

NeuroConstruct: 3D Reconstruction and Visualization of Neurites in Optical Microscopy Brain Images

Parmida Ghahremani¹, Saeed Boorboor¹, Pooya Mirhosseini, Chetan Gudisagar, Mala Ananth, David Talmage², Lorna W. Role, and Arie E. Kaufman¹, *Fellow, IEEE*

Abstract—We introduce NeuroConstruct, a novel end-to-end application for the segmentation, registration, and visualization of brain volumes imaged using wide-field microscopy. NeuroConstruct offers a Segmentation Toolbox with various annotation helper functions that aid experts to effectively and precisely annotate micrometer resolution neurites. It also offers an automatic neurites segmentation using convolutional neuronal networks (CNN) trained by the Toolbox annotations and somas segmentation using thresholding. To visualize neurites in a given volume, NeuroConstruct offers a hybrid rendering by combining iso-surface rendering of high-confidence classified neurites, along with real-time rendering of raw volume using a 2D transfer function for voxel classification score versus voxel intensity value. For a complete reconstruction of the 3D neurites, we introduce a Registration Toolbox that provides automatic coarse-to-fine alignment of serially sectioned samples. The quantitative and qualitative analysis show that NeuroConstruct outperforms the state-of-the-art in all design aspects. NeuroConstruct was developed as a collaboration between computer scientists and neuroscientists, with an application to the study of cholinergic neurons, which are severely affected in Alzheimer’s disease.

Index Terms—Wide-field microscopy, neuron morphology, segmentation, registration, hybrid volume rendering, CNN

1 INTRODUCTION

ADVANCES in optical microscopy (OM) have driven the field of neuroanatomy and the acquisition of high-resolution 3D images of the brain across multiple spatial scales. Using new techniques and tools for reconstruction, visualization, and analysis of these 3D images, neuroscientists can now study in detail the structural and functional connectivity underlying the brain. This technology is widely used to diagnose diseases caused by neuron degeneration, such as cholinergic neurons in Alzheimer’s disease [1].

There are various techniques for imaging brain sections, such as electron (EM), confocal, two-photon, and light wide-field microscopy (WFM). Amongst neuroscientists, WFM is preferred for experimental studies due to its large field-of-view and fast image acquisition. Imaging a 40× slice of a sample using confocal microscopy would take 15 hours, but only 1.5 hours on WFM. Moreover, WFM automatically moves the sample stage, resulting in sequential image acquisition without manually readjusting the sample orientation for every field-of-view. These advantages come with

a trade-off. Due to its optics, images acquired using WFM suffer from a degraded contrast between foreground and background, low signal-to-noise ratio (SNR), and poor axial resolution (Fig. 1). The microscope limited chamber size and the adverse affect of increased light scattering in thicker samples compel neuroscientists to physically slice thin sections of specimens. Consequently, a brain study is constrained to analyzing individual sections of a specimen.

To study and diagnose brain diseases, neuroscientists explore the structure and function of the nervous system. In brain studies using WFM images, due to its optics and limited chamber size, neuroscientists face three challenges:

- *Segmentation of neuronal structures*: The primary information neuroscientists expect from WFM brain volumes are neuronal structures. The inherent WFM limitations, such as out-of-focus blurring and the absence of distinctive set of intensity values differentiating foreground (neurons) from background (blurring artifacts and brain tissue), leads to failure of current neuron tracing methods.
- *Registration of neuronal structures*: Following the segmentation, the reconstruction of the entire brain specimen as a full volume enables a complete understanding of neuron morphologies. However, registration of brain sections is complicated, as the physical slicing results in non-rigid deformations on captured images.
- *Visualization of neuronal structures*: The WFM limitations make visualization parameters adjustment complex and time-consuming, and directly applying rendering techniques do not yield effective results.

• *Parmida Ghahremani, Saeed Boorboor, Pooya Mirhosseini, Chetan Gudisagar, and Arie E. Kaufman are with the Department of Computer Science, Stony Brook University, Stony Brook, NY 11794-2424 USA.*

E-mail: {pghahremani, sboorboor, semirhossein, fchetan, ari}@cs.stonybrook.edu.

• *Mala Ananth, David Talmage, and Lorna W. Role are with the National Institutes of Health, Bethesda, MD 20892 USA.*

E-mail: {mala.ananth, david.talmage, lorna.role}@nih.gov.

Manuscript received 5 Jan. 2021; revised 10 Aug. 2021; accepted 19 Aug. 2021.

Date of publication 3 Sept. 2021; date of current version 27 Oct. 2022.

(Corresponding author: Parmida Ghahremani.)

Recommended for acceptance by C. Wang.

Digital Object Identifier no. 10.1109/TVCG.2021.3109460

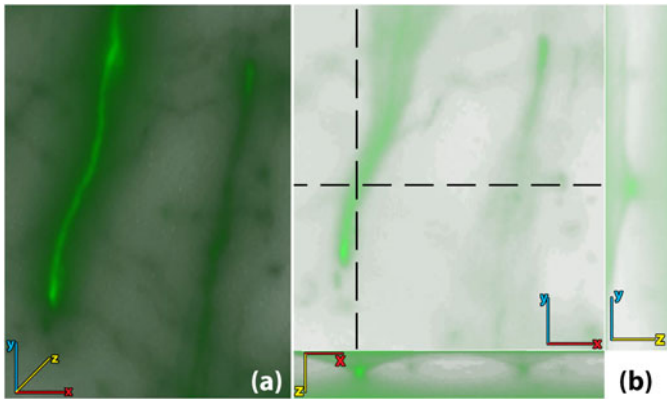


Fig. 1. WFM images are volumes obtained by focusing at different depths of thinly sliced specimen [2]. (a) Volume rendering of unprocessed WFM brain image. (b) Top-left: 2D cross-sectional view of the volume in x - y plane. Top-right: 2D cross-section in y - z plane at the vertical dashed line. Bottom: 2D cross-section in x - z plane at the horizontal dashed line. The cross-sections show how out-of-focus light occludes low-intensity features, making it difficult to analyze 3D structures.

Moreover, segmentation and tracing techniques are limited to classifying features captured within the designed algorithm or trained model. Thus, a complimentary approach is required.

The primary goal of NeuroConstruct is reconstruction of neuronal structures in WFM whole brain specimens that enables the exploration of the nervous system. To achieve this goal, we address all challenges and present NeuroConstruct, a novel end-to-end application to reconstruct neuronal structures by performing tasks of segmenting, registering, and visualizing neuronal structures in brain volumes.

To overcome the segmentation and visualization problem, NeuroConstruct offers a novel Segmentation Toolbox. It provides simultaneous 2D cross-sectional views and 3D volume rendering of image stacks along with real-time user-drawn annotations. It also provides novel annotation functions to help experts annotate neurons in 3D brain images efficiently. We further implemented automatic neuron segmentation using a nested CNN that hires skip pathways for connecting the encoder and decoder to compute feature maps and segment neurites using the extracted maps combined with image processing techniques. CNNs have achieved breakthrough performance in various segmentation tasks. Their primary issue is requiring a vast amount of labeled data for training. Due to the high density of neurons in brain images, their manual annotation in 3D image stacks requires tremendous time and effort. We introduce a workflow to speed up ground-truth generation.

The robustness of deep-learning models dramatically depends on the accuracy and availability of sufficient training data. Biologist's workflow is subjected to experimental variations, and their data has immense biological variability. Thus, the infeasibility of capturing sufficient training data covering all neuronal variations can result in the model failing to segment a neurite for which it was not trained. Therefore, we devise a hybrid approach to visualize the extracted neurites along with possible unsegmented neurites. Specifically, our model generates a per-voxel confidence score of the classification as a neurite. In our hybrid visualization, we first render the iso-surface of high-confidence neurites using a user-adjusted confidence threshold,

calling it the *structural representation mode*. Next, we introduce a 2D transfer function (TF) of voxel confidence versus raw data voxel intensity, calling it the *fusion mode*. By interactively manipulating the TF, experts visualize possible neurites in the raw volume below the confidence threshold.

To overcome the registration challenge, NeuroConstruct provides a Registration Toolbox for coarse-to-fine registration of depth-adjacent brain sections along with visualization of brain sections to be registered and the aligned sections. Using the brain overall structural anatomy, it first estimates a global rigid-body transformation that coarsely registers adjacent sections. Then, we introduce a novel method that maximizes sparsely labeled neurites morphological continuity in a user-selected region-of-interest (ROI). We estimate the trajectories of severed neurites at interfaces between slices, using an ellipsoid as the approximate location where these neurites continue in the adjacent section.

The contributions (to the best of our knowledge) of this paper are as follows:

- First end-to-end application for reconstructing and visualizing neurites in densely-structured WFM images.
- Novel 3D Segmentation Toolbox for streamlining segmentation of neurites with features including brushing, erasing, optical flow, snap, gamma correction, skeletonizing.
- Novel CNN model for segmenting the neurites in low-resolution densely structured WFM images.
- Novel algorithm for registering depth-adjacent brain sections using a coarse-to-fine sequential process.
- First hybrid visualization technique that combines segmentation results with the raw input volume.

2 RELATED WORK AND BACKGROUND

2.1 Biological Background

The human brain has 80–100 billion neurons, and the nervous system groups neurons into different neurite morphology. Studies have shown that in mice brain, an axonal arbor of a single cholinergic neuron, including its terminal branches, is as long as 30 cm [3]. Given the extensive branching of cholinergic projections, conventional specimen preparation and imaging techniques make it difficult to analyze their full expanse and intricate features. Beyond the genetic labeling novelty, a 3D reconstruction of the circuitry is required for understanding the cholinergic connectome.

2.2 Segmentation

Based on the motivation behind the scientific investigation, visualizing neuronal structures is more significant than rendering voxel intensity values of raw volumes. Our previous work [2] presents a preprocessing method for meaningful rendering of neurons. However, a more robust solution (e.g., neuron segmentation) is required for extracting neurites for visualization and registration purposes.

Neuron segmentation is a challenging task in neurobiology, due to the low quality of images and high complexity of neuron morphology. To tackle this challenge, a number of manual or semi-automatic segmentation tools have been developed, such as NeuroLucida [4], V3D [5], ManSegTool [6],

SIGEN [7]. These tools rely on tedious manual operations, making the segmentation of complex neurites in large volumes nearly impossible.

Since neurons have a branching tree structure, many methods have been hired for tracing dendritic branches and axonal trees, such as optimal seed-points pathfinding [8], [9], [10], model fitting [11], [12], fast marching [13], [14], and distance-tree hierarchical pruning [15]. Most of these methods require an ideal manually- or automatically-generated set of seeds. Manual marker placement is tedious, while automatic seed generation is greatly affected by image low quality, noisy patterns, and broken structures.

