Automatically Segmenting and Reconstructing Neurons in SEM images

Qiang Rao[1], Hua Han[1,2], Weifu Li[1,3], Lijun Shen[1,2], Xi Chen[1,2], Qiwei Xie[1,2]
Institute of Automation, Chinese Academy of Sciences Beijing, 100190, China[1]
CAS Center for Excellence in Brain Science and Intelligence Technology[2]
Applied Mathematics and the Faculty of Mathematics and Statistics, Hubei University, Wuhan, 430062, China[3]
hua.han@ia.ac.cn

***Abstract*—Neuronal networks reconstruction is a big challenge in the neuroscience. Recent developments in volume electron microscopy (EM) imaging have enabled us to obtain large amounts of brain tissues image data. Analysis of the tremendously huge neuronal EM images based on automated method would be of vital importance. In this paper, we propose a method that Deep Convolutional Neural Network (DCNN) is used for neuronal boundary detection; and then, with the membrane detection probability map (MDPM) generated by DCNN, a marker-controlled watershed method is applied to segment neurons in the EM images over the MDPM. Semi-automated and fully automated 3D reconstruction methods are employed to connect the sections of the corresponding segmentations belonging to each neuron. Finally, we have reconstructed dense neurons in** $8000\times8000\times1796$ **consecutive EM images stack of drosophila mushroom body with resolution of** $4nm\times4nm\times50nm$ **with automated method and several sparse neurons with semi-automated method.**

***Index Terms*—Neuronal networks reconstruction, DCNN, watershed, neuronal boundary detection, drosophila mushroom body.**

I. Introduction

The relation between the structure and brain function is still poorly understood. Neuroscientists are willing to look into the fine structure of neuronal networks. Electron Microscopy (EM) has become one of the main tools for the acquisition of neuronal image data, which at synaptic resolution of several nanometers. Roughly, $1mm^3$ volume of brain tissue generates about 2000TB EM image data and the whole mouse brain generates about 60PB. Analysing the magnanimity neuronal EM images and reconstruction of the neuronal networks, i.e., connectome(connectomic), have become a big challenge [1, 2].

Much work has been done to the reconstruction of neuronal networks [5–7], where [5] invented the automated tape collecting ultramicrotome Scanning Electron Microscopy (ATUM-SEM) which is currently one of best methods suited for dense reconstruction of substantial volumes of neuronal tissue at synaptic level. Jain [8] emphasized that machine learning would be crucial technology for connectomics. Deep Neural Network (DNN) [9], especially, Deep Convolutional Neural Network (DCNN) [10] performs well in settings with little prior algorithmic knowledge about the classification task. The automatic analysis of large scale of data with artificial intelligence (AI) techniques has become such a heat point that machine learning has been used in the EM image analysis, such as mitochondria segmentation [11] and synapse segmentation [12]. On the whole, in terms of reconstruction, the data analysis procedure can be divided into three steps: (1) EM image stitching [13]; (2) neuronal segmentation in 2D EM image; (3) neuronal networks reconstruction. neuronal segmentation relies on the image segmentation [14],which is based on the detection of membrane over EM images. For membrane detection, [15] used the artificial neural network (ANN) as the pixel-wise classifier, which got a stencil of pixels surrounding the pixel as the input. The ANN classifier got binary outputs that labeled the membrane after trained on large amounts of data. However, many uncontrollable issues result in the homogeneous of neuronal boundaries color distribution in the stage of sample preparation and imaging under electron microscopy. The high overlapping of color distribution of intracellular and extracellular as well as the membranes cause it hard to get the characteristic description of the neuronal boundaries accordingly difficult to segment [7].

Connectome research (connectomics) has a number of competing objectives. On the one hand, investigators prefer an organism small enough that the connectome can be obtained in a reasonable amount of time, this argues for a small creature. On the other hand, one of the main utilities of a connectome is to relate structure and behavior, so an animal with a large behavioral repertoire is desirable. It's also very helpful to use an animal with a large existing community of experimentalists, and many available genetic tools. Drosophila looks very good on these counts [3, 4]. After preparation, a sample of the drosophila brain tissue is typically cut into 50-nanometers slices by the ATUM, each slice is then recorded as a 2D grayscale image in the SEM with pixel size about $5nm\times5nm$.

