PointTree: Automatic and accurate reconstruction of long-range axonal projections of single-neuron

Lin Cai[1,2], Taiyu Fan[1,2], Xuzhong Qu[1,2], Ying Zhang[1,2], Xianyu Gou[1,2], Quanwei Ding[1,2,3], Weihua Feng[1,2], Tingting Cao[1,2], Xiaohua Lv[1,2], Xiuli Liu[1,2], Qing Huang[1,2,3], Tingwei Quan[1,2,*], Shaoqun Zeng[1,2].

[1]Britton Chance Center for Biomedical Photonics, Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China.

[2]MOE Key Laboratory for Biomedical Photonics, Wuhan National Laboratory for Optoelectronics, Huazhong University of Science and Technology, Wuhan, Hubei, 430074, China.

[3]School of Computer Science & Engineering, Hubei Key Laboratory of Intelligent Robot, Wuhan Institute of Technology, Wuhan, Hubei, 430205, China.

*Corresponding author. Email: quantingwei@hust.edu.cn

Abstract

Single-neuron axonal projections reveal the route map of neuron output and provide a key cue for understanding how information flows across the brain. Reconstruction of single-neuron axonal projections requires intensive manual operations in tens of terabytes of brain imaging data, and is highly time-consuming and labor-intensive. The main issue lies in the need for precise reconstruction algorithms to avoid reconstruction errors, yet current methods struggle with densely distributed axons, focusing mainly on skeleton extraction. To overcome this, we introduce a point assignment-based method that uses cylindrical point sets to accurately represent axons and a minimal information flow tree model to suppress the snowball effect of reconstruction errors. Our method successfully reconstructs single-neuron axonal projections across hundreds of GBs (Gigabytes) images with an average of 80% F1-score, while current methods only provide less than 40% F1-score reconstructions from a few hundred MBs (Megabytes) images. This huge improvement is helpful for high-throughput mapping of neuron projections.

Introduction

Neuronal axons in general project to different brain regions, and their projection distribution is an essential cue for neuron type identification, neuronal circuit construction, and deeper insight into how information flows in the brain[1-4]. Advances in optical imaging and molecular labeling techniques[5-10] have allowed us to observe the entire mouse brain at single-axon resolution, and provided the database for the study of neuronal projection patterns[11-18]. However, the reconstruction of these long-range projected axons still requires extensive manual annotation in tens of TBs volumetric images[7,19-22], this labor-intensive process creates a major bottleneck for high-throughput mapping of neuronal projections[23].

The difficulties in reconstructing the long-range projections of neurons are as follows. On the one hand, while molecular labeling techniques can shed light on a very small fraction of neurons, a significant fraction of neuronal axons are still densely distributed due to the morphological complexity of neurons. The identification of densely distributed axons is considered an open problem in the field[23-25], which still has no good solution. On the other hand, during neuron reconstruction, reconstruction errors accumulate and a single reconstruction error can result in an entire branch being connected erroneously to other neurons or missing[26]. Therefore, effective large-scale reconstruction of neurons requires extremely high identification accuracy of dense axons. The contradictions between these two aspects seem hard to reconcile.

The current neuron reconstruction frameworks focus on how to accurately extract skeletons of neurites and establish the connections between skeletons[2,27]. The BigNeuron project[28] conducts a

systematic evaluation of 35 automatic neuron reconstruction algorithms, all of which are based on tracing neurite skeletons and can be divided into two categories: local and global approaches. In the local approach[29-32], the localization of the next skeleton point requires computation of the signal anisotropy of the image region near the current skeleton point. Localization errors typically occur when this image region contains other neurite signals. The global approach[24,33,34] first generates multiple seed points that are commonly located at the neurite centerline, and then establishes connections between these seed points for generating the neurite skeleton. This connection relies mainly on spatial location information, resulting in densely distributed neurites being connected to each other erroneously. While deep learning is widely used in neuron reconstruction[35-38],-mainly for neuronal image segmentation and signal intensity enhancement to reduce reconstruction errors-even ideal segmentation with all neurite centers identified and their signal enhanced still exhibit significant reconstruction errors with skeleton-based methods (**Fig. S1**).

To address the problem of error accumulation during neuron reconstruction, it is common practice to utilize statistical information of neuron morphology, such as the angle between two neurites, to identify and remove spurious connections between the reconstructed neurites. This strategy[24,39] achieves 80% reconstruction accuracy from GB-scale images under two critical constrains: (1) precise identification of neurite terminals and branch points is required for accurate angle computation and morphological analysis, and (2) somatic locations are required as critical information to remove some links between the reconstructed neurites to ensure that each cell body can be mapped to the root node of a single tree structure. However, for long-range axonal reconstruction across hundreds of GB-scale images, the strategy is not effective to eliminate the accumulation of errors due to factors such as the position of the axon at a distance from the soma, and slight morphological differences between axon junction and termination. Consequently, current long-range projection reconstruction methods are semi-automatic and require substantial human intervention[19,21,22,40].

Here, we propose a new neuron reconstruction method called PointTree, which aims at how to assign foreground points in neuronal images to their own neurons. In the workflow, we design a constrained Gaussian clustering method to partition the foreground region of a neuronal image into a series of columnar regions whose centerline belongs to only a single neurite. This operation essentially eliminates the interference of different neurites in the dense reconstruction. In addition, each columnar region is characterized by a minimal envelope ellipsoid for constructing connections between columnar regions, which forms the neurite shapes. Based on the reconstructed shapes, we design a minimal information flow tree model to suppress the cumulative reconstruction error. Using the proposed method, we successfully achieve accurate reconstruction of long-range projections of neurons across hundreds of gigabytes of volumetric image.

Results

The architecture and principles of PointTree

In the design of PointTree, we have developed a series of optimization problems to assign foreground points in data blocks to their respective neurites. Firstly, the segment network is utilized for each data block to obtain foreground points. Subsequently, we apply a constrained Gaussian clustering method[41] to partition the foreground points into columnar regions and determine their geometrical parameters by solving the minimum-volume covering ellipsoids problem[42]. Using these geometrical parameters, we construct a 0-1 assignment problem[43] to establish links between these columnar regions. Finally, skeletons are extracted from these linked columnar regions to reduce data redundancy by using region growing[44]. The key procedures for neuron reconstruction are presented in **Figure 1A.**

In addition, PointTree employed the statistical prior information to reduce the reconstruction errors. At the branching point (node) of the neurites, it can be divided into three segments of neurite skeletons. The segment entering the node forms two angles with the other two segments exiting the node respectively. The node angle is defined as the smaller angle between the entering segment and

each exiting segment (**Figure 1B**). With node angle, we can identify the single complete neurite and its corresponding node angles. The skeleton of the neurite is generally smooth, with very few sudden directional changes and even fewer at the nodes. So, the node angles should be as small as possible. For neuronal branches, the node angles are uniquely determined when the root node is given, and the sum of the negative cosine of these node angles expressed by information flow value is small when the root node is correctly identified. This rule is defined as minimal information flow tree (MIFT).

In image blocks of densely distributed neurites, we used semi-automatic software[21] extracting 208 neuronal branches and identifying their root nodes. For each branch, we calculated their information flow values as the ground-truth information flow values (**Figure 1C**). To validate MIFT, we looped through all possible structure of these branches by changing the root node in order to compute the maximum and minimum information flow values (**Figure 1C**). It is evident that, for most neuronal branches (195/208), the ground-truth values of the information flow achieve the minimum value, suggesting that MIFT rule is reasonable. We utilized MIFT to modify skeleton structure and remove spurious connections between reconstructed neurites (**Figure 1D and Fig.S2**), both for reconstructions within individual blocks and for the fused reconstruction in adjacent blocks.

PointTree has the capability to separate densely distributed neurites. When dealing with two parallel neurites in close proximity to each other, their shapes can be represented by a series of columnar regions (**The left panels of Figure 1E**). We have modified the Gaussian clustering algorithm by constraining the estimated mean and covariance parameters so that the cluster shape approaches a columnar shape. Additionally, foreground points within the same cluster are connected to each other. These two features ensure that the central line in the columnar region belongs to only a single neurite, which is crucial for separating densely packed neurites. Furthermore, we utilize the minimum volume covering ellipsoid to extract shape information of the columnar regions for constructing their connections. These designs enable PointTree to successfully reconstruct packed neurites. In contrast, skeleton-based local methods rely on determining the position of the next skeleton point based on the shape anisotropy of the region. This often leads to localization errors when there are two neurite image signals within a region (**The middle panels of Figure 1E**). When it comes to skeleton-based global methods, although seed points can be located at individual neurite centers, accurately constructing connections between these seed points proves challenging due to the reliance on distance between points and susceptibility to interference from densely distributed neurites. (**The right panels of Figure 1E**).

The merits of PointTree in dense reconstruction

In dense reconstruction, one of the main concerns is how well to separate densely distributed neurites which behaves as crossover and closely paralleled neurites. These neurites can be manually identified by visualization with different view angles (**Fig.S3**). We compared PointTree with several skeleton-based methods such as neuTube[45], PHDF[46], NGPST[39] and MOST[47] in performing this task. We manually labeled the locations where neurites are crossover or closely parallel from five 256×256×256 image blocks. For fair comparison, all methods are performed on segmented images derived from the segmentation network. **Figure 2A** illustrates the process of PointTree's separation of crossover and closely paralleled neurites. PointTree can successfully separate the densely distributed neurites in a range of 71.4 % and 91.7%, while these skeleton-based methods only separate 25.0% densely distributed neurites (**Figure 2B**) at most. We also present the comparison of PointTree and other methods on some reconstruction examples in which multi crossover neurites (**Figure 2C**) and closely paralleled neurites are involved. PointTree provides the perfect reconstruction while other methods fail to reconstruct these neurites.

Furthermore, we present the quantitative results derived from PointTree and five widely used skeleton-based reconstruction methods. including APP2, neuTube, NGPST, PHDF, MOST. Eight 256×256×256 image blocks that include many densely distributed neurites are of the testing dataset. All reconstruction algorithms are performed on the segmentation images of these testing dataset.

We give the intuitive reconstruction comparisons (**Figure 3A**). PointTree provides the reconstruction close to the ground truth. The skeleton-based methods generate lots of reconstruction errors and incorrectly combine multi neurites into a single branch. The quantitative reconstructions suggest that PointTree is far superior to skeleton-based methods (**Figure 3B**). For PointTree, the average precision is above 90%, both recall and f1-score are above 85%. The skeleton-based methods cannot provide the good solution to separate the densely neurites. The f1-score of these reconstructions ranges from 30% to 40%, which indicates the ineffective reconstructions.

Reconstruction of data with different signal-to-noise ratios

In the field of neuronal reconstruction, data acquired by different imaging systems often exhibit varying signal-to-noise ratio (SNR) characteristics. For some low-SNR datasets, severe noise interference makes it difficult even for human observers to accurately identify neurite structures. To systematically evaluate PointTree's reconstruction performance across datasets with different SNRs, we selected and analyzed data from three imaging systems: light sheet microscopy[48] (LSM), fluorescent micro-optical sectioning tomography[49] (fMOST), and high-definition fluorescent micro-optical sectioning tomography[50] (HD-fMOST), with SNR ranges of 2–7, 6–12, and 9–14, respectively (**Figure 4A**).