Recently, a few learning models have been developed to automatically trace neurons in OM images. Chen *et al.* [16] trained a self-learning method using user-provided neuron reconstructions. Li *et al.* [17] hired 3D CNNs for segmenting neurites, which suffer from relatively long computation. Zhou *et al.* [18] developed an open-source toolbox, Deep-Neuron, with a 2D CNN followed by 3D mapping. Many of these methods have shown to perform well in segmenting a single neuron on high-resolution images. However, they cannot faithfully reconstruct complex neuron morphology in images with medium to low quality.

There is also a vast amount of research on segmenting neuronal membranes in EM images. Deep learning models have shown an outstanding performance in automatic neurites segmentation [19], [20], [21]. However, due to the limited availability of ground-truth data, they suffer from over- and under-segmentation. Haehn *et al.* [22] developed desktop applications for proofreading the automatic algorithm segmentations. Unfortunately, these methods are unapplicable to WFM images due to the differences in neuron visual representation, details level, and images quality.

2.3 Registration and Alignment

Volume reconstruction for OM images of brain specimens utilizes intensity- or feature-based methods. Intensity-based approaches select a pair of representative images from adjacent sub-volumes and compute a correlation measure to estimate their relative spatial registration [23], [24], [25]. These methods do not enhance registration accuracy at a finer morphological scale. Also, imaging artifacts, uneven contrast, and large datasets are potential bottlenecks for these methods. Feature-based methods use specific structures knowledge, which needs preprocessing for producing geometrical features as registration landmarks. Landmarks registration methods [26], [27] are fast and scale up easily with higher-order transformation models. Tsai *et al.* [28] presented microscopy slices montage synthesis by utilizing generic alignment cues from multiple fluorescence channels without landmarks segmentation. Yigitsoy and Navab [29] proposed tensor voting based structure propagation for multi-modal medical images mosaicing.

Lee and Bajcsy [30] registered and reconstructed depth adjacent sub-volumes acquired by a confocal microscope by connecting the 3D trajectories of salient cylindrical structures at the sub-volume boundaries. This method is ineffective on sparsely labeled samples due to the lack of continuous structures to be segmented for the proposed trajectory fusion. Dercksen *et al.* [31] proposed the alignment of filamentous structures by tracing the filaments and matching the traced

endpoints to find an optimal transform. NeuronSticher [32] aligns neurite tips at stack boundaries using triangulation. However, these tips are identified from neuron-tracing reconstructions, relying on tip selection.

2.4 Visualization

Recently introduced tools for the reconstruction, visualization, and analysis of complex neural connection maps enable neuroscientists to gain insights into the underlying brain morphology and function. We refer the reader to a survey [33] of techniques for macro-, meso-, and micro-scale connectivity visualization for connectomics. Volume rendering has been developed for 3D reconstruction and visualization of brain microscopy images. Mosaliganti *et al.* [34] developed axial artifacts correction and 3D reconstruction of cellular structures from OM images. Nakao *et al.* [35] discussed interactive visualization and proposed a TF design for two-photon microscopy volumes based on feature spaces. Wan *et al.* [36] described an interactive rendering tool for confocal microscopy data, combining the rendering of multi-channel volume and polygon mesh data.

Beyer *et al.* [37] presented ConnectomeExplorer for interactive 3D visualization and query-guided visual analysis of large volumetric EM datasets. Hadwiger *et al.* [38] designed scalable multi-resolution virtual memory architecture for visualizing petascale volumes imaged as a continuous stream of high-resolution EM images. Haehn *et al.* [39] developed a scalable platform for visualizing registration parameters and steps for fine-tuning the alignment computation, visualizing segmentation of 2D nano-scale images with overlaid layers, and interactive visualization for proofreading EM images. Neuroglancer [40] is a WebGL-based visualization framework for volumetric EM data. These methods are designed specifically for confocal, two-photon, or EM data. When applied to WFM, they do not yield qualitatively accurate neuron projections. Our previous work [2] discussed the challenges related to WFM volume visualization and introduced a workflow for its meaningful rendering.

3 NEUROCONSTRUCT OVERVIEW

NeuroConstruct is an end-to-end application for neuron reconstruction and visualization for WFM images. It consists of four main components: Segmentation of neuronal structures, proofreading reconstructed structures, registration of brain sections, and visualization of the reconstructed neurons and raw data using hybrid volume rendering. We present a fast and efficient ground-truth data generation pipeline. NeuroConstruct provides an interactive Segmentation Toolbox for automatically segmenting neurons and proofreading the segmentations. It also renders the reconstructed volume by combining the segmentation results with the raw input volume. NeuroConstruct also presents a Registration Toolbox for automatic coarse-to-fine registration of depth-adjacent 3D brain sections.

NeuroConstruct enables neuroscientists to study brain sections of interest thoroughly. After the whole brain section is acquired, the user can follow a 4-step pipeline (Fig. 2): (1) automatically and manually coarse-aligning whole brain sections using the Registration Toolbox, (2) automatically and manually fine-aligning an ROI from the coarse-aligned

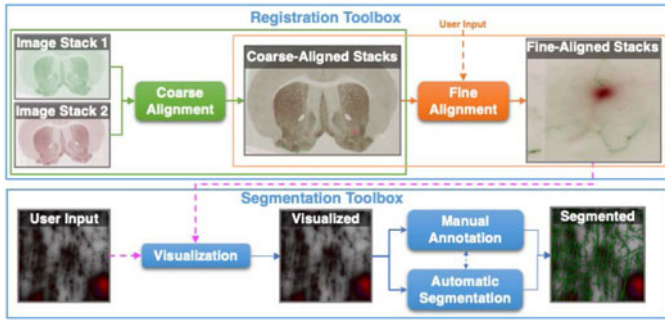


Fig. 2. Overview of the application. After acquiring a whole-brain section, experts can use NeuroConstruct to perform a thorough analysis of the neuronal structures. They can follow the pipeline shown in the figure. In the registration toolbox, given depth-adjacent whole brain sections, NeuroConstruct automatically coarse-aligns the sections. Then, users have to select an ROI to start the fine-alignment process. NeuroConstruct provides features for manual refinement of coarse and fine registration and allows users to save the coarse-aligned brain sections and finely registered region for further analysis. Next, users can load the registered ROI in the segmentation toolbox to visualize and segment the neuronal structures manually and automatically using the manual annotation features and automatic segmentation methods.

stacks using the Registration Toolbox, (3) visualizing the 3D and 2D views of the fine-aligned stacks using the Segmentation Toolbox (4) automatically and manually segmenting the neuronal structures using the provided functions in the Segmentation Toolbox. These result in a complete and precise reconstruction and visualization of neuronal data.

Users can also choose to use each component, including coarse-alignment, and fine-alignment in the Registration Toolbox, and visualization, segmentation, and manual annotation in the Segmentation Toolbox independently without the need to go through the whole pipeline. The components individual usability helps experts to perform the desired analysis, including visualization, segmentation, and registration on any WFM of interest independently or go through certain steps in the pipeline in the desired manner.

Both segmentation and registration can be done on the whole brain sections. However, because of the high density of neurons and the large size of the brain sections, it takes a tremendous amount of time to segment the neurons in a brain section, and neuroscientists might not infer useful information by analyzing a large segmented brain section.

4 DATA PREPARATION

4.1 Biological Prep

To train and test our application, we used two types of samples generated by our neuroscientists (Institutional Animal Care and Use Committee approval# 1618). For segmentation components, we used densely labeled samples obtained from a transgenic mouse line where a fusion protein of tau and a green fluorescent protein was under the control of the ChAT promoter (ChAT-tauGFP mouse). This allowed for labeling all cholinergic fibers and cell bodies throughout all the slices within the sample. For registration components, we used sparsely labeled samples obtained from a knock-in mouse where cre-recombinase expressed exclusively in cholinergic neurons (ChAT-IRES-Cre mice). Mice were transcardially perfused with 4% PFA, and brain tissue was harvested and sucrose equilibrated for cryosectioning. Samples were

serially sectioned at $20\ \mu\text{m}$ thickness and serially imaged on a WFM scanner (*Olympus VS-120*). Imaging was conducted with a $40\times$ objective with z-step of $1\ \mu\text{m}$. WFM images are volumes obtained by focusing at different depths of a thinly sliced specimen, which means that each section is an image stack of 2D slices.

4.2 Ground Truth Generation

To generate ground-truth of reconstructed neurites in 3D WFM images, we annotated neurites in several brain section regions using automatic segmentation and manual refinement. Due to a large number of neurites in brain samples, generating a ground-truth dataset is nearly impossible. To solve this, we have developed a novel approach for efficient neuron segmentation in image stacks. We used a small set of 2D strips to train a 2D CNN model for generating preliminary neurite segmentations to be proofread using the Segmentation Toolbox. This technique speeds up the ground-truth generation process significantly. Our data generation pipeline consists of 3 main steps:

Step 1 (2D-Strip Annotation). In WFM images, neurites in $x-y$ plane have a tree-shaped structure. In $y-z/x-z$ planes, neurites are bright blobs with light projections going outwards (Fig. 1). Due to neuron high density and image low SNR, distinguishing the weak neurons from the background in $x-y$ plane is hard and neuron annotation in this plane is time-consuming. Therefore, we used $x-z$ and $y-z$ views for neuron annotation. Two neuroscientists spent about 40 hours each to annotate one hundred of 2D strips of $y-z/x-z$ planes with approximately 30 neurons in each strip resulting in a set of 3000 individual neuron representations.

Step 2 (Preliminary Segmentation). We used the annotated strips to train a U-Net [41] based network consisting of contracting and extracting paths. The contraction path consists of five downsampling components with two 3×3 convolutions, each followed by a parameter rectified linear unit (PReLU) and a 2×2 max pooling with stride 2. We used dropout layers with a rate of 0.5 after the last two components to reduce overfitting. The expansive path consists of feature map upsampling followed by a 2×2 convolution, merged with the correspondingly cropped feature map from the contracting path, and two 3×3 convolutions, each followed by a PReLU. Last, a pixel-wise softmax applied on the resultant image followed by a Poisson loss function to classify pixels into neurite or background.

Each 2D strip size is 25×512 , which is the number of adjacent stacks in z -direction (depth) by volume width or length. We train the network with input images (512×512) tiled with side by side strips with spacing. It makes the segmentation process approximately 15 times faster than feeding the network, a single strip per image. Our dataset with 3000 neuron representations introduces to the network a large variety of shape, size, and intensity of neurons. To infuse more spatial neuronal information, we augmented the strips by applying flip and zoom transformations. We trained our network with 300 images, each containing 15 individual augmented 25×512 strips. Under-segmentation is a major concern, which is the artifact of lack of neurons projection traveling in z -direction in $x-z$ and $y-z$ planes.

