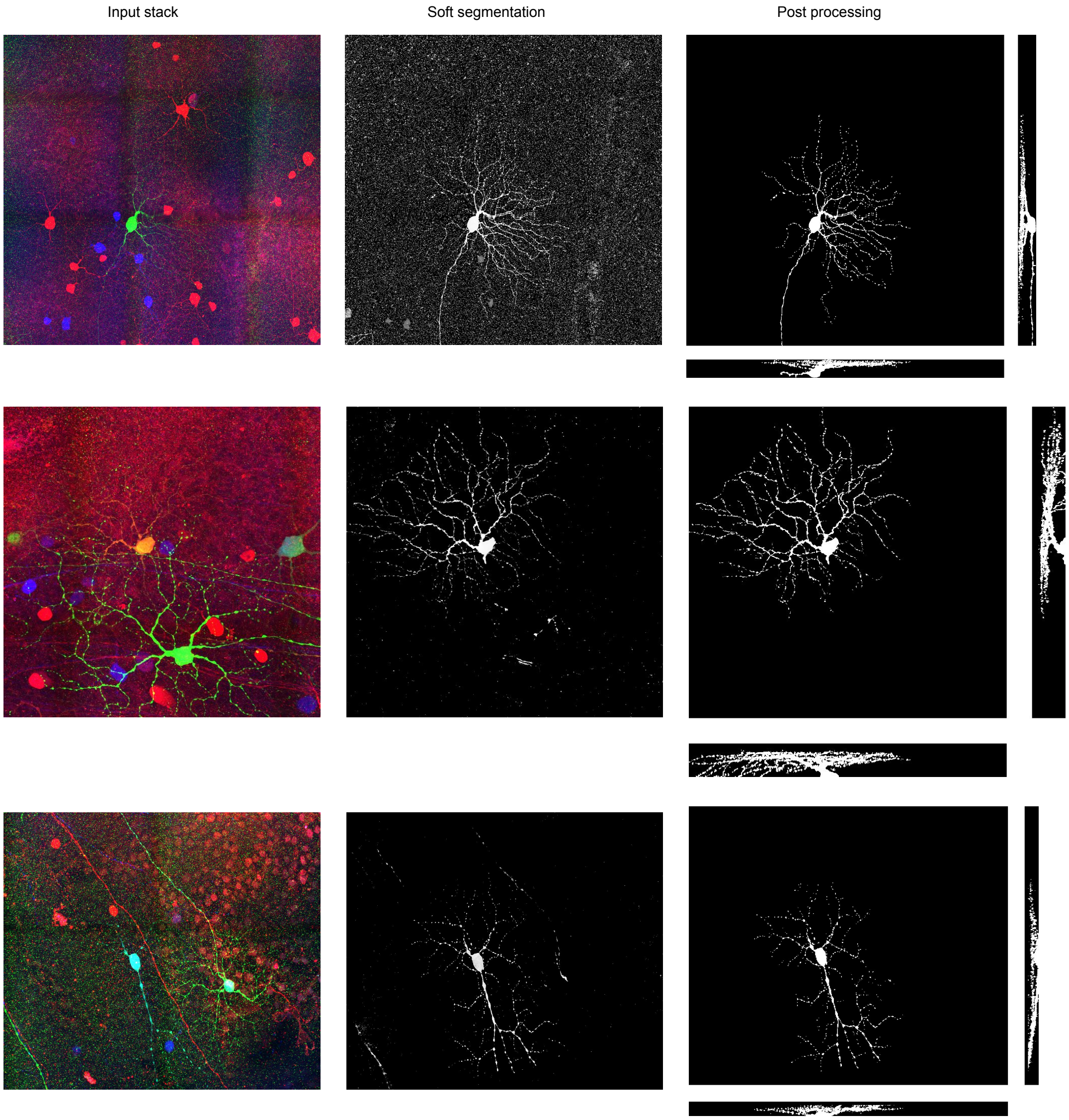


Figure 20: Obtaining hard segments: An example of green (first row), orange (middle row) and cyan (last row) labeled neuron case. Max-projection of the image stack, soft segmentation of neurons, and hard segment after post processing shows successfully 3d Reconstruction performed by our algorithm from noisy input images.

4.4 Discussion

Quantitative experiment results for both 6 (Table 2) and 8 neuron (Table 3) cases show successful segmentation of spectral viral labeled neurons very close to ground truth values. For synthetic

