Journal of Neuroscience Methods 226 (2014) 88–102

Contents lists available at ScienceDirect

Journal of Neuroscience Methods

journal homepage: www.elsevier.com/locate/jneumeth

Computational Neuroscience

A modular hierarchical approach to 3D electron microscopy image segmentation

Ting Liu [a,b], Cory Jones [a,c], Mojtaba Seyedhosseini [a,c], Tolga Tasdizen [a,b,c,*]

[a] Scientific Computing and Imaging Institute, University of Utah, United States
[b] School of Computing, University of Utah, United States
[c] Department of Electrical and Computer Engineering, University of Utah, United States

HIGHLIGHTS

- Proposed a novel automatic hierarchical method for 2D EM image segmentation.
- Proposed a novel semi-automatic method for 2D EM image segmentation with minimal user intervention.
- Proposed a 3D linking method for 3D neuron reconstruction using 2D segmentations.
- Achieved close-to-human 2D segmentation accuracy.
- Achieved state-of-the-art 3D segmentation accuracy.

ARTICLE INFO

Article history:
Received 5 November 2013
Received in revised form 9 January 2014
Accepted 13 January 2014

Keywords:
Image segmentation
Electron microscopy
Hierarchical segmentation
Semi-automatic segmentation
Neuron reconstruction

ABSTRACT

The study of neural circuit reconstruction, i.e., connectomics, is a challenging problem in neuroscience. Automated and semi-automated electron microscopy (EM) image analysis can be tremendously helpful for connectomics research. In this paper, we propose a fully automatic approach for intra-section segmentation and inter-section reconstruction of neurons using EM images. A hierarchical merge tree structure is built to represent multiple region hypotheses and supervised classification techniques are used to evaluate their potentials, based on which we resolve the merge tree with consistency constraints to acquire final intra-section segmentation. Then, we use a supervised learning based linking procedure for the inter-section neuron reconstruction. Also, we develop a semi-automatic method that utilizes the intermediate outputs of our automatic algorithm and achieves intra-segmentation with minimal user intervention. The experimental results show that our automatic method can achieve close-to-human intra-segmentation accuracy and state-of-the-art inter-section reconstruction accuracy. We also show that our semi-automatic method can further improve the intra-segmentation accuracy.

1. Introduction

1.1. Motivation

Connectomics (Sporns et al., 2005), i.e., neural circuit reconstruction, is drawing attention in neuroscience as an important method for studying neural circuit connectivity and the implied behaviors of nervous systems (Briggman et al., 2011; Bock et al., 2011; Seung, 2011). It has also been shown that many diseases are highly related to abnormalities in neural circuitry. For instance, changes in the neural circuitry of retina can lead to corruption of retinal cell class circuitry, and therefore retinal degenerative diseases can be found by ultrastructural cell identity and circuitry examination, which at the same time implies certain strategies for vision rescue (Peng et al., 2000; Marc et al., 2003, 2007, 2008; Jones et al., 2003, 2005; Jones and Marc, 2005).

To study the connectivity of a nervous system, image analysis techniques are widely adopted as an important approach. For image acquisition, electron microscopy (EM) provides sufficiently high resolution on nanoscale to image not only intra-cellular structures but also synapses and gap junctions, which are required for neural circuit reconstruction. In this paper, the EM images we use for neuron segmentation and reconstruction are acquired using serial section transmission electron microscopy (ssTEM) (Anderson et al., 2009; Chklovskii et al., 2010), serial block-face scanning electron microscopy (SBFSEM) (Denk and Horstmann, 2004), and serial section scanning electron microscopy (ssSEM) (Horstmann et al., 2012). The use of ssTEM offers a relatively wide field of view for cell identification. However, since specimen sections have to be

* Corresponding author at: Scientific Computing and Imaging Institute, University of Utah, Salt Lake City, UT, United States. Tel.: +1 801 581 3539.
E-mail address: tolga@sci.utah.edu (T. Tasdizen).

0165-0270/$ – see front matter © 2014 Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.jneumeth.2014.01.022

cut prior to imaging, thin sections are difficult to acquire, and the image volume can be very anisotropic with as high as 2 nm in-plane resolution and approximately 50 nm z-resolution. Image deformations can be observed between sections to some extent even after image alignment, which is a challenging problem due to the ambiguity between structure changes and coordinate transformations (Tasdizen et al., 2010). These problems are to some degree avoided using SBFSEM, in which sections as thin as 25 nm are cut and removed, and the remaining block face is used for imaging. Since the remaining block is relatively solid and stable, the section shifts and deformations are generally smaller than ssTEM. On the other hand, the lower intra-section resolution (about 10 nm) is a potential drawback of SBFSEM. Recently, a novel approach based on ssSEM was introduced (Horstmann et al., 2012). This method combines serial sectioning specimen preparation and the scanning electron microscopy imaging technique, and is feasible to generate high resolution ultrastructural images of larger volumes than ssTEM. With the recent development of the automatic tape-collecting ultramicrotome (ATUM), sections can be made as thin as 25 nm without increasing section damage risk (Schalek et al., 2012). Another newly introduced imaging technique, focused ion beam scanning electron microscopy (FIBSEM) (Knott et al., 2008), can generate isotropic image data with as high as 10 nm z-resolution. However, currently, the ion beam milling used for removing sections can work on a maximum area of only $100 \times 100\,\mu m$, and thus it is not feasible for imaging substantially large tissue volumes, which is crucial to eventually map the entire human brain neural connectivity.

The image datasets generated by EM are on terabyte scale (Anderson et al., 2011), and dense manual analysis can take decades (Briggman and Denk, 2006) and is thus infeasible. Therefore, automatic or semi-automatic image-based connectome reconstruction techniques that can extensively reduce human workloads are highly required. Currently, fully automatic EM image segmentation and connectome reconstruction remain as a challenging problem because of the complex ultrastructural cellular textures and the considerable variations in shape and physical topologies within and across image sections (Jain et al., 2007; Lucchi et al., 2010). Even though segmentation of intra-cellular structures (e.g., mitochondria, synapses) is of importance, our work focuses on neuron segmentation and reconstruction as a first step for automating connectomics.

1.2. Related work

For fully automatic neuron segmentation and reconstruction using EM images, there are two general approaches. One approach focuses on segmenting neurons in 2D images and making intersection linkings for 3D reconstruction, which is suggested by the current anisotropy of most EM image volumes except for FIBSEM. As for 2D neuron segmentation, several unsupervised attempts were made. Anisotropic directional filtering is applied to enhance membrane continuity (Tasdizen et al., 2005; Jurrus et al., 2008), but fails to detect membranes with sufficient accuracy and it cannot remove intra-cellular structures. Kumar et al. (2010) introduced the radon-like features to suppress undesirable intra-cellular structures, but can achieve only moderate accuracy performance. On the other hand, supervised learning methods have proven successful in detecting membranes and segmenting neurons in 2D. Jain et al. (2007) trained a convolutional network to classify pixels as membrane or non-membrane. Mishchenko (2009) used a single-layer artificial neural network with Hessian eigenspace features to identify cell boundaries. Ciresan et al. (2012) applied deep neural networks (DNN) for membrane detection and achieved remarkable results. Laptev et al. (2012) used SIFT flow to align adjacent image sections and incorporate both intra- and inter-section pixel information for membrane detection. Seyedhosseini and Tasdizen (2013) proposed a multi-class multi-scale series contextual model that utilizes both cross- and inter-object information under a serial neural network framework for the detection of membranes and other cellular structures simultaneously. Given membrane detection results, the classifier output can simply be thresholded (Jurrus et al., 2010; Seyedhosseini et al., 2011; Ciresan et al., 2012) to acquire region segmentation. Other more sophisticated methods were proposed to further improve the segmentation results. Kaynig et al. (2010) proposed a graph-cut framework with perceptual grouping constraints to enhance closing of membrane gaps. Liu et al. (2013) used a merge forest structure to incorporate inter-section information to improve 2D neuron segmentation. For 3D linking given 2D segmentation, Yang and Choe (2009) proposed a graph-cut framework to trace 2D contours in 3D. Kaynig et al. (2010) exploited geometrical consistency constraints and used the expectation maximization algorithm to optimize the 3D affinity matrix. Vitaladevuni and Basri (2010) considered the 3D linking as co-clustering each pair of adjacent sections and formulated it as a quadratic optimization problem.

The other group of methods seek to achieve 2D segmentation and 3D reconstruction at the same time. Jain et al. (2011) proposed a reinforcement learning framework to merge supervoxels into segmentations. Andres et al. (2012) proposed a graphical model framework to incorporate both supervoxel face and boundary curve information for 3D supervoxel merging. Vazquez-Reina et al. (2011) generated multiple 2D segmentation hypotheses and formulated the 3D segmentation fusion into a Markov random field framework. Similarly, Funke et al. (2012) used tree structures to represent 2D segmentation hypotheses and achieved 2D segmentation and 3D reconstruction simultaneously by solving an integer linear programming problem with constraints. Recently, Nunez-Iglesias et al. (2013) also used an reinforcement learning framework to learn a merging policy for superpixel agglomeration, which achieved notable result in isotropic FIBSEM image volume segmentation.

In addition to the automatic segmentation methods, there are several semi-automatic methods that utilize user input to improve segmentation in some way. Chklovskii et al. (2010) and EyeWire (Seung, 2013) over-segment images into 2D or 3D regions, and then let a user manually merge these regions together to form the final cell. Post-automatic manual correction methods such as these are sometimes called proofreading methods. Other examples of semi-automatic segmentation include (Vazquez et al., 1998; Vu and Manjunath, 2008; Macke et al., 2008; Jurrus et al., 2009; Jeong et al., 2009), which use various input from the user to initialize automatic methods.

The attempts to segment image volumes directly in 3D require the data to be close to isotropic in order to achieve acceptable results. Although the generation of image data with isotropic resolution becomes possible with the emergence of new EM imaging techniques, e.g., FIBSEM (Knott et al., 2008), the image volume size that can be generated is currently limited. A majority of current data sets of interest are anisotropic, for which direct 3D segmentation approaches may not be suitable.

We develop a two-step approach for 2D segmentation and 3D reconstruction, which is suitable for both anisotropic and isotropic data. Independent of a specific membrane detection algorithm, our method takes membrane probability maps as input for superpixel generation, and uses a hierarchical merge tree structure to represent the merging of multiple region hypotheses. We use supervised classification techniques to quantify the likelihoods of the hypotheses, based on which we acquire intra-section region segmentation via constrained optimization or user interaction. Then we apply a supervised learning based inter-section linking procedure for 3D neuron reconstruction. We show that our automatic method outperforms other state-of-the-art methods and achieves close-to-human errors in terms of 2D segmentation and state-of-the-art 3D

reconstruction accuracy. We also show that our semi-automatic method can further improve the 2D segmentation result with minimal user intervention. Compared with the other most recent work by Nunez-Iglesias et al. (2013), it is noteworthy that the use of node potential in our method (see Section 2.2.5) utilizes higher order information to make merging decisions than only thresholding boundary classifier output. Also, our merge tree framework makes it convenient to incorporate prior knowledge about segmentation, which is not easily achievable in the reinforcement learning framework (Nunez-Iglesias et al., 2013).

2. Methods

Let $\mathcal{X} = \{x_i\}$ be an input image. A segmentation $\mathcal{S} = \{s_i\}$ assigns each pixel an integer label that is unique for each object. A ground truth segmentation $\mathcal{G} = \{g_i\}$ is usually generated by human experts and is considered as the gold standard. This notation applies to arbitrarily dimensional images. In this paper, we refer to a 2D image as an image or a section, and a sequence of 2D images as an image stack. The accuracy of a segmentation $\mathcal{S}$ is measured based on its agreement with the ground truth $\mathcal{G}$. The measurement of agreement is introduced in Section 2.1.