Step 3 (Refinement). We generated a 3D neuronal segmentation dataset using the trained model in Step 2 and proofread

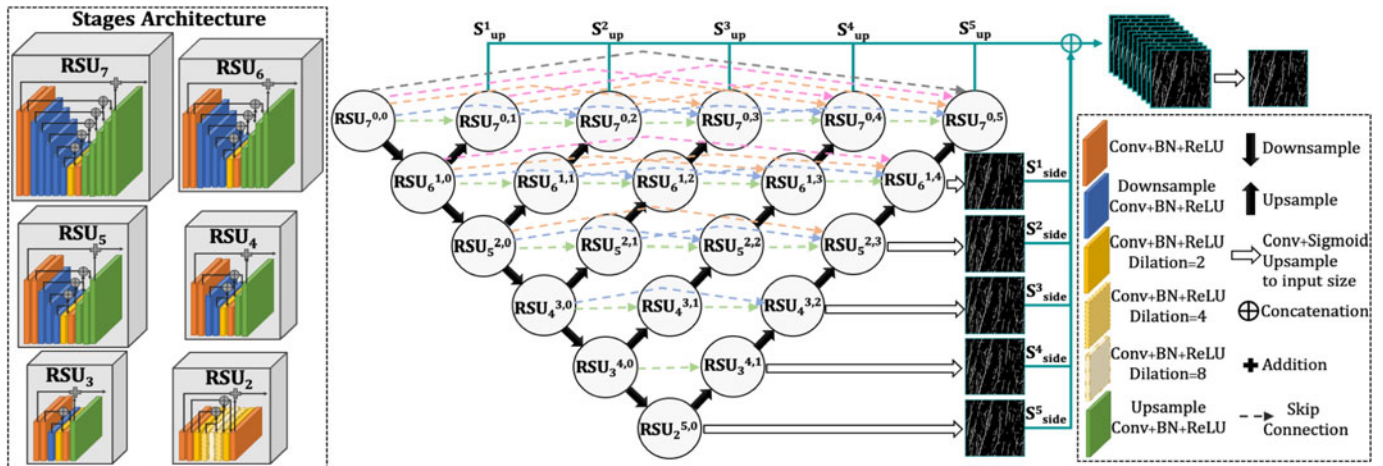


Fig. 3. Architecture of the 3D CNN segmentation model. It consists of contracting and expansion paths. The network input is a batch of grayscale images and its output is a probability map of the same size. Each output pixel represents the probability of the input pixel being part of a neurite.

the segmented results to create an accurate ground-truth set using the Segmentation Toolbox (explained in Sec. 6).

5 SEGMENTATION

The neuron segmentation process consists of two steps: Neurite segmentation and soma segmentation. We segment neurites using our designed model. For segmenting somas, we use a thresholding technique combined with the segmented neurites as a guide. This section describes the neuron segmentation process, including the dataset and the proposed model, and soma segmentation technique.

5.1 Data

We created a ground-truth dataset of WFM image stacks for training and testing purposes. We cropped six regions of size $25 \times 512 \times 512$ randomly from the brain medial septum region with the size of $25 \times 48000 \times 33000$ and annotated using the pipeline explained in Sec. 4.2, to be used as ground-truth. This dataset covers many variations in neuron morphology due to the large size of the cropped regions and the high density of neurons. The segmentation model was trained, tested, and validated on 3, 2, and 1 image stacks, respectively. We also conducted a qualitative analysis to evaluate the accuracy of reconstructed neurites, performed by domain experts. Using our trained model, we segmented neurons for eight image stacks (never before seen by the model) with sizes varying between $25 \times 512 \times 512$ and $21 \times 1024 \times 1024$ randomly cropped from the medial septum and cortical sections of five different brains.

5.2 Proposed Network for Neurite Segmentation

We propose a nested encoder-decoder network with re-designed skip pathways for connecting the encoder and decoder sub-networks (similar to U-Net++ [42]) for a precise semantic segmentation, and stacking U-structure (similar to U²-Net [43]) for salient object detection. Fig. 3 shows an overview of the network architecture. Our network is a two-level nested U-structure consisting of 21 stages. Each stage is configured with a ReSidual U-block (RSU) introduced by [43]. In our network, the feature maps follow a

path through a dense stage block to fuse the output from the previous stage layer of the same dense block with the corresponding up-sampled output of the lower dense block. The re-designed skip pathways enable more similar semantic level feature maps between the encoder and decoder, making the optimization problem easier, and the use of RSU blocks enables a more efficient extraction and aggregation of intra-stage multi-scale contextual features.

Architecture of the Network. Our network consists of 21 stages. Each stage is formed from an RSU block with a specific height. We represent each stage as $RSU_L^{i,j}$, where i is the index of the down-sampling layer along the encoder of the big U-structure, j is the index of the up-sampling layer along the decoder of the big U-structure, and L is the number of encoder layers in the U-Net-like structure of the RSU block, except in the stage with $L = 2$. Since the resolution of feature maps in the last encoding stage is relatively low, further down-sampling results in the loss of useful information. Therefore, in the last encoder stage ($L = 2$), we only use dilated convolutions, having the same resolution as its input feature maps. As shown in Fig. 3, our network consists of a sub-network of encoder stages which is the backbone of the network, a sub-network of decoder stages, skip pathways, and a saliency map fusion module. The fusion module is responsible for generating the probability map. The network generates five side output saliency probability maps $S_{side}^5, S_{side}^4, S_{side}^3, S_{side}^2, S_{side}^1$ from stages $RSU_2^{5,0}, RSU_3^{4,1}, RSU_4^{3,2}, RSU_5^{2,3}, RSU_6^{1,4}$ by a 3×3 convolution layer, up-sampling layer and a sigmoid function, and five top output saliency probability maps $S_{top}^1, S_{top}^2, S_{top}^3, S_{top}^4, S_{top}^5$ from stages $RSU_7^{0,1}, RSU_7^{0,2}, RSU_7^{0,3}, RSU_7^{0,4}, RSU_7^{0,5}$ by a 1×1 convolution layer and a sigmoid function. Then, the final saliency map S_{fuse} is generated by concatenating all side and top output saliency maps, followed by a 1×1 convolution layer and sigmoid function.

RSU Block. Each block consists of 3 main components: (1) Input convolution layer, transforming the input feature map x to an intermediate map $\mathcal{F}(x)$ for local feature extraction. (2) U-Net based encoder-decoder structure with input of the intermediate feature map $\mathcal{F}(x)$ that extracts the multi-scale contextual information $\mathcal{U}(\mathcal{F}(x))$, where \mathcal{U} represents the

U-Net-like structure. The structure with larger height results in richer local and global features. (3) Residual connection fusing the output from the local features $\mathcal{F}(x)$ and multi-scale features $\mathcal{U}(\mathcal{F}(x))$ using the formula: $\mathcal{F}(x) + \mathcal{U}(\mathcal{F}(x))$. RSU blocks enables capturing of the fine details that might be lost by direct up-sampling with large scales.

We define RSU blocks as $\mathcal{G}_L = RSU-L(C_{in}, M, C_{out})$, where C_{in}, C_{out}, M are respectively the sizes of input channel, output channel, and RSU internal layers channel. L is the number of layers in the encoder of the RSU block, except for $L = 2$. As mentioned earlier, due to the low resolution of feature maps in the last encoding stage, we use dilated convolutions instead of down-sampling. So, in this stage, the defined L do not denote the number of layers in the encoding stage. We formulate the skip pathways as follows:

$$RSU_L^{i,j} = \begin{cases} \mathcal{G}_L(RSU_{L+1}^{i-1,j}) & j = 0, \\ \mathcal{G}_L([\![RSU_L^{i,m}]_{m=0}^{j-1}, \mathcal{UP}(RSU_{L-1}^{i+1,j-1})\!]) & j > 0, \end{cases} \quad (1)$$

where function $\mathcal{UP}(\cdot)$ denotes an up-sampling layer, and $[\]$ denotes the concatenation layer. Stages at level $j = 0$ receive one input from the previous layer. Stages at level $j > 0$ receive $j + 1$ inputs, where j input(s) are the output(s) of the previous j stage(s) in the same pathway and the last input is the up-sampled output from the lower pathway.

Training. The training regiment is as follows:

- *Input Data.* We trained the network with 512×512 images. To add more variation in neuron morphology, we applied elastic deformations (e.g., random rotation and scaling), resulting in 540 images of size 512×512 .

To generate a universal model for segmenting neuron morphology that encompasses images captured from different sections of the brain using any imaging techniques, we adjust the pixel intensities through Eq. 2:

$$Adjusted_k = \left(\frac{I_k - \min I_k}{\max I_k - \min I_k} \right)^\gamma \quad (2)$$

where k is the image index in the image stack, I_k is the intensity array of the k th image in the stack, $\min I_k$ and $\max I_k$ are respectively the minimum and maximum intensities in the corresponding image, and γ is the intensity adjustment parameter. Using Eq. 2, we map the intensity range to $[0, 1]$, leading to a universal WFM model.

- *Parameters.* We trained the network using Adam optimizer with a batch size of 4 for 2D training for 100 epochs. The initial learning rate was set to 0.001. We define the loss function as:

$$\mathcal{L} = \sum_{i=1}^N w_{top}^n \ell_{top}^m + w_{side}^n \ell_{side}^n + w_{fuse} \ell_{fuse} \quad (3)$$

where $N = 5$, ℓ_{side}^n and ℓ_{top}^m denote the loss of the side output saliency map S_{side}^n and the top output saliency map S_{top}^m , w_{side}^n and w_{top}^m denote their weights, and ℓ_{fuse} is the final fusion output saliency map S_{fuse} with its corresponding weight w_{fuse} . We use binary cross-

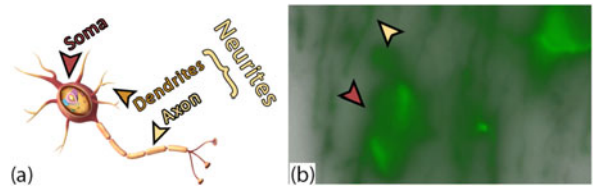


Fig. 4. (a) Neuron structure. (b) Region cropped from the brain medial septum. A soma is a bright green blob.

entropy loss to calculate loss of each output (ℓ), defined as:

$$\ell = - \sum_{i=0}^M y_i \log(y'_i) + (1 - y_i) \log(1 - y'_i) \quad (4)$$

where M is the number of samples, y_i is the label for i , and y'_i is the predicted label for sample i . It takes 186 min. to train the network with defined parameters using the training set of 540 images with a size of 512×512 .

- *Output Data.* The model generates images of the same size as the input (512×512). Each output pixel represents the probability of the input pixel being part of a neurite, showing the network confidentiality in classifying the pixel as foreground (with close to zero for background pixels). Weak neurites have lower probability than strong ones. This probability map enables experts to study neuron morphology in detail, as they can study weak and strong neurites in one whole structure or independently.