Our work uses the DCNN to get MDPM, based on which marker-controlled watershed is utilized to segment the neurons on the 2D EM images. The watershed performs well on the image segmentation despite the weak boundaries, however,it easily produces over-segmentation. By using the marker-controlled watershed on the MDPM, we get considerably fair segmentation results. Having got the segmentation, we connect the segmentations in sequence of images which are stitched by [13] to form the 3D reconstruction of neuronal networks. Our automated approach is effective that it rarely relays on manual operation, able to generate dense neuronal reconstruction simultaneously, which is mainly based on adjacent connection between pair of consecutive neuron segmentations; our semi-automated approach relays on the manual labeled neuron skeletons, with which, we can produce sparse and fine reconstruction. With our proposed method, we segment the neurons on the 2D EM images from drosophila brain cortex and drosophila mushroom body, on which our method shows fairly fine performance; In the following sections, we will describe our method in detail and some experimental results of the method. Finally, we will show parts of the automated reconstruction results of the drosophila mushroom body neuronal networks.

II. Neuron segmentation method

In this section we will describe the neuron membrane detection method based on the deep convolutional neural network and watershed based segmentation method. In the following, we will describe in details the procedure and work flow of those methods in subsections.

A. Boundary description

The segmentation of neuron is the base of reconstruction of neuronal networks. Neuron is a type of special cell, whose boundary is the membrane and the neural structure can also be separated by the membrane. Our segmentation method is based on the detection of membrane, which contours the internal segmentation of a neuron. We will firstly give the explicit definition of membrane namely the neuronal boundary in the EM image. Supposing a given point (x, y) that belongs to the membrane of neuron Neu_i in the EM image, the membrane can be defined as

$$\begin{aligned} M(Neu_i) =&\{(x,y)|(x,y) \notin intra\&\&(x,y) \notin \text{extra} \\ &\&\&\text{d(M(Neu}_i\text{))} \cap intra \neq \emptyset\}, \end{aligned} \tag{1}$$

where M denotes the pixels set of membrane that belongs to the neuron Neu_i on a slice of EM image, *intra* and *extra* is the shorthand of intracellular and extracellular that present the inside and outside of the cell respectively. On the EM image, the membrane is the connected image area that is stained contrast the other portion which can be labeled by experts showing in Fig. 1.

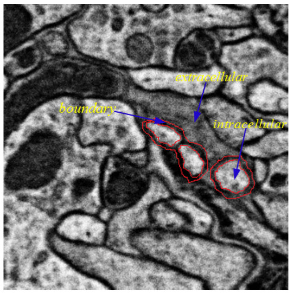


Fig. 1. Boundaries between the extracellular and intracellular regions or between intracellular regions.

Intracellular space and extracellular space are separated by membrane which usually has some unique features in the EM images. In some neural tissues, most of the neural structures densely bunch together where neural regions are tightly adjacent to other neighbour neural regions in EM image. The membrane becomes the only separation between neurons while no extracellular portion. There is a visual illustration in Fig. 2. In this case, we can define the boundary of cell similar to [16]. Supposing there are a set of non-overlapping neural segmentations $C = \{c_i\}$ from a $2D$ EM image, where c_i is the set of pixels of segmentation related to the certain neuron. Denote e_i as the area (connected pixels set) that expands from c_i to all the edges of neighbouring neurons segmentations $c_{j,j\in N_i}$, where N_i denotes the neighbouring neural segmentations adjacent to neuron i. In this case, the boundary of membrane between c_i can be defied as

$$\begin{aligned} M(c_i) =&\{(x,y)|(x,y) \in d(c_i)\&\& \\ &d(c_i) \cap c_{j,j\in N_i} \neq \emptyset\&\&(x,y) \notin d(c_i)\}, \end{aligned} \tag{2}$$

where $d(\cdot)$ denotes the expanding function which can be used to expand the c_i to e_i until e_i is adjacent to the edges of $c_{j,j\in N_i}$ and intersect with all the $c_{j,j\in N_i}$ regions. In practice we can use the image dilation operation as the expanding function. Similar to c_i, $c_{j,j\in N_i}$ can also be expanded as in equation 3.

$$\begin{aligned} M(c_{j,j\in N_i}) =&\{(x,y)|(x,y) \in d(c_{j,j\in N_i})\&\& \\ &d(c_{j,j\in N_i}) \cap c_i \neq \emptyset\&\&(x,y) \notin d(c_{j,j\in N_i})\}. \end{aligned} \tag{3}$$

Accordingly, the boundary between c_i and $c_{j,j\in N_i}$ can be defined as:

$$boundary(c_i, c_{j,j\in N_i}) = M(c_i) \cap M(c_{j,j\in N_i}). \tag{4}$$

After all, we can define the boundaries between all the pairs of segmented neuronal regions in the 2D EM images. In the following, we will describe the feature of boundary at pixel level.