2.1. Segmentation accuracy metric

For both purposes of learning and evaluating the segmentation, a metric that is sensitive to correct region separation but less sensitive to minor shifts in the boundaries is desired. Traditional pixel classification error metrics are not effective in this regard. Instead, we chose to use the modified Rand error metric from the ISBI EM Challenge (Arganda-Carreras et al., 2012, 2013).

The adapted Rand error is based on the pairwise pixel metric introduced by Rand (1971). For both the original Rand index and the adapted Rand error, the true positives (*TP*), true negatives (*TN*), false positives (*FP*) and false negatives (*FN*) are computed as:

$$TP = \sum_i \sum_{j>i} \lambda\left(s_i = s_j \wedge g_i = g_j\right), \tag{1}$$

$$TN = \sum_i \sum_{j>i} \lambda\left(s_i \neq s_j \wedge g_i \neq g_j\right), \tag{2}$$

$$FP = \sum_i \sum_{j>i} \lambda\left(s_i = s_j \wedge g_i \neq g_j\right), \tag{3}$$

$$FN = \sum_i \sum_{j>i} \lambda\left(s_i \neq s_j \wedge g_i = g_j\right), \tag{4}$$

where $\lambda(\cdot)$ is a function that returns 1 if the argument is true and 0 if the argument is false. Each of these calculations compares the labels of a given pixel pair in $\mathcal{S}$, (s_i, s_j), with the corresponding pixel pair in $\mathcal{G}$, (g_i, g_j), for all possible pixel pairs in the image. *TP*, which counts the number of pixels for which $s_i = s_j$ and $g_i = g_j$, and *TN*, which counts the number of pixels for which $s_i \neq s_j$ and $g_i \neq g_j$, correspond to accurately segmented pixel pairs. *FP*, which counts the number of pixels which $s_i = s_j$ but $g_i \neq g_j$, corresponds to under-segmented pixel pairs, and *FN*, which counts the number of pixels for which $s_i \neq s_j$ but $g_i = g_j$, corresponds to over-segmented pixel pairs. These pairs are considered across a single image for 2D evaluation and across a full image stack for 3D evaluation.

In the original Rand index, the metric is computed by dividing the number of true pairs both positive and negative by the total number of possible pairs in the image. This results in values between 0 and 1 with 1 being an indication of a perfect segmentation. The adapted Rand error, however, is 1 minus the *F*-score computed from these results using precision and recall where

$$\text{Precision} = \frac{TP}{TP + FP}, \tag{5}$$

$$\text{Recall} = \frac{TP}{TP + FN}, \tag{6}$$

$$F\text{-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}, \tag{7}$$

$$\textit{Adapted}\ \text{Rand error} = 1 - F\text{-score}. \tag{8}$$

Once again the values of the adapted Rand error are between 0 and 1, but with 0 now representing a perfect segmentation.

One advantage of using this metric is that when the error is high, the precision and recall provide additional information regarding the type of error that is present. When precision is high and recall is low, this is an indication that the image has been more over-segmented. Conversely if recall is high and precision is low, the indication is that the image is under-segmented resulting from regions being merged that should have been separate. Errors such as boundaries between regions being in the shifted locations or a mix of under- and over-segmentations will result in a more balanced value for precision and recall.

This metric is particularly useful in segmentation of EM images because it can be computed accurately independent of the number of regions segmented. One implementation consideration for our application is that all pixels where g_i corresponds to boundary pixels are ignored. The reason for excluding these pixels is that if the proposed region entirely encompasses the ground truth region and the boundary between regions falls within the true boundary, then the segmentation is accurate for the purposes of this application.

2.2. Intra-section segmentation[1]

In this section, we propose a fully automatic method for 2D EM image segmentation. We take a membrane detection probability map as input and transform it into an initial segmentation using the watershed algorithm. Then starting with the initial segmentation, we use an iterative algorithm to build the merge tree as a representation of region merging hierarchy. Non-local features are computed and a boundary classifier is trained to give predictions about region merging, based on which we assign a node potential to each node in the merge tree. We resolve the merge tree using a greedy optimization approach under a pixel consistency constraint and generate the final segmentation. Furthermore, we present a semi-automatic merge tree resolution approach to improve the segmentation accuracy while minimizing the user time consumption.

2.2.1. Pixel-wise membrane detection

Our intra-section segmentation method uses pixel-wise membrane detection probability maps as input. A membrane detection map $\mathcal{B} = \{b_i\}$ is a probability image with each pixel intensity $b_i \in [0, 1]$ as shown in Fig. 1b. Such membrane detection maps are commonly obtained using machine learning algorithms, such as (Jain et al., 2007; Laptev et al., 2012; Ciresan et al., 2012; Seyedhosseini et al., 2013). We can obtain a segmentation of the cells in the image by simply thresholding the probability map. With this approach, however, a few false negatives can cause significant under-segmentation errors. To address this problem, we propose a

[1] The preliminary version of the fully automatic method was presented in ICPR 2012 (Liu et al., 2012).

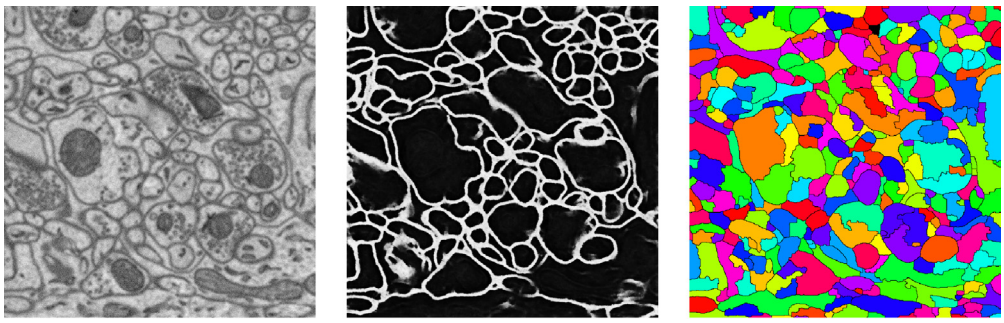

Fig. 1. Example of (a) an original EM image, (b) a membrane detection probability map, and (c) an initial segmentation. Each connected component in the initial segmentation is considered as an initial region. Some boundaries that are too faint to see in (b) result in regions since we aim for an initial over-segmentation.

region-based method for 2D segmentation. Our method is independent of how the membrane detection probability maps are obtained. In Section 3, we experiment with the membrane detection probability maps learned with cascaded hierarchical models (CHM) (Seyedhosseini et al., 2013) and deep neural networks (DNN) (Ciresan et al., 2012), and we show that our method can significantly improve the segmentation accuracy over thresholding either of these probability maps.

2.2.2. Initial segmentation

Based on the probability maps, we generate an initial segmentation for each image, which over-segments the image into superpixels. It is important to use the membrane detection probability maps instead of the original intensity images, because the former usually removes intra-cellular structures and preserves cell boundaries. Many methods can be applied to generate the superpixel initial segmentation (Shi and Malik, 2000; Felzenszwalb and Huttenlocher, 2004; Levinshtein et al., 2009; Veksler et al., 2010; Achanta et al., 2012). Our method uses, but is not limited to, the morphological watershed implementation (Beare and Lehmann, 2006) provided in the Insight Segmentation and Registration Toolkit (ITK) (Yoo et al., 2002), because of its high computational efficiency and good adherence to image boundaries. The basic idea of the watershed algorithm (Beucher and Lantuejoul, 1979) is to consider an image as a 3D terrain map with pixel intensities representing heights. As the rain falls into the terrain, water flows down along the steepest path and forms lakes in the basins, which correspond to regions or segments of the image, and the borders between lakes are called watershed lines. In terms of implementation, local minima of the image are used as seeds, from which regions are grown based on intensity gradients until the region boundaries touch. Fig. 1 shows an example of an initial segmentation. In practice, we blur the membrane detection probabilities with a Gaussian filter and ignore the local minima with dynamics below threshold t_w to avoid too many initial regions. Meanwhile, by using a small value of t_w, we can ensure over-segmentation. Also, we keep the one-pixel wide watershed lines as background, and use these points as boundary points between foreground regions.

2.2.3. Merge tree construction

Given a set of disjoint regions $\mathcal{R} = \{r_i\}$, in which each region r_i is a point set, and the background point set (watershed lines) r_b, we define the boundary between two regions r_i and r_j as a set of background points that are in the 4-connected neighborhoods of points from both of the two regions, respectively:

$$boundary(r_i, r_j) = \{p \in r_b \mid \exists p_{i'} \in r_i,\ p_{j'} \in r_j,\ \text{s.t.} p \in N_4(p_{i'}) \cap N_4(p_{j'})\}, \tag{9}$$

where $N_4(\cdot)$ represents the set of 4-connected neighbor points of a given point. If $boundary(r_i, r_j)$ is not the empty set, we say that region r_i and r_j are neighbors. Next, we define a merge of m regions $\{r_1, r_2, \ldots, r_m\}$ as the union of these region point sets as well as the set of boundary points between these regions:

$$merge(r_1, r_2, \ldots, r_m) = \left(\bigcup_{i=1}^{m} r_i\right) \cup \left(\bigcup_{i=1}^{m-1} \bigcup_{j=i+1}^{m} boundary(r_i, r_j)\right). \tag{10}$$

Notice that $merge(r_1, r_2, \ldots, r_m)$ is also a region. As an example shown in Fig. 2, region r_i and r_j merge to form region r_k, which consists of the points from both region r_i and r_j and also the boundary points between them. Next, we define a merging saliency function $f_{\mathcal{X},\mathcal{B}} : \mathcal{R}^m \to \mathbb{R}$, which takes a set of m regions and uses the pixel information from the original image and/or the membrane detection probability map to determine the merging saliency of the regions, which is a real number. Larger saliency indicates the regions are more likely to merge. In practice, we consider merging only two regions. In determining the merging saliency, we find that the membrane detection probability performs more accurately and consistently than the original image intensity. As an example

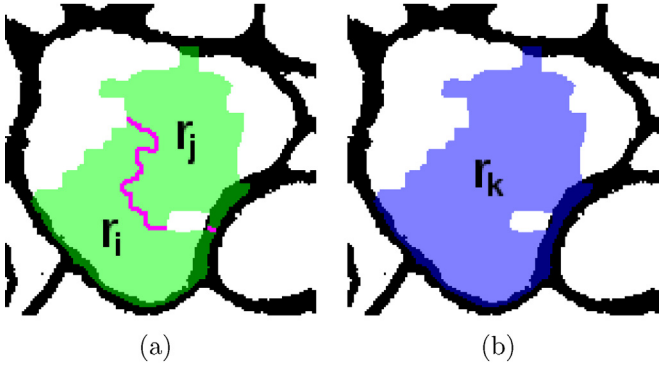

Fig. 2. Example of region merging. (a) Two merging regions r_i and r_j (green) and their boundary (magenta), and (b) the merging result region r_k (blue) are overlaid to the ground truth segmentation. The boundary is dilated for visualization purposes. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

shown in Fig. 1, some intra-cellular structures, e.g., mitochondria, look even darker than the membranes in the original image, and thus using these original intensities could give false indications about region merging saliency. In the membrane detection probability map, however, the strengths of such pixels are suppressed and the membrane pixels are distinguished. Also, we find that the median of the membrane probabilities of the boundary points between two regions is a good indication of region merging saliency and gives a more robust boundary measure than other statistics, such as minimum and mean. Thus, the merging saliency function is specified as

$$f_{\mathcal{B}}(r_i, r_j) = 1 - \text{median}(\{b_p \in \mathcal{B} \mid p \in \textit{boundary}(r_i, r_j)\}). \quad (11)$$

In practice, some regions in the initial segmentation could be too small to extract meaningful region-based features (see Section 2.2.4), so we optionally conduct a pre-merging step to merge initial regions smaller than a certain threshold t_{a_1} pixels to one of its neighbor regions that yields the largest merging saliency according to Eq. (11). We also pre-merge a region if its size is smaller than a certain threshold t_{a_2} pixels and its average intensity from the membrane detection probability map of all its pixels is above a certain threshold t_p.