Experimental results demonstrate that, thanks to the powerful feature extraction capability of the deep learning network, the trained neural network achieves satisfactory segmentation performance (**Third row in Figure 4B**) even on low-SNR data (**First two columns in Figure 4B, top row**), laying a solid foundation for subsequent accurate reconstruction (**Bottom row in Figure 4B**). Quantitative analysis reveals that PointTree delivers stable reconstruction performance across all SNR levels. Specifically: for LSM data (sample size n=25, mean SNR=5.01), average precision=96.0%, recall=88.7%, and f1-score=91.0%; for fMOST data (sample size n=25, mean SNR=8.68), average precision=95.8%, recall=87.3%, and f1-score=90.0%; for HD-fMOST data (sample size n=25, mean SNR=11.4), average precision=98.1%, recall=91.0%, and f1-score=93.3% (**Figure 4A**).

Notably, in low-SNR LSM data, background regions contain more artifactual signals (**First panel in Figure 4C**) due to similar intensity distributions between background and foreground points. In contrast, high-SNR datasets (fMOST and HD-fMOST) exhibit cleaner background features with distinct intensity separation between background noise and neurite signals (**Second and third panel in Figure 4C**). This observation highlights the critical impact of SNR on reconstruction quality while simultaneously validating the robustness of PointTree, which is aided by the segmentation network, across diverse SNR conditions.

Restrain error accumulation in the reconstruction

In order to achieve accurate axon reconstruction, it is essential to effectively suppress the snowballing accumulation of reconstruction errors. The performance of the minimal information flow tree (MIFT) in retraining the reconstruction errors is evaluated in this study. **Figure 5A** presents six 512×512×512 image blocks and their reconstructions using PointTree in the first column. The reconstruction fusing procedure is then performed on these axonal reconstructions (**Figure 5A**). By employing MIFT to revise the reconstructions and remove false connections between axons, reasonable reconstructions are achieved. In contrast, when the same fusion procedure is conducted without MIFT to revise the reconstruction, almost all axons are incorrectly connected together (**Bottom-right panel in Figure 5A**).

We furthermore measure the enhancement in the reconstruction accuracy achieved by MIFT (**Figure 5B**). For the initial reconstructions from six image blocks, the average of f1-score is about 0.86. By using MIFT, the average of f1-score is above 0.8 for the reconstructions from two image blocks which are generated with the first fusion. In the second fusion (**Top-right panel in Figure 5A**), f1-score still keeps 0.79. In contrast, without MIFT, the first fusion leads to a drop of about f1-score of 0.3. After the second fusion, f1-score is less than 0.2. We also present some reconstruction examples after two fusions in **Figure 5C**, which are close to the ground truth. These

results suggest that MIFT model take consideration of the proper structure of axons and thus can restrain the error communications in the reconstruction fusion process.

Long-range axonal projections reconstruction

We applied PointTree for long-range axon reconstruction. The testing image block has the size of 11226×8791×1486 voxels and includes axons from eight neurons (**Figure 6A**). We also used GTree to manually reconstruct these neurons as the ground-truth reconstruction (**Figure 6B**). Except for the labeling of training data for segmentation network and of the axon starting points of a single neuron, the whole reconstruction process is totally automatic. The results show PointTree successfully recovered the axonal morphology of these eight neurons without manual interference (**Figure 6C** and **Movies S1 & S2**), and we compared these reconstructions with ground truth (**Fig.S4**). The average precision is above 85% and the average recall and f1-score are above 80% (**Figure 6E**). In addition, we presented the axons reconstructions from two image blocks (**Figures 6C$_1$&C$_2$**) which include a large number of densely distributed axons. This reconstruction performance suggests that the point assignment and the minimal information flow tree mode, as the two key strategies in PointTree, perform well in long-range axonal reconstruction.

We also applied PointTree to process another 10739×11226×3921 image blocks collected with HD-fMOST system[50]. The high signal-to-noise ratio in this optical system results in a significantly extended dynamic range of the signal. PointTree can effectively deal with this case, and all 14 long-range projections are successfully reconstructed (**Fig.S5**). The quantitative results suggest that the average f1-score is above 90% (**Table S1**).

Despite the need to solve multiple large-scale optimization problems, the reconstruction speed using PointTree is generally faster than the imaging speed. For instance, in a typical scenario involving 254 image blocks with 512×512×512 voxels, the total time required for reconstruction is approximately 44 minutes. Even for a larger dataset comprising 821 image blocks with 512×512×512 voxels and including a significant number of sparsely distributed neurites, the total time cost amounts to about 60 minutes (**Table S2**). It should be noted that the time cost does not increase linearly as data volume increases due to the influence of neurite density on overall reconstruction time. In summary, PointTree demonstrates remarkable speed in reconstructing long-range axons (**Movie S3**).

Discussion

We have presented an automated method for reconstructing the long-range projections of neurons. In this study, we address the problem of mutual interference among densely distributed neurites and the cumulative error during reconstruction by designing reconstruction method based on point set assignment and the minimal information flow tree, respectively. As a result, our approach enables accurate reconstruction of long-range neuron projections from hundreds of gigabytes of data. This advance significantly enhances the efficiency of whole-brain-scale neuron reconstruction, bridging the substantial gap between factory-level generation of whole-brain-scale neuronal imaging data and tens of hours required to reconstruct one neuron.

Our approach is performed on image foregrounds where the segmented neurites have a fixed radius approximately equal to the total size of the three voxels. In this case, we can estimate the total number of foreground points (voxels) and set a suitable number of columnar regions for ensuring the anisotropy of each columnar region, which is based on the fact that the union of columnar regions equals to the foreground region. The anisotropy of the columnar regions will reduce the difficulty in establishing their connection. The requirement that all segmented neurites have a relatively fixed radius can be fulfilled. For all neurites, the value of their voxels decreases as these voxels deviate from the nearest centerline. The deep learning network is able to grasp this feature and segment only the neurite centerline and its neighborhood. Typically, in reconstructions of neurons whose projections are distributed over hundreds to thousands of GBs of data, less than GB-sized images with labels are needed as training data. The labeling process takes a few hours, which is negligible for semi-automatic reconstruction of all neurons in the whole volume images.

We propose a new reconstruction mode centered on point set assignment instead of the current reconstruction mode focused on skeleton extraction. In the current reconstruction paradigm, most deep networks are used to enhance the signal-to-noise ratio of neuronal images and do not address well the issue of signal interference during skeleton extraction. In contrast, our reconstruction approach is based on directly processing the foreground points generated by the deep learning network. With continued advances in deep learning techniques, the generality and accuracy of image segmentation will be continuously enhanced, thereby significantly boosting the application scope of our method in various scenarios. Essentially, our method can be applied to any skeleton tracking-based application scenario and effectively eliminate dense signal interference.

Our method still generates a few reconstruction errors. This is due to the following three aspects. First, our method directly handles image foregrounds, which leads to reconstruction errors when some neurites with weak image intensities are not identified. Second, relying solely on foreground point information and rule-based judgment methods may generate some connection errors when establishing connections between neurites. Finally, the minimal information flow tree's fundamental assumption, that axons should be as smooth as possible does not always hold true. In fact, real axons can take quite sharp turns leading the algorithm to erroneously separate a single continuous axon into disjoint fibers (**Fig. S7**). Therefore, for the automatic reconstruction of neurons on a brain-wide scale, further work is needed to enhance the imaging intensity and incorporate soma shapes and raw image signals for neurites connection recognition.

Materials and Methods

Data collections

The test datasets are collected through the preparation of two kinds of samples. For one C57BL/6 male mouse, 100 nl AAV-Cre virus and 100 nl of AAV-EF1α-DIO-EYFP virus were injected into the VPM nucleus at the same time. 21 days later, the chemical sectioning fluorescence tomography(CSFT) system[49] was used to acquire imaging data (**Figures 1-6**), more details can be seen in the reference[51]. For one C57BL/6J male mouse, 100 nl of AAV-YFP was injected into the motor area. 21 days later, high-definition fluorescent micro-optical sectioning tomography (HD-fMOST) was used to acquire imaging data[50] (**Fig.S5**).

Generation of foreground points

Our reconstruction method performs on the image foregrounds. Here, we used UNet3D[52] for image stacks segmentation without network structure modification. The detailed information about UNet3D can be found in the reference[52]. Considering the requirement that the network output, the segmented neurites, have the relatively fixed radius, we calculate the distance field of the neurite's skeleton as the ground-truth for supervise the network. Initially, the semi-automatic software GTree was utilized to extract the neurites skeleton and subsequently interpolate the skeleton points. The interpolation operation ensured that the distance between any skeleton point and its nearest point was less than 1 μm. Subsequently, the interpolated skeleton points were used as centers to mark spherical regions with a radius of 5 voxels. These spherical regions served as candidate areas for foreground. Within these candidate areas, the distance from each point to its nearest interpolated skeleton point was calculated. Finally, the distances are mapped into Gaussian kernel distances which forms the Gaussian density map. This map normalized by maximum value leads to the distance field map to supervise UNet3D output.

In the training stage, Adam optimizer is used with an initial learning rate at 3e-4. The input image size is 128×128×128. Batch size is set to 1, the L1-norm is used as loss function to train the network. We presented the reconstructions from two kinds of fMOST datasets. One is from the reference[51] and the other is from the reference[50]. Therefore, we created two sets of training data, each consisting of 20 512×512×512 image blocks (each divided into 64 image blocks of size 128×128×128). In each set, 10 image blocks contain densely distributed neurites, while the other 10 blocks contain sparsely distributed neurites. In the predicting stage, we applied the threshold operation to the

distance field image. The voxels whose values are more than 0.5 are regarded as the foreground points.

Neuron Reconstruction based on Points assignment

For the image stack, we allocated the foreground points to their respective neurites and established connections between neurites by constructing three optimization models: (1) the constrained Gaussian mixture model divides the foreground points into a set of points, each of which has a column shape; (2) the minimum-volume covering ellipsoids model extracts the features of the column-shaped point set; (3) the 0-1 assignment optimization model establishes connections between the column-shaped point sets, resulting in the shapes of individual neurites, and then builds connections between the reconstructed neurites.