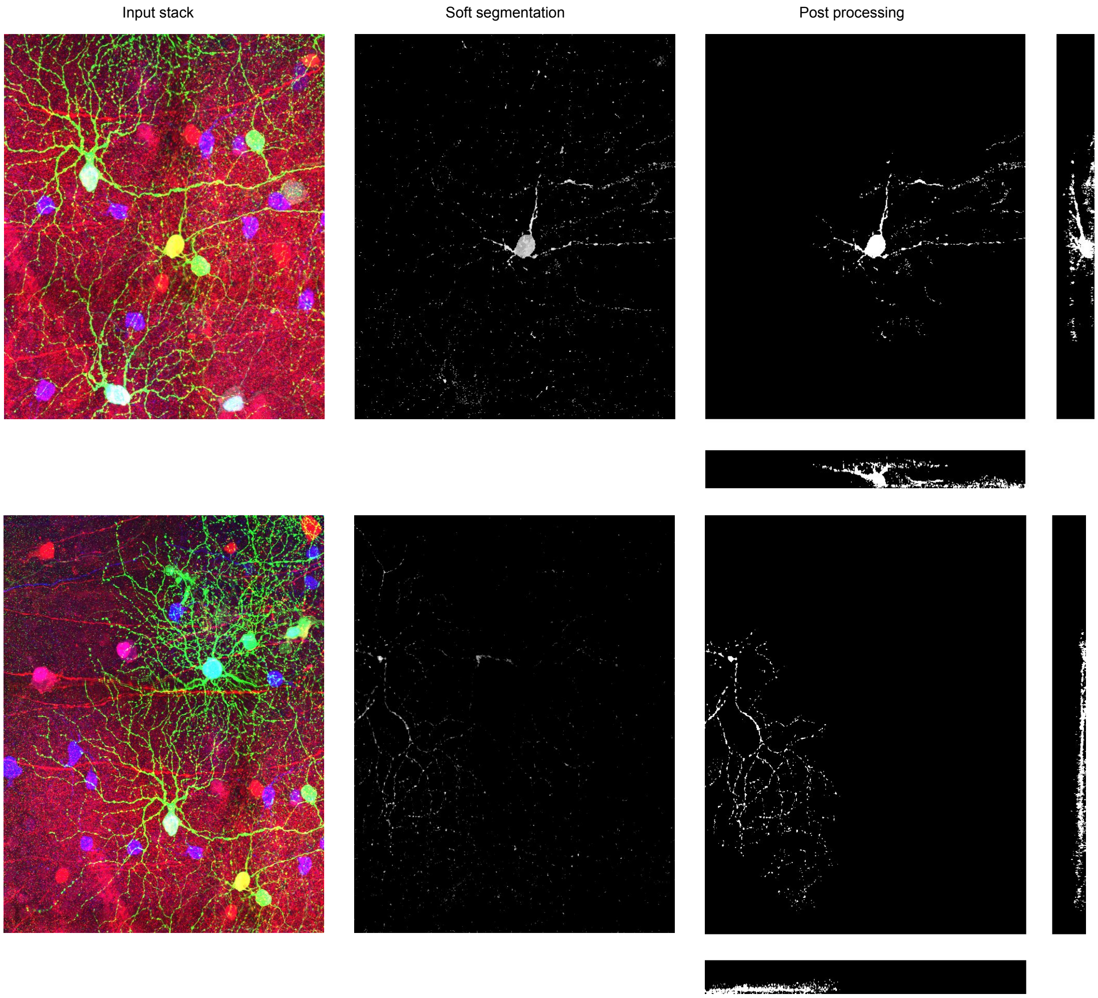


Figure 21: Obtaining hard segments: Challenging case, example of yellow (first row) and magenta (second row) labeled neurons. For magenta case, last 20 planes are excluded not to include other neuron somas. Due to unspecific labeling of these two neurons, results are not satisfactory.

dataset, it is not even required to carry our further post-processing, and labels can be obtained by simply thresholding operation. The success of the proposed pipeline can be attributed to its flexibility as soft layers computed by inverting the modified Porter-Duff model preserve available information. On the other hand, pixel-level classification (i.e., hard segmentation) strictly assigns a label to a pixel considering local neighbourhood. As can be seen from Figure 13 and 14, [1] is unable to preserve very thin dendritic trees and at the overall the segmented neurons are partitioned. In terms of computation time, the proposed approach also enables much faster processing comparing state of the art (Table 4). We also observe that when our preprocessing is replaced with the one proposed in [1], total processing time decreases an order of magnitude.

As stated at the outset, we face four main challenges in viral labeled confocal image stacks. Densely occluded structure of neurons and their thin dendritic structures are addressed by the design of soft segmentation, yet heavily background noise and inhomogeneous diffusion of viral vectors decrease the quality of results. To enhance the details without distorting dendritic arbors, BaSIC background subtraction [23] method is used and a weighted L1 norm total variation flattening algorithm is proposed. In Figure 15, results after BaSIC background correction (column C) show superiority comparing classical rolling-ball algorithm (column B). As can be seen from (D), the algorithm is able to correct non-uniform illumination caused by stitching as well. It can also shift the baseline intensity along the depth dimension for non-biased segmentation of neurons (D). As our algorithm is applied to each 2d image plane, correction of baseline drift along z directly enhances the algorithm accuracy and processing speed greatly. The most surprising is the success of the proposed image flattening algorithm. When applied to confocal stacks (Figure 16) or natural images (18), it induces sparsity and flattens low-frequency color variations while preserving very subtle details. This can be better observed in Figure 17 where the intensity profile of a spectral channel along the yellow line plotted. Here, piecewise constant recovery is clearly visible.

Since the proposed algorithm is based on convex polyhedron of data points and corresponding segments extracted for each vertex of this polyhedron, the method is able to extract segments that are otherwise ignored by partitioning based algorithms such as k-means, fuzzy c-means and meanshift. In Figure 19, for example, the neuron at the 3rd column is only segmented by our method but failed to be recognized by others.