To represent the hierarchy of region merging, we use a full binary tree structure $T = (\mathcal{N}, \mathcal{E})$, which we call a merge tree. In a merge tree, node $n_i^d \in \mathcal{N}$ represents a region $r_i \in \mathcal{R}$ in the image, where d denotes the depth in the tree at which this node occurs. The leaf nodes represent the regions from the initial over-segmentation. A non-leaf node corresponds to a region that is formed by one or more merges; the root node corresponds to the whole image as one single region. An undirected edge $e_{ij} \in \mathcal{E}$ between node n_i and its parent node n_j means region r_i is a subregion of region r_j, and a local structure $(\{n_j^d, n_{i_1}^{d+1}, n_{i_2}^{d+1}\}, \{e_{i_1 j}, e_{i_2 j}\})$ represents region r_j is the merging result of region r_{i_1} and r_{i_2}. Fig. 3 shows a merge tree example with the corresponding initial segmentation shown in Fig. 3a. Fourteen initial regions (region 1–14) correspond to the leaf nodes in the merge tree (node 1–14). The non-leaf nodes are the merging result of their descendant regions. For example, region 15 is the merging result of region 4 and 10. The root node 27 corresponds to the whole image.

To construct a merge tree, we start with the initial regions $\mathcal{R} = \{r_1, r_2, \ldots, r_{N_0}\}$, the corresponding leaf nodes $\mathcal{N} = \{n_1^{d_1}, n_2^{d_2}, \ldots, n_{N_0}^{d_{N_0}}\}$, and an empty edge set $\mathcal{E}$. First, we extract the set of all region pairs that are neighbors, $\mathcal{P} = \{(r_i, r_j) \in \mathcal{R} \times \mathcal{R} \mid i < j, r_j \in N_{\mathcal{R}}(r_i)\}$, where $N_{\mathcal{R}}(r_i)$ denotes the neighbor region set of r_i. Then we pick the region pair (r_{i^*}, r_{j^*}) from $\mathcal{P}$ that yields the largest merging saliency function output as

$$(r_{i^*}, r_{j^*}) = \underset{(r_i, r_j) \in \mathcal{P}}{arg\max} \; f_{\mathcal{B}}(r_i, r_j). \quad (12)$$

Let r_k be the merging result of r_{i^*} and r_{j^*}. A node n_k corresponding to r_k is added as the parent of node n_{i^*} and n_{j^*} into $\mathcal{N}$ and edges e_{i^*k} and e_{j^*k} are added to $\mathcal{E}$. Meanwhile, we append $\mathcal{R}$ with r_k, remove any region pair related to r_{i^*} or r_{j^*} from $\mathcal{P}$, and add all the region pairs concerning r_k into $\mathcal{P}$. This process is repeated until there is no region pair left in $\mathcal{P}$, and $T = (\mathcal{N}, \mathcal{E})$ is the merge tree.

2.2.4. Boundary classifier

In a merge tree, each non-leaf node and its children represent a potential region merging $(\{n_k^d, n_i^{d+1}, n_j^{d+1}\}, \{e_{ik}, e_{jk}\})$, the probability of which is needed for selecting the best segments. One possible solution is to use the merging saliency function output, which, however, relies only on the boundary cues, and cannot utilize the abundant information from the two merging regions. Instead, we train a boundary classifier to give such predictions by outputting the probability of region merging. The boundary classifier takes 88 non-local features computed from each pair of merging regions, including region geometry, image intensity, and texture statistics from both original EM images and membrane detection probability maps, and merging saliency information (see Appendix B for a summary of features). One advantage of our features over the local features commonly used by pixel classifiers (Laptev et al., 2012; Seyedhosseini et al., 2013) is that our features are extracted from regions instead of pixels and thus can be more informative. For instance, we use geometric features to incorporate region shape information for the classification procedure, which is not feasible for pixel classifiers.

To generate training labels that indicate whether the boundary between two regions exists or not, we utilize the ground truth segmentation of the training data. As shown in Fig. 2, when deciding whether region r_i and region r_j should merge to become region r_k, we compute the adapted Rand errors (see Section 2.1) for both merging (ε_k) and keeping split (ε_{ij}) against the ground truth. Either case with smaller error deviates less from the ground truth and should thus be adopted. Therefore, the training label is determined automatically as

$$y_{ij \to k} = \begin{cases} +1(\text{boundary exists, not merge}) & \text{if} \varepsilon_{ij} \le \varepsilon_k \\ -1(\text{boundary not exist, merge}) & \text{otherwise.} \end{cases} \quad (13)$$

The boundary classifier is not limited to a specific supervised classification model. We choose the random forest (Breiman, 2001) among various other classifiers because of its fast training and remarkable performance. Different weights are assigned to positive/negative samples to balance their contributions to the training process. The weights for positive sample w_{pos} and for negative samples w_{neg} are determined as

$$w_{\text{pos}} = \begin{cases} N_{\text{neg}}/N_{\text{pos}} & \textit{if} \;\; N_{\text{pos}} < N_{\text{neg}} \\ 1 & \text{otherwise} \end{cases}, \quad (14)$$

$$w_{\text{neg}} = \begin{cases} N_{\text{pos}}/N_{\text{neg}} & \text{if} \;\; N_{\text{pos}} > N_{\text{neg}} \\ 1 & \text{otherwise} \end{cases}, \quad (15)$$

where N_{pos} and N_{neg} are the number of samples with positive and negative labels, respectively.

2.2.5. Automatic merge tree resolution

After creating the merge tree, the task of generating a final segmentation of the image becomes choosing a subset of nodes in the merge tree. The boundary classifier predicts the probability for each potential merge, based on which we define a local potential for each node that represents the likelihood that it is a correct segmentation. Considering (a) a region exists in the final segmentation because it neither splits into smaller regions nor merges into others and (b) each prediction the boundary classifier makes depends only on the local merge tree structure, we define the potential for a node n_i^d as

$$P_i = p_{i_{c_1}, i_{c_2} \to i} \cdot (1 - p_{i, i_s \to i_p}), \quad (16)$$

where $p_{i_{c_1}, i_{c_2} \to i}$ is the probability that the child $n_{i_{c_1}}^{d+1}$ and $n_{i_{c_2}}^{d+1}$ merge to node n_i^d, and $p_{i, i_s \to i_p}$ is the probability that node n_i^d merges with its sibling $n_{i_s}^d$ to the parent $n_{i_p}^{d-1}$. Both of these probabilities come from the boundary classifier. As an example shown in Fig. 3, the potential of node 24 is $P_{24} = p_{20,21 \to 24}(1 - p_{2,24 \to 26})$. Since there are no children for a leaf node, its potential is computed as the square of 1 minus the probability that it merges with its sibling. Similarly, the potential of the root node is the square of the probability that its children merge.

Given the node potentials, we would like to choose a subset of nodes in the merge tree that have high potentials and preserve

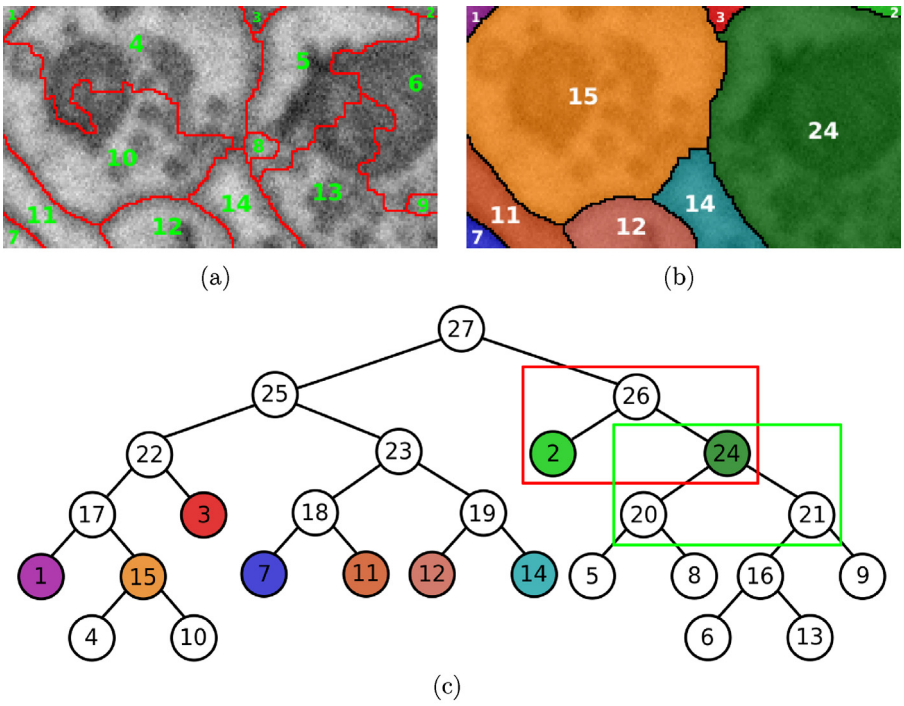

Fig. 3. Example of (a) an initial segmentation, (b) a consistent final segmentation, both overlaid with an original EM image, and (c) a corresponding merge tree. The leaf nodes have identical labels as the initial regions, and the colored nodes correspond to regions in the final segmentation. As an example of node potential computation described in Section 2.2.5, the potential of node 24 equals the probability that node 20 and 21 merge to node 24 (the green box), while node 2 and 24 do not merge to node 26 (the red box). (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

the pixel consistency of the final segmentation. Pixel consistency requires that any pixel should have exactly one label in the final segmentation, either a background label or one of the foreground labels, each of which is assigned to a unique connected component. Consequently, if a node in the merge tree is chosen, all of its ancestors and descendants cannot be selected; if a node is not chosen, one of its ancestors or a set of its descendants must be selected. In other words, exactly one node should be selected on the path from every leaf node to the root node. Fig. 3 shows an example: the colored nodes are picked to form a consistent final segmentation in Fig. 3. The other nodes cannot be picked, because we cannot label any pixel more than once. For example, if node 4 (or 10) is also picked along with node 15, the pixels in region 4 (or 10) would be labeled as both 15 and 4 (or 10), which violates the pixel consistency by definition.

Under the pixel consistency constraint, we use a greedy optimization approach to resolve the merge tree. First, we pick the node with the highest potential in the merge tree. Then the ancestors and descendants of the picked node are removed as inconsistent choices. This procedure is repeated until every node in the merge tree is either picked or removed. The set of picked nodes form a consistent 2D segmentation.

2.2.6. Semi-automatic merge tree resolution

On each of the data sets we have applied the automatic intra-section segmentation method to, the improvement over optimally thresholding the automatic intra-section segmentation has been significant. Depending on the data set, however, the final segmentation may not be sufficiently accurate for the application. To overcome situations where the resulting accuracy for a given data set is insufficient, we also developed a method for allowing manual input in resolving the merge tree. The goal with this method is to minimize the time requirement of the manual input while still improving the accuracy of the final result.