B. Pixel-wise feature extraction

Our work is on the base of DCNN which is used as the pixel-wise classifier. DCNN comes from CNN, the early work that used convolutional neural network (CNN) is in the restoration

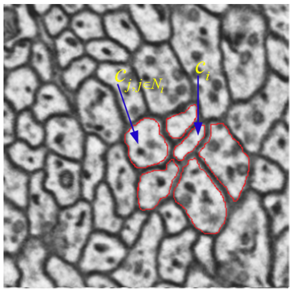


Fig. 2. Neurons that bunch densely together, $c_{j,j\in N_i}$ is one of the neural regions adjacent to c_i.

of EM image [17] where Jain used the supervised trained CNN for low-level image processing that restoring noisy of degraded images for segmentation. We use the DCNN to extract features that can be used in the pixel-wise classification task, which is the former step of neuron segmentation. DCNN learns from large amounts of training data and can extract highly abstract features by multi-layers of convolutions and non-linear operations. DCNN is from CNN and composes of the deeply hierarchical layers of the elementary layers of CNN. CNN is kind of ANN and different from the general ANN, the most important and basic layers of CNN are convolutional layers. In CNN, the convolutional filter's weights are replicated over 2D space and the non-linear transformation is over the 2D plane of output of 2D convolutions. The outstanding advantage of CNN is weight sharing which comes from the fact that the image statistics are stationary and combinations of features that are relevant in one region of an image are also relevant in other regions [18]. Because of the imposed weight sharing, DCNN has been successfully used in many high-level image processing problem, such as the image labeling problem. In the following, our method will show that CNN can also be used as a general method for low-level image processing of image segmentation.

The Neuron membrane as the boundary of a neuron segmentation is a set of pixels on the EM image, which can be denoted as M_i. The detection of membrane is equivalent to find the certain set of M_i on the image of I. In fact, because of the similarities of pixels in M_i, the detection of membrane can be transformed into the task of pixel-wise classification. To this thought, we need to get the features of each pixel. For a single given pixel point (x, y), we use its contextural information to construct the features of this point, and usually the surrounding pixels of this point is the natural contexture. The work of [15] used a stencil centred around a pixel to extract it's contexture while the DCNN we use as the feature extractor can learn directly from the image data and we extract a square image block centred around a pixel point as its contexture which illustrated in Fig. 3.

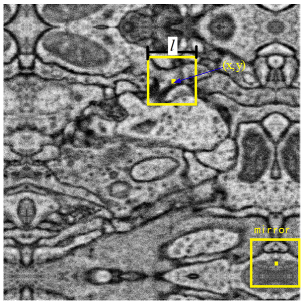
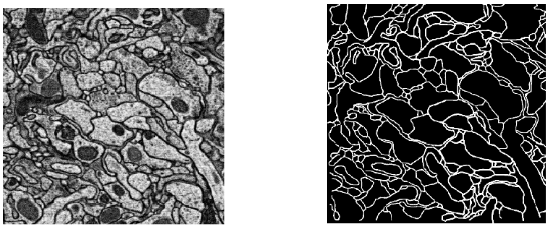


Fig. 3. Extracting a contexture square block of image centering a point.



(a) Original EM image

(b) Ground truth

Fig. 4. Images used for obtaining labeled data set; (a): for generating contexture of each point; (b): for generating label of each pixel.

As demonstrated in Fig. 3, for a given pixel point (x, y), we get a square image block that centred around it with size of $l \times l$, using the raw intensity value of the block as the input of DCNN. l is an odd number to enforce symmetry. When a pixel is close to the image border, it's square image block will include pixels outside boundary of the image; in this case, we firstly mirror along all the sides of the origin image the inside boundary with $\lfloor l/2 \rfloor$ pixels. After mirrored the boundaries, the origin image's size will increase $l - 1$ in both sides. Thus, for all of the pixels of the origin image, we can get a $l \times l$ image block. The scale of l is dependent on the resolution of EM image and the quality of EM imaging as well as many other unknown factors; therefore, l is an adjustable parameter that may have effect on the recognition accuracy.