In Figure 20, we apply our algorithm in an end-to-end fashion to two image stacks and reconstruct pseudo-green and pseudo-orange labelled neurons. Column A shows the maximum projection of the image stack, column B is the max projection of soft segmentation result and column C illustrates corresponding x, y and z maximum projection of masks obtained after our post-processing. The results show the successful reconstruction of whole neuron volume which can be directly used for neuron type identification. Figure 21, on the other hand, illustrates reconstruction of two neurons with nonspecific viral labeling. The resulting segments are not as satisfactory as in the previous figure.

5 Conclusion and Future Work

In this study, we developed a soft segmentation algorithm by estimating weighted Porter-Duff opacity layers for each fluorescent label in order to extract dendritic trees without damaging them. Number of soft segments (i.e., convex polyhedron vertices) are determined by progressive hulls and optimization is performed with a ridge and squared $TV - \ell_2$ regularized L-BFGS-B based least-squares framework. Soft segmentation is augmented by a novel weighted ℓ_1 total-variation image flattening and a post-processing pipeline designed using nonlinear morphological operators.

Firstly, a fluorescent image prior is proposed by stating that actual fluorescence emitting objects are distributed on the outer surface of the manifold formed by confocal images. Therefore, fluorescent images are inherently different than reflective-component dominated natural images. Exploiting this prior, expected fluorescent labels are estimated using vertices of convex hull of image data points. In our case, these pseudo-color vertices enable discovery of distinct fluorescent labels even if they are present in a single pixel.

Four main challenges are realized in viral labeled confocal images. Preprocessing consisting of background subtraction, contrast enhancement and image flattening is put forward to address these issues. Furthermore, image flattening is implemented using a modular proximal algorithm that gives very promising results and decreases computational load up to 50 times comparing state of the art [27]. We also show effectiveness of our image flattening via replacing preprocessing proposed in [1] by ours and observe 8 to 10 times increase in processing speed as tabulated in Table 4.

Soft segmentation algorithm is quantitatively evaluated using two synthetically generated image stacks with 6 and 8 neurons. Our implementation is also compared with previous 2d and 3d image segmentation techniques. We observe clear increase in segmentation accuracy in all three metrics used. Qualitative inspection of results also show that the soft segmentation is a risk-free way of reconstructing neurons compared to hard segmentation. Processing speed of soft segmentation is increased by initializing segments from results of the previous plane in z-stack. As consecutive frames have similar pixel values, this heuristic encourage fast convergence to local minimum.

We speculate that the method can be further improved by:

- Formulating the optimization as a least absolute deviations (LAD) problem instead of least-squares. It is known that for high dimensional data, euclidean loss may not give desired outcome (i.e., curse of dimensionality).
- Applying sparsity enforcing regularizers in soft segmentation, $TV - \ell_1$ and ℓ_1, particularly to induce natural inpainting and smoothness.
- Studying global fluorescence priors that can be employed to recover non-local statistics during optimization stage.
- Exploiting circular natural of hue to remove spectral salt and pepper noise. We observed that

some image stacks have high impulsive noise that cannot be removed by image flattening due to their high gradients. We apply 3d median filter in our post-processing, but this problem can also be resolved by employing a weighted or hue-based circular median filter.

- Using conditional random fields (CRF) as the last step in soft segmentation. In recent years, CRF has been shown to improve pixel-level details in segmentation tasks. Therefore, it would be possible to consult graph-based correction after the main segmentation step.

Bibliography

[1] Sümbül, U., Roossien, D., Cai, D., Chen, F., Barry, N., Cunningham, J. P., Boyden, E., & Paninski, L. 2016. Automated scalable segmentation of neurons from multispectral images. In *Advances in Neural Information Processing Systems*, 1912–1920.

[2] Meijering, E., Carpenter, A. E., Peng, H., Hamprecht, F. A., & Olivo-Marin, J.-C. 2016. Imagining the future of bioimage analysis. *Nature biotechnology*, 34(12), 1250.

[3] Zeng, H. & Sanes, J. R. 2017. Neuronal cell-type classification: challenges, opportunities and the path forward. *Nature Reviews Neuroscience*, 18(9), 530.

[4] Sharpee, T. O. 2014. Toward functional classification of neuronal types. *Neuron*, 83(6), 1329–1334.

[5] Swanson, L. W. & Lichtman, J. W. 2016. From cajal to connectome and beyond. *Annual review of neuroscience*, 39, 197–216.

[6] Masland, R. H. 2012. The neuronal organization of the retina. *Neuron*, 76(2), 266–280.

[7] Baden, T., Berens, P., Franke, K., Rosón, M. R., Bethge, M., & Euler, T. 2016. The functional diversity of retinal ganglion cells in the mouse. *Nature*, 529(7586), 345.

[8] Dhande, O. S., Stafford, B. K., Lim, J.-H. A., & Huberman, A. D. 2015. Contributions of retinal ganglion cells to subcortical visual processing and behaviors. *Annual review of vision science*, 1, 291–328.

[9] Farrow, K. & Masland, R. H. 2011. Physiological clustering of visual channels in the mouse retina. *Journal of neurophysiology*, 105(4), 1516–1530.