5.3 Soma Detection

Soma is the spherical neuron part containing the nucleus. Fig. 4a shows the neuron structure. As shown in Fig. 4b, soma has higher contrast than neurites. This feature allows us to segment somas using thresholding techniques without the need to create a ground-truth dataset, training networks, and spending time and resources. Using this feature, we extract the somas from the image stacks by thresholding the voxel intensities using multi-level otsu's method [44]. We consider two threshold levels (t_1, t_2) to classify the image into three classes; the background voxels are in the range 0 to $t_1 - 1$, the neurite voxels most probably are in the range t_1 to $t_2 - 1$, and the soma voxel (which are the brightest voxels) are in the range t_2 to 256. Some strong neurites might have high contrast similar to somas resulting in misannotation. We use the generated segmentation mask by the model consisting of dendrites and axons to remove the segmented neurites that are mistakenly segmented as somas.

6 SEGMENTATION TOOLBOX

NeuroConstruct provides a Segmentation Toolbox for segmenting and visualizing neurons in small crops of WFM image stacks, as our neuroscientists required a toolbox that visualizes neurons in fine detail with the highest possible resolution enabling them to study complex brain morphology. As shown in Fig. 5 and the accompanying video, the Toolbox offers three options and various features for efficient and accurate neurite segmentation: (1) annotating neurites

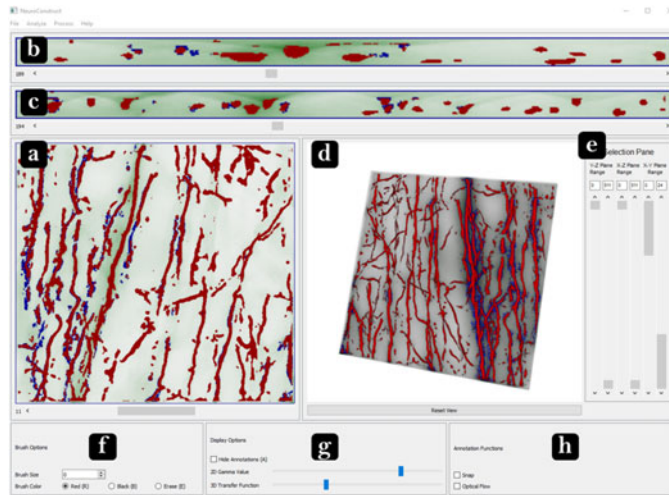


Fig. 5. Segmentation Toolbox screenshot. The main menu has four options: *File*, *Analyze*, *Process*, *Help*. *File* option provides loading image stacks and/or annotations, and saving annotations (shown in red). *Analyze* option includes the Skeletonize feature. *Process* options are 2D and 3D segmentation and registration of the image stacks. *Help* menu has instructions on Toolbox use. (a) 2D cross-sectional view of the volume in $x-y$ plane. (b) 2D cross-sectional view in $x-z$ plane. (c) 2D cross-sectional view of the volume in $y-z$ plane. (d) 3D volume of a WFM image stack. (e) Area selection frame to specify an ROI to focus on by adjusting corresponding sliders. (f) Brush options frame including size, color, and eraser. (g) Display options frame containing gamma value selection sliders for 2D and 3D views and annotation display options. (h) Annotation functions frame, containing the helper functions for neurite annotation, including snap and optical flow features.

from scratch, (2) loading an annotation file and refining the annotations, (3) selecting NeuroConstruct Segmentation method and proofreading the results.

6.1 Visualization of Image Stacks

The Toolbox provides four simultaneous views of the image stacks to assist neuroscientists in effectively and confidently mark neurites in the raw WFM volume: The first three are 2D cross-sectional views in the $x-y$, $x-z$, and $y-z$ planes. The fourth view is rendering of the 3D volume and iso-surface of annotated neurites. These four views provide a comprehensive exploration of the image stacks and no limitation in visualizing and annotating any structure.

6.2 Annotation Features

The user can annotate the neurites and somas with two different colors, red and black. When the user draws over any 2D cross-sectional slice, all views are updated simultaneously to follow the user's drawing. The Toolbox allows the user to zoom in and out in all views. The user can specify an ROI to focus on. In the 2D views, the Toolbox bounds the selected ROI with a blue rectangle, and in the 3D view, it only renders the specified ROI. If a user prefers to manually annotate an image stack from scratch or manually refine an automatically segmented stack, the Toolbox offers novel features to ease the manual annotation process. We describe below some of these functions:

Optical Flow. This feature helps the user to annotate neurites more efficiently. As shown in Fig. 1, a neurite in $x-z$ and $y-z$ planes is a circular bright region. When the user clicks on a neurite, the Toolbox finds the smallest 2D region

surrounding the selected neurite and applies the Pyramidal Lucas-Kanade optical flow algorithm [45]. Then, it finds the most similar regions (based on pixels intensities) to the corresponding neurite in the following image slices. Roberts *et al.* [46] hired a similar approach for estimating optimal volumetric pathway through the image stack connecting the computed 2D segmentations using dense optical flow. Their approach requires 2D segmentation on the first and last image slices to compute the minimum distance from pixels in these slices, while we compute the neuron path using the user-specified segmentation on the first slice and find the most similar regions in the following slices.

Snap. A neurite might be visible in several image slices with different levels of contrast and brightness. However, a pixel is considered a neurite if it is unblurry and has higher contrast than pixels in adjacent slices. In some cases, due to the low contrast between neurites and the background and image blurriness, it is challenging to select the correct slices containing the neurite. The Toolbox helps the user to find those slices. The user specifies the ROI containing the corresponding neurite in $x-y$ plane and clicks on the neurite in one of the slices. The Toolbox searches for the sharpest point with the highest contrast in the specified slices and selects the 2D region with the highest intensity in a virtual cylinder perpendicular to the view plane from the corresponding point with a brush size diameter.

3D Skeleton. The Toolbox employs user's annotation to generate a 3D skeleton (cf. [47]). The algorithm defines an octree over the annotations and examines $3 \times 3 \times 3$ neighborhood of voxels. It iteratively proceeds until only strips of voxels are left. This method outperformed various methods we tested for 3D skeleton generation, as they failed to create the skeleton of weak neurons with thin regions.

Gamma Correction. The user can change the contrast of 2D views using the corresponding slider. It transforms linear color mapping to a non-linear space and adjusts the contrast using the user-specified gamma values.

TF. The user can change the contrast and color of the rendered volume using a slider. To adjust the rendered volume contrast, the Toolbox uses the following novel TF:

- $\gamma = 0$: Defining n bins with equal length for the TF, where n is the number of bins.
- $0 < \gamma \leq 1$: Breaking the TF into bins with length defined as $a_i = 2^\gamma \times a_{i-1}$, where a_i is the i th bin, and $\sum_{i=0}^n a_i = 1$.

We empirically found that $n = 20$ works well on our data. We provided our experts with our TF and gamma correction methods applied to the same image stacks. After a thorough comparison between these techniques, they remarked that our TF outperforms gamma correction in terms of neurite visual representation in blurry images with low contrast.

6.3 Visualization of Automatic Segmentation

The primary information neuroscientists expect to visualize in brain WFM volumes are neuronal structures – namely, tubular-shaped neurites and blob-like somas. The inherent challenges of blurred WFM data and the absence of distinctive set of voxel intensity values that can differentiate foreground (neuronal data) from background (blurring artifacts and brain tissue) makes the task of adjusting visualization

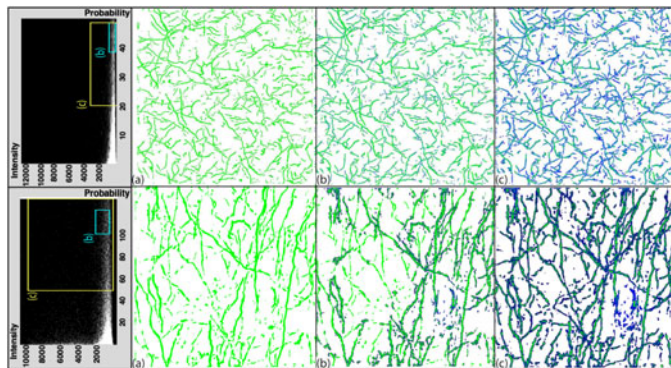


Fig. 6. Neuronal data hybrid visualization. Top row shows the cortical region and bottom row shows the medial septum region. Green represents iso-surface rendering of the automatic segmentation result, and blue represents direct volume rendering of raw data using 2D TF mapping. For both datasets: (a) Structural mode: iso-surface rendering of the high-confidence voxels. (b)-(c) Hybrid Fusion mode: the iso-surfaces in (a) combined with direct volume rendering of the raw data using two different ranges selected on the 2D TF mappings shown on the left.

parameters or directly applying surface rendering techniques to the raw volume tedious and disfavored by neuroscientists. Finding optimal visualization parameters for rendering both high- and low-intensity neurites while suppressing similar intensity-valued blurring artifacts becomes more complicated in densely populated samples. Thus, we use the output of our segmentation pipeline as an integral component for visualizing neurons.

The robustness of deep learning models greatly depends on the accuracy and availability of sufficient ground-truth training data. For training, we attempted to accurately annotate neurites of varying intensity, trajectory, morphology, and biological labeling. However, using real data, there is large biological diversity and within biologists' workflow there are many experimental variations from data preparation methods to imaging modality input parameters and conditions. Thus, in a practical scenario, preparing a training set covering all variations is an ambitious goal to achieve.

In NeuroConstruct, we combine the ability to use visualization methods to render essential features within a dataset and the binary mask from our segmentation model as two complimentary inputs to what we term as, hybrid visualization. Using hybrid visualization, users can recover missing neurites by exploring the correlation between the raw volume scalar values and the segmentation confidence score for each voxel, determined by our model, which we represent as a 2D TF. Given a 3D image stack, NeuroConstruct first segments each image stack, and then combines them to create the final 3D segmentation mask corresponding to the output. The computed segmentation mask is a volume, where each voxel represents a confidence score between 0 and 1, where 0 is background and 1 represents a neurite with full confidence. Using this score, we provide two visualization modes, structural and fusion (see Fig. 6 and Fig. S1, which can be found on the Computer Society Digital Library at <http://doi.ieeecomputersociety.org/10.1109/TVCG.2021.3109460>, in Supplementary Material). The structural mode renders the segmentation mask as an iso-surface. Using a slider, users interactively choose a minimum confidence score as the minimum threshold value for reconstructing an iso-surface using marching cubes.

To visualize possible neurite structures in the raw volume with scores below the threshold confidence, we offer the hybrid fusion mode that simultaneously renders iso-surfaces extracted in the structural mode along with selected voxels from raw volume, using a 2D TF that represents the raw volume scalar intensity versus segmentation confidence score. Users can draw a rectangular region on the 2D TF for direct volume rendering (DVR) of the selected voxels, as in Fig. 6, and select points on the 2D TF to render iso-surface of the selected voxels. By observing domain experts, we determined to use structural rendering along with DVR with the 2D TF as a default preset, as it is more effective to select an entire voxel range. One limitation of such a correlation is that we assume that the model outputs a level of reasonable confidence for neuron voxels, compared to background and noise voxels. A future work to overcome this limitation can be model re-training using a refined set of annotations generated by this hybrid visualization mode.