C. Boundary recognition

In this subsection, we recognize membrane of neurons by using DCNN, which is able to automatically extract high level abstract features from the raw image intensity values. As mentioned above, for every pixel point $I(x, y)$, we can get a square image block b as the contexture of that point and label c of which according to manually labeled ground-truth. By sampling on the labeled images, we can obtain an origin labeled data set $T = \{(b_1, c_1), (b_2, c_2), (b_3, c_3), ..., (b_n, c_n)\}$, where subscript of b and c indicate the different samples. One of the portions of labeled set T is used to train the DCNN. The neuron membrane recognition model can be formulated as

$$\begin{aligned} &c(x, y) = f_{cnn}(fea(I(x, y))) \\ &c \in \{0 : nonmembrane, 1 : membrane\} \end{aligned} \tag{5}$$

For any given pixel point $I(x, y)$, we can use the extracted features $fea(I(x, y))$ that is from b of $I(x, y)$ forwards through DCNN trained hierarchical layers. The trained DCNN classifier will predict the labels of instances from the another portion of T.

*1) **Training procedure**:* Given some EM images, we manually labeled the membrane with expert experience to get the ground-truth as shown in Fig. 4 .

the pixel overlapped on the membrane is labeled to positive samples, the others (intracellular and extracellular) are the negative samples. There are approximately 1.5 million positive instances from eight 1000×1000 training EM images. Because of the sparsity of membrane, positive instances are 4 time smaller than negative instances. In order to balance the positive and negative samples, we use all the positive instances and samples on the negative pixels to let ration of the number of positive instances to negative equal 1; Finally, we obtained 3 million labeled data. we separate the labeled data to training set about 2.5 million and validation set about 0.5 million. We directly test on the testing images, thus, the testing set is not needed. The validation set are not used in the training but used in all the duration of the training processing; to some degree, the validation is mainly used for us to observe the generalization ability and helps us choose or modify the training parameters. On the training processing, DCNN converges after many epoches, However, the DCNN is not convex function, thus it may converge to the local minimum point or overfit on the training set; in this case, we will use error testing on the validation set to modify training parameters or do early-stoping for the sake of anti-overfitting.

*2) **Testing procedure**:* After the DCNN is trained, we get the DCNN model as the pixel-wise classifier. The testing is implemented directly on the test image I_{test} to get the predicted label L_{test}. According to the ground-truth G_{test} of I_{test}, the test error can be calculated with the formulation:

$$pixel - error = \frac{sum(L_{test} \odot G_{test})}{h \times w}, \tag{6}$$

where $\odot$ denotes XOR operation, $sum(L_{test} \odot G_{test})$ represents calculating non-zero number of $L_{test} \odot G_{test}$, h and w denote the hight and width of image L_{test} respectively. The $pixel - error$ is the membrane detection error which will be used in the evolution of segmentation.

In the most of binary classification tasks, there are usually two probabilities to be calculated by the classifier:
(1): The probability of the pixel belonging to each class;
(2): Probability of a boundary dividing the adjacent regions. In the graph cut [19], (1) was used as the unary and (2) as weight. In our work we will mainly utilize the probability in (1). We apply the DCNN that uses the $softmax$ as the output layer to the image membrane detection task. $softmax$ is formulated as

$$softmax(a) = \arg\max_j \left(\frac{e^{-w_j{}^t a}}{\sum_i e^{-w_i{}^t a}}\right), \tag{7}$$

where a denotes the feature vector of image x, w_i is the regression weight vector, i, j denote the classes. The $softmax$ outputs class that has the maximum regression value. Initially, the DCNN outputs a binary membrane detection map where each pixel represents class; benefiting from the property of $softmax$, we modify the output of DCNN in the test stage where we let the DCNN outputs the statistic salience of membrane namely the membrane detection probability map(MDPM)

$$Prob(a) = \frac{e^{-w_1{}^t a}}{\sum_i e^{-w_i{}^t a}}, \tag{8}$$

with the MDPM, we can get the binary classified membrane map by thresholding, however, there isn't a global threshold to get the finest membrane detection map. On the one hand, our goal is to segment the neurons on the EM images, while the binary map is not good enough for the fine segmentation; On the other hand, membrane detection probability map filters the most of the non-membrane portions which interfere the segmentation and retains the weakly membrane yet. To this thought, we apply post-segmenting algorithm on the MDPM. In the next section, we will introduce the implementation of segmentation on the MDPM with mark-controlled watershed.