Constrained Gaussian mixture model

The three-dimensional Gaussian function exhibits an ellipsoidal shape in space, which we have utilized to approximate the columnar shape of local neurites. In this study, Gaussian distribution mixture functions with K components are employed to approximate the shape of all neurites in an image block. The component number K is obtained by point density and will be discussed later. Given the foreground points $x_1, x_2, \cdots, x_n$, for each foreground points x_i, the probability density function $P(x_i)$ is calculated as follow:

$$P(x_i) = \sum_{j=1}^{K} \pi_j N(x_i \mid \mu_j, \Sigma_j) \tag{1}$$

Here $N(x_i|\mu_j, \Sigma_j)$ is the Gaussian density function with mean value μ_j and covariance matrix Σ_j. Weight π_j is the regularization parameter. $N(x_i|\mu_j, \Sigma_j)$ is given by the formula:

$$N(x_i|\mu_j, \Sigma_j) = \frac{1}{2\pi^{3/2} |\Sigma_j|^{1/2}} e^{-\frac{1}{2}(x_i-\mu_j)^T \Sigma_j^{-1} (x_i-\mu_j)^T} \tag{2}$$

Based on probability density function, the conditional probability can be computed as:

$$p_{i,j} = P(x_i \mid cluster_j) = \frac{\pi_j N(x_i \mid \mu_j, \Sigma_j)}{\sum_{j=1}^{K} \pi_j N(x_i \mid \mu_j, \Sigma_j)} \qquad j = (1, 2, ..., K) \tag{3}$$

Here $p_{i,j}$ is the conditional probability for x_i to assign to the *j*-th cluster. If $p_{i,k}$ is the maximum value among $\{p_{i,1}, ... p_{i,K}\}$, the foreground point x_i will be assigned to the *k*-th cluster. All the points assigned to the *k*-th cluster form a columnar region. Considering that both the number of foreground points and component number are large, we have added some constrained conditions for Gaussian mixture model as follows:

$$\sum_{j=1}^{K} \pi_j = 1 \tag{4}$$

$$I(\mu_j) \geq \varepsilon_0, \ \ |\Sigma_j| \leq \varepsilon_1 \tag{5}$$

$\sum_{j=1}^{K} \pi_j = 1$ refers to the fact that the total probability distribution normalizes to 1. $I(\cdot)$ represents the signal intensity from segment image, ε_0 is the minimum signal intensity of foreground points and is set to 128 in the algorithm. $I(\mu_i) \geq \varepsilon_0$ restrain the center of the gaussian distribution to be a foreground point. $|\Sigma_j| \leq \varepsilon_1$ restrain the determinant of the covariance matrix which control the suitable number of foreground points for each columnar region. ε_1 is set to cube of three times the average diameter of neurite.

Maximum likelihood is employed to estimate the parameters of Gaussian mixture model and the final optimization problem is formed as follow:

$$(\pi_j^*, \mu_j^*, \Sigma_j^*)_{j=1,2,\cdots,K} = \arg\max \prod_{i=1}^{n} P(x_i) = \arg\max \prod_{i=1}^{n} \left(\sum_{j=1}^{K} \pi_j N(x_i|\mu_j, \Sigma_j) \right) \tag{6}$$

$$s.t. \sum_{j=1}^{K} \pi_j = 1, \ I(\mu_j) \geq \varepsilon_0, \ \ |\Sigma_j| \leq \varepsilon_1 \tag{7}$$

In solving this optimization problem, we employ peak density algorithm[53] to compute density for each foreground points and sort them in descending order. We first select a point as seed point, and the foreground points within a radius of 5 centered on it will be excluded. Then we continue selecting seed points until all foreground points are either selected or excluded. The selected K seed points represents the initial K components. We select signal points from the median (based on density) to both sides as seed points which can decrease the situations that seed points lies in the center of a crossover or the edge of neurites, this strategy can make the generated columnar regions be more reasonable. The positions of the K seed points are set to the initial $(\mu_1, \mu_2, \cdots, \mu_K)$. The initial setting of covariance matrix is the identity matrix. The constrained Gaussian mixture model was solved by EM algorithm[54] , the EM algorithm is divided into two steps:

Estep: For each point x_i, compute its probability within each gaussian distribution using the probability density function:

$$p_{i,j} = \frac{\pi_j N(x_i \mid \mu_j, \Sigma_j)}{\sum_{j=1}^{K} \pi_j N(x_i \mid \mu_j, \Sigma_j)} \tag{8}$$

Mstep: Update the mean value, covariance matrices and weight vectors.

$$\pi_j = \frac{\sum_{i=1}^{n} p_{i,j}}{n} \tag{9}$$

$$\mu_j = \frac{\sum_{i=1}^{n} p_{i,j} x_i}{\sum_{i=1}^{n} p_{i,j}} \tag{10}$$

$$\Sigma_j = \frac{\sum_{i=1}^{n} p_{i,j} (x_i - \mu_j)(x_i - \mu_j)^T}{\sum_{i=1}^{n} p_{i,j}} \tag{11}$$

Besides, the constrained gaussian mixture model possesses additional constraints: $I(\mu_j) \geq \varepsilon_0$ and $|\Sigma_j| \leq \varepsilon_1$. After finish M-step, μ_j with $I(\mu_j) < \varepsilon_0$ are selected. Eigenvalue decomposition are applied on Σ_j and obtain eigenvalues $(\gamma_1, \gamma_2, \gamma_3)$ in descending order and eigenvectors (v_1, v_2, v_3). μ_j is updated along v_1 and $-v_1$ to generate two new cluster with mean value and covariance matrices $(u_{j,1}, \Sigma_{j,1})$ and $(u_{j,2}, \Sigma_{j,2})$ as follow:

$$u_{j,1} = u_j + v_1 \cdot \frac{\gamma}{2} \tag{12}$$

$$u_{j,2} = u_j - v_1 \cdot \frac{\gamma}{2} \tag{13}$$

$$\Sigma_{j,1} = \frac{\sum_{i=1}^{n} p_{i,j} (x_i - \mu_{j,1})(x_i - \mu_{j,1})^T}{\sum_{i=1}^{n} p_{i,j}} \tag{14}$$

$$\Sigma_{j,2} = \frac{\sum_{i=1}^{n} p_{i,j} (x_i - \mu_{j,2})(x_i - \mu_{j,2})^T}{\sum_{i=1}^{n} p_{i,j}} \tag{15}$$

For $\Sigma_j > \varepsilon_1$, it will be updated as follow:

$$\Sigma_j^{'} = \frac{\varepsilon_1}{\Sigma_j} \Sigma_j \tag{16}$$

Iteration of Estep and Mstep will continue until the *k*-th result $\{\mu^k, \Sigma^k\}$ and (*k*-1)-th result satisfy the stopping criteria:

$$\left\| \frac{u^k - u^{k-1}}{u^{k-1}} \right\| < \boldsymbol{\varepsilon} \text{ and } \left\| \frac{\Sigma^k - \Sigma^{k-1}}{\Sigma^{k-1}} \right\| < \boldsymbol{\varepsilon} \tag{17}$$

Here the division represents element-wise division and $\|\bullet\|$ denotes L_2^1-norm and $\boldsymbol{\varepsilon}$ is set to 0.01.

Shape characterization of columnar regions

After deriving the columnar regions through solving the constrained Gaussian mixture model, it is imperative to characterize their geometric shape (terminals and centerlines). For this purpose, we calculate the minimum-volume ellipsoids that can fully encompasses each individual columnar region. For $c \in R^3, Q \in S_{++}^3$, a three-dimensional ellipsoid can be defined as follow[42]:

$$E_{c,Q} := \{x \in R^3 \mid (x-c)^T Q(x-c) \le 1\} \tag{18}$$

Here, c is the center of ellipsoid, Q represents the geometric shape, S_{++}^3 denotes the convex cone of 3×3 symmetric positive definite matrices. The volume of $E_{c,Q}$ is given by the formula:

$$Volume(E_{c,Q}) = \frac{\pi^{3/2}}{\Gamma(3/2+1)} \frac{1}{\sqrt{det(Q)}} \tag{19}$$

Here, $\Gamma(\cdot)$ is the standard gamma function of calculus, $det(Q)$ means the determinant of matrix Q. Minimizing the volume of $E_{c,Q}$ is equivalent to minimizing $\det(Q^{-1/2})$. Therefore, for a columnar region with foreground points $P\{x_1, x_2, \ldots x_m\}$, we define the target function as follow:

$$P1: \quad (c^*, Q^*) = \arg\min_{c,Q} \det(Q^{-1/2}) \tag{20}$$

$$s.t. (x_i - c)^T Q(x_i - c) \le 1, \; i=1,2,...,m \tag{21}$$

$$c \in CHull(P), \quad Q \in S_{++}^3 \tag{22}$$

Here $c \in CHull(P_i)$ restrain the solved center of ellipsoid to locate within the smallest convex hull formed by the clustering points. To solve this problem, a variable substitution $A = Q^{1/2}$ and $y = Q^{1/2}c$ were applied to equation (20) and equation (21), the original problem $P1$ can be transformed into a convex optimization problem as follow:

$$P2: \quad (A^*, y^*) = \arg\min_{A,y} \; - ln\, det(A) \tag{23}$$

$$s.t. (Ax_i - y)^T (Ax_i - y) \le 1, \quad i=1,2,...,m \tag{24}$$

$$A \in S_{++}^3 \tag{25}$$

Through adding the logarithmic barrier function, we can obtain the following formula:

$$P3: \quad (A^*, y^*, \theta^*) = \arg\min_{A,y,\theta} \; - ln\, det(A) - \theta \sum_{i=1}^{m} ln(z_i) \tag{26}$$

$$s.t. (Ax_i - y)^T (Ax_i - y) + z_i = 1, \quad i=1,2,...,m \tag{27}$$

$$A \in S_{++}^3, \; z_i > 0 \tag{28}$$

As θ varies in the interval $(0, \infty)$, the solution of $P3$ changes. When θ approaches 0, the optimal solution of $P3$ tends to the optimal solution of $P2$. By adding the dual multipliers d_i which satisfies $d_i \cdot z_i = \theta$, the optimality conditions can be written as:

$$\sum_{i=1}^{m} d_i [(Ax_i - y)^T x_i^T + x_i (Ax_i - y)^T] - A^{-1} = 0 \tag{29}$$

$$\sum_{i=1}^{m} d_i (y - Ax_i) = 0 \tag{30}$$

$$(Ax_i - y)^T (Ax_i - y) + z_i = 1 \; i = 1,2,...,m \tag{31}$$

$$\sum_{i=1}^{m} d_i \cdot z_i = \theta, \quad i = 1, 2, ..., m \tag{32}$$

$$d_i,\ z_i \geq 0 \tag{33}$$

At this point, the error between the solution of the system of equations and the optimal solution of $P3$ is less than $d^T z$. Through equation (30), the explicit expression for solving y can be obtained as follow:

$$y = \frac{AXd}{e^T d} \tag{34}$$

Here, X stands for a $3 \times m$ matrix $[x_1 \mid x_2 \mid ... \mid x_m]$, e stands for vector of ones $(1,1,...,1)_{1\times m}^T$ and d stands for $(d_1, d_2, ..., d_m)_{1\times m}^T$. Substitute equation (34) into equation (29), the equation for matrix A can be obtained by:

$$(XDX^T - \frac{Xdd^T X^T}{e^T d})A + A(XDX^T - \frac{Xdd^T X^T}{e^T d}) = A^{-1} \tag{35}$$

Here D stands for a $m \times m$ diagonal matrix $Diag(d_1, d_2, ..., d_m)$. And the explicit expression for A is formed as

$$A = A(d) = [2(XDX^T - \frac{Xdd^T X^T}{e^T d})]^{-1/2} \tag{36}$$

And explicit expression for y:

$$y = \frac{[2(XDX^T - \frac{Xdd^T X^T}{e^T d})]^{-1/2} Xd}{e^T d} \tag{37}$$

Through substitute the above two equation to the system of equations (29)-(33), variables *A* and y are eliminated. The following system of equations with only variables *d* and *z* can be obtained:

$$f(d) + z - e = 0 \tag{38}$$

$$Dz - \theta e = 0 \tag{39}$$

$$d_i,\ z_i \geq 0 \tag{40}$$

Here $f(d)$ is nonlinear function of variable d:

$$f_i(d) = (x_i - \frac{Xd}{e^T d})[2(XDX^T - \frac{Xdd^T X^T}{e^T d})]^{-1} \cdot (x_i - \frac{Xd}{e^T d})\ i = 1, 2, ..., m \tag{41}$$

For a fixed barrier parameter θ, we employ Newton's method to solve the system of equations. We use $\nabla_d f(d)$ to represent the Jacobian matrix of $f(d)$. Thus, the Jacobian matrix of the system of equations can be computed as follow:

$$\begin{bmatrix} \nabla_d f(d) & I \\ Z & D \end{bmatrix} \tag{42}$$

And the newton's direction is written as:

$$\Delta(d) = (\nabla_d f(d) - D^{-1}Z)^{-1}(h_1 - D^{-1}h_2) \tag{43}$$

$$\Delta(z) = D^{-1}h_2 - D^{-1}Z(\nabla_d f(d) - D^{-1}Z)^{-1}(h_1 - D^{-1}h_2) \tag{44}$$

$$h_1 = e - z - f(d),\ h_2 = \theta e - Dz \tag{45}$$

With initial $(d_0,\ z_0)$, iterate with $(d_n,\ z_n) = (d_{n-1},\ z_{n-1}) + \tilde{\beta}(\Delta(d_{n-1}),\ \Delta(z_{n-1}))$ to obtain the final optimal solution, $\tilde{\beta}$ represents the newton's step. Detailed process can see the pseudo code as follow:

Algorithm 1 Compute Newton's direction.