7 REGISTRATION

In an ideal scenario, undisturbed imaging of biological samples would allow for the best reconstruction of cholinergic axonal projections. However, current experimental methods for imaging using a WFM require the physical sectioning of brain specimens. To this end, NeuroConstruct presents a coarse-to-fine pipeline for the registration and alignment of neuronal fibers across the sliced sections.

7.1 Coarse Registration

In neuroscientists' workflow, physically sectioned brain slices specimen are placed on a glass-slide at arbitrary positions and orientations and imaged separately using WFM. Therefore, an intuitive initial step is to coarsely register the individual brain section volumes. For coarsely registering the image stacks, we tried several methods (e.g., intensity-based [23], [24], [25]) that perform well in registering slices at a coarse level, but they do not align neurites within slices at a finer, morphological scale. We also attempted to adapt the montage/mosaic stitching alignment (primarily used to align microscopy tiles in x - y direction) [28], [29]. Our method draws insights from the tensor voting method [29].

We take as input a series of depth-adjacent sections and use the tissue structure to estimate a global rigid-body transformation between adjacent sections. The series of 3D sections have to be transformed into a single reference coordinate system. So, as an initialization step, users need to specify a reference brain section. Using the referenced section, we determine the slice representing the tissue largest spatial extents and consider that slice image space as the reference coordinate system for subsequent transformations.

For estimating rigid transformations between sections, the interfacing z -slices from each depth-adjacent section pair are used as representatives for the coarse registration. Some approaches suggest using slices with the highest contrast within the 3D sections [30], [48], as the end slices usually have low signal values. However, since NeuroConstruct registers several sections, we expect changes in the brain outer boundary across the sequence. Thus, to avoid cascading errors, we choose interfacing slices with the smallest variation in the outer boundary between adjacent sections.

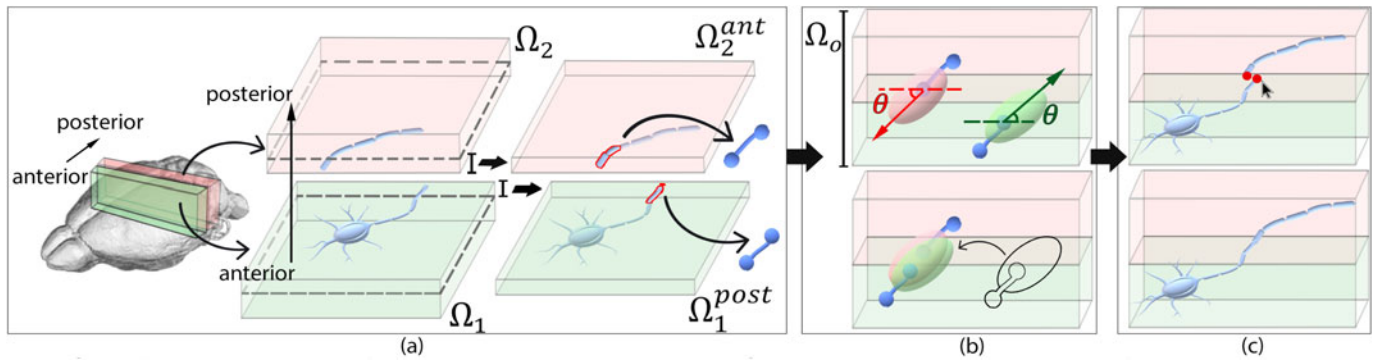


Fig. 7. Steps of our neurite alignment (fine registration) in two depth-adjacent ROI sections. (a) We determine a subset of neurites whose trajectory suggests continuity beyond the interface of its section. We then estimate a direction of its propagated trajectory into the depth-adjacent section and define an ellipsoidal region around its end-point, representing a 3D space of possible continuing locations, shown in the top of (b). Then, we use ICP on the ellipsoids point cloud representation to estimate a rigid-body transformation, as shown in the bottom of (b). Finally, we present the user with an interface to verify the results and correct for any misaligned neurites (c).

Finally, a rigid-body transformation is estimated using Mattes mutual information (MI) [49] with a regular-step gradient descent optimizer. MI for intensity-based registration uses joint probability distribution of pixel samples between images and define a mapping for a set of pixels based on value similarity by reducing the entropy between the distributions, signaling the images are likely better aligned. This method is well-suited for our coarse alignment as the brain tissue outer-boundary or internal regions vary in perimeter across the anterior to posterior, and geometry-based registration will not yield effective results. In applying MI, we adopted a multi-resolution approach to avoid reaching a local minimum due to noise and tiling artifacts in a sparsely labeled sample. We doubled the step size for each subsequent higher resolution to avoid the optimizer from focusing on progressively smaller regions.

7.2 Fine Alignment

Following coarse registration, we utilize the coherency between neurites geometric structures across depth-adjacent sections. Our fine alignment method has 3 major steps: (1) ROI selection, (2) automatic neurite trajectory propagation, and (3) automatic trajectory alignment. The main idea is to adopt a feature-based registration to maximize the neurites morphological continuity in neighboring brain sections.

ROI Selection. A critical limitation of micrometer resolution microscopy images of brain samples is its large spatial extent. For a computationally faster and memory-efficient implementation of the fine alignment method, we ask the user to mark an ROI.

Neurite Trajectory Propagation. Common methods for registering microscopic images introduce fiduciary landmarks during image acquisition, which are then registered to reconstruct the complete volume. However, this adds complexity to neuroscientists' workflow. In our fine alignment approach, we have developed a novel method that uses linearly extrapolated neurites trajectories to infer their corresponding continuity, beyond the section slicing interface. This correspondence between neurites is used to estimate the necessary transformation parameters that spatially align depth-adjacent sections. To formally define our approach:

- Let Ω_1 and Ω_2 be two adjacent sections, where the z -slices order from Ω_1 to Ω_2 are in anterior to posterior direction.
- Let Ω_1^{post} and Ω_2^{ant} be the z -slices corresponding to the posterior sub-section of Ω_1 and the anterior sub-section of Ω_2 , respectively. In our implementation, the extracted thickness of Ω_1^{post} and Ω_2^{ant} are set to be $5 \mu\text{m}$.
- Let Ω_o be the region defined by $\Omega_1^{post} \cup \Omega_2^{ant}$. This is essentially the interfacing region Ω_1 and Ω_2 .

Broadly, we determine a subset of neurites from Ω_1 and Ω_2 whose trajectory within the section suggests continuity beyond the section interface and establish a correspondence between the candidates with a similar trajectory in the depth-adjacent sections. Using this information, we solve the alignment problem in Ω_o , by maximizing overlaps between linearly extrapolated trajectories of the neurites. Fig. 7 provides an illustration of our fine alignment algorithm.

First, for each two serial sections, we locate neurites with trajectories propagating into the other section and estimate a propagated trajectory direction. We first use the segmentation mask volume (see Sec. 5) to find a 3D line segment that passes through the connected-component voxels of each extracted propagating neurite from Ω_1^{post} or Ω_2^{ant} . Next, using the line segments, we determine propagation direction as a vector $v_{i,j}$, for each propagating neurite. Since the goal is to extrapolate the propagating trajectory, we did not consider the entire neurite in the section (Ω_1 or Ω_2) but rather limited to the z -thickness defined for Ω_1^{post} and Ω_2^{ant} .

Finally, each neurite trajectory propagation in the overlapping 3D space Ω_o is defined by an ellipsoidal region around its end-point close to the section interfacing slice, using the estimated vector $v_{i,j}$ as the major axis (see Fig. 7). The choice of an ellipsoid to represent the trajectory propagation is to accommodate possible neurite signal-loss and non-rigid deformation from physically slicing the brain. Because of hydration and dehydration of sample preparation, in addition to the uniform brain growing and shrinking, tissue distortion may occur. Therefore, we center the ellipsoid at the end-point of a propagating neurite to maximize the search space for estimating a transformation that aligns depth-adjacent sections. The ellipsoid major axis is aligned parallel to the estimated trajectory propagation

vector $v_{i,j}$, and the other two axes perpendicular to it. In our implementation, the ellipsoid major axis length is twice the thickness defined for Ω_1^{post} and Ω_2^{ant} and the other two axes length equals to the neurite thickness. This length ensures that the ellipsoid propagates to the depth-adjacent section.

Trajectory Alignment. After ellipsoid generation, the final step is to maximize propagating neurites alignment. We translate this problem into a simpler point-cloud registration, where each point on the ellipsoid surface represents a potential location for the continued neurite trajectory. By discretizing the ellipsoidal regions as 3D point clouds and estimating a rigid-body transformation using ICP, we apply the transformation to the entire section and concatenate with its depth-adjacent section to create an aligned sub-volume.

Each potential correspondence measure ω in the ICP estimation is assigned using $\gamma_{m,n} = \omega_1 D_{m,n} + \omega_2 |\theta_{1,m} - \theta_{2,n}|$, where m and n are the j th indices of the two sets trajectory vector, D is the euclidean distance between the two vectors start points, and $\omega_{1,2}$ are weights. $\omega_{i,j}$ is determined as the vector projection angle $v_{i,j}$ on the x - y plane (top in Fig. 7b).

To establish a potential one-to-one correspondence between the trajectories, we apply a weighted bipartite matching approach using the Hungarian algorithm [50] that minimizes the cost across all of the potential assignments. Using these, a transformation is measured based on Arun *et al.*'s method [26] that maps the propagated trajectories from one section to the other using $\mathbf{p}_{fixed}^{(i)} = \mathbf{R}\mathbf{p}_{moving}^{(i)} + \mathbf{T}$, where $i \in 1, \dots, N$ points, $\mathbf{p}_{fixed}^{(i)}$ is the i th fixed point, $\mathbf{p}_{moving}^{(i)}$ is the i th moving point, \mathbf{R} is a 3×3 rotation matrix, and \mathbf{T} is a 3×1 translation vector. In our implementation, we consider Ω_1 to be the *fixed* section if its position is anterior to the reference slice (determined in the coarse registration step, Sec. 7.1) and the *moving* section otherwise.

8 REGISTRATION TOOLBOX

NeuroConstruct offers a Registration Toolbox which provides coarse and fine alignment registration and visualization of brain sections. We conducted a study to find the proper way of visualizing large image stacks with GBs of data. We provided our neuroscientists with two visualization options: (1) visualization of the image stacks maximum intensity projection (MIP), and (2) visualization of the raw image stack combined with a slider for moving between the slices. Our experts preferred MIP visualization as they could verify the registration result faster and easier.