Using the Visualization Toolkit (Schroeder et al., 2006), we present the user with an image of the original image overlaid with the proposed segmentation. The user will then be presented with an unresolved region with the highest node potential in the merge tree, and specify whether it is good, over-segmented (if the region is too small), under-segmented (if the region is too large), or poorly segmented (if the region contains portions of multiple regions, but no complete region). Fig. 4 shows what the interface (Fig. 4a) and each segmentation possibility (Figs. 4b–4e) look like.

If the region is specified as good, the region is labeled in the label image and the ancestors and descendants are removed from the merge tree as described previously. The next region presented to the user follows the greedy approach as described in Section 2.2.5. Ideally this will be the dominant answer and it will take minimal time for the user to resolve the tree.

If the region is instead specified as either under-segmented or over-segmented, the proper portions of the tree are removed and the next region is presented according to this pick. When it is over-segmented, the descendants are removed and the parent is presented as the next region; when it is under-segmented, the ancestors are removed and the highest potential child is presented as the next region. Doing this allows the user to immediately resolve each region which will require less time than having to revisit and re-examine the region. If the parent in the case of over-segmented or the children in the case of under-segmented have already been removed from the tree structure, then the assumption of the correct segmentation of this region being in the tree fails and the next node presented once again follows the same greedy approach as above.

Finally, if the region is specified as bad, this is an indication that the correct segmentation for this region is not present in the tree. The ancestors and descendants along with the region presented are all removed from the tree and the next region presented will again follow the greedy approach. This will lead to some regions of the image being left unresolved. To accommodate this scenario,

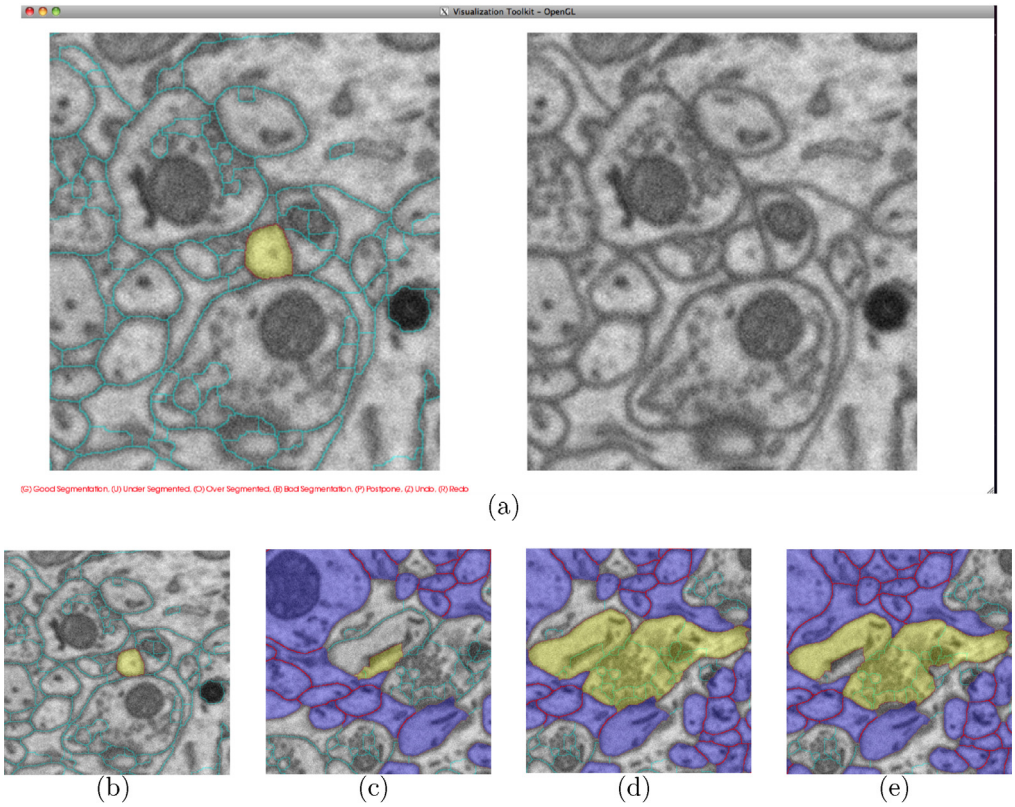

Fig. 4. Example of (a) the manual interface as the user sees it, (b) a good segmentation, (c) an over-segmentation, (d) an under-segmentation, and (e) a bad segmentation. The yellow area is the current region of interest. The blue areas are the regions that have been resolved. The cyan lines show region boundaries in the initial segmentation.

once all nodes in the tree have been visited or removed, the user is presented with the initial regions that are not yet part of a labeled region and allowed to manually combine these regions into the correct segmentations. Fig. 5 shows a flowchart illustrating the entire procedure of the semi-automatic merge tree resolution.

Our goal with this method is to be able to manually improve the segmentation quality while minimizing the time required by the user. By presenting the highest potential regions selected by the automatic method first, there will be many cases early on where the user can tell it is a good segmentation with a quick glance and it takes less than a second per cell to resolve. When the region is incorrect, having the user specify the type of segmentation results in removing many more possible regions while again limiting the amount of input required by the user.

2.3. Inter-section reconstruction

In order to reconstruct the wiring diagram of connections in a nervous system, we need not only the intra-section segmentation of each image slice, but also the inter-section reconstruction by grouping image pixels that belong to the same neurons together in 3D. We propose a fully automatic serial linking method that finds the spatial adjacency matrix for 2D segmentation regions across image slices. The output of our method is a label image stack, in which each reconstructed neuron is labeled uniquely. In other words, 2D regions that are on different sections but belong to the same neuron receive identical labels in the output.

Suppose we have a stack of N_s consecutive images and their 2D segmentations $\mathcal{S} = \{s_i^z \mid z \in \mathbb{Z}, 1 \le z \le N_s\}$, where z indicates on which section a 2D region is and i distinguishes regions within the same section. First, we generate a set of candidate links $\mathcal{L} = \{(s_i^{z_i}, s_j^{z_j}) \in \mathcal{S} \times \mathcal{S} \mid z_i < z_j\}$ that are between pairs of 2D regions that are on different sections. We use two kinds of links: adjacent links and non-adjacent links. Adjacent links are between region pairs on two adjacent sections, with which we seek to serially connect sequences of 2D regions into 3D bodies. In real EM data sets, image blurring and corruption occasionally happen on individual sections, which could lead to errors in 2D segmentation using our method. While it may not affect the overall 2D accuracy by too much, the error could be significantly amplified if it accidentally breaks a 3D body in the middle. To avoid such interrupted neuron tracing, we use non-adjacent links to compensate for the occasional adjacent missing links. Here we consider non-adjacent links only between regions on every other section, because large intersection shape variations due to high data anisotropy can make it difficult to reliably trace a neuron when skipping more sections. Moreover, we consider only pairs of regions whose orthographic projections overlap or are within a certain centroid distance t_{cd}. Then, we train a section classifier to predict whether each candidate link is true. Eventually, we threshold the section classifier predictions to eliminate undesirable candidates and incorporate the adjacent and non-adjacent links to generate the final label image stack.

2.3.1. Section classifier

For each candidate link $(s_i^{z_i}, s_j^{z_j})$, we train a section classifier with the ground truth data to give a prediction about its correctness. The section classifier takes 230 geometric features extracted from each region pair, including centroid distance, region sizes, overlapping, and other shape similarity information. We use the SIFT flow algorithm (Liu et al., 2011) to align consecutive sections and include features from both the aligned and non-aligned sections. A summary of features is provided in Appendix B.

To generate the training labels for the section classifier, we utilize the ground truth data, which gives the true 3D links between the ground truth 2D segmentation regions. Since our 2D

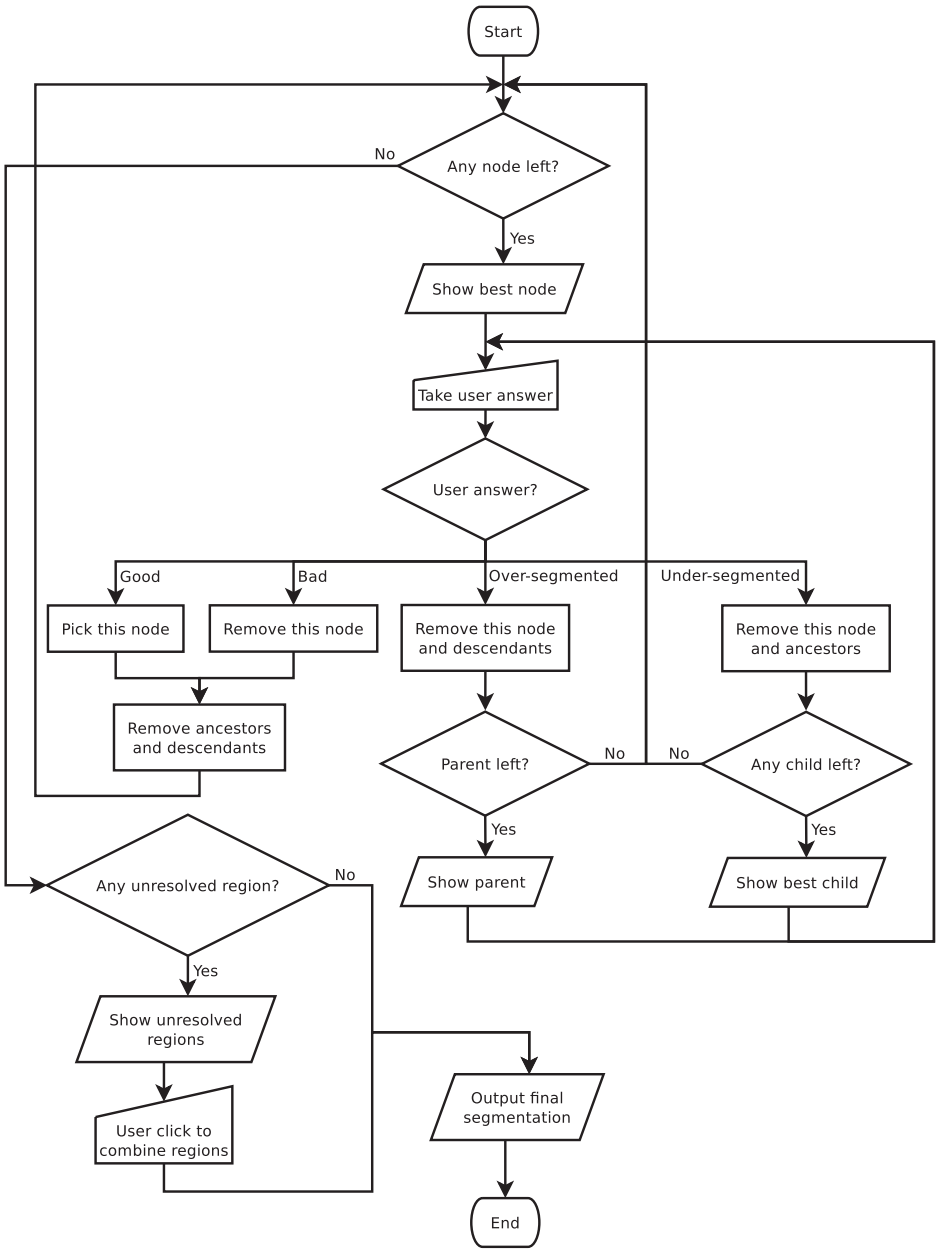

Fig. 5. Flowchart of the semi-automatic merge tree resolution procedure.

segmentation does not match the ground truth perfectly, each 2D segmentation region is matched to one ground truth region that shares the largest overlapping area. Then all the links between the ground truth regions that have at least one corresponding 2D segmentation region are regarded as the true links. All the other candidate links are considered as false links. Note that it is important to use the training 2D segmentation results instead of the ground truth 2D segmentations to generate the training link samples for the section classifier, because the training link samples using the training 2D segmentation results resemble better the testing data than those using the ground truth 2D segmentation and thus the section classifier can be better generalized for the testing samples.