[10] Martersteck, E. M., Hirokawa, K. E., Evarts, M., Bernard, A., Duan, X., Li, Y., Ng, L., Oh, S. W., Ouellette, B., Royall, J. J., et al. 2017. Diverse central projection patterns of retinal ganglion cells. *Cell reports*, 18(8), 2058–2072.

[11] Roska, B. & Werblin, F. 2001. Vertical interactions across ten parallel, stacked representations in the mammalian retina. *Nature*, 410(6828), 583.

[12] Sanes, J. R. & Masland, R. H. 2015. The types of retinal ganglion cells: current status and implications for neuronal classification. *Annual review of neuroscience*, 38, 221–246.

[13] Sümbül, U., Song, S., McCulloch, K., Becker, M., Lin, B., Sanes, J. R., Masland, R. H., & Seung, H. S. 2014. A genetic and computational approach to structurally classify neuronal types. *Nature communications*, 5, 3512.

[14] Sümbül, U., Zlateski, A., Vishwanathan, A., Masland, R. H., & Seung, H. S. 2014. Automated computation of arbor densities: a step toward identifying neuronal cell types. *Frontiers in neuroanatomy*, 8, 139.

[15] Armañanzas, R. & Ascoli, G. A. 2015. Towards the automatic classification of neurons. *Trends in neurosciences*, 38(5), 307–318.

[16] Callaway, E. M. 2008. Transneuronal circuit tracing with neurotropic viruses. *Current opinion in neurobiology*, 18(6), 617–623.

[17] Nassi, J. J., Cepko, C. L., Born, R. T., & Beier, K. T. 2015. Neuroanatomy goes viral! *Frontiers in neuroanatomy*, 9, 80.

[18] Chalfie, M., Tu, Y., Euskirchen, G., Ward, W. W., & Prasher, D. C. 1994. Green fluorescent protein as a marker for gene expression. *Science*, 263(5148), 802–805.

[19] Jennings, J. H. & Stuber, G. D. 2014. Tools for resolving functional activity and connectivity within intact neural circuits. *Current Biology*, 24(1), R41–R50.

[20] Livet, J., Weissman, T. A., Kang, H., Draft, R. W., Lu, J., Bennis, R. A., Sanes, J. R., & Lichtman, J. W. 2007. Transgenic strategies for combinatorial expression of fluorescent proteins in the nervous system. *Nature*, 450(7166), 56.

[21] Hadjieconomou, D., Rotkopf, S., Alexandre, C., Bell, D. M., Dickson, B. J., & Salecker, I. 2011. Flybow: genetic multicolor cell labeling for neural circuit analysis in drosophila melanogaster. *Nature methods*, 8(3), 260.

[22] Pan, Y. A., Freundlich, T., Weissman, T. A., Schoppik, D., Wang, X. C., Zimmerman, S., Ciruna, B., Sanes, J. R., Lichtman, J. W., & Schier, A. F. 2013. Zebrabow: multispectral cell labeling for cell tracing and lineage analysis in zebrafish. *Development*, 140(13), 2835–2846. doi:10.1242/dev.094631.

[23] Peng, T., Thorn, K., Schroeder, T., Wang, L., Theis, F. J., Marr, C., & Navab, N. 2017. A basic tool for background and shading correction of optical microscopy images. *Nature Communications*, 8, 14836.

[24] Porter, T. & Duff, T. 1984. Compositing digital images. In *ACM Siggraph Computer Graphics*, volume 18, 253–259. ACM.

[25] Tan, J., Lien, J.-M., & Gingold, Y. 2017. Decomposing images into layers via rgb-space geometry. *ACM Transactions on Graphics (TOG)*, 36(1), 7.

[26] Barbero, A. & Sra, S. 2014. Modular proximal optimization for multidimensional total-variation regularization. URL: http://arxiv.org/abs/1411.0589.

[27] Bi, S., Han, X., & Yu, Y. 2015. An l 1 image transform for edge-preserving smoothing and scene-level intrinsic decomposition. *ACM Transactions on Graphics (TOG)*, 34(4), 78.

[28] Liu, S., Qi, L., Qin, H., Shi, J., & Jia, J. 2018. Path aggregation network for instance segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8759–8768.

[29] Chen, L.-C., Zhu, Y., Papandreou, G., Schroff, F., & Adam, H. 2018. Encoder-decoder with atrous separable convolution for semantic image segmentation. *arXiv preprint arXiv:1802.02611*.

[30] He, K., Gkioxari, G., Dollár, P., & Girshick, R. 2017. Mask r-cnn. In *Computer Vision (ICCV), 2017 IEEE International Conference on*, 2980–2988. IEEE.

[31] Bansal, A., Chen, X., Russell, B., Gupta, A., & Ramanan, D. 2016. Pixelnet: Towards a general pixel-level architecture. *arXiv preprint arXiv:1609.06694*.

[32] Syu, J.-H., Wang, S.-J., & Wang, L.-C. 2017. Hierarchical image segmentation based on iterative contraction and merging. *IEEE Transactions on Image Processing*, 26(5), 2246–2260.

[33] Liu, T., Seyedhosseini, M., & Tasdizen, T. 2016. Image segmentation using hierarchical merge tree. *IEEE Transactions on Image Processing*, 25(10), 4596–4607.