The Registration Toolbox offers sections coarse alignment and user-selected ROI fine alignment and visualizes the base, moving, and transformed moving image in the corresponding viewer (see Fig. 8). The user can then manually refine the registration parameters using the provided translation and rotation options. Rendering image stack MIP speeds up the manual alignment process as the transformation is only applied on the MIP images and enables an interactive user experience. For accelerating the computation, the final registration parameters are applied on the whole image stack, in two stages: (1) before starting fine alignment, and (2) while saving the user-requested registered images.

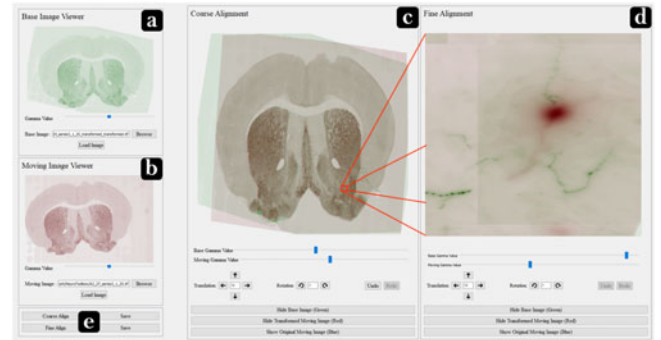


Fig. 8. Screenshot of the Registration Toolbox. (a) Base image viewer. (b) Moving image viewer. (c) Coarse alignment frame, showing the result. Users can adjust image contrast using provided sliders and refine the rotation or translation parameters to reach the desired registration. Using the provided buttons at the bottom, users can hide/show the base, moving, and transformed moving image to verify the registration result more precisely. (d) Fine alignment frame, showing the result on the user-selected ROI. It provides the same features as the coarse-alignment frame, including manual alignment parameters and display options.

9 IMPLEMENTATION

The NeuroConstruct consists of several components, Segmentation and Registration Toolbox, segmentation training models, coarse and fine alignment registration, and structural and hybrid visualization modes. The Segmentation and Registration Toolbox, fine alignment algorithm, and the structural and hybrid visualization modes were implemented using python, VTK [51] and Qt libraries. The coarse alignment module was implemented in C++ using ITK [52] and VTK [51]. The CNN model was implemented using python and Keras and trained and tested on a desktop with an NVIDIA Quadro RTX 6000 GPU, which was also used for all NeuroConstruct implementations.

10 EVALUATIONS AND RESULTS

10.1 NeuroConstruct Evaluation

We were in close contact with our neuroscientists to design our system, add features and refine functionalities based on their needs. These domain experts (co-authors) thoroughly evaluated the system using various scenarios. As the first use case, they followed the 4-step pipeline, starting by coarse-aligning, fine-aligning the brain sections through the Registration Toolbox, visualizing, and automatically and manually segmenting the neurons through the Segmentation Toolbox to reconstruct the whole WFM brain sections acquired from different regions of the brain. In other use cases, they used each component individually. After a careful review of NeuroConstruct, our neuroscientists found it “exclusive” and “specialized”. They said “the framework provides an end-to-end solution for segmentation, registration, and visualization of serially collected WFM images. Upon image acquisition, data can be registered using the Registration Toolbox, and then each individual image can be segmented for a clean representation of the signal to the background. At each of these steps, the critical component is an automated solution that can be entirely modified by user input. These user-updated segmentations on the registered volumes essentially enable tracing individual projections across a volume, facilitating the study of cholinergic neurons

affected by Alzheimer's disease." They also evaluated each component and features of the toolboxes separately (see Sec. 10.2 and 10.3).

10.2 Segmentation Toolbox Evaluation

Our neuroscientists used the Segmentation Toolbox to annotate and trace projections within WFM sections. They evaluated the Toolbox, including the hybrid volume rendering and annotation features, and compared the Toolbox with available tools. They believe that segmentation and volume rendering offer incomparable tools for neurite segmentation and subsequent visualization. Several features are unique to our Toolbox that our experts have not found in other programs they have used (e.g., ImageJ [53], FluoRender [36]).

Hybrid Volume Rendering Evaluation. Using the hybrid volume rendering, users visualize and have ultimate control over the auto-segmentation result. The Toolbox then provides a user-friendly interface for identifying signals from the background and improving the segmentation. Our experts believe that "both of these features allow users to visualize a full resolution image, which is critical during segmentation." Overall, they found this visualization ideal, as it overlaid the segmentation on the raw data, allowing users to adjust and visualize the segmentation based on what they desire to visualize in the raw. They also remarked "this allowed for segmenting different neuron types, initially strongest, and the option to include weaker ones."

Annotation Features Evaluation. The Toolbox offers an option to control the contrast of all views. Our experts found this feature helpful, saying that "the gamma slider allowed us to incorporate all fibers weak and strong, or just to annotate and understand morphological or trajectory in the strongest fibers." The sliders allow the user to choose an ROI to focus on and visualize it across X, Y, and Z individually. To improve upon or erode the annotation, the Toolbox provides a simple, user-friendly solution, a paintbrush-like feature. Our experts said "this intuitive tool makes it exceptionally easy for users to improve the segmentation on their own in fluid strokes, following a projection along its path. The ability to draw, trace, and erase on a slice by slice and pixel by pixel basis provided us complete control over how refined or simple we wanted the annotations to be, and the brush size option allows us to carefully follow and mimic the projection morphology and path." Our experts expressed their interest in *snap* by saying that "the toolbox offers a smart-adjustment to the simple paintbrush concept by updating the view as you draw to the point where the signal is sharpest for every click. This is especially helpful as projection paths in real data are hardly ever straight or within the same optical section but instead come in and out of view, making tracing them challenging!" Using the updated segmentation, a skeletonized image can be generated and overlaid onto the 3D rendering for simplified, clean visualization. Our experts remarked that "while the annotation feature allowed for visualization of neuron morphology, the skeleton view allowed for a simplified way to visualize neurite trajectory. These two features worked in complement to enhance 3D visualization."

10.3 Registration Toolbox Evaluation

Our experts registered and visualized brain sections using the Registration Toolbox and believe that it provides a

unique solution for serial rendering of WFM images. The base and moving images are loaded in two separate frames for initial coarse alignment. Our experts said: "the loaded images offer a good deal of flexibility and can be loaded as a maximum intensity projection of a 3D image (image contains X, Y, and Z information) or a sub-stack of the XYZ image." The Toolbox then computes coarse alignment and offers to move and rotate the image and visualize the base and moving image individually. Our experts believe that "this allows users control over the ultimate coarse alignment solution." Upon completion of coarse alignment, users can save it and, ultimately, the new volume (two registered images). This is in our experts' interest as it provides an essential intermediate point at which further registration can be done with the series next image, or the new volume can be separately inspected. At this point, images fine alignment can be conducted upon selection of a small relevant ROI. Similar to coarse alignment, the fine-alignment is visualized within the Toolbox and can be moved/rotated for ultimate control over the fine-alignment solution. The final aligned volume (coarse + fine) can be saved for further analysis.

Overall, our experts found a few features of this Toolbox as a unique solution for neuroscientists. "First, the toolbox completes segmentation of the images as a step before registration, which provides a complementary set of information to use in visualization and analysis. Second, the down-sampling conducted by the toolbox is a temporary solution for ease of working with large images. Upon saving, the solutions for both coarse- and fine-alignment are applied to each Z step within the stack, thereby providing the end volume with full image resolution. Finally, the registration toolbox offers a great deal of flexibility to incorporate the user's preference into the registration with the use of arrow toggles to nudge the image as well as toggles to rotate the image. These components of the registration toolbox make it a powerful tool for registration of serial WFM images."

10.4 Segmentation Method Evaluation

Quantitative Analysis. Evaluation Metrics. We evaluated the trained model using several metrics. We define TP as pixels correctly segmented as neurite, TN as pixels correctly segmented as background, FP as background pixel segmented as neurite, and FN as neurite pixels segmented as background. We then compute *precision*: $\frac{TP}{TP+FP}$, *recall*: $\frac{TP}{TP+FN}$, *f1-score*: $2 \times \frac{\text{recall} \times \text{precision}}{\text{recall} + \text{precision}}$, *accuracy*: $\frac{TP+TN}{(TP+TN+FP+FN)}$, *IOU*: $\frac{(TP)}{(TP+FP+FN)}$, and *Dice*: $\frac{(2TP)}{(2TP+FP+FN)}$.

Testing Set. Our testing set contains 50 images of size 512×512, cropped from the medial septum, and manually segmented using the pipeline in Sec. 4. The manual segmentations are binary masks, where 0 represents background and 1 represents neurite. The network output is a probability map within the [0,1] range, where neurite pixels have a higher probability than background pixels.

Speed. The trained model segments a 512×512 image in 0.28 seconds. Running time scales up linearly by the volume depth and every 512 increment in width and length, as each input 3D image stack is broken into 2D image stacks of size 512×512 with step size of 1, 512, and 512 over *z*-, *y*-, and *x*-axis, respectively.

TABLE 1
A Comparative Analysis of our Model Against
State-of-the-Art Models

Method	Dice	IOU	Accuracy	Precision	Recall	F1-score
U ² -Net	86.48 ± 2.21	76.24 ± 3.32	98.92 ± 0.57	82.02 ± 3.05	91.54 ± 2.65	86.48 ± 2.21
U-Net3+	86.36 ± 2.61	76.08 ± 3.91	98.93 ± 0.56	82.68 ± 2.80	90.43 ± 3.13	86.36 ± 2.61
U-Net3+	86.81 ± 2.29	76.76 ± 3.46	98.94 ± 0.56	81.55 ± 2.92	92.83 ± 1.85	86.81 ± 2.29
Ours	87.26 ± 2.09	77.46 ± 3.19	98.97 ± 0.55	82.03 ± 2.71	93.25 ± 1.62	87.26 ± 2.09

Comparison. We applied several automatic neuron tracing techniques on our testing set, including App2 [15], Smart Tracing [16], Rivulet [54], TRemap [55], and NueroGPS-Tree [10], but they failed at reconstructing neuronal structures in the absence of a soma in densely-structured low-resolution images. We compared the performance of our model against three state-of-the-art models including U²-Net [43], U-Net+ [42], U-Net3+ [56], summarized in Table 1. You can find precision-recall curve for these models in Fig. S2, available online in Supplementary Material. As seen, our model achieves the best overall performance in segmenting neurites. As shown in Fig. 9, our model detects strong and weak neurites and only misses extremely weak ones.

We also evaluated our model against these three models on various tasks, including nuclei segmentation and liver-tumor segmentation with larger datasets (consisting of thousands of images). According to all metrics, our network had better performance and more robustness on various segmentation tasks, compared to other networks.