We choose the random forest algorithm (Breiman, 2001) as the classifier. The training data can be very imbalanced, since there can be several candidate links from any given region but usually only one or two of them are true. Therefore, it is important to balance the training process by assigning different weights according to Eqs. (14) and (15). Two separate classifiers are trained for the adjacent and non-adjacent links, respectively.

2.3.2. Linking

We use the output probability for each candidate link as a weight and apply a thresholding strategy to preserve the most reliable links and remove all other candidates. We use a threshold t_{adj} for adjacent links and a separate threshold t_{nadj} for non-adjacent links. For a 2D segmentation region s_i^z, in the forward directions of z, we pick the adjacent links $(s_i^z, s_{j'}^{z+1})$ whose weights are above t_{adj}, and if there is no adjacent link picked for s_i^z, we pick the non-adjacent links $(s_i^z, s_{j'}^{z+2})$ whose weights are above t_{nadj}. Similarly, in the backward direction, the adjacent links (s_k^{z-1}, s_i^z) with weights above t_{adj} are picked, and if there are no such links for s_i^z, we pick the non-adjacent links $(s_{k'}^{z-2}, s_i^z)$ with weights above t_{nadj}. Finally, we force one candidate adjacent link with the largest weight to be preserved for every region without any picked links, on the grounds that the

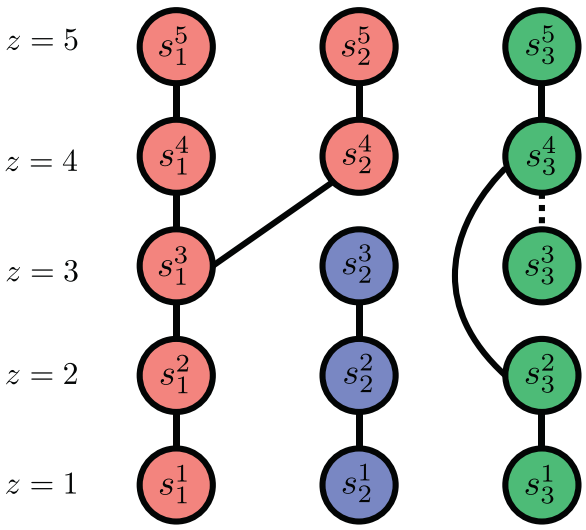

Fig. 6. Example of linking 2D segmentation regions in 3D. Picked links with weight above thresholds are drawn in solid lines, and the dashed line between s_3^3 and s_3^4 indicates their connection is forced to avoid s_3^3 being isolated. Colors indicate there are three reconstructed 3D bodies. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

z-resolutions of our EM data sets are all sufficiently high to have a neuron appear in more than one section and therefore there should not be any isolated 2D region within an image stack. An artificial example is shown in Fig. 6: there are five sections with three 2D segmentation regions on each section; links with weights larger than the thresholds are drawn in solid lines. Since s_3^2 has no adjacent links with weights above t_{adj} in the forward direction of z, the non-adjacent link between s_3^2 and s_3^4 whose weight is larger than t_{nadj} is picked. Region s_3^3 is forced to be connected to s_3^4, because it has no picked links and its strongest adjacent link is (s_3^3, s_3^4).

With the preserved links, we can partition $\mathcal{S}$ into subsets $\{\mathcal{S}_1, \mathcal{S}_2, \ldots\}$ such that any pair of 2D regions in each subset are connected directly or indirectly by the preserved links. Each subset contains all the 2D regions of one reconstructed 3D body. Finally, we relabel regions in $\mathcal{S}$ by assigning a unique label to all the regions in each partitioning subset $\mathcal{S}_i$, and thus generate the final label image stack as a representation of all the reconstructed neurons. As an example shown in Fig. 6, $\{s_1^1, s_1^2, s_1^3, s_1^4, s_2^4, s_1^5, s_2^5\}$, $\{s_2^1, s_2^2, s_2^3\}$ and $\{s_3^1, s_3^2, s_3^3, s_3^4, s_3^5\}$ constitute three reconstructed 3D bodies, respectively, according to the preserved links.

3. Results

In this section, we validate our proposed methods using three data sets. For intra-section segmentation, we compare our automatic method with other state-of-the-art methods and also show the improvement using our semi-automatic method. For inter-section reconstruction, we compare our automatic method with other state-of-the-art methods. We use the adapted Rand error as the evaluation metric, which is described in detail in Section 2.1. The implementation of our methods uses ITK (Yoo et al., 2002) and OpenCV[2] library. Our random forest classifier implementation is a derivation from (Jaiantilal, 2009). All our experiments were performed on a computational server with 80 2.4GHz Intel CPUs (160 cores with hyper-threading) and 750 GB of memory, and a maximum of 100 cores were used simultaneously.

3.1. Data sets

We used three data sets in the experiments.

3.1.1. Drosophila VNC data set

The Drosophila first instar larva ventral nerve cord (VNC) data set (Cardona et al., 2010, 2012) contains 60 512×512 images acquired using ssTEM at a resolution of $4 \times 4 \times 50$nm/pixel. This data set was used in ISBI 2012 EM Challenge (Arganda-Carreras et al., 2012) with the ground truth 2D segmentation of 30 consecutive images as the training set and the other 30 consecutive images as the testing set. The 3D ground truth is also publicly available for the training set. For the 2D segmentation experiment, we trained our automatic algorithm with the 30 training images, tested on the 30 testing images and submitted our results to the ISBI 2012 EM Challenge server for evaluation. For the 3D reconstruction experiment, we divided the training set into three stacks with 10 consecutive images in each stack, and performed a three-fold cross validation: we used two stacks to train our intra- and inter-section automatic algorithms, and used the other one stack for testing.

3.1.2. Mouse neuropil data set

The whole mouse neuropil data set (Deerinck et al., 2010) is a stack of 400 images of size 4096×4096 acquired using SBFSEM. The resolution is $10 \times 10 \times 50$nm/pixel. A subset of 70 700×700 images was cropped and the 2D segmentation was annotated by an expert for performance evaluation. A subset of 14 images was randomly selected to train our automatic algorithms and the remaining 56 images were used for testing. We also had two non-expert subjects use our semi-automatic program for 2D segmentation. We did not evaluate our inter-section reconstruction algorithm using the data set due to the absence of the 3D ground truth.

3.1.3. Mouse cortex data set

Also known as the AC4 data set, the whole mouse cortex data set is a stack of 1850 images of 4096×4096 pixels acquired by ssSEM at a resolution of $3 \times 3 \times 30$nm/pixel. The images were downsampled by a factor of 2 in x–y and two subsets of $1024 \times 1024 \times 100$ pixels were cropped and used in ISBI 2013 EM Challenge (Arganda-Carreras et al., 2013) as the training and testing set, respectively. The 2D and 3D ground truths of 100 images were provided for the training set. We used the membrane detection probability maps generated with the DNN (Ciresan et al., 2012) and trained our automatic intra-section segmentation and inter-section reconstruction algorithm using the training set, tested on the testing set, and submitted our results to the ISBI 2013 EM Challenge server for 3D adapted Rand error evaluation.

3.2. Intra-section segmentation experiments

In this section, we first tested our automatic 2D segmentation method with the Drosophila VNC data set and the mouse neuropil data set. We also show on the mouse neuropil data set the results of the semi-automatic procedure used by the two non-experts to compare with the automatic results.

3.2.1. Drosophila VNC data set

For the Drosophila VNC data set, we used both the membrane detection probability maps generated using the CHM[3] (Seyedhosseini et al., 2013) and the DNN (Ciresan et al., 2012)

[2] http://opencv.org.

[3] A CHM implementation is at http://www.sci.utah.edu/ mseyed/Mojtaba_Seyedhosseini/CHM.html.

Table 1
Precisions and recalls of initial segmentations of the Drosophila VNC data set training images using different probability maps. The watershed threshold t_w is 0.03 for the CHM and 0.01 for the DNN probability maps.

Probability map	Precision	Recall
CHM	0.9967	0.5054
DNN	0.9971	0.9704

with 30 training slices. The watershed local minimum dynamic threshold t_w is used as 0.03 and 0.01 for the CHM and the DNN probability maps, respectively. The precisions and recalls of the initial segmentations of training images are shown in Table 1. The high precisions and relatively low recalls indicate we ensured initial over-segmentation. For both data sets and probability maps, we applied the pre-merging with parameters $t_{a_1} = 50$, $t_{a_2} = 200$ and $t_p = 0.5$. For the random forest used as our boundary classifier, we used $\theta_t = 255$ trees with $\theta_s = 70$ percent of all training samples randomly selected to train each tree, and at each decision tree node the square root of the number of features ($\theta_f = \lfloor\sqrt{88}\rfloor = 9$) were examined for the most informative one to build a branching. It took approximately 2 minutes to finish the training and testing.

The testing results are shown in Table 2 along with the results of other state-of-the-art methods from various groups for comparison. All the results are also available on the ISBI 2012 EM Challenge online leader board (Arganda-Carreras et al., 2012), which currently still accepts submissions for evaluation after the challenge. Note that the challenge evaluation system thresholds the resulting images at 11 thresholds uniformly distributed from 0 to 1 and picks the best results. Therefore, the resulting images as probability maps were thresholded at the best thresholds, while other resulting images as hard segmentations, such as ours, yield identical results with different thresholds. The "human" entries were generated by two human observers.

According to Table 2, our approach outperforms other state-of-the-art methods by a large margin. We can see that applying our approach improves the adapted Rand error by over 4.8 percent for the CHM membrane detection probability maps (comparing Entry 3 and 8). Our approach also improves the DNN membrane detection results by over 2.4 percent (comparing Entry 2 and 5). Using either probability maps, our approach yields even smaller errors than a human observer (comparing Entry 2, 3, and 4). We claim, based on the results, that our approach can improve 2D segmentation accuracy from thresholding membrane probability maps with best thresholds independent of the membrane detection algorithms. Fig. 7 shows four example testing results of both pixel-wise membrane detections and our 2D segmentation approach. Our approach closes boundary gaps in the membrane detection, which may lead to under-segmentation errors. Our approach also removes some undesirable intra-cellular structures. Since our automatic result is already at the human level, we did not apply our semi-automatic approach to this data set.

Table 2
Automatic intra-section segmentation testing results (adapted Rand error) on the Drosophila VNC data set. "Human" represents the manually labeled results by a human observer. The results are also available on the ISBI 2012 EM Challenge online leader board (Arganda-Carreras et al., 2012). Our group is under "SCI".

No.	Group	Adapted Rand error
1	Human 1	0.002109
2	Our approach (with DNN probability maps)	**0.02581**
3	Our approach (with CHM probability maps)	**0.02805**
4	Human 2	0.02995
5	IDSIA (DNN) (Ciresan et al., 2012)	0.05040
6	MLL-ETH (Laptev et al., 2012)	0.06392
7	MIT (Jain et al., 2007)	0.07361
8	SCI (CHM) (Seyedhosseini et al., 2013)	0.07662
9	CellProfiler (Kamentsky et al., 2011)	0.08697
10	Harvard Coxlab	0.08996
11	UCL CoMPLEX	0.09028
12	Harvard GVI (Kaynig et al., 2010)	0.09129
13	UCL Vision and Imaging Science	0.09807

Table 3
Intra-section segmentation results (adapted Rand error) on the mouse neuropil data set.