Input: $(d,\ z,\ \theta)$ satisfying $d,\ z > 0$, $\theta \geq 0$

1. $A^{-2}(d) = [2(XDX^T - \frac{Xdd^T X^T}{e^T d})]$

2. $\Sigma(d) = (X - \frac{Xde^T}{e^T d})A^2(d)(X - \frac{Xde^T}{e^T d})$

3. $\nabla_d f(d) = -2(\frac{\Sigma(d)}{e^T d} + \Sigma(d) \circ \Sigma(d))$

4. $(\Delta(d),\ \Delta(z)) = ((\nabla_d f(d) - D^{-1}Z)^{-1}(h_1 - D^{-1}h_2),\ D^{-1}h_2 - D^{-1}Z\Delta(d))$

Output: $(\Delta(d),\ \Delta(z))$

Algorithm 2 Process of solving P2

Input: $\{x_1,\ x_2, ..., x_m\}$

1. $r = 0.99$, $(d_0, z_0) = (\frac{3}{2m}e, e - f(d_0))$

2. $E = -det(A(d))$

3. while $(|e - f(d) - z| > \varepsilon_1$ or $\frac{d^T z}{E} > \varepsilon_2)$

4. $\theta = \frac{d^T z}{10m}$

5. $(\Delta(d),\ \Delta(z)) = Compute_Newton_direction(d,\ z)$

6. $\bar{\beta} = max\{\beta \mid (d,\ z)\ + \beta(\Delta(d),\ \Delta(z) \geq 0)\}$

7. $\tilde{\beta} = min(r\bar{\beta},\ 1)$

8. $(d,\ z) = (d,\ z) + \tilde{\beta}(\Delta(d),\ \Delta(z))$

9. $E = -det(A(d))$

Output: $Q = A(d)^2,\ c = A(d)^{-1}y(d)$

With the solved optimal solution of $(Q,\ c)$, we then check whether c is located within the convex hull of the input point set $\{x_1,\ x_2, ..., x_m\}$. If it is not, constrained gaussian mixture model will be applied to partition it into two subsets and solve the minimum-volume covering ellipsoids problem again in the two subsets. Through solving the above minimum-volume covering ellipsoids problem, we can characterize the columnar regions more accurate.

Note that from constrained GMM, each cluster has the corresponding mean and covariance matrix of points in the cluster. These two values essentially describe the shape of the cluster. However, if these two values directly replace c^* and Q^*, the exported ellipsoid may only encompass a part of points in the cluster. For covering all points in the cluster, all elements in the covariance matrix are

needed to proportionally enlarged, but the volume of the corresponding ellipsoid is not minimum. These two cases will reduce the accuracy of the connections between clusters, i.e., columnar regions. So, we introduce the minimum-volume covering ellipsoid model to extract the shape of columnar region.

Skeleton generation using 0-1 assignment model

The 0-1 assignment model[43] can robustly and accurately establish connections between particles in live-cell imaging[55]. It is particularly effective in handling cases where particles are densely distributed, merged, or split. We analogize column regions to particles, and apply the 0-1 assignment model to build the connections between column regions. For the *i*-th columnar region, the center and the two endpoints of the longest axis of its minimum-volume covering ellipsoid are denoted by $c_i,\ t_{i,0},\ t_{i,1}$. The direction refers to the pointing of the center point towards $t_{i,k}$, k equals to 0 or 1. According to the direction and the endpoints, we design the cost matrix for building the 0-1 assignment model.

$$C=\begin{bmatrix} c(t_{1,0},t_{1,0}) & c(t_{1,0},t_{1,1}) & \cdots & c(t_{1,0},t_{n,1}) & \\ c(t_{1,1},t_{1,0}) & c(t_{1,1},t_{1,1}) & \cdots & c(t_{1,1},t_{n,1}) & D \\ \vdots & \vdots & \vdots & \vdots & \\ c(t_{n,1},t_{1,0}) & c(t_{n,1},t_{1,1}) & \cdots & c(t_{n,1},t_{n,1}) & \\ & & & & \\ & D & & & D \end{bmatrix}_{4n\times 4n} \tag{46}$$

$$c(t_{i,i0},t_{j,j0})=\begin{cases} 100 & if\,(i=j) \\ \dfrac{norm(t_{i,i0},t_{j,j0})}{\left(0.5\times\left(\dfrac{\theta(t_{i,i0},t_{j,j0})}{3}+1.001\right)\right)^4} & if\,(i\neq j) \end{cases} \tag{47}$$

$$\begin{aligned}\theta(t_{i,i0},t_{j,j0})=&\left\langle dir(c_i,t_{i,i0}),dir(c_i,t_{j,j0})\right\rangle+\left\langle dir(c_j,t_{j,j0}),dir(c_j,t_{i,i0})\right\rangle \\ &-\left\langle dir(c_i,t_{i,i0}),dir(c_j,t_{j,j0})\right\rangle\end{aligned} \tag{48}$$

Here D is $2n\times 2n$ auxiliary matrix all element of which are all set 100. Both $i0$ and $j0$ in EQ (47) are equal to 0 or 1, labeling the two endpoints of the longest axis of the ellipsoid. $norm(t_{i,i0},t_{j,j0})$ represents the Euclidean distance between $t_{i,i0}$ and $t_{j,j0}$. $\theta(t_{i,i0},t_{j,j0})$ describes the angle between two ellipsoids, i.e., two columnar regions. $dir(c_i,t_{i,i0})$ represents the line from point c_i to $t_{i,i0}$. $\left\langle dir(c_i,t_{i,i0}),dir(c_i,t_{j,j0})\right\rangle$ represents cosine angle between the two lines. The threshold of 100 in D in EQ (46) and EQ (47) is an experimental value designed to ensure that the terminal points of neurites do not connect to more than one other terminal points.

After set the cost matrix, the 0-1 assignment problem is defined as follow:

$$A=\arg\min_A \sum_{i=1}^{4n}\sum_{j=1}^{4n} A_{ij}C_{ij} \tag{49}$$

$$s.t.\ \sum_{i=1}^{4n} A_{i,j}=1 \quad (j=1,2,\cdots,4n) \tag{50}$$

$$\sum_{j=1}^{4n} A_{i,j}=1 \quad (i=1,2,\cdots,4n) \tag{51}$$

Here A represents the connectivity matrix between different terminals of columnar regions: if $A_{i,j}=1$, then establish connection between terminal *i* and terminal *j*, if $A_{i,j}=0$, then establish no connection between terminal *i* and terminal *j*. $\sum_{i=1}^{4n} A_{i,j}=1$ $(j=1,2,\cdots,4n)$ and $\sum_{j=1}^{4n} A_{i,j}=1$ $(i=1,2,\cdots,4n)$ restrain each terminal establish connection with at most one other terminal. The Lapjv algorithm[43] is utilized to solve this optimization problem and the shapes of

individual neurites in block images are formed. Furthermore, we employ the region growing method to generate skeletons from the reconstructed shape, achieving the neurites reconstruction from individual image blocks.

Minimal information flow tree for revising the reconstruction

The minimal information flow tree model is designed to modify the topology of skeletons, eliminate incorrect connections, and decompose them into multiple branches. When given an input skeleton file such as the swc file[56], we convert it into a binary tree structure with following steps.

Step1: select the neurite skeleton S_1. S_1 has the largest length in the neurite skeletons that connect with each other. One of its terminal nodes are recorded as the head node n_1.

Step2: generate the initial tree structure. Starting at head node n_1, search the linking nodes along the skeleton S_1, denoted by $n_1^{s_1}, n_2^{s_1}, \cdots, n_{k_1}^{s_1}$. The topology structure is $n_i \to leftnode = n_{i+1}^{s_1}$.

Step3: generate new structure induced by the linking node $n_1^{s_1}$. $n_1^{s_1}$ is regarded as the head node and its corresponding neurite skeleton is denoted by S_2. Let $n_1^{s_2}, n_2^{s_2}, \cdots, n_{k_2}^{s_2}$ represent the linking nodes in skeleton S_2 .The corresponding topology structure is $n_1^{s_1} \to rightnode = n_1^{s_2}$, $n_i^{s_2} \to leftnode = n_{i+1}^{s_2}$.

Step4: repeat the operation in **Step3** for dealing with the linking nodes $n_2^{s_1}, \cdots,$ $n_{k_1}^{s_1}$. The corresponding topology structures are added into the total tree structure. After obtaining the tree structures induced by linking nodes in S_1, use the operation in **Step3** to generate the tree structures induced by linking nodes in S_2. Continue in this manner until all linking nodes have been processed.

To gain a better understanding of the above process, we have provided a demonstration of how to generate the corresponding binary tree from the skeletons of neurites **(Fig.S6)**.

For the skeletons of neurites in an image block, the corresponding number of binary tree structures will be generated. We use the MIFT model to merge or split these binary structures. Suppose that an image stack contains m skeletons all of which have K nodes, denoted by $n_1, \cdots, n_{K-1}, n_K$. The connections among these nodes are stored in a matrix W with $K \times K$ elements. $W_{i,j} = 0$ indicates that there is no connection between node i and node j. $W_{i,j} = -1$ indicates that $j \to headnode = i$, $W_{i,j} = -2$ indicates that $j \to leftnode = i$, $W_{i,j} = -3$ indicates that $j \to rightnode = i$.

The information flow can be computed as follow:

$$W^* = \arg\min_W \sum_{i=1}^{K} f(W, n_i) \tag{52}$$

$$f(W, n_i) = \cos\left(\theta(n_i \to headnode, n_i, n_i \to leftnode)\right) \tag{53}$$

Here the optimization objective function in EQ (53) is called information flow. $\theta(\cdot)$ is the angle between flow from $n_i \to headnode$ to n_i and flow from n_i to $n_i \to leftnode$. To minimize optimization problem while ensuring that the topology matrix W does not exhibit abnormal values, we adopt the strategy of dynamic programming to update topology matrix W. Briefly, we calculate the other two possible angles $\theta(n_i \to headnode, n_i, n_i \to rightnode)$ and $\theta(n_i \to leftnode, n_i, n_i \to rightnode)$ at the first linking node n_i. The minimum information flow is selected and W is updated. Following the updated W, the next branching node is found and information flow and W is updated. The updating process iterates until all nodes are updated. The final root nodes $\{r_1, r_2, ..., r_m\}$ are obtained (node satisfies $W(r_t, i) = 0 \; or \; -1 (i = 1, ...n)$ is set root node). The pseudo-code for solving the optimization problem is provided below:

Algorithm 1 Generation of Minimal Information Flow Tree

Graph defines tree topology of the nodes, t_node->left represent the left child node of t_node, t_node->right represent the right child node of t_node, t_node->head represent the head node of t_node.