Expert Analysis. Our model reconstructs the complete neuronal structure including weak neurites blended into the background due to the image low contrast in densely structured image stacks. Our experts were provided with crops from different brain regions to evaluate how our segmentation improved neuron visualization as compared to raw data. Despite the brain region, our experts reported that there was significant recovery of fibers in the images as compared to raw data. Their typical workflow would involve creating MIP of images and visualizing them in ImageJ [53]. Using ImageJ, despite increasing gamma value in the raw data, they were unable to visualize some of the fibers that were appropriately segmented by our model. However, for evaluating the segmentation results generated by our model, they created MIPs of both the raw and segmented data and overlaid the images in ImageJ. Our experts were excited to see the dramatic improvement in recovery between raw and segmented images that could now be visualized using their typical workflow (ImageJ). See Fig. S3, available online in Supplementary Material for 2 examples.

10.5 Registration Method Evaluation

NeuroConstruct was used to register six individual serial sections from the mouse brain specimen (see Sec. 4). Each section was 16bit, 21937×35616 pixels ×47 TIFF images, totaling approximately 55GBs per stack. The coarse registration took 25 minutes. We cropped a 5000×5000 ROI of size 6GB. The automatic fine alignment process for six ROI stacks took 33 minutes. In total, coarse-to-fine registration of a 300 μm sample took 68 minutes. We also tested a use-case for re-registration after correcting a misaligned stack, taking an additional 25 minutes. In Fig. S4, available online in Supplementary Material, we show two focused neurons, studied by our experts for evaluation. Following are their discussions:

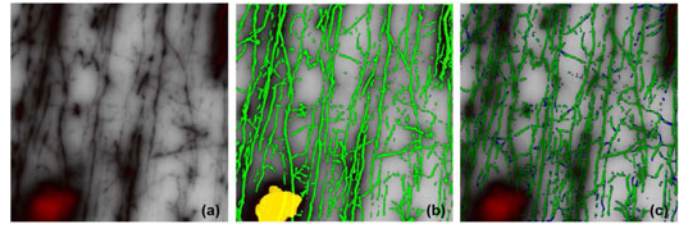


Fig. 9. The neuron segmentation of a densely structured WFM image of a brain medial septum section. Green, yellow, and blue colors represent reconstructed neurites, reconstructed somas, and ground-truth segmentation, respectively. (a) Raw image. (b) Reconstructed neurons overlaid on the raw image. (c) Reconstructed neurites and the ground-truth segmentation overlaid on the raw image.

Utility of NeuroConstruct registration in domain science. Advances in genetic labeling have allowed us to selectively label specific neuronal subsets. The next frontier is to follow these neurons through 3D to allow complete reconstruction of single neurons and their entire arbors (dendritic branches (shown in Fig. 4)). Without a proper reconstruction method, quantitative analysis of WFM data typically does not make claims about individual neurons and their trajectories, but instead about neuron populations. Normal workflow includes subsampling stacks across a region to get a density of labeled neurons and processes from anterior to posterior. These processes would not come from a single soma, but instead gives a population estimate of labeled fibers. Using NeuroConstruct, we register serial slices, reconstruct and isolate single neurons and their labeled arbors. Our registration now allows for a number of applications and analyses that were previously impossible with WFM data.

Quantitative analysis of arbor length. Prior to registration, we were only able to visualize somas and extensions from single stacks and unable to determine the arbors morphology through the anterior to posterior axis. This allows us to visualize the single neuron arbors complexity through serial sections, and compute extensions length. Registration helped to recover 5255 μm of the processes length from the selected sub-ROI in Fig. S4, available online in Supplementary Material.

Quantitative analysis of arbor branch points. Using the information gathered from the registered dataset, another analysis of interest is the number and location of branch points from the soma. The morphological diversity of these branches could tune different aspects of cell communication. Prior to registration, analysis on a single stack would have shown one or two branches. Following NeuroConstruct reconstruction process, it is evident that there are 24 branch nodes visible. The analysis determines the distance between branch points and the soma, and arbor complexity.

Analysis of fiber integrity. A key feature in neuroanatomy study is the health and dramatic deterioration and fragmentation of cholinergic fibers in specific brain regions during brain diseases, such as Alzheimer's. The fibers integrity is critical to maintain neural connectivity and function. Using the registered dataset and the ability to visualize weak fibers from the segmentation, experts now visualize alterations to fiber morphology including swelling and fragmentation. For example, we determine how far from the soma the fragmentation began in affected neuron processes, and how much of the process remains intact and healthy.

11 CONCLUSION AND FUTURE WORK

We presented NeuroConstruct, a novel end-to-end application for the segmentation, registration, and visualization of brain volumes, imaged using WFM. NeuroConstruct combines deep learning 3D segmentation methods, a novel algorithm for coarse-to-fine registration of brain sections, and a hybrid approach to visualize recovered neurites. To generate a ground-truth set to train and test the model, we developed the Segmentation Toolbox, facilitating annotation of WFM image stacks. We evaluated NeuroConstruct quantitatively and qualitatively, along with experts' analysis. Our results show that NeuroConstruct outperforms the state-of-the-art in all aspects of the design, including segmentation, registration, and visualization of neurons in WFM images.

Our application is designed with the goal of helping neuroscientists to study cholinergic neurons in WFM images and perform well in reconstruction and visualization of WFM neurons. We tested our segmentation model on confocal and two-photon microscopic images. Although our network was trained using WFM image stacks, it could recover most of the neurites in these stacks perfectly well. In the future, we will revise the NeuroConstruct to create a universal application that reconstructs and visualizes images acquired by other imaging modalities. We will incorporate more features into our Segmentation Toolbox to increase efficiency and simplicity of annotating WFM neurons. Note that the reconstruction quality can be further improved by training the model with a larger variety of neurites.

ACKNOWLEDGMENTS

This work was supported in part by the NSF under Grants CNS1650499, OAC1919752, ICER1940302, and IIS2107224 and in part by the Intramural Research Program of NIH, NINDS, and NIMH.

REFERENCES

- [1] T. H. Ferreira-Vieira, I. M. Guimaraes, F. R. Silva, and F. M. Ribeiro, "Alzheimer's disease: Targeting the cholinergic system," *Curr. Neuropharmacol.*, vol. 14, pp. 101–115, 2016.
- [2] S. Boorboor, S. Jadhav, M. Ananth, D. Talmage, L. Role, and A. Kaufman, "Visualization of neuronal structures in wide-field microscopy brain images," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 1018–1028, Aug. 2018.
- [3] E. C. Ballinger, M. Ananth, D. A. Talmage, and L. W. Role, "Basal forebrain cholinergic circuits and signaling in cognition and cognitive decline," *Neuron*, vol. 91, pp. 1199–1218, 2016.
- [4] J. R. Glaser and E. M. Glaser, "Neuron imaging with neuroLuCida—PC-based system for image combining microscopy," *Comput. Med. Imag. Graph.*, vol. 14, pp. 307–317, 1990.
- [5] H. Peng, Z. Ruan, F. Long, J. H. Simpson, and E. W. Myers, "V3d enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets," *Nat. Biotechnol.*, vol. 28, no. 4, pp. 348–353, 2010.
- [6] C. Magliaro, A. L. Callara, N. Vanello, and A. Ahluwalia, "A manual segmentation tool for three-dimensional neuron datasets," *Front. Neuroinform.*, vol. 11, 2017, Art. no. 36.
- [7] H. Ikeno, A. Kumaraswamy, K. Kai, T. Wachtler, and H. Ai, "A segmentation scheme for complex neuronal arbors and application to vibration sensitive neurons in the honeybee brain," *Front. Neuroinform.*, vol. 12, 2018, Art. no. 61.
- [8] H. Peng, F. Long, and G. Myers, "Automatic 3D neuron tracing using all-path pruning," *Bioinformatics*, vol. 27, pp. 239–247, 2011.
- [9] M. H. Longair, D. A. Baker, and J. D. Armstrong, "Simple neurite tracer: Open source software for reconstruction, visualization and analysis of neuronal processes," *Bioinformatics*, vol. 27, pp. 2453–2454, 2011.
- [10] T. Quan *et al.*, "NeuroGPS-tree: Automatic reconstruction of large-scale neuronal populations with dense neurites," *Nat. Methods*, vol. 13, pp. 51–54, 2016.
- [11] Y. Wang, A. Narayanaswamy, C.-L. Tsai, and B. Roysam, "A broadly applicable 3D neuron tracing method based on open-curve snake," *Neuroinformatics*, vol. 9, pp. 193–217, 2011.
- [12] S. Li *et al.*, "Optimization of traced neuron skeleton using lasso-based model," *Front. Neuroanatomy*, vol. 13, 2019, Art. no. 18.
- [13] S. Basu and D. Racoceanu, "Reconstructing neuronal morphology from microscopy stacks using fast marching," in *Proc. IEEE Int. Conf. Image Process.*, 2014, pp. 3597–3601.
- [14] J. Yang, M. Hao, X. Liu, Z. Wan, N. Zhong, and H. Peng, "FMST: An automatic neuron tracing method based on fast marching and minimum spanning tree," *Neuroinformatics*, vol. 17, pp. 185–196, 2019.
- [15] H. Xiao and H. Peng, "APP2: Automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree," *Bioinformatics*, vol. 29, pp. 1448–1454, 2013.
- [16] H. Chen, H. Xiao, T. Liu, and H. Peng, "SmartTracing: Self-learning-based neuron reconstruction," *Brain Inform.*, vol. 2, pp. 135–144, 2015.
- [17] R. Li, T. Zeng, H. Peng, and S. Ji, "Deep learning segmentation of optical microscopy images improves 3D neuron reconstruction," *IEEE Trans. Med. Imag.*, vol. 36, no. 7, pp. 1533–1541, Jul. 2017.
- [18] Z. Zhou, H.-C. Kuo, H. Peng, and F. Long, "DeepNeuron: An open deep learning toolbox for neuron tracing," *Brain Inform.*, vol. 5, pp. 1–9, 2018.
- [19] A. Fakhry, T. Zeng, and S. Ji, "Residual deconvolutional networks for brain electron microscopy image segmentation," *IEEE Trans. Med. Imag.*, vol. 36, no. 2, pp. 447–456, Feb. 2016.
- [20] M. Januszewski *et al.*, "High-precision automated reconstruction of neurons with flood-filling networks," *Nat. Methods*, vol. 15, pp. 605–610, 2018.
- [21] K. Luther and H. S. Seung, "Learning metric graphs for neuron segmentation in electron microscopy images," in *Proc IEEE Int. Symp. Biomed. Imag.*, 2019, pp. 244–248.
- [22] D. Haehn *et al.*, "Design and evaluation of interactive proofreading tools for connectomics," *IEEE Trans. Vis. Comput. Graph.*, vol. 20, no. 12, pp. 2466–2475, Dec. 2014.
- [23] P. Bajcsy, S.-C. Lee, A. Lin, and R. Folberg, "Three-dimensional volume reconstruction of extracellular matrix proteins in Uveal Melanoma from fluorescent confocal laser scanning microscope images," *J. Microsc.*, vol. 221, pp. 30–45, 2006.
- [24] T. Ju *et al.*, "3D volume reconstruction of a mouse brain from histological sections using warp filtering," *J. Neurosci. Methods*, vol. 156, pp. 84–100, 2006.
- [25] H. Liang, N. Dabrowska, J. Kapur, and D. S. Weller, "Whole brain reconstruction from multilayered sections of a mouse model of status epilepticus," in *Proc. Asilomar Conf. Signals, Syst., Comput.*, 2017, pp. 1260–1263.
- [26] K. S. Arun, T. S. Huang, and S. D. Blostein, "Least-squares fitting of two 3D point sets," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PAMI-9, no. 5, pp. 698–700, Sep. 1987.
- [27] J. Luck, C. Little, and W. Hoff, "Registration of range data using a hybrid simulated annealing and iterative closest point algorithm," in *Proc. IEEE Int. Conf. Robot. Automat.*, 2000, pp. 3739–3744.
- [28] C.-L. Tsai *et al.*, "Robust, globally consistent and fully automatic multi-image registration and montage synthesis for 3D multi-channel images," *J. Microsc.*, vol. 243, pp. 154–171, 2011.
- [29] M. Yigitsoy and N. Navab, "Structure propagation for image registration," *IEEE Trans. Med. Imag.*, vol. 32, no. 9, pp. 1657–1670, Sep. 2013.
- [30] S.-C. Lee and P. Bajcsy, "Trajectory fusion for 3D volume reconstruction," *Comput. Vis. Image Understanding*, vol. 110, pp. 19–31, 2008.
- [31] V. J. Dercksen, H.-C. Hege, and M. Oberlaender, "The filament editor: An interactive software environment for visualization, proof-editing and analysis of 3D neuron morphology," *Neuroinformatics*, vol. 12, pp. 325–339, 2014.
- [32] H. Chen, D. M. Iacone, N. M. da Costa, E. S. Lein, T. Liu, and H. Peng, "Fast assembling of neuron fragments in serial 3D sections," *Brain Inform.*, vol. 4, 2017, Art. no. 183.
- [33] H. Pfister *et al.*, "Visualization in connectomics," in *Scientific Visualization*, Berlin, Germany: Springer, 2014, pp. 221–245.
- [34] K. Mosaliganti *et al.*, "Reconstruction of cellular biological structures from optical microscopy data," *IEEE Trans. Vis. Comput. Graph.*, vol. 14, no. 4, pp. 863–876, Jul./Aug. 2008.