[34] Levinkov, E., Uhrig, J., Tang, S., Omran, M., Insafutdinov, E., Kirillov, A., Rother, C., Brox, T., Schiele, B., & Andres, B. 2017. Joint graph decomposition & node labeling: Problem, algorithms, applications. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*.

[35] Gao, L., Song, J., Nie, F., Zou, F., Sebe, N., Shen, H. T., et al. 2016. Graph-without-cut: An ideal graph learning for image segmentation. In *AAAI*, 1188–1194.

[36] Cheung, G., Magli, E., Tanaka, Y., & Ng, M. 2018. Graph spectral image processing. *arXiv preprint arXiv:1801.04749*.

[37] Wang, X., Tang, Y., Masnou, S., & Chen, L. 2015. A global/local affinity graph for image segmentation. *IEEE Transactions on Image Processing*, 24(4), 1399–1411.

[38] Tang, M., Ben Ayed, I., Marin, D., & Boykov, Y. 2015. Secrets of grabcut and kernel k-means. In *Proceedings of the IEEE International Conference on Computer Vision*, 1555–1563.

[39] Marin, D., Tang, M., Ayed, I. B., & Boykov, Y. Y. 2017. Kernel clustering: density biases and solutions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[40] Fang, C., Liao, Z., & Yu, Y. 2018. Piecewise flat embedding for image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[41] Hell, B., Kassubeck, M., Bauszat, P., Eisemann, M., & Magnor, M. 2015. An approach toward fast gradient-based image segmentation. *IEEE Transactions on Image Processing*, 24(9), 2633–2645.

[42] Kong, S. & Fowlkes, C. 2017. Recurrent pixel embedding for instance grouping. *arXiv preprint arXiv:1712.08273*.

[43] Xia, X. & Kulis, B. 2017. W-net: A deep model for fully unsupervised image segmentation. *arXiv preprint arXiv:1711.08506*.

[44] Xie, S. & Tu, Z. 2015. Holistically-nested edge detection. In *Proceedings of the IEEE international conference on computer vision*, 1395–1403.

[45] Kokkinos, I. 2015. Pushing the boundaries of boundary detection using deep learning. *arXiv preprint arXiv:1511.07386*.

[46] Maninis, K.-K., Pont-Tuset, J., Arbeláez, P., & Van Gool, L. 2016. Convolutional oriented boundaries. In *European Conference on Computer Vision*, 580–596. Springer.

[47] Marin, D., Zhong, Y., Drangova, M., & Boykov, Y. 2015. Thin structure estimation with curvature regularization. In *Proceedings of the IEEE International Conference on Computer Vision*, 397–405.

[48] Fan, Q., Yang, J., Hua, G., Chen, B., & Wipf, D. 2018. Revisiting deep intrinsic image decompositions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 8944–8952.

[49] Lettry, L., Vanhoey, K., & van Gool, L. 2018. Deep unsupervised intrinsic image decomposition by siamese training. *arXiv preprint arXiv:1803.00805*.

[50] Nestmeyer, T. & Gehler, P. V. 2017. Reflectance adaptive filtering improves intrinsic image estimation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, volume 2, 4.

[51] Zhao, Q., Tan, P., Dai, Q., Shen, L., Wu, E., & Lin, S. 2012. A closed-form solution to retinex with nonlocal texture constraints. *IEEE transactions on pattern analysis and machine intelligence*, 34(7), 1437–1444.

[52] Lafferty, J., McCallum, A., & Pereira, F. C. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data.

[53] Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., & Yuille, A. L. 2018. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence*, 40(4), 834–848.

[54] Hariharan, B., Arbeláez, P., Girshick, R., & Malik, J. 2015. Hypercolumns for object segmentation and fine-grained localization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 447–456.

[55] Tai, Y.-W., Jia, J., & Tang, C.-K. 2007. Soft color segmentation and its applications. *IEEE transactions on pattern analysis and machine intelligence*, 29(9).

[56] Chang, H., Fried, O., Liu, Y., DiVerdi, S., & Finkelstein, A. 2015. Palette-based photo recoloring. *ACM Transactions on Graphics (TOG)*, 34(4), 139.

[57] Lin, S., Fisher, M., Dai, A., & Hanrahan, P. 2017. Layerbuilder: Layer decomposition for interactive image and video color editing. *arXiv preprint arXiv:1701.03754*.

[58] Aksoy, Y., Aydin, T. O., Smolić, A., & Pollefeys, M. 2017. Unmixing-based soft color segmentation for image manipulation. *ACM Transactions on Graphics (TOG)*, 36(2), 19.

[59] Zhang, Q., Xiao, C., Sun, H., & Tang, F. 2017. Palette-based image recoloring using color decomposition optimization. *IEEE Transactions on Image Processing*, 26(4), 1952–1964.

[60] Chuang, Y.-Y., Curless, B., Salesin, D. H., & Szeliski, R. 2001. A bayesian approach to digital matting. In *null*, 264. IEEE.

[61] Xu, N., Price, B., Cohen, S., & Huang, T. 2017. Deep image matting. In *Computer Vision and Pattern Recognition (CVPR)*.

[62] Levin, A., Rav-Acha, A., & Lischinski, D. 2008. Spectral matting. *IEEE transactions on pattern analysis and machine intelligence*, 30(10), 1699–1712.