Input: N: $\{N_0, N_1, ..., N_k\}$, Graph head: $\{N_0\}$

$Set = \{N_0\}$

While $|Set| > 0$:

$t_node = Set[0]$

calculate three possible information flow

$res = calc_three_directions(t_node)$

if $(res == 0)$:

maintain original structure.

$Set[0] = t_node->left$

$Set.push_back(t_node->right)$

if $(res == 1)$:

change the position of t_node's two child node.

$Exchange_child(t_node)$

$Set[0] = t_node->left$

$Set.push_back(t_node->right)$

if $(res == 2)$:

Information flows from t_node->left to t_node->right, update the structure along t_node->left and t_node->head, generate new head if possible.

$New_node = Reverse_head(t_node)$

$Set[0] = New_node$

Output: N: $\{N_0, N_1, ..., N_k\}$, Graph head: $\{N_0^{'}, N_1^{'}, ..., N_m^{'}\}$.

Please note that the model has the capability to merge binary trees. When two branches of neurites have identifiable root nodes, and one root node is in close proximity to the skeleton points on the other branch of neurites, the root node does not contribute to the calculation of information flow without fusion. However, after fusion, the root node becomes a linking node in the other branch of neurites, resulting in an additional negative information flow value. In this merging process, a threshold is required to be set. When the minimum distance between the root node of a branch of neurites and the skeleton point of the other branch of neurites is less than 8 for individual image blocks or less than 8,12,16 for fused image blocks respectively, these two branches are merged. When splitting a branch of neurites, the minimal information flow tree model is also applied to both individual and fused image blocks.

The fusion of neurites reconstruction

By using MIFT model to revise the neurites reconstruction in individual image blocks, the root nodes and leaf nodes of a branch of neurites can be extracted directly. Here, we use 0-1 assignment model to merge the reconstructions between two adjacent image blocks. For two adjacent image blocks P and Q, the neurite skeleton nodes which locate near the common boundary are extracted as $\{p_1, p_2, ...p_m\}$, $\{q_1, q_2, ...q_n\}$ and the cost matrix is constructed as follow:

$$C = \begin{bmatrix} c(p_1,q_1) & \cdots & c(p_1,q_n) & \\ \vdots & \ddots & \vdots & D_{m\times m} \\ c(p_m,q_1) & \cdots & c(p_m,q_n) & \\ & D_{n\times n} & & D_{n\times m} \end{bmatrix}_{(m+n)\times(m+n)} \tag{54}$$

$$c(p_i,q_j) = d(p_i,q_j)\times\left(2-\theta\left(L(p_i), L(q_j)\right)\right) \tag{55}$$

Here $D_{m\times m}$, $D_{n\times n}$, $D_{n\times m}$ are auxiliary matrix which the values are all set 20. $d(p_i,q_j)$ represents the Euclidean distance between terminal p_i and q_j. $L(p_i)$ and $L(q_j)$ are fitted lines from the skeleton points near p_i and q_j. $\theta(L(p_i), L(q_j))$ represents the cosine value of their angle. Thus, the 0-1 assignment problem is formed as follow:

$$A = \arg min_A \sum_{i=1}^{m+n}\sum_{j=1}^{m+n} A_{i,j}\cdot C_{i,j} \tag{56}$$

$$s.t. \sum_{i=1}^{m+n} A_{i,j} = 1\left(j=1,2,\ldots m+n\right) \tag{57}$$

$$\sum_{j=1}^{m+n} A_{i,j} = 1\left(i=1,2,\ldots m+n\right) \tag{58}$$

Here A represents the connectivity relationship between nodes, if $A_{i,j}=1$, there is connection between block P's node i and block Q's node j, if $A_{i,j}=0$, there is no connection between block P's node i and block Q's node j. $\sum_{i=1}^{m+n} A_{i,j} = 1\left(j=1,2,\ldots m+n\right)$ and $\sum_{j=1}^{m+n} A_{i,j} = 1\left(i=1,2,\ldots m+n\right)$ restrict each node connect to one other node at most. With solved matrix A, the neurite skeletons of adjacent blocks can be merged and fused skeleton structures can be obtained.

Statistical Analysis

In this study, three commonly used metrics defined in[39] were used, including precision, recall and f1-score are computed to measure the fidelity between the reconstruction results and the ground truth. They are defined as follow:

$$precision(R,G)=\frac{|R\cap G|}{|R|}=\frac{|TP|}{|R|} \tag{59}$$

$$recall(R,G)=\frac{|R\cap G|}{|G|}=\frac{|TP|}{|G|} \tag{60}$$

$$f1-score(R,G)=2\cdot\frac{precision\times recall}{precision+recall} \tag{61}$$

R represents the point set of reconstructed neurons, G represents the point set of the ground truth, $|\cdot|$ represents the number of points of a set. The three metrics are first computed on each individual neuron, and then averaged by weighting each neuron with its point number of its ground truth neuritis.

We also calculated the signal-to-ratio (SNR) of the data using the following method: For a given data block B and its corresponding ground-truth skeleton S, we first densify the skeleton S by using linear interpolation to ensure that the Euclidean distance between adjacent skeleton points is than 1 voxel. Next, we expand each skeleton point in the densified skeleton $S^{'}$ into a spherical mask with a radius of 3 voxels. The resulting region serves as the foreground $mask$. Finally, SNR is computed with mean intensity of foreground points and standard deviation of background points as follow:

$$Mean_{foreground}=\sum_{x\in B}I(x)\times\sigma_1(x)/\sum_{x\in B}\sigma_1(x) \tag{62}$$

$$Mean_{backgroud}=\sum_{x\in B}I(x)\times\sigma_2(x)/\sum_{x\in B}\sigma_2(x) \tag{63}$$

$$Std_{backgroud}=\sqrt{\sum_{x\in B}(I(x)-Mean_{background})^2\times\sigma_2(x)/\sum_{x\in B}\sigma_2(x)} \tag{64}$$

$$\sigma_1(x)=\begin{cases}1 & if\,(x\in S^{'})\\ 0 & if\,(x\notin S^{'})\end{cases} \tag{65}$$

$$\sigma_2(x)=\begin{cases}1 & if\,(x\notin mask)\\ 0 & if\,(x\in mask)\end{cases} \tag{66}$$

Here, $I(x)$ represents the signal intensity of the voxel at position x, the SNR is calculated by $Mean_{foregroud}$ and $Std_{backgroud}$ by the following formula:

$$SNR=10\log_{10}(Mean_{foreground}/Std_{background}) \tag{67}$$

Acknowledgments

We thank the members of the Britton Chance Center for Biomedical Photonics for advice and help in experiments.

Funding: This work was supported by National Natural Science Foundation of China (32471146) and the project N20240194.

Author contributions:

Methodology: T.Q., L.C.

Investigation: T.Q., L.C., T.F., X.Q., X.G., W.F.

Validation: L.C., T.F., Q.D, T.C, Y.Z.

Supervision: S.Z., T.Q., X.L., X.L., Q.H.

Writing: T.Q., L.C., T.F.

Competing interests:

Authors declare that they have no competing interests.

Data and materials availability:

The data for Figure 1C, Figure 2, Figure 3, Figure 4, Figure 5, Figure 6 and Fig.S5 is available in https://zenodo.org/records/15589145. The raw image blocks are extremely large and can be available on request from the corresponding author. The training code of the segmentation network is available on Github: https://github.com/FateUBW0227/Seg_Net. The software of PointTree and its user guideline are available on https://zenodo.org/records/15589145.

References

1 Parekh, R. & Ascoli, G. A. Neuronal morphology goes digital: a research hub for cellular and system neuroscience. *Neuron* **77**, 1017-1038 (2013).

2 Meijering, E. Neuron tracing in perspective. *Cytometry Part A* **77**, 693-704 (2010).

3 Zingg, B. *et al.* Neural networks of the mouse neocortex. *Cell* **156**, 1096-1111 (2014).

4 Huang, Z. J. & Luo, L. It takes the world to understand the brain. *Science* **350**, 42-44 (2015).

5 Cai, R. *et al.* Panoptic imaging of transparent mice reveals whole-body neuronal projections and skull–meninges connections. *Nature neuroscience* **22**, 317-327 (2019).

6 Chung, K. & Deisseroth, K. CLARITY for mapping the nervous system. *Nature methods* **10**, 508-513 (2013).

7 Economo, M. N. *et al.* A platform for brain-wide imaging and reconstruction of individual neurons. *elife* **5**, e10566 (2016).

8 Kim, T. H. & Schnitzer, M. J. Fluorescence imaging of large-scale neural ensemble dynamics. *Cell* **185**, 9-41 (2022).

9 Li, A. *et al.* Micro-optical sectioning tomography to obtain a high-resolution atlas of the mouse brain. *Science* **330**, 1404-1408 (2010).

10 Osten, P. & Margrie, T. W. Mapping brain circuitry with a light microscope. *Nature methods* **10**, 515-523 (2013).

11 Zeng, H. What is a cell type and how to define it? *Cell* **185**, 2739-2755 (2022).

12 Xu, F. *et al.* High-throughput mapping of a whole rhesus monkey brain at micrometer resolution. *Nature biotechnology* **39**, 1521-1528 (2021).

13 Qiu, S. *et al.* Whole-brain spatial organization of hippocampal single-neuron projectomes. *Science* **383**, eadj9198 (2024).

14 Peng, H. *et al.* Morphological diversity of single neurons in molecularly defined cell types. *Nature* **598**, 174-181 (2021).

15 Gao, L. *et al.* Single-neuron projectome of mouse prefrontal cortex. *Nature Neuroscience* **25**, 515-529 (2022).

16 Foster, N. N. *et al.* The mouse cortico–basal ganglia–thalamic network. *Nature* **598**, 188-194 (2021).

17 Muñoz-Castañeda, R. *et al.* Cellular anatomy of the mouse primary motor cortex. *Nature* **598**, 159-166 (2021).

18 Sun, Q. *et al.* A whole-brain map of long-range inputs to GABAergic interneurons in the mouse medial prefrontal cortex. *Nature neuroscience* **22**, 1357-1370 (2019).

19 Winnubst, J. *et al.* Reconstruction of 1,000 projection neurons reveals new cell types and organization of long-range connectivity in the mouse brain. *Cell* **179**, 268-281. e213 (2019).

20 Friedmann, D. *et al.* Mapping mesoscale axonal projections in the mouse brain using a 3D convolutional network. *Proceedings of the National Academy of Sciences* **117**, 11068-11075 (2020).

21 Zhou, H. *et al.* GTree: an open-source tool for dense reconstruction of brain-wide neuronal population. *Neuroinformatics* **19**, 305-317 (2021).