- [35] M. Nakao *et al.*, "Visualizing in vivo brain neural structures using volume rendered feature spaces," *Comput. Biol. Med.*, vol. 53, pp. 85–93, 2014.
- [36] Y. Wan, H. Otsuna, C.-B. Chien, and C. Hansen, "FluoRender: An application of 2D image space methods for 3D and 4D confocal microscopy data visualization in neurobiology research," in *Proc. IEEE Pacific Vis. Symp.*, 2012, pp. 201–208.
- [37] J. Beyer, A. Al-Awami, N. Kasthuri, J. W. Lichtman, H. Pfister, and M. Hadwiger, "ConnectomeExplorer: Query-guided visual analysis of large volumetric neuroscience data," *IEEE Trans. Vis. Comput. Graph.*, vol. 19, no. 12, pp. 2868–2877, Dec. 2013.
- [38] M. Hadwiger, J. Beyer, W. Jeong, and H. Pfister, "Interactive volume exploration of petascale microscopy data streams using a visualization-driven virtual memory approach," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 12, pp. 2285–2294, Dec. 2012.
- [39] D. Haehn *et al.*, "Scalable interactive visualization for connectomics," *Informatics*, vol. 4, 2017, Art. no. 29.
- [40] WebGL-based viewer for volumetric data. Accessed: Oct. 12, 2020. [Online]. Available: <https://github.com/google/neuroglancer>
- [41] O. Ronneberger, P. Fischer, and T. Brox, "U-Net: Convolutional networks for biomedical image segmentation," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2015, pp. 234–241.
- [42] Z. Zhou, M. M. R. Siddiquee, N. Tajbakhsh, and J. Liang, "Unet++: A nested u-net architecture for medical image segmentation," in *Proc. Deep Learn. Med. Image Anal. Multimodal Learn. Clin. Decis. Support*, 2018, pp. 3–11.
- [43] X. Qin, Z. Zhang, C. Huang, M. Dehghan, O. R. Zaiane, and M. Jagersand, "U2-Net: Going deeper with nested U-structure for salient object detection," *Pattern Recognit.*, vol. 106, 2020, Art. no. 107404.
- [44] P.-S. Liao *et al.*, "A fast algorithm for multilevel thresholding," *J. Inf. Sci. Eng.*, vol. 17, no. 5, pp. 713–727, 2001.
- [45] B. D. Lucas and T. Kanade, "An iterative image registration technique with an application to stereo vision," in *Proc. 7th Int. Joint Conf. Artif. Intell.*, vol. 2, pp. 674–679, 1981.
- [46] M. Roberts *et al.*, "Neural process reconstruction from sparse user scribbles," in *Proc. Int. Conf. Med. Image Comput. Comput.-Assist. Interv.*, 2011, pp. 621–628.
- [47] T.-C. Lee, R. L. Kashyap, and C.-N. Chu, "Building skeleton models via 3D medial surface axis thinning algorithms," *Graphical Models Image Process.*, vol. 56, pp. 462–478, 1994.
- [48] U. Bagci and L. Bai, "Automatic best reference slice selection for smooth volume reconstruction of a mouse brain from histological images," *IEEE Trans. Med. Imag.*, vol. 29, no. 9, pp. 1688–1696, Sep. 2010.
- [49] D. Mattes, D. R. Haynor, H. Vesselle, T. K. Lewellen, and W. Eubank, "PET-CT image registration in the chest using free-form deformations," *IEEE Trans. Med. Imag.*, vol. 22, no. 1, pp. 120–128, Jan. 2003.
- [50] H. W. Kuhn, "The Hungarian method for the assignment problem," *Nav. Res. Logist. Quart.*, vol. 2, no. 1/2, pp. 83–97, 1955.
- [51] W. J. Schroeder, B. Lorensen, and K. Martin, *The Visualization Toolkit: An Object-oriented Approach to 3D Graphics*. New York, NY, USA: Kitware, 2004.
- [52] L. Ibanez, W. Schroeder, L. Ng, and J. Cates, *The ITK Software Guide*, 1st ed. New York, NY, USA: Kitware, 2003.
- [53] W. S. Rasband, "ImageJ: Image processing and analysis in java," *Astrophysics Source Code Library*, vol. 1, pp. 06013, Jun. 2012.
- [54] S. Liu, D. Zhang, S. Liu, D. Feng, H. Peng, and W. Cai, "Rivulet: 3D neuron morphology tracing with iterative back-tracking," *Neuroinformatics*, vol. 14, pp. 387–401, 2016.
- [55] Z. Zhou, X. Liu, B. Long, and H. Peng, "TRemap: Automatic 3D neuron reconstruction based on tracing, reverse mapping and assembling of 2D projections," *Neuroinformatics*, vol. 14, pp. 41–50, 2016.
- [56] H. Huang *et al.*, "UNET 3+: A full-scale connected UNet for medical image segmentation," in *Proc. IEEE Int. Conf. Acoust., Speech Signal Process.*, 2020, pp. 1055–1059.



Parmida Ghahremani received the BS degree in computer engineering from the Sharif University of Technology, Iran. She is currently working toward the PhD degree in computer science from Stony Brook University. Her research interests include computer vision, biomedical imaging, deep learning, and computer graphics.



Saeed Boorboor received the BSc (Hons.) degree in computer science from the School of Science and Engineering, Lahore University of Management Sciences, Pakistan. He is currently working toward the PhD degree in computer science from Stony Brook University. His research interests include scientific visualization, biomedical imaging, and computer graphics.



Pooya Mirhosseini received the BSc degree in software engineering from the Sharif University of Technology and the MSc degree in computer science from Stony Brook University. He was a research assistant with Visualization Lab. He is currently a software engineer with Apple in the Bay Area. His research interests include embrace low-level GPU optimization, GPGPU, visualization, and virtual/augmented reality.



Chetan Gudisagar received the BTech degree from the National Institute of Technology Karnataka, Surathkal, India, and the MSc degree in computer science from Stony Brook University, New York. He is currently working on an open source project on a consistency platform called CorfuDB with VMware. His research interests include distributed systems, backend development, and data science.



Mala Ananth received the BS degree in biology and the PhD degree in neuroscience from Stony Brook University in 2011 and 2019, respectively. From 2013, she was a research assistant with Brookhaven National Laboratory. She is currently a postdoctoral research fellow with the National Institute of Neurological Disorders and Stroke. Her research interests include the heterogeneity of cell types in age-related cognitive decline.



David Talmage received the BA degree in biology from the University of Virginia and the PhD degree in genetics from the University of Minnesota. He was a postdoctoral with the Rockefeller University and Harvard Medical School. He is currently a senior scientist with the National Institute of Mental Health, NIH. He has authored or co-authored more than 70 peer-reviewed articles that have been cited nearly 4000 times.



Lorna W. Role received the AB degree in applied mathematics and the PhD degree in physiology from Harvard University. He was a postdoctoral in pharmacology with Harvard Medical School and the Washington University School of Medicine. She was an assistant professor with Columbia University in 1985 and became a professor. In 2008, he was a SUNY distinguished professor and chair with the Department of Neurobiology and Behavior, and co-director of Neurosciences Institute, Stony Brook University. She is currently the scientific director

and senior investigator with NINDS, NIH. She was the recipient of many awards and honors, including fellow of the American Association for the Advancement of Science in 2011 and fellow of the American College of Neuropsychopharmacology in 2009.



Arie E. Kaufman (Fellow, IEEE) received the PhD degree in computer science from Ben-Gurion University, Israel, in 1977. From 1999 to 2017, he was a chair with the Computer Science Department. He is currently a distinguished professor of computer science, the director of the Center of Visual Computing, and the chief scientist of the Center of Excellence with Wireless and Information Technology, Stony Brook University. He has conducted research for more than 40 years in visualization, graphics, and imaging and has authored or co-

authored more than 350 refereed papers. He was the founding editor-in-chief of the *IEEE Transactions on Visualization and Computer Graphics*, 1995–1998. He is an ACM fellow, National Academy of Inventors fellow, and the recipient of the IEEE Visualization Career Award in 2005, and inducted into the IEEE Visualization Academy in 2019, and the Long Island Technology Hall of Fame in 2013.

▷ **For more information on this or any other computing topic, please visit our Digital Library at www.computer.org/csdl.**