D. Watershed segments neurons

The MDPM is the statistical salience of membrane on large amounts of data to some degree. On the MDPM, binary map can be obtained by thresholding. As mentioned above, if we apply the global threshold on the MDPM, lots of detailed information will lost ,which will result in the coarse segmentation of neuron. Because of the fact that the weak and intense pixel values intricately spread across the the MDPM which results from non-uniform of membrane in term of intensity in color. With big threshold, many membranes will be classified as non-membrane, which results in unclosing of neuron membrane and causes under-segmentation, while, with small threshold, many interferences come out and cause over-segmentation. Thus, as there is no good global threshold that solves this dilemma, we consider using other segmentation method over the MDPM.

MDPM reflects the membrane recognition salience of each pixel of the EM image. The membrane is the boundary of neuron based on which we segment the neuron on MDPM by utilizing the watershed [20], which is sensitive to weak boundaries and able to get segmentation line with width of one pixel. Watershed method is on the base of morphological operations and with regions growing from the seed points(often set as the local minimums). The excessive local minimums on the MDPM which result from the weak probabilities of weak membrane or non-membrane cause sever over-segmentation if we directly use the watershed algorithm over the local minimums. Our algorithm overcomes the problem by utilizing the marker-controlled watershed [21]. The segmentation procedure consists of the following steps:
(a) Removing the small connecting bridges between otherwise unconnected image regions by morphologically opening and closing by reconstruction using a structure element on the MDPM;
(b)generating the markers for watershed using a threshold on the output of (a);
(c)Applying watershed algorithm on the output of (a) based on the markers getting from (b).

III. 3D reconstruction method

For the reconstruction of neuronal networks structures, we need not only to segment out the neurons on the 2D EM stack of images but also decide which neuron each voxel belonging to in 3D. In the 3D reconstruction of neurons, we need accomplish the 3D segmentation at first. We can get the 2D segmentation of each neuron on the sequence of EM images with our proposed method. In the following, we will connect the sequence of 2D segmentations to form the 3D neuronal structures. Our method is based on the fusion of 2D segmentations and the neuronal skeletons.

Supposing there is a given sequence of K_N slices of EM images and their 2D neuronal segmentations $S = \{s_i^z | z \in \mathrm{Z}, 1 \leq z \leq K_N\}$, where z denotes the sequential index of segmentations, i denotes the index of distinct segmentation within the same slice. To reconstruct the Neu_i from the sequence of EM images, we have two policies; the first method: labeling the the skeleton of the neuron Neu_i in the sequence of EM images according to the position and coordinate of neuron on the EM image. the skeleton being denoted as l_i, we formulate the reconstruction of Neu_i based on the sequence of segmentations S as

$$Neu_i = \{s_j^z | s_j^z \cap l_i \neq \emptyset, 1 \leq z \leq K_N\}, \tag{9}$$

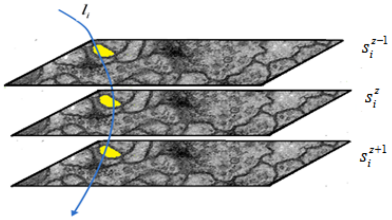


Fig. 5. Fusion of segmentations and skeleton in the sequence of 2D EM images.

where in the sequence of images, $\cap$ denotes the intersecting in the 3D space and $s_j^z \cap l_i \neq \emptyset$ represents that skeleton l_i pass through the region of s_j^z. The segmentations that belong to the same neuron are connected to form 3D reconstruction under the constraint of skeleton l_i. The second method: utilizing the semi-automated skeleton tracing strategy, for the sequence of EM images, connecting the two adjacent images serially, supposing two segmented EM images s^z,s^{z+1} and a distinct segmentation s_j^z of s^z, we need to decide the segmentation s_j^{z+1} belonging to the same neuron to which s_j^z belongs on slice s^{z+1}. Let c_j^z denote the geometrical centroid of s_j^z, then

$$s_j^{z+1} = \underset{s_i^{z+1}}{argue}\{\mathrm{c}_j^{z+1} \cap s_i^{z+1} \neq \emptyset\}. \tag{10}$$

Finally, we connect the segmentations on each slice together to reconstruct neurons. It should be pointed out that we only need to manually label the start point of skeleton on the first slice of the sequence, then the segmentations and skeleton automatically generating interactively.

IV. Experiments

In this section, we will mainly introduce the experimental results of segmentation and reconstruction.