22 Wang, Y. *et al.* TeraVR empowers precise reconstruction of complete 3-D neuronal morphology in the whole brain. *Nature communications* **10**, 3474 (2019).
23 Zeng, H. & Sanes, J. R. Neuronal cell-type classification: challenges, opportunities and the path forward. *Nature Reviews Neuroscience* **18**, 530-546 (2017).
24 Li, R. *et al.* Precise segmentation of densely interweaving neuron clusters using G-Cut. *Nature communications* **10**, 1549 (2019).
25 Lichtman, J. W. & Denk, W. The big and the small: challenges of imaging the brain's circuits. *Science* **334**, 618-623 (2011).
26 Helmstaedter, M. Cellular-resolution connectomics: challenges of dense neural circuit reconstruction. *Nature methods* **10**, 501-507 (2013).
27 Peng, H. *et al.* BigNeuron: large-scale 3D neuron reconstruction from optical microscopy images. *Neuron* **87**, 252-256 (2015).
28 Manubens-Gil, L. *et al.* BigNeuron: a resource to benchmark and predict performance of algorithms for automated tracing of neurons in light microscopy datasets. *Nature Methods* **20**, 824-835 (2023).
29 Choromanska, A., Chang, S.-F. & Yuste, R. Automatic reconstruction of neural morphologies with multi-scale tracking. *Frontiers in neural circuits* **6**, 25 (2012).
30 Li, S. *et al.* Brain-wide shape reconstruction of a traced neuron using the convex image segmentation method. *Neuroinformatics* **18**, 199-218 (2020).
31 Yang, J., Gonzalez-Bellido, P. T. & Peng, H. A distance-field based automatic neuron tracing method. *BMC bioinformatics* **14**, 1-11 (2013).
32 Peng, H., Long, F. & Myers, G. Automatic 3D neuron tracing using all-path pruning. *Bioinformatics* **27**, i239-i247 (2011).
33 Xiao, H. & Peng, H. APP2: automatic tracing of 3D neuron morphology based on hierarchical pruning of a gray-weighted image distance-tree. *Bioinformatics* **29**, 1448-1454 (2013).
34 Türetken, E., González, G., Blum, C. & Fua, P. Automated reconstruction of dendritic and axonal trees by global optimization with geometric priors. *Neuroinformatics* **9**, 279-302 (2011).
35 Zhou, Z., Kuo, H.-C., Peng, H. & Long, F. DeepNeuron: an open deep learning toolbox for neuron tracing. *Brain informatics* **5**, 1-9 (2018).
36 Li, Q. & Shen, L. 3D neuron reconstruction in tangled neuronal image with deep networks. *IEEE transactions on medical imaging* **39**, 425-435 (2019).
37 Huang, Q. *et al.* Weakly supervised learning of 3D deep network for neuron reconstruction. *Frontiers in Neuroanatomy* **14**, 38 (2020).
38 Liu, Y., Wang, G., Ascoli, G. A., Zhou, J. & Liu, L. Neuron tracing from light microscopy images: automation, deep learning and bench testing. *Bioinformatics* **38**, 5329-5339 (2022).
39 Quan, T. *et al.* NeuroGPS-Tree: automatic reconstruction of large-scale neuronal populations with dense neurites. *Nature methods* **13**, 51-54 (2016).
40 Gao, L. *et al.* Single-neuron analysis of dendrites and axons reveals the network organization in mouse prefrontal cortex. *Nature Neuroscience* **26**, 1111-1126 (2023).
41 Reynolds, D. A. Gaussian mixture models. *Encyclopedia of biometrics* **741** (2009).
42 Sun, P. & Freund, R. M. Computation of minimum-volume covering ellipsoids. *Operations Research* **52**, 690-706 (2004).
43 Volgenant, A. Linear and semi-assignment problems: a core oriented approach. *Computers & Operations Research* **23**, 917-932 (1996).
44 Harris, R. Models of regional growth: past, present and future. *Journal of economic surveys* **25**, 913-951 (2011).

45 Feng, L., Zhao, T. & Kim, J. neuTube 1.0: a new design for efficient neuron reconstruction software based on the SWC format. *eneuro* **2** (2015).
46 Radojević, M. & Meijering, E. Automated neuron tracing using probability hypothesis density filtering. *Bioinformatics* **33**, 1073-1080 (2017).
47 Wu, J. *et al.* 3D BrainCV: simultaneous visualization and analysis of cells and capillaries in a whole mouse brain with one-micron voxel resolution. *Neuroimage* **87**, 199-208 (2014).
48 Stelzer, E. H. *et al.* Light sheet fluorescence microscopy. *Nature Reviews Methods Primers* **1**, 73 (2021).
49 Wang, X. *et al.* Chemical sectioning fluorescence tomography: high-throughput, high-contrast, multicolor, whole-brain imaging at subcellular resolution. *Cell Reports* **34** (2021).
50 Zhong, Q. *et al.* High-definition imaging using line-illumination modulation microscopy. *Nature methods* **18**, 309-315 (2021).
51 Zhang, H. *et al.* Cross-streams through the ventral posteromedial thalamic nucleus to convey vibrissal information. *Frontiers in Neuroanatomy* **15**, 724861 (2021).
52 Çiçek, Ö., Abdulkadir, A., Lienkamp, S. S., Brox, T. & Ronneberger, O. in *Medical Image Computing and Computer-Assisted Intervention–MICCAI 2016: 19th International Conference, Athens, Greece, October 17-21, 2016, Proceedings, Part II 19.* 424-432 (Springer).
53 Wei, X., Peng, M., Huang, H. & Zhou, Y. An overview on density peaks clustering. *Neurocomputing* **554**, 126633 (2023).
54 McLachlan, G. J. & Krishnan, T. *The EM algorithm and extensions*. (John Wiley & Sons, 2007).
55 Jaqaman, K. *et al.* Robust single-particle tracking in live-cell time-lapse sequences. *Nature methods* **5**, 695-702 (2008).
56 Cannon, R. C., Turner, D. A., Pyapali, G. & Wheal, H. An on-line archive of reconstructed hippocampal neurons. *Journal of neuroscience methods* **84**, 49-54 (1998).

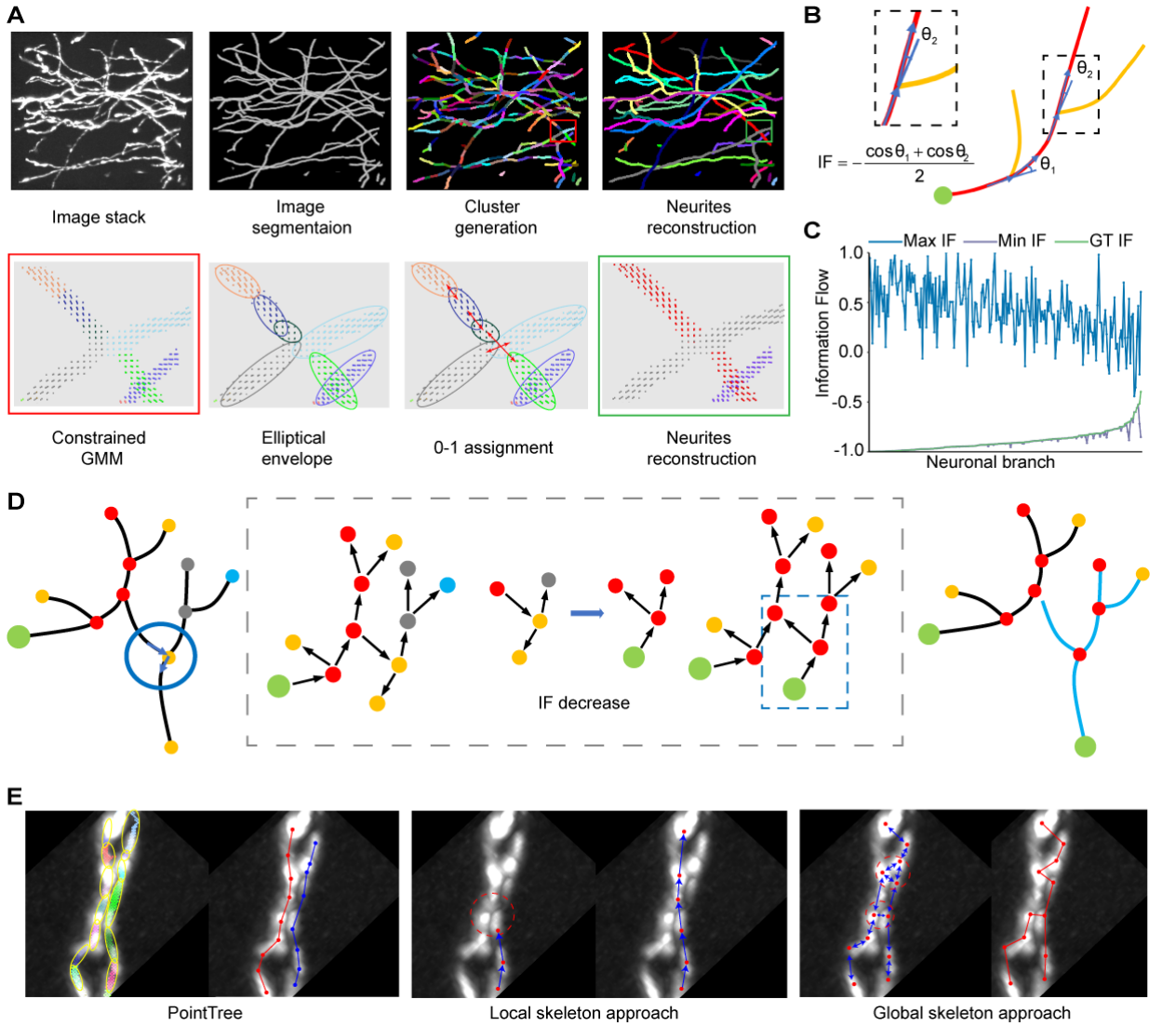

Figure 1. **Summary and principle of PointTree**. (**A**) The reconstruction procedure of PointTree involves the generation, clustering, and connection of foreground points (the first row). Within this procedure, three optimization problems are designed to allocate the foreground points into their respective neurites (the second row). (**B**) Schematic diagram of information flow score calculation. In a neurite branch with a fixed root node (green circle), the information flow score is calculated based on the assumption that a neurite has few directional changes. The assumption determines the neurite directly connecting to the root node (red), resulting in two branch angles used to calculate the information flow score. (**C**) Statistical analysis of the consistency between the minimum information flow and the real situation. For 208 neurite branches, the information flow scores are calculated as ground truth according to their manually determined skeletons and root nodes. These scores are then displayed in ascending order. The root nodes of neurite branches are changed to generate both maximum and minimum information flow scores. (**D**) One neurite branch is decomposed into two by minimizing the total information flow scores. (**E**) Performance of different methods on separating closely paralleled neurites. In PointTree, a single neurite is represented by a series of ellipsoids whose centerlines are not simultaneously located within different neurites. They are connected using ellipsoid shape which results in perfect reconstruction (Left). However, skeleton-based methods fail to separate two closely paralleled neurites due to interference from other signals (Red circle in middle) or connections being interfered with by another neighboring skeleton point (Red circle in right).