[63] Sommer, C., Straehle, C., Koethe, U., & Hamprecht, F. A. 2011. Ilastik: Interactive learning and segmentation toolkit. In *Biomedical Imaging: From Nano to Macro, 2011 IEEE International Symposium on*, 230–233. IEEE.

[64] Bas, E. & Erdogmus, D. 2010. Piecewise linear cylinder models for 3-dimensional axon segmentation in brainbow imagery. In *Biomedical Imaging: From Nano to Macro, 2010 IEEE International Symposium on*, 1297–1300. IEEE.

[65] Bas, E., Erdogmus, D., Draft, R., & Lichtman, J. W. 2012. Local tracing of curvilinear structures in volumetric color images: Application to the brainbow analysis. *Journal of Visual Communication and Image Representation*, 23(8), 1260–1271.

[66] Shao, H.-C., Cheng, W.-Y., Chen, Y.-C., & Hwang, W.-L. 2012. Colored multi-neuron image processing for segmenting and tracing neural circuits. In *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 2025–2028. IEEE.

[67] Kim, T.-Y., Kang, M.-S., Kim, M.-H., & Choi, H.-K. 2012. Segmentation of neuronal axons in brainbow images. *Journal of Korea Multimedia Society*, 15(12), 1417–1429.

[68] Hsu, Y. & Lu, H. H.-S. 2013. Brainbow image segmentation using bayesian sequential partitioning. *World Academy of Science, Engineering and Technology, International Journal of Biomedical and Biological Engineering*, (12).

[69] Baglietto, S., Kepiro, I. E., Hilgen, G., Sernagor, E., Murino, V., & Sona, D. 2017. Segmentation of retinal ganglion cells from fluorescent microscopy imaging. In *BIOIMAGING*, 17–23.

[70] Chien-Chang, C., Hung-Hui, J., Meng-Yuan, T., & Lu, H. H.-S. 2018. Unsupervised learning and pattern recognition of biological data structures with density functional theory and machine learning. *Scientific Reports (Nature Publisher Group)*, 8, 1–11.

[71] Roossien, D. H., Webb, J. M., Sadis, B. V., Yan, Y., Min, L. Y., Dizaji, A. S., Bogart, L. J., Mazuski, C., Huth, R. S., Stecher, J. S., et al. 2018. Neuronal morphology and projection analysis by multispectral tracing in densely labeled mouse brain. *bioRxiv*, 230458.

[72] Chen, J., Banerjee, S., Grama, A., Scheirer, W. J., & Chen, D. Z. 2017. Neuron segmentation using deep complete bipartite networks. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 21–29. Springer.

[73] Yan, Y., Sadis, B. V., Roossien, D. H., Corso, J. J., & Cai, D. 2018. A recurrent neural network approach for automated neural tracing in multispectral 3d images. *bioRxiv*, 230441.

[74] Hastie, T. & Stuetzle, W. 1989. Principal curves. *Journal of the American Statistical Association*, 84(406), 502–516.

[75] Sinop, A. K. & Grady, L. 2007. A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm. In *Computer Vision, 2007. ICCV 2007. IEEE 11th International Conference on*, 1–8. IEEE.

[76] Longair, M. H., Baker, D. A., & Armstrong, J. D. 2011. Simple neurite tracer: open source software for reconstruction, visualization and analysis of neuronal processes. *Bioinformatics*, 27(17), 2453–2454.

[77] Wright, S. & Nocedal, J. 1999. Numerical optimization. *Springer Science*, 35(67-68), 7.

[78] Boyd, S. & Vandenberghe, L. 2004. *Convex optimization*. Cambridge university press.

[79] Luenberger, D. G., Ye, Y., et al. 1984. *Linear and nonlinear programming*, volume 2. Springer.

[80] Papadimitriou, C. H. & Steiglitz, K. 1998. *Combinatorial optimization: algorithms and complexity*. Courier Corporation.

[81] Kall, P., Wallace, S. W., & Kall, P. 1994. *Stochastic programming*. Springer.

[82] Kappes, J., Andres, B., Hamprecht, F., Schnorr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B., Lellmann, J., Komodakis, N., et al. 2013. A comparative study of modern inference techniques for discrete energy minimization problems. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 1328–1335.

[83] LeCun, Y., Chopra, S., Hadsell, R., Ranzato, M., & Huang, F. 2006. A tutorial on energy-based learning. *Predicting structured data*, 1(0).

[84] Byrd, R. H., Lu, P., Nocedal, J., & Zhu, C. 1995. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific Computing*, 16(5), 1190–1208.

[85] Floudas, C. A. & Gounaris, C. E. 2009. A review of recent advances in global optimization. *Journal of Global Optimization*, 45(1), 3–38.

[86] Liu, D. C. & Nocedal, J. 1989. On the limited memory bfgs method for large scale optimization. *Mathematical programming*, 45(1-3), 503–528.

[87] Broyden, C. G. 1970. The convergence of a class of double-rank minimization algorithms 1. general considerations. *IMA Journal of Applied Mathematics*, 6(1), 76–90.

[88] Fletcher, R. 1970. A new approach to variable metric algorithms. *The computer journal*, 13(3), 317–322.