A. Segmentation

In this experiment, the EM images are obtained from portion of the drosophila brain tissue. The tissue specimen is specially stained to make the neuronal membrane contrast against the other tissue. The data generating procedure is as follows: (1) The block of stained tissue being cut into slices at thickness of 50nm by automated tap-collecting ultramicrotome(ATUM). (2) Putting the thin slices on the wafers sequentially. (3) Imaging on the wafers by Scanning Electron Microscopy(SEM) to produce sequences of EM images. In order to observe the neurons at synaptic level, specimen is imaged at resolution of $5nm \times 5nm$, as mentioned above, we use 2.5 million training instances from 3 million label data to train the DCNN. By experimenting on multiple parameters, we choose a group of parameters that presents relatively good performance in terms of pixel-error. The architecture of DCNN is reported in Table I.

TABLE I
DCNN architecture

$Layer$	$Type$	$MapsAndNeurons$	$KernelSize$
0	input	1 map of 65×65	$--$
1	convolutional	48 maps of 60×60	4×4
2	max-pooling	48 maps of 30×30	2×2
3	convolutional	48 maps of 28×28	3×3
4	max-pooling	48 maps of 14×14	2×2
5	convolutional	48 maps of 10×10	5×5
6	max-pooling	48 maps of 5×5	2×2
7	Fully-connected	250	$--$
8	Fully-connected	2	$--$

We let the contexture image block edge size l to be 65 and utilize the max-pooling deep convolutional neural network to generate the MDPM. As aforementioned, the MDPM is segmented by watershed algorithm. Some results of segmentation are illustrated in Fig. 6. To evaluate the segmentation of the test image, usually, there are three metrics in relation to the ground-truth:

Pixel error: as defined in section II-C2, which directly gives the score of the pixels dissimilarity of segmentation relating to ground truth.

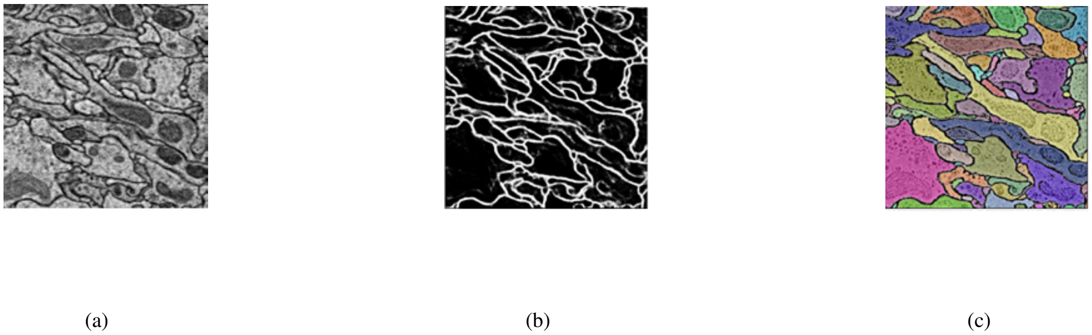
Rand error: defined as $1 - F_{rand}$, where F_{rand} represents the F_1 score of Rand index [22], which directly measures the clusters of segmentations against the ground truth.

Warp error: metric that measures the topological disagreements [23] between proposed segmentation and ground truth.

The quantized errors of the segmentation results of our method are in the Table II.

B. Reconstruction

We firstly get the MDPM by DCNN from origin image and then apply the marker-controlled segmentation procedure to the MDPM for getting sequence of segmentations of neurons on the 2D EM images. The neuronal reconstruction is on



(a)

(b)

(c)

Fig. 6. (a) The original EM image; (b) The membrane detection probability map; (c) The segmented image over the membrane detection probability map.

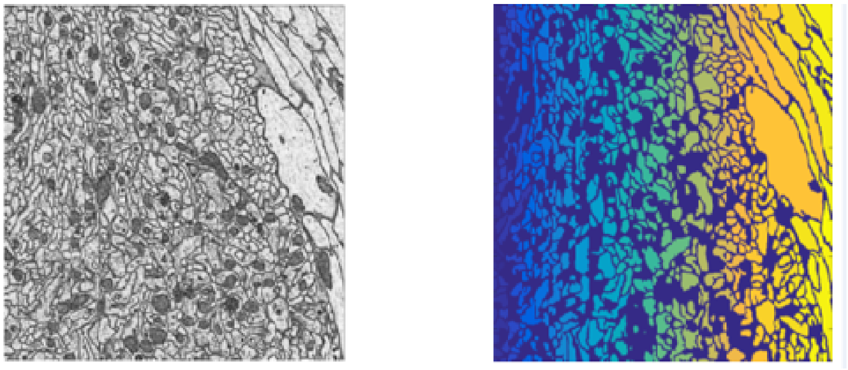


Fig. 7. Left: one slice of drosophila mushroom body; Right: segmented results.