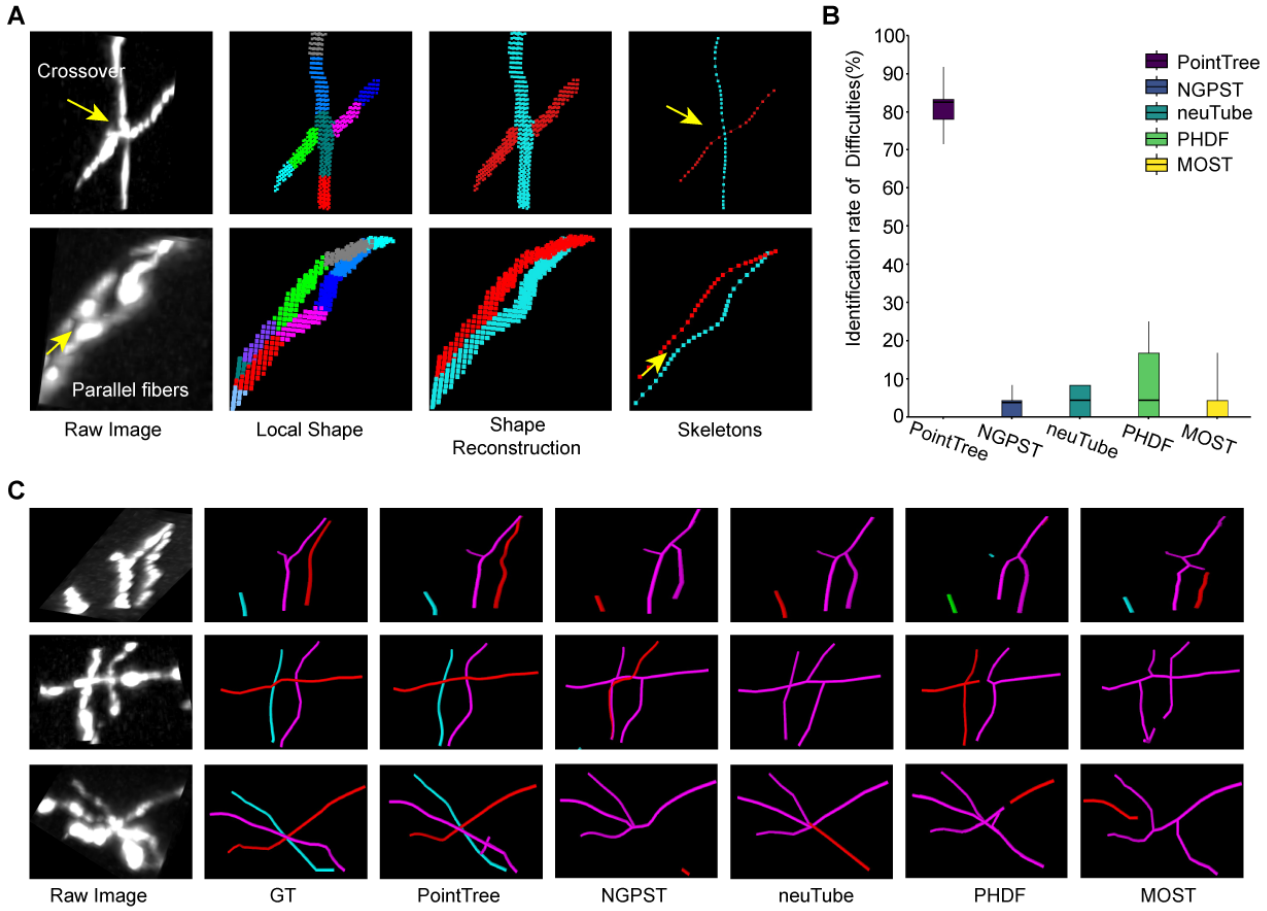

Figure 2. Performance of PointTree on crossover and closely paralleled neurites. (**A**) The reconstruction process of crossover and closely paralleled neurites. (**B**) Quantitative evaluation of PointTree and several skeleton-based methods on identifying closely distributed neurites. The box plots present the statistical information in which the horizontal line in the box, the lower and upper borders of the box represent the median value, the first quartile (Q1) and the third quartile (Q3) respectively. The vertical black lines indicate 1.5 × IQR. (**C**) Three reconstruction examples derived from PointTree and several skeleton-based methods.

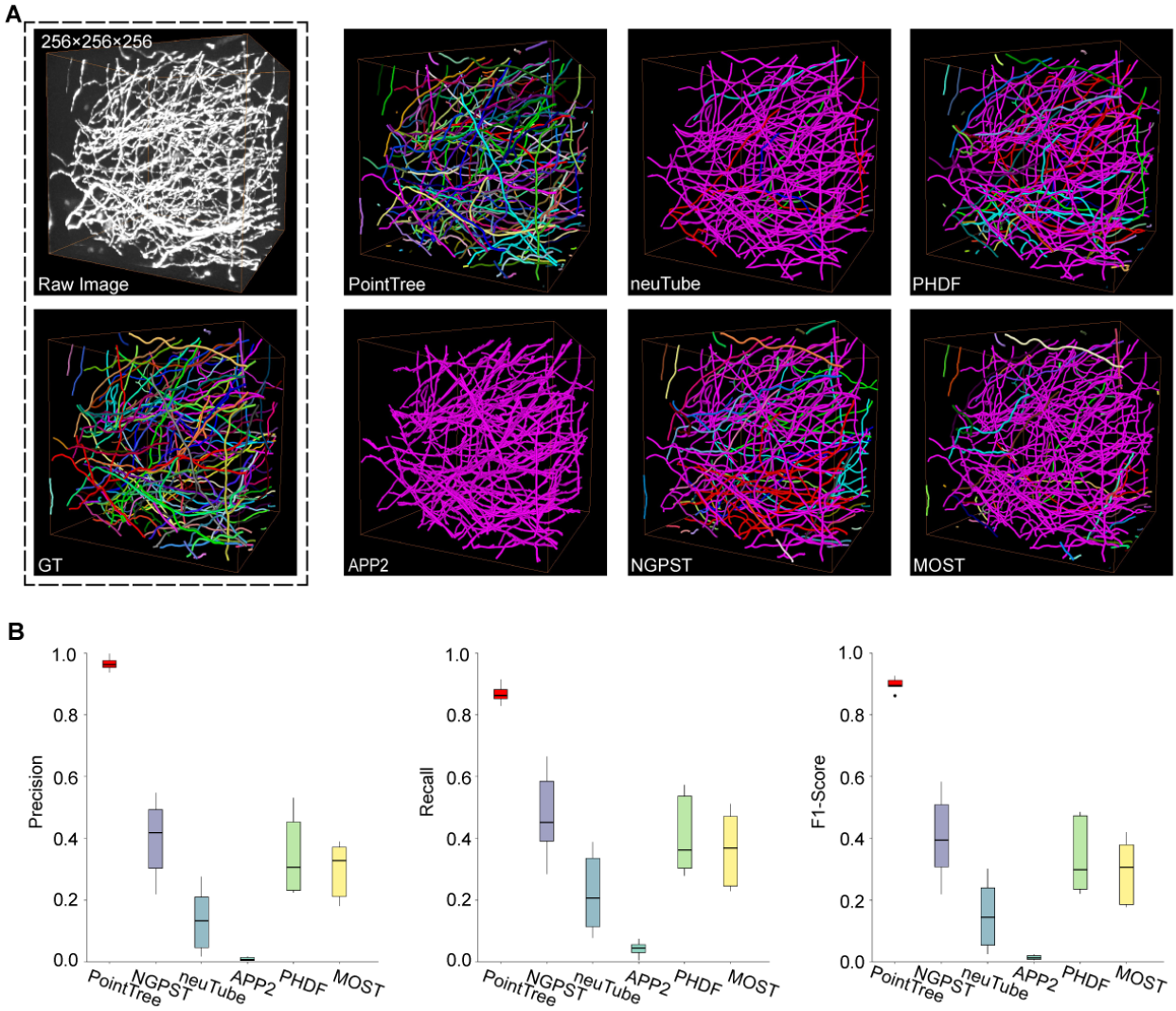

Figure 3. **Comparison of reconstruction methods on image blocks containing densely distributed neurites.** (**A**) Comparison of reconstruction performance among six methods, including PointTree, NGPST, neuTube, APP2, PHDF, and MOST. Individual neurite branches are delineated in different colors. (**B**) Quantitative evaluation of reconstruction performance using precision, recall, and f1-score. The box plots display these three evaluation indexes (n=8). In the box, the horizontal line represents the median value. The box shows the interquartile range (IQR) from the first quartile (Q1) to the third quartile (Q3). The vertical lines indicate 1.5× IQR.

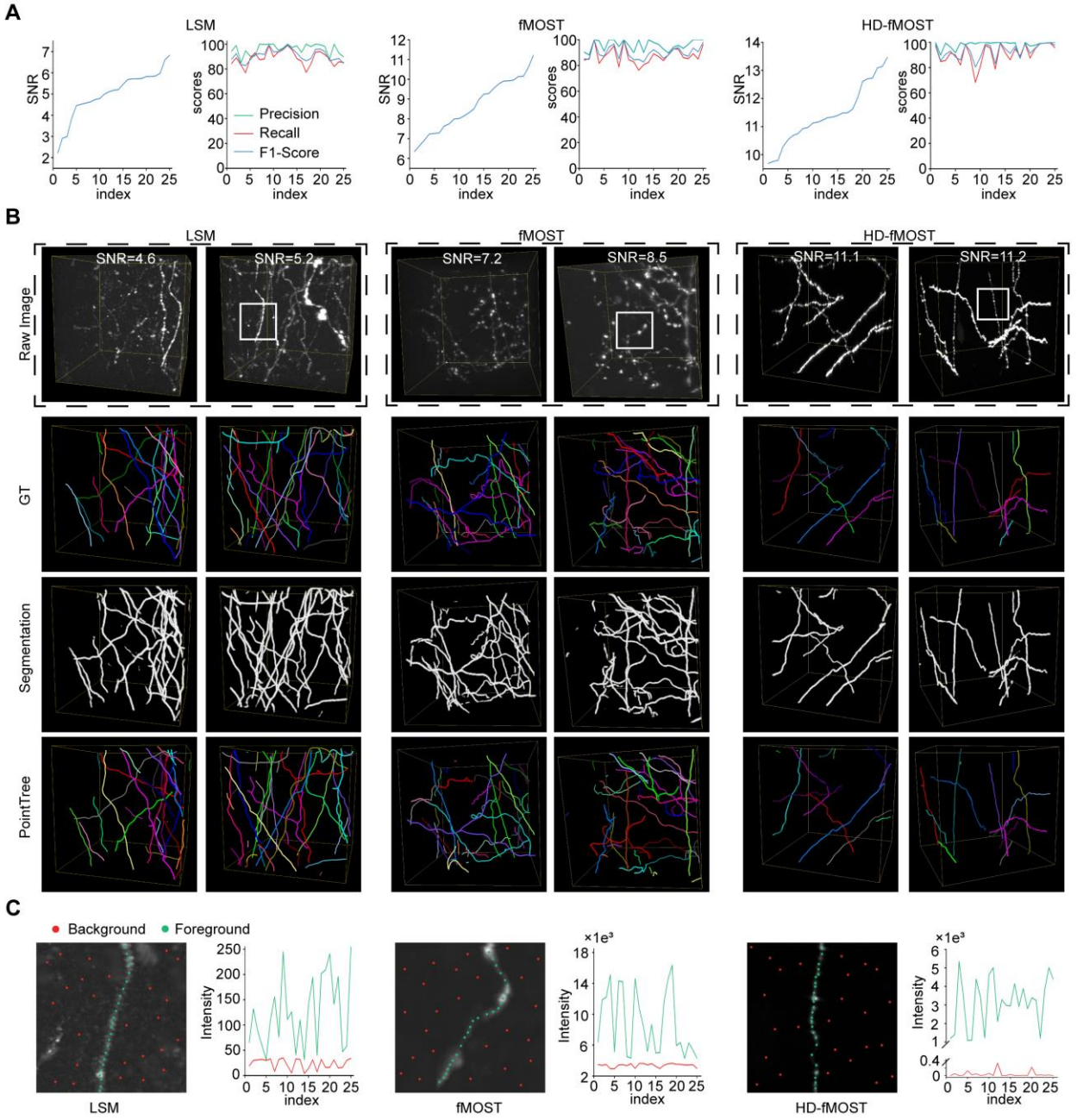

Figure 4. Reconstruction performance of PointTree across data with different signal-to-noise ratios. (**A**) Data blocks from light sheet microscopy (LSM), fluorescent micro-optical sectioning tomography (fMOST) and high-definition fluorescent micro-optical sectioning tomography (HD-fMOST) are selected. SNR and corresponding reconstruction scores with PointTree are drawn with line charts. Each dataset is of sample size n =25 and each data block size of 128×128×128. (**B**) shows reconstruction performance of PointTree on different datasets. (**C**) The zoomed-in view displays the region marked by white box in the first column of (B), with 25 foreground points and 25 background points sampled respectively. The signal intensities of both the foreground points and background points are plotted in the adjacent line charts.