No.	Approach	Training error	Testing error
1	Thresholding probability maps	0.07298	0.2023
2	Automatic merge tree	0.04128	0.1288
3	Semi-automatic merge tree (User 1)	0.05106	0.07546
4	Semi-automatic merge tree (User 2)	0.06594	0.04556

3.2.2. Mouse neuropil data set

For the mouse neuropil data set, since the membrane detection results using the DNN (Ciresan et al., 2012) are not available, we experimented only with the probability maps generated by the CHM (Seyedhosseini et al., 2013). The membrane detection classifier was trained with 14 images that were randomly selected. We trained our algorithm with the same images and tested on the rest of the stack. We used $t_w = 0.02$ to threshold the watershed local minima, and the same parameters for the pre-merging ($t_{a_1} = 50$, $t_{a_2} = 200$ and $t_p = 0.5$) and the random forest ($\theta_t = 255$, $\theta_s = 0.7$ and $\theta_f = 9$) as the Drosophila VNC experiments. It took about 14 min to finish the experiment. Two non-expert users used our semi-automatic software to manually resolve the automatically generated merge trees and generated 2D segmentations. The adapted Rand errors of each approach on the training and testing set are shown in Table 3, in which we also compare our results with thresholding the membrane detection probability maps at the best threshold (Entry 1).

In general, the mouse neuropil data set is a more difficult data set, because of its larger variations of cell shapes and more complex intra-cellular structures. Our automatic approach again improves over the membrane detection probabilities significantly by over 7.3 percent. Moreover, the error can be largely reduced (5.3 percent and 8.3 percent) by the users with our semi-automatic approach. Based on these results, we argue that, with the help of our semi-automatic approach, even non-expert users with little training can segment the images very well by merely answering a reasonable number of automatically generated questions. We expect that the result can be even further improved by experts using our semi-automatic approach while saving a large amount of time. Fig. 8 shows four example testing images of our automatic approach compared with our semi-automatic results by User 2. Our semi-automatic approach can fix segmentation errors, such as under-segmentation due to massive boundary missing in pixel-wise membrane detection, with the intervention of even non-expert users without much training. These users required a mean time of 2.2 seconds per cell per section and a median time of 1.6 seconds per cell per section. This results in an approximate total resolution time of 5 hours for this $7 \times 7 \times 3.5$ μm block.

3.3. Inter-section reconstruction experiments

In this section, we validate our automatic 3D reconstruction method with the Drosophila VNC data set and the mouse cortex data set.

3.3.1. Drosophila VNC data set

Due to the absence of the ground truth data for the testing set used in the ISBI 2012 EM Challenge, we performed a three-fold cross validation experiment by dividing the Drosophila VNC training set into three substacks of 10 consecutive sections. Two substacks were

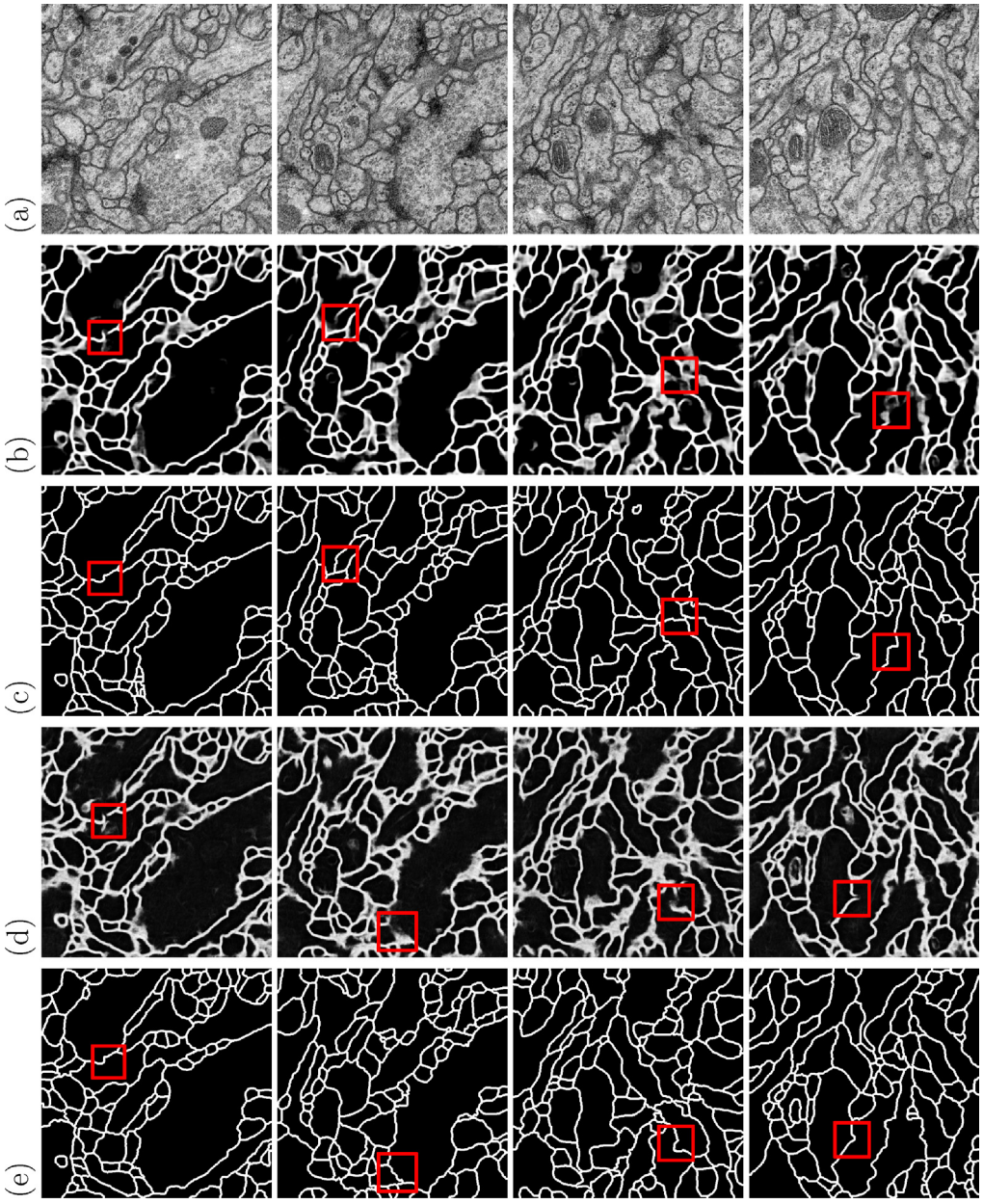

Fig. 7. Automatic intra-section segmentation testing results (four sections) on the Drosophila VNC data set. Rows: (a) original EM images, (b) DNN (Ciresan et al., 2012) membrane detection, (c) segmentation by our automatic approach using the DNN results, (d) CHM (Seyedhosseini et al., 2013) membrane detection, and (e) segmentation by our automatic approach using the CHM results. The resulting cell boundaries are dilated for visualization purposes. The red squares denote the missing boundaries fixed by our approach.

used for training and the other one substack was tested each time. The membrane detection probability maps were generated using the CHM (Seyedhosseini et al., 2013) trained with the two training substacks. For the 2D automatic segmentation, we used $t_w = 0.02$ and the same other parameters ($t_{a_1} = 50$, $t_{a_2} = 200$ and $t_p = 0.5$) as in Section 3.2.1. For the 3D reconstruction, we used $t_{cd} = 50$ for the section classifier for adjacent links and $t_{cd} = 100$ for the section classifier for non-adjacent links. The section classifier random forest has the same specifications ($\theta_t = 255$, $\theta_s = 0.7$) as the boundary classifier except that the number of features examined at each tree node is $\theta_f = \lfloor\sqrt{230}\rfloor = 15$. We used $t_{adj} = 0.5$ to threshold the adjacent links, and a higher threshold $t_{nadj} = 0.95$ to threshold non-adjacent links in order to keep only the most reliable ones. The 2D segmentation and 3D reconstruction each took approximately 3 minutes. Table 4 shows both the 2D and 3D results for each fold and the average.

Table 4
Automatic intra-section segmentation and inter-section reconstruction results (adapted Rand error) of cross validation experiments on the Drosophila VNC data set.

Fold	Intra-section		Inter-section	
	Training	Testing	Training	Testing
1	0.009633	0.04871	0.04946	0.1306
2	0.01161	0.06328	0.04119	0.1507
3	0.01174	0.03726	0.03855	0.09794
Avg.	0.01099	0.04975	0.04307	0.1264

3.3.2. *Mouse cortex data set*

The mouse cortex data set is the standard data set used in the ISBI 2013 EM Challenge for 3D segmentation method

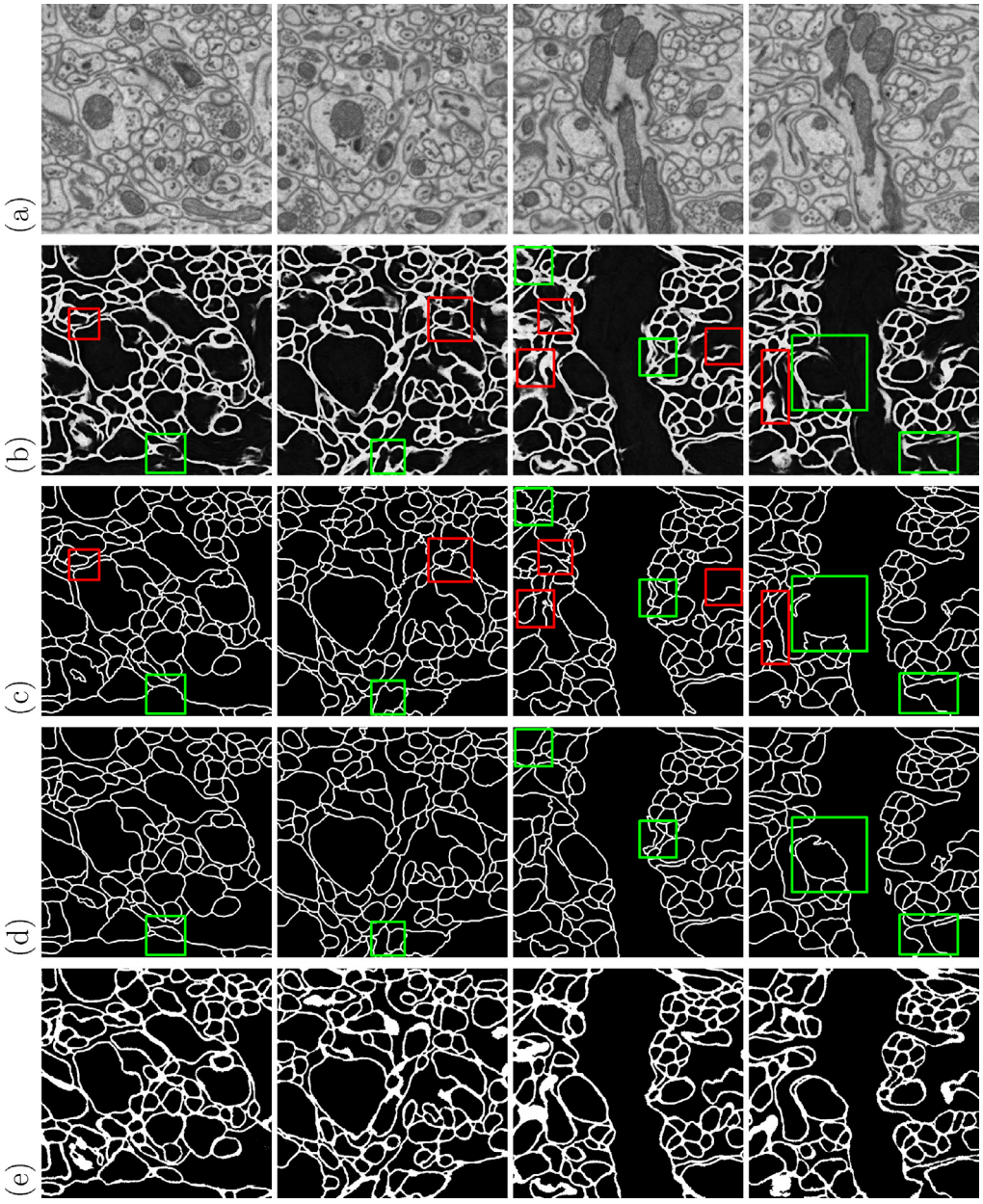

Fig. 8. Automatic and semi-automatic intra-section segmentation testing results (four sections) on the mouse neuropil data set. Rows: (a) original EM images, (b) CHM (Seyedhosseini et al., 2013) membrane detection, (c) automatic segmentation results, (d) semi-automatic segmentation results by a non-expert user, and (e) ground truth images. The resulting cell boundaries are dilated for visualization purposes. The red rectangles denote the missing boundaries fixed by the automatic approach. The green rectangles denote the segmentation errors further corrected by the semi-automatic approach.