TABLE II
QUANTIZED ERROR OF SEGMENTATION

Metric	*Error*
Pixel – error	0.134
Warping – error	0.00573
Rand – error	0.0379

the base of neuronal segmentation. We obtained 1850 slices generated by ATUM with thickness of 50 nanometers and imaged under SEM, by which, we get a sequence of 1793 images with the size of 8000×8000. By using the first method described in section III, we sparsely reconstruct a few partial neurons(Fig. 8) from drosophila mushroom body in Fig. 7, the 2D segmentations are extracted from 500 consecutive EM images from the full 1793 images. we densely reconstruct neurons by using the second method described in the section III which is semi-automated but weakly relies on the manual interference. The dense reconstructions (Fig. 9) come from the 500 consecutive segmented slices of the full 1793 images.

V. Discussion and Conclusion

The strength of deep neural network contributed much to our approach of neuronal membrane detection. The DCNN is trained by online back-propagation to become a very powerful pixel-wise membrane detector with considerably high accuracy. The combination of DCNN and watershed enables us to get the very fine segmentations of neurons in the EM images. Based on the segmentation, 3D reconstruction is implemented by automated and semi-automated method. Our approach gives a full procedure for reconstruction with high flexibility and feasibility.

We applied our method to the reconstruction of neurons in the drosophila mushroom body. We believe that the procedure of our proposed method can be a general method for the reconstruction of neuronal networks.

Our reconstruction method fully depends on the sequence of image segmentation, despite the strength of DCNN, the image segmentation is yet not solved perfectly. There is still much work to be done about the accuracy and efficiency.

Despite the high performance of GPU implementation of deep convolutional neural networks, the pixel-wise recognition

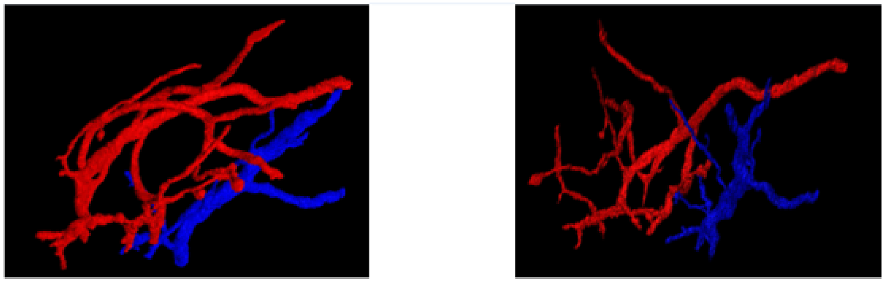


Fig. 8. Visualization of the sparse reconstruction of drosophila by semi-automated method.

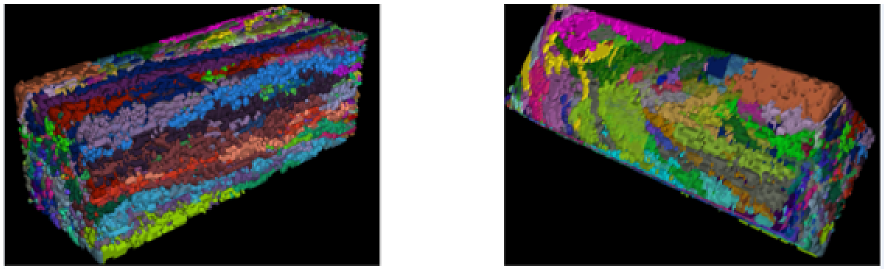


Fig. 9. Visualization of the dense reconstruction of drosophila by automated method.

of membrane is still very time-consuming. What's more, the connection method of segmentation for reconstruction is also not of high accuracy. There is much work for improving the overall 3D reconstruction accuracy in the future.

Acknowledgment

We thank Dr. Yu Kong, Yang Yang, Danqian Liu and Ke Zhang(Institute of Neuroscience, CAS) for sample preparation and sectioning. We thank Mr. Lixin Wei and colleagues (Institute of Automation, CAS) for Zeiss Supra55 SEM and technical support. This paper is supported by Special Program of Beijing Municipal Science&Technology Comission (No. Z161100000216146), Strategic Priority Research Program of the CAS (No. XDB02060001), National Science Foundation of China (NO. 61101219, NO. 61201050, NO. 61306070) and Institute of Automation, CAS for 3D Reconstruction of Brain Tissue at Synaptic Level (NO. Y3J2031DZ1).