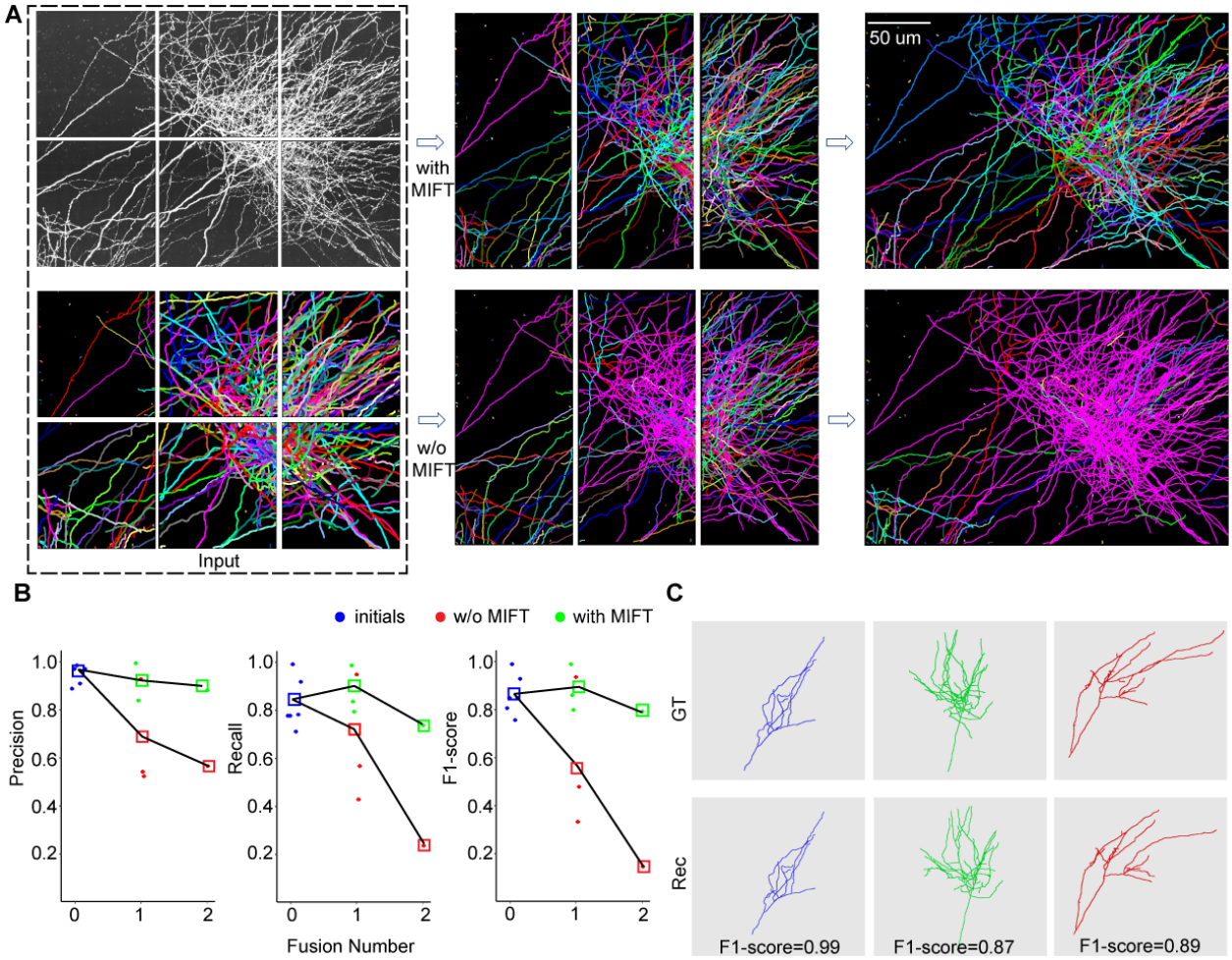

Figure 5. **Minimal information flow tree effectively restrains the accumulation of reconstruction errors.** (**A**) Reconstruction comparisons in the fusion process with MIFT and without MIFT are shown. Both image blocks and neurites reconstructions are displayed using maximum projection along the z-direction. Two fusion procedures are performed, and the final fusion reconstructions are presented in the third column. (**B**) The variation in reconstruction accuracy during the fusion process with MIFT and without MIFT is illustrated. Blue points represent the initial reconstruction accuracy from six image blocks, while green points and red points denote the merged reconstruction accuracy with MIFT and without MIFT, respectively. The squares represent the mean values of the evaluation indexes. (**C**) The skeletons of three neurite branches from the final merged reconstructions with MIFT are shown. Additionally, corresponding ground-truth reconstructions and reconstruction evaluations are also presented.

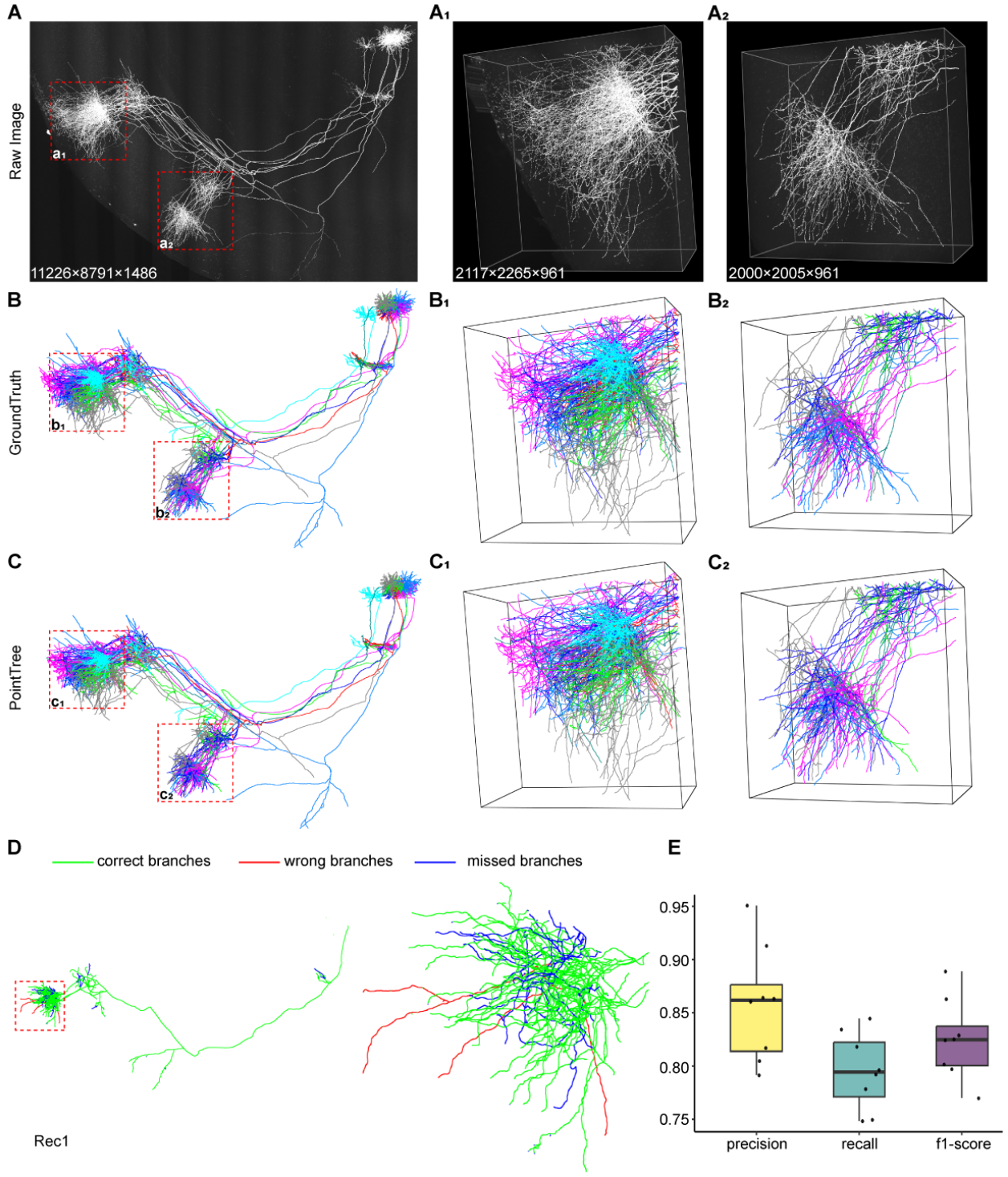

Figure 6. **Long-range axonal reconstruction using PointTree.** (**A**) The image block contains eight neurons in the ventral posteromedial thalamic region. The projection of these neurons includes a large number of densely distributed axons, which are enlarged in A_1 and A_2. (**B**) The reconstruction of the eight neurons is achieved by annotators with semi-automatic software GTree, serving as ground-truth reconstruction to evaluate automatic algorithms. The reconstructions B_1 and B_2 correspond to the image blocks A_1 and A_2. (**C**) Automatic reconstruction with PointTree results in reconstructions of the densely distributed axons, which are enlarged in C_1 and C_2. (**D**) A comparison between automatic reconstruction and ground-truth reconstruction of axonal projection for one neuron is shown. Green indicates consistent reconstruction, blue indicates missed branches, and red denotes branches from other neurons. (**E**) Quantitative analysis of long-range projections for these neurons is presented. Statistical information is displayed in boxes, while black points represent the accuracy of the reconstructions for these neurons.

Movie S1.

Reconstructed long-range axonal projections and raw image data shown in Fig6, individual axonal projections are delineated in different colors.

Movie S2.

Trace one of the reconstructed projections shown in Fig6.

Movie S3.

Example run of PointTree on Windows.