evaluation. The membrane detection probability maps from the DNN (Ciresan et al., 2012) were used as the input. We used $t_w = 0.01$ for the watershed initial segmentation generation, and $t_{a_1} = 50$, $t_{a_2} = 1000$ and $t_p = 0.5$ for the pre-merging. Also, we used the same parameters ($t_{cd} = 50$ for adjacent links and $t_{cd} = 100$ for non-adjacent links) for candidate link generation and the same random forest specifications ($\theta_t = 255$, $\theta_s = 0.7$ and $\theta_f = 15$) as the Drosophila VNC experiment. The adjacent links were thresholded at $t_{adj} = 0.85$, and the non-adjacent links were thresholded at $t_{nadj} = 0.95$. The 2D segmentation took about 75 minutes, and the 3D reconstruction took about 126 min. In Table 5, we show our 3D adapted Rand error in comparison with other groups from the ISBI 2013 EM Challenge (Arganda-Carreras et al., 2013). Selected testing results of 3D neuron reconstruction are shown in Fig. 9.

Table 5
Automatic inter-section reconstruction segmentation testing result (adapted Rand error) on the mouse cortex data set.

No.	Group	Adapted Rand error
1	Human	0.05998
2	Janelia Farm FlyEM (Nunez-Iglesias et al., 2013)	0.1250
3	Our approach	**0.1315**
4	Singapore ASTAR	0.1665
5	Harvard Rhoana (Kaynig et al., 2013)	0.1726
6	Brno University of Technology SPLab	0.4665

4. Discussion and conclusions

We developed a fully automatic two-step approach for neuron spatial structure segmentation and reconstruction using EM

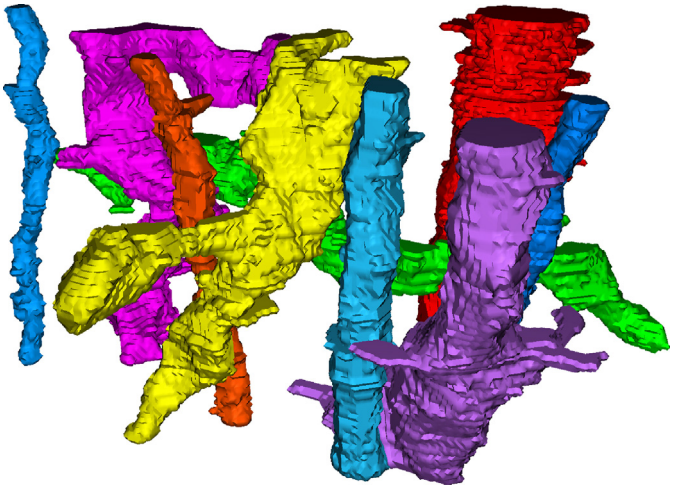

Fig. 9. Examples of reconstructed 3D neurons of the mouse cortex data set testing image stack. Different neurons are distinguished by color. The visualization was generated using TrakEM2 (Cardona et al., 2012) in Fiji (Fiji Is Just ImageJ) (Schindelin et al., 2012).

images. We proposed a hierarchical 2D segmentation method with the merge tree structure and supervised classification that uses membrane probability maps as input. Next, we used a supervised linking method to acquire inter-section reconstruction of neurons. We also designed a semi-automatic 2D segmentation method that takes advantage of the automatic intermediate results and improves 2D segmentation with minimized user interaction.

According to the experimental results, our automatic merge tree approach improves the 2D neuron segmentation accuracy substantially over thresholding the membrane probability maps at the best thresholds. By using superpixels instead of pixels as the unit element, we are able to compute non-local region-based features, which give richer information about a segmentation. Also, the use of the merge tree structure presents the most plausible segmentation hypotheses in a more efficient way than using a general graph structure, and it transforms the problem of acquiring final segmentation from considering all possible region combinations to choosing a set of best answers from the most likely choices given. The way that the node potentials are evaluated incorporates both lower and higher merging level information, and thus the impact of single boundary classification error can be alleviated. Meanwhile, the nature of our method is suitable for parallelization without any modification, and with more careful implementation, the memory and time usage of applying our method can be even further reduced. As we can see so far, one major concern about using the automated algorithm based on the merge tree structure is its inability to fix incorrect region merging orders. According to the experimental results, however, we argue that boundary median probability is a robust merging saliency metric, which helps generate correct merging order for most cases. Also, with further improvement of membrane detection algorithms, we will have more consistent membrane probability maps as input, and the occurrence of incorrect merging orders that actually leads to incorrect segmentation will be further suppressed.

The semi-automatic 2D segmentation approach we proposed makes full use of the intermediate results from our automatic method and thus minimizes the interaction needed from users. By giving answers to a limited number of questions with fixed multiple choices, a user without expertise can achieve full 2D segmentation at an average speed of about 2 seconds per cell. Also, by allowing a user to override the fixed merge tree structure, we can fix the segmentation error due to occasional incorrect region merging order.

Considering only region pairs with overlap or within a certain centroid distance, the complexity of our automatic 3D reconstruction approach is linear to the number of 2D segmented regions. The non-adjacent linking option helps avoid the breakup of a 3D tracing due to occasional bad 2D segmentation. In the experiments, we often used high thresholds for link weights to avoid incorrect linkages, and we observed that our method tends to generate over-segmentation results with considerably high precision but relatively low recall. This indicates a major amount of reconstruction errors we encounter can be fixed by merging the reconstructed 3D body pieces, which can be achieved by user interaction or another automatic procedure, for which potentially more powerful volume features can be extracted and utilized. This could be a future direction for improving the overall 3D neuron reconstruction accuracy.

Acknowledgment

This work was supported by NIH 1R01NS075314-01 (TT, MHE) and NSF IIS-1149299 (TT). We thank the Albert Cardona Lab at the Howard Hughes Medical Institute Janelia Farm Research Campus for providing the Drosophila VNC data set, the National Center for Microscopy and Imaging Research at the University of California, San Diego for providing the mouse neuropil data set, the Jeff Lichtman Lab at Harvard University for providing the mouse cortex data set, and Alessandro Guisti and Dan Ciresan at the Dalle Molle Institute for Artificial Intelligence for providing the deep neural networks membrane detection probability maps. We also thank the editor and reviewers whose comments greatly helped improve the paper.

Appendix A. Summary of parameters

The parameters used in our methods are summarized in Table A.6.

Table A.6
Summary of parameters.

Initial 2D segmentation (Section 2.2.2)
t_w: Watershed initial local minimum dynamic threshold.
t_{a_1}: Min. region area threshold for pre-merging.
t_p: Min. average probability threshold for pre-merging.
t_{a_2}: Max. region area threshold for pre-merging (used with t_p).
3D linking (Section 2.3)
t_{cd}: Max. centroid distance threshold for candidate links.
t_{adj}: Adjacent link weight threshold.
t_{nadj}: Non-adjacent link weight threshold.
Random forest classifier (Section 2.2.4, 2.3.1)
θ_t: Number of trees.
θ_s: Portion of all samples used in training each decision tree.
θ_f: Number of features examined at each node.

Appendix B. Summary of classifier features

The categories of features used for boundary classification (Section 2.2.4) are summarized in Table B.7. For the texton features, 7×7 patches are used and k-means clustering[4] is used for learning the texture dictionary of 100 bins (words).

The section classifier (Section 2.3.1) feature categories are summarized in Table B.8.

[4] We used a parallel k-means implementation from http://users.eecs.northwestern.edu/ wkliao/Kmeans/.

Table B.7
Summary of boundary classifier (Section 2.2.4) feature categories. Features are generated between a pair of merging regions within a section. In the table, "statistics" refers to minimum, maximum, mean, median, and standard deviation.

Geometry
Region areas
Region perimeters
Region compactness
Boundary length
Boundary curvatures
Intensity (of original images and probability maps)
Boundary intensity histogram (10 bins)
Boundary intensity statistics
Region intensity histogram (10 bins)
Region intensity statistics
Region texton histogram (100 bins)
Merging saliencies

Table B.8
Summary of section classifier (Section 2.3.1) feature categories. Features are generated between a pair of regions in different sections.

Geometry
Region areas
Region perimeters
Region overlap
Region centroid distance
Region compactness
Hu moment shape descriptors (Hu, 1962)
Shape convexity defects
Bounding boxes
Fitted ellipses and enclosing circles
Contour curvatures

References

Achanta R, Shaji A, Smith K, Lucchi A, Fua P, Susstrunk S. SLIC superpixels compared to state-of-the-art superpixel methods. IEEE Transactions on Pattern Analysis and Machine Intelligence 2012;34:2274–82.

Anderson JR, Jones BW, Watt CB, Shaw MV, Yang JH, DeMill D, et al. Exploring the retinal connectome. Molecular Vision 2011;17:355.

Anderson JR, Jones BW, Yang JH, Shaw MV, Watt CB, Koshevoy P, et al. A computational framework for ultrastructural mapping of neural circuitry. PLoS Biology 2009;7:e1000074.

Andres B, Koethe U, Kroeger T, Helmstaedter M, Briggman KL, Denk W, et al. 3D segmentation of SBFSEM images of neuropil by a graphical model over supervoxel boundaries. Medical Image Analysis 2012;16:796–805.

Arganda-Carreras I, Seung HS, Cardona A, Schindelin J. Segmentation of neuronal structures in EM stacks challenge – ISBI 2012. http://brainiac2.mit.edu/isbi_challenge/; 2012 [accessed 01.11.13].

Arganda-Carreras I, Seung HS, Vishwanathan A, Berger D. 3D segmentation of neurites in EM images challenge – ISBI 2013. http://brainiac2.mit.edu/SNEMI3D/; 2013 [accessed 01.11.13].

Beare R, Lehmann G. The watershed transform in ITK – discussion and new developments. Insight Journal 2006, http://hdl.handle.net/1926/202.

Beucher S, Lantuejoul C. Use of watersheds in contour detection. In: International Workshop on Image Processing: Real-Time Edge and Motion Detection/Estimation; 1979.

Bock DD, Lee WCA, Kerlin AM, Andermann ML, Hood G, Wetzel AW, et al. Network anatomy and in vivo physiology of visual cortical neurons. Nature 2011;471:177–82.

Breiman L. Random forests. Machine Learning 2001;45:5–32.

Briggman KL, Denk W. Towards neural circuit reconstruction with volume electron microscopy techniques. Current Opinion in Neurobiology 2006;16:562–70.