[89] Goldfarb, D. 1970. A family of variable-metric methods derived by variational means. *Mathematics of computation*, 24(109), 23–26.

[90] Shanno, D. F. 1970. Conditioning of quasi-newton methods for function minimization. *Mathematics of computation*, 24(111), 647–656.

[91] Parikh, N., Boyd, S., et al. 2014. Proximal algorithms. *Foundations and Trends® in Optimization*, 1(3), 127–239.

[92] Boyd, S., Parikh, N., Chu, E., Peleato, B., Eckstein, J., et al. 2011. Distributed optimization and statistical learning via the alternating direction method of multipliers. *Foundations and Trends® in Machine learning*, 3(1), 1–122.

[93] Douglas, J. & Rachford, H. H. 1956. On the numerical solution of heat conduction problems in two and three space variables. *Transactions of the American mathematical Society*, 82(2), 421–439.

[94] Goldstein, T. & Osher, S. 2009. The split bregman method for l1-regularized problems. *SIAM journal on imaging sciences*, 2(2), 323–343.

[95] Chambolle, A. & Pock, T. 2011. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of mathematical imaging and vision*, 40(1), 120–145.

[96] Beck, A. & Teboulle, M. 2009. A fast iterative shrinkage-thresholding algorithm for linear inverse problems. *SIAM journal on imaging sciences*, 2(1), 183–202.

[97] Geman, D. & Yang, C. 1995. Nonlinear image recovery with half-quadratic regularization. *IEEE Transactions on Image Processing*, 4(7), 932–946.

[98] Yin, W. 2016. Proximal operators and proximal algorithms, ucla mathematics 285j lecture notes.

[99] Natarajan, B. K. 1995. Sparse approximate solutions to linear systems. *SIAM journal on computing*, 24(2), 227–234.

[100] Donoho, D. L. 2006. Compressed sensing. *IEEE Transactions on information theory*, 52(4), 1289–1306.

[101] Tibshirani, R. 1996. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 267–288.

[102] Candes, E., Tao, T., et al. 2007. The dantzig selector: Statistical estimation when p is much larger than n. *The Annals of Statistics*, 35(6), 2313–2351.

[103] Rudin, L. I., Osher, S., & Fatemi, E. 1992. Nonlinear total variation based noise removal algorithms. *Physica D: nonlinear phenomena*, 60(1-4), 259–268.

[104] Gousseau, Y. & Morel, J.-M. 2001. Are natural images of bounded variation? *SIAM Journal on Mathematical Analysis*, 33(3), 634–648.

[105] Muller, M. 2006. *Introduction to confocal fluorescence microscopy*, volume 69. SPIE press.

[106] Wu, Q., Merchant, F., & Castleman, K. 2010. *Microscope image processing*. Elsevier.

[107] Lin, Z., Liu, R., & Su, Z. 2011. Linearized alternating direction method with adaptive penalty for low-rank representation. In *Advances in neural information processing systems*, 612–620.

[108] Perona, P. & Malik, J. 1990. Scale-space and edge detection using anisotropic diffusion. *IEEE Transactions on pattern analysis and machine intelligence*, 12(7), 629–639.

[109] Tomasi, C. & Manduchi, R. 1998. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, 839–846. IEEE.

[110] Xu, L., Lu, C., Xu, Y., & Jia, J. 2011. Image smoothing via l 0 gradient minimization. In *ACM Transactions on Graphics (TOG)*, volume 30, 174. ACM.

[111] Strekalovskiy, E. & Cremers, D. 2014. Real-time minimization of the piecewise smooth mumford-shah functional. In *European Conference on Computer Vision*, 127–141. Springer.

[112] Davies, P. L. & Kovac, A. 2001. Local extremes, runs, strings and multiresolution. *Annals of Statistics*, 1–48.

[113] Steidl, G., Didas, S., & Neumann, J. 2005. Relations between higher order tv regularization and support vector regression. In *International Conference on Scale-Space Theories in Computer Vision*, 515–527. Springer.

[114] Sander, P. V., Gu, X., Gortler, S. J., Hoppe, H., & Snyder, J. 2000. Silhouette clipping. In *Proceedings of the 27th annual conference on Computer graphics and interactive techniques*, 327–334. ACM Press/Addison-Wesley Publishing Co.

[115] Platis, N. & Theoharis, T. 2003. Progressive hulls for intersection applications. In *Computer Graphics Forum*, volume 22, 107–116. Wiley Online Library.

[116] Duff, T. 2017. Deep compositing using lie algebras. *ACM Transactions on Graphics (TOG)*, 36(4), 120a.

[117] Byrd, R. H., Nocedal, J., & Schnabel, R. B. 1994. Representations of quasi-newton matrices and their use in limited memory methods. *Mathematical Programming*, 63(1-3), 129–156.

[118] Zhu, C., Byrd, R. H., Lu, P., & Nocedal, J. 1997. Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. *ACM Transactions on Mathematical Software (TOMS)*, 23(4), 550–560.

[119] Unnikrishnan, R. & Hebert, M. 2005. Measures of similarity. In *Application of Computer Vision, 2005. WACV/MOTIONS'05 Volume 1. Seventh IEEE Workshops on*, volume 1, 394–394. IEEE.