References

[1] M. Helmstaedter, "Cellular-resolution connectomics: challenges of dense neural circuit reconstruction," Nature Methods, vol. 10, no. 6, pp. 501-507, 2013.

[2] K. Masaaki, et al, "Automated transmission-mode scanning electron microscopy (tSEM) for large volume analysis at nanoscale resolution," Plos One, vol. 8, no. 3, e59573-e59573, 2013.

[3] T. Shin-Ya, et al, "A visual motion detection circuit suggested by Drosophila connectomics," Nature, vol. 500, no. 7461, pp. 175-181, 2013.

[4] A. A Paul, et al, "The brain activity map project and the challenge of functional connectomics," Neuron, vol. 74, no. 6, pp. 970-974, 2012.

[5] K. Narayanan, et al, "Saturated Reconstruction of a Volume of Neocortex," Cell, Vol. 162, no. 3, pp. 648-661, 2015.

[6] V. Kaynig, et al, "Large-Scale Automatic Reconstruction of Neuronal Processes from Electron Microscopy Images," Medical Image Analysis, vol. 22, no. 1, pp. 77C88, 2015.

[7] M. Berning, K. M. Boergens, and M. Helmstaedter, "SegEM: Efficient Image Analysis for High-Resolution Connectomics," Neuron ,vol. 87, no. 6, pp. 1193-1206, 2015.

[8] V. Jain, H. S. Seung, and S. C. Turaga, "Machines that learn to segment images: a crucial technology for connectomics," Current Opinion in Neurobiology, vol. 20, no. 5, pp. 653-666, 2010.

[9] J. Schmidhuber, "Deep learning in neural networks: An

overview," Neural Networks the Official Journal of the International Neural Network Society, vol. 61, pp. 85-117, 2015.

[10] Krizhevsky, Alex, I. Sutskever, and G. E. Hinton, "ImageNet Classification with Deep Convolutional Neural Networks," Advances in Neural Information Processing Systems, vol. 25, no. 2, pp. 2009-2012, 2012.

[11] L. Aurlien, et al, "Supervoxel-based segmentation of mitochondria in em image stacks with learned shape features," IEEE Transactions on Medical Imaging, vol. 31, no. 2, pp. 474-486, 2012.

[12] B. Carlos, et al, "Learning context cues for synapse segmentation," International Conference on Medical Image Computing and Computer-Assisted Intervention Springer-Verlag, pp. 585-592, 2012.

[13] X. Chen, Q. Xie, L. Shen, and H. Han, "Wrinkle Image Registration for Serial Microscopy Sections," 2015 11th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS) IEEE Computer Society, 2015, pp. 23-26.

[14] A. Radhakrishna, et al, "SLIC superpixels compared to state-of-the-art superpixel methods," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 34, no. 11, pp. 2274-2282, 2012.

[15] J. Elizabeth, et al. "Semi-automated neuron boundary detection and nonbranching process segmentation in electron microscopy images," Neuroinformatics, vol. 11, no. 1, pp. 5-29, 2013.

[16] T. Liu, C. Jones, M. Seyedhosseinia, and T. Tasdizena, "A modular hierarchical approach to 3D electron microscopy image segmentation," Journal of neuroscience methods, vol. 226, no. 8, pp. 88-102, 2014.

[17] V. Jain, et al, "Supervised Learning of Image Restoration with Convolutional Networks," Computer Vision, IEEE 11th International Conference on IEEE, pp. 1-8, 2007.

[18] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, Scene Parsing with Multiscale Feature Learning, Purity Trees, and Optimal Covers, in Proc. of the International Conference on Machine Learning (ICML'12), Edinburgh, Scotland, pp. 575-582, 2012.

[19] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol.23, no. 11, pp. 1222-1239, 2001.

[20] Meyer, Fernand, "Topographic distance and watershed lines," Signal Processing , vol. 38, no. 1, pp. 113-125, July 1994.

[21] X. Shengzhou, L. Hong, and S. Enmin, "Marker-controlled watershed for lesion segmentation in mammograms," Journal of Digital Imaging, vol.24, no. 5, pp. 754-763, 2011.

[22] W.M. Rand, "Objective criteria for the evaluation of clustering methods," Journal of the American Statistical association, vol.66, no. 336, pp. 846-850, 1971.

[23] V.Jain, et al, "Boundary Learning by Optimization with Topological Constraints," Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on IEEE, San Francisco, CA, pp. 2488-2495, 2010.