Briggman KL, Helmstaedter M, Denk W. Wiring specificity in the direction-selectivity circuit of the retina. Nature 2011;471:183–8.

Cardona A, Saalfeld S, Preibisch S, Schmid B, Cheng A, Pulokas J, et al. An integrated micro-and macroarchitectural analysis of the Drosophila brain by computer-assisted serial section electron microscopy. PLoS Biology 2010;8:e1000502.

Cardona A, Saalfeld S, Schindelin J, Arganda-Carreras I, Preibisch S, Longair M, et al. TrakEM2 software for neural circuit reconstruction. PLoS One 2012;7:e38011.

Chklovskii DB, Vitaladevuni S, Scheffer LK. Semi-automated reconstruction of neural circuits using electron microscopy. Current Opinion in Neurobiology 2010;20:667–75.

Ciresan D, Giusti A, Gambardella LM, Schmidhuber J. Deep neural networks segment neuronal membranes in electron microscopy images. Advances in Neural Information Processing Systems 2012;25:2852–60.

Deerinck TJ, Bushong EA, Lev-Ram V, Shu X, Tsien RY, Ellisman MH. Enhancing serial block-face scanning electron microscopy to enable high resolution 3-D nanohistology of cells and tissues. Microscopy and Microanalysis 2010;16:1138–9.

Denk W, Horstmann H. Serial block-face scanning electron microscopy to reconstruct three-dimensional tissue nanostructure. PLoS Biology 2004;2:e329.

Felzenszwalb PF, Huttenlocher DP. Efficient graph-based image segmentation. International Journal of Computer Vision 2004;59:167–81.

Funke J, Andres B, Hamprecht FA, Cardona A, Cook M. Efficient automatic 3D-reconstruction of branching neurons from EM data. In: IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE; 2012. p. 1004–11.

Horstmann H, Körber C, Sätzler K, Aydin D, Kuner T. Serial section scanning electron microscopy (S3EM) on silicon wafers for ultra-structural volume imaging of cells and tissues. PLoS One 2012;7:e35172.

Hu MK. Visual pattern recognition by moment invariants. IRE Transactions on Information Theory 1962;8:179–87.

Jaiantilal A. Classification and regression by randomforest-matlab; 2009, Available at: http://code.google.com/p/randomforest-matlab

Jain V, Murray JF, Roth F, Turaga S, Zhigulin V, Briggman KL, et al. Supervised learning of image restoration with convolutional networks. In: IEEE international conference on computer vision (ICCV) 2007:1–8.

Jain V, Turaga SC, Briggman K, Helmstaedter MN, Denk W, Seung HS. Learning to agglomerate superpixel hierarchies. Advances in Neural Information Processing Systems 2011:648–56.

Jeong WK, Beyer J, Hadwiger M, Vazquez A, Pfister H, Whitaker RT. Scalable and interactive segmentation and visualization of neural processes in em datasets. IEEE Transactions on Visualization and Computer Graphics 2009;15:1505–14.

Jones BW, Marc RE. Retinal remodeling during retinal degeneration. Experimental Eye Research 2005;81:123–37.

Jones BW, Watt CB, Frederick JM, Baehr W, Chen CK, Levine EM, et al. Retinal remodeling triggered by photoreceptor degenerations. Journal of Comparative Neurology 2003;464:1–16.

Jones BW, Watt CB, Marc RE. Retinal remodelling. Clinical and Experimental Optometry 2005;88:282–91.

Jurrus E, Hardy M, Tasdizen T, Fletcher PT, Koshevoy P, Chien CB, et al. Axon tracking in serial block-face scanning electron microscopy. Medical Image Analysis 2009;13:180–8.

Jurrus E, Paiva AR, Watanabe S, Anderson JR, Jones BW, Whitaker RT, et al. Detection of neuron membranes in electron microscopy images using a serial neural network architecture. Medical Image Analysis 2010;14:770–83.

Jurrus E, Whitaker R, Jones BW, Marc RE, Tasdizen T. An optimal-path approach for neural circuit reconstruction. In: 5th IEEE International Symposium on Biomedical Imaging: from Nano to Macro ISBI. IEEE; 2008. p. 1609–12.

Kamentsky L, Jones TR, Fraser A, Bray MA, Logan DJ, Madden KL, et al. Improved structure, function and compatibility for cellprofiler: modular high-throughput image analysis software. Bioinformatics 2011;27:1179–80.

Kaynig V, Fuchs T, Buhmann JM. Neuron geometry extraction by perceptual grouping in stem images. In: IEEE conference on computer vision and pattern recognition (CVPR). IEEE; 2010. p. 2902–9.

Kaynig V, Fuchs TJ, Buhmann JM. Geometrical consistent 3D tracing of neuronal processes in ssTEM data. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI. Springer; 2010. p. 209–16.

Kaynig V, Vazquez-Reina A, Knowles-Barley S, Roberts M, Jones TR, et al. Large-scale automatic reconstruction of neuronal processes from electron microscopy images, 2013. Available from: arXiv:1303.7186.

Knott G, Marchman H, Wall D, Lich B. Serial section scanning electron microscopy of adult brain tissue using focused ion beam milling. Journal of Neuroscience 2008;28:2959–64.

Kumar R, Vázquez-Reina A, Pfister H. Radon-like features and their application to connectomics. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW). IEEE; 2010. p. 186–93.

Laptev D, Vezhnevets A, Dwivedi S, Buhmann JM. Anisotropic ssTEM image segmentation using dense correspondence across sections. In: Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012. Springer; 2012. p. 323–30.

Levinshtein A, Stere A, Kutulakos KN, Fleet DJ, Dickinson SJ, Siddiqi K. Turbopixels: fast superpixels using geometric flows. IEEE Transactions on Pattern Analysis and Machine Intelligence 2009;31:2290–7.

Liu C, Yuen J, Torralba A. SIFT flow: dense correspondence across scenes and its applications. IEEE Transactions on Pattern Analysis and Machine Intelligence 2011;33:978–94.

Liu T, Jurrus E, Seyedhosseini M, Ellisman M, Tasdizen T. Watershed merge tree classification for electron microscopy image segmentation. In: 21st international conference on pattern recognition (ICPR). IEEE; 2012. p. 133–7.

Liu T, Seyedhosseini M, Ellisman M, Tasdizen T. Watershed merge forest classification for electron microscopy image stack segmentation. In: Proceedings of the 2013 IEEE international conference on image processing; 2013.

Lucchi A, Smith K, Achanta R, Lepetit V, Fua P. A fully automated approach to segmentation of irregularly shaped cellular structures in EM images. In: Medical image computing and computer-assisted intervention – MICCAI 2010. Springer; 2010. p. 463–71.

Macke JH, Maack N, Gupta R, Denk W, Schölkopf B, Borst A. Contour-propagation algorithms for semi-automated reconstruction of neural processes. Journal of Neuroscience Methods 2008;167:349–57.

Marc RE, Jones B, Watt C, Vazquez-Chona F, Vaughan D, Organisciak D. Extreme retinal remodeling triggered by light damage: implications for age related macular degeneration. Molecular Vision 2008;14:782.

Marc RE, Jones BW, Anderson JR, Kinard K, Marshak DW, Wilson JH, et al. Neural reprogramming in retinal degeneration. Investigative Ophthalmology & Visual Science 2007;48:3364–71.
Marc RE, Jones BW, Watt CB, Strettoi E. Neural remodeling in retinal degeneration. Progress in Retinal and Eye Research 2003;22:607–55.
Mishchenko Y. Automation of 3d reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs. Journal of Neuroscience Methods 2009;176:276–89.
Nunez-Iglesias J, Kennedy R, Parag T, Shi J, Chklovskii DB. Machine learning of hierarchical clustering to segment 2D and 3D images. PLoS ONE 2013;8:e71715.
Peng YW, Hao Y, Petters RM, Wong F. Ectopic synaptogenesis in the mammalian retina caused by rod photoreceptor-specific mutations. Nature Neuroscience 2000;3:1121–7.
Rand WM. Objective criteria for the evaluation of clustering methods. Journal of the American Statistical Association 1971;66:846–50.
Schalek R, Wilson A, Lichtman J, Josh M, Kasthuri N, Berger D, et al. ATUM-based SEM for high-speed large-volume biological reconstructions. Microscopy and Microanalysis 2012;18:572–3.
Schindelin J, Arganda-Carreras I, Frise E, Kaynig V, Longair M, Pietzsch T, et al. Fiji: an open-source platform for biological-image analysis. Nature Methods 2012;9:676–82.
Schroeder W, Martin K, Lorensen B. The visualization toolkit: an object-oriented approach to 3D graphics. 4th ed. Kitware, Inc. (formerly Prentice-Hall); 2006.
Seung HS. Neuroscience: towards functional connectomics. Nature 2011;471:170–2.
Seung HS. Eyewire. http://www.eyewire.org; 2013 [accessed 24.10.13].
Seyedhosseini M, Kumar R, Jurrus E, Giuly R, Ellisman M, Pfister H, et al. Detection of neuron membranes in electron microscopy images using multi-scale context and radon-like features. In: Medical image computing and computer-assisted intervention – MICCAI 2011. Springer; 2011. p. 670–7.
Seyedhosseini M, Sajjadi M, Tasdizen T. Image segmentation with cascaded hierarchical models and logistic disjunctive normal networks. In: Proceedings of IEEE international conference on computer vision. IEEE; 2013.
Seyedhosseini M, Tasdizen T. Multi-class multi-scale series contextual model for image segmentation. IEEE Transactions on Image Processing 2013;22:4486–96.
Shi J, Malik J. Normalized cuts and image segmentation. IEEE Transactions on Pattern Analysis and Machine Intelligence 2000;22:888–905.
Sporns O, Tononi G, Kötter R. The human connectome: a structural description of the human brain. PLoS Computational Biology 2005;1:e42.
Tasdizen T, Koshevoy P, Grimm BC, Anderson JR, Jones BW, Watt CB, et al. Automatic mosaicking and volume assembly for high-throughput serial-section transmission electron microscopy. Journal of Neuroscience Methods 2010;193:132–44.
Tasdizen T, Whitaker R, Marc RE, Jones BW. Enhancement of cell boundaries in transmission electron microscopy images. ICIP 2005(II):129.
Vazquez L, Sapiro G, Randall G. Segmenting neurons in electronic microscopy via geometric tracing. In: International Conference on Image Processing, ICIP 98. IEEE; 1998. p. 814–8.
Vazquez-Reina A, Gelbart M, Huang D, Lichtman J, Miller E, Pfister H. Segmentation fusion for connectomics. In: IEEE international conference on computer vision (ICCV). IEEE; 2011. p. 177–84.
Veksler O, Boykov Y, Mehrani P. Superpixels and supervoxels in an energy optimization framework. In: Computer Vision-ECCV 2010. Springer; 2010. p. 211–24.
Vitaladevuni SN, Basri R. Co-clustering of image segments using convex optimization applied to em neuronal reconstruction. In: IEEE conference on computer vision and pattern recognition (CVPR). IEEE; 2010. p. 2203–10.
Vu N, Manjunath B. Graph cut segmentation of neuronal structures from transmission electron micrographs. In: 15th IEEE international conference on image processing, ICIP 2008. IEEE; 2008. p. 725–8.
Yang HF, Choe Y. Cell tracking and segmentation in electron microscopy images using graph cuts. In: IEEE international symposium on biomedical imaging: from nano to macro, ISBI'09. IEEE; 2009. p. 306–9.
Yoo TS, Ackerman MJ, Lorensen WE, Schroeder W, Chalana V, Aylward S, et al. Engineering and algorithm design for an image processing api: a technical report on ITK – the insight toolkit. Studies in Health Technology and Informatics 2002:586–92.