[120] Meilă, M. 2005. Comparing clusterings: an axiomatic view. In *Proceedings of the 22nd international conference on Machine learning*, 577–584. ACM.

[121] Sternberg, S. R. 1983. Biomedical image processing. *Computer*, 16(1), 22–34.

[122] Martin, D., Fowlkes, C., Tal, D., & Malik, J. July 2001. A database of human segmented natural images and its application to evaluating segmentation algorithms and measuring ecological statistics. In *Proc. 8th Int'l Conf. Computer Vision*, volume 2, 416–423.

A Retinabow Protocol

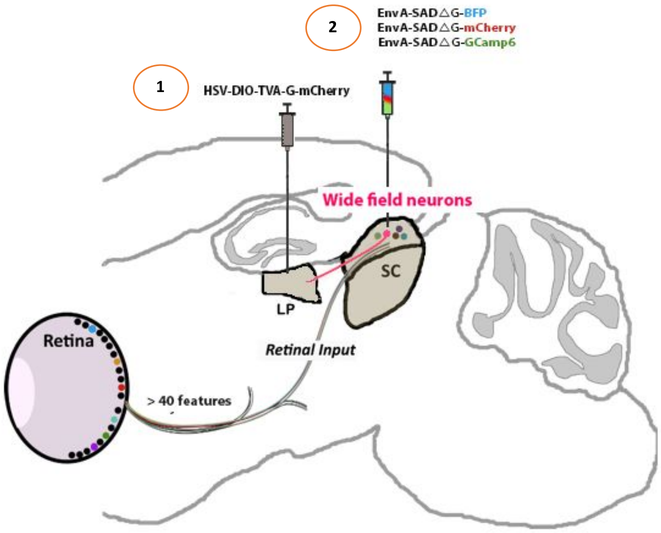


Figure 22: Schematic of the targeted neural circuits from retina to lateral posterior nuclei (LP) via superior colliculus (SC). Firstly HSV virus vectors are injected to identify wide-field neurons which take inputs from retinal ganglion cells. Then, three virus vectors corresponding red, green and blue band emission are injected to create stochastic multi-color expression in ganglion cells

Injection of Rabbies Vectors

At the outset, floxed HSV was injected to the LP. This results in expression of TVA and protein G in wide-field neurons that project to the LP. We wait 21 days for injections to be expressed. Then, a mixture of different EnvA-coated and G-deleted fluorescent rabies virus were injected to SC that will interact with neurons containing the TVA and protein G (Figure 22). Retinal ganglion cells having synaptic connections with wide-field neurons will be labeled with different fluorescent proteins randomly. This occur due to the retrograde transynaptic motion of rabies virus vectors. At the eighth day following the rabies virus injection, retina was dissected and fixed in the PFA. Afterwards, the immunostaing is performed to amplify emitted fluorescent signal (Figure 23).

Immunostaining

The primary antibodies recognizing BFP, mCherry, GCaMP6 and ChaT (for Starburst Amacrine cells

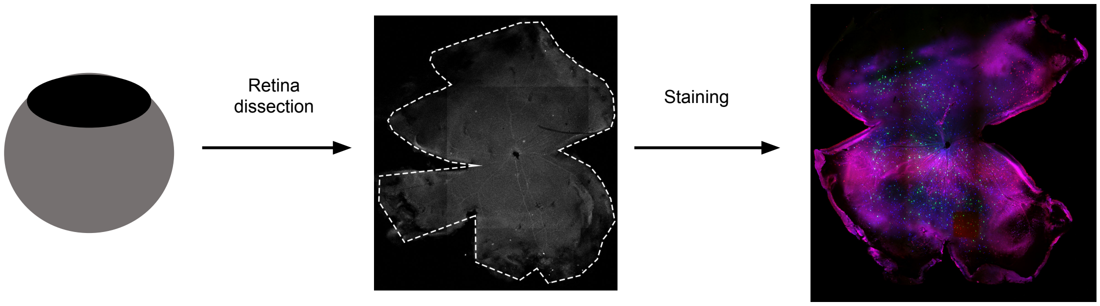


Figure 23: Schematic of the processing of the retina. Immunostaining is necessary to amplify fluorescence emission.

labelling) were firstly prepared and incubated with the whole-mount retina for 5-7 days. Then relative secondary antibodies conjugated with different fluorescent proteins (405, 555, 488 and 633) were added and incubated overnight. After mounting, the retina will be ready for image acquisition.

Confocal Laser Scanning Microscopy Imagining

Zeiss LSM 710 microscope is used for confocal imaging of retinas. Images of the whole retina for overview purposes were captured with a 10x (plan-APOCHROMAT 0.45 NA, Zeiss) objective with settings: Zoom 0.7, 4x4-tiles with 0 to 15% overlap, 2.37 μm/pixel resolution. Close-up retinal ganglion cell images used for processing purposes obtained with 63x (plan-APOCHROMAT 1.4 NA, Zeiss) objective with settings: Zoom 0.7, 2x2-tiles (this is adjusted depending on the cell size) with 0 to %15 overlap. Resulting images have XY-resolution of approximately 0.38 μm/pixel and a Z-resolution between 0.25 and 0.35μm/pixel. The Z-stacks scanned over approximately 50 μm in depth.