Membrane extraction using two-step classification and post-processing

Xiao Tan[1]*, Changming Sun[1], and Tuan D. Pham[2]

[1]CSIRO Mathematics Informatics & Statistics
Locked Bag 17, North Ryde, NSW 1670, Australia

[2]Research Center for Advanced Information Science and Technology
Aizu Research Cluster for Medical Engineering and Informatics
The University of Aizu, Aizu-Wakamatsu, Fukushima 965-8580, Japan

{xiao.tan,changming.sun}@csiro.au,tdpham@u-aizu.ac.jp

Abstract. This paper presents an algorithm for membrane structure extraction in electron microscopy (EM) stacks, which is one of the critical tasks in the brain structure analysis. We first classify the pixels into membrane and non-membrane classes by adaptive threshold classification using edge strength, then dark blobs which may be incorrectly classified into the membrane class are eliminated from the class by checking its co-occurrence in the neighboring images of the stack. After that, we extract more features and use them to train a SVM for further classification on pixels. Finally, post-processing is applied on the probability map obtained from the SVM to recover some membrane structures which may be misclassified into the non-membrane class and to eliminate some dark regions which are incorrectly classified into the membrane class. Experimental evaluation shows that our method performs well on the test datasets.

Key words: Membrane extraction, EM image analysis, SVM, Pixel classification

1 Introduction

The electron microscopy (EM) stacks analysis is indispensable to the study of brain structure and function. Many researches have been conducted in the area of the application of EM to brain study [2–4]. Accurate membrane detection method of the EM stacks with high robustness to noise and image alignment errors is required for biological studies such as neuronal development and reconstruction of neuropile [1]. Membrane extraction is to classify the pixels into two classes, i.e., membrane and non-membrane classes. The most widely used algorithm to carry out this duo classification is the machine learning based algorithm. Such algorithm learns from a training dataset, and then run the classification

* Ph.D student of University of New South Wales in Canberra, Australia

on the testing datasets. The result is given in the form of the probability image where each pixel has a value between 0 (100% membrane certainty) and 1 (100% non-membrane certainty). As some global information such as the membrane shape, and the consistency of membrane in the neighborhood may be difficult to be incorporated into this pixel based method, we use a post-processing step to improve the result. The rest of the paper is organized as following. We introduce our two-step classification in Sect. 2 and Sect. 3. Post-processing is described in Sect. 4. The experimental result and quantitive evaluation are given in Sect. 5. The conclusion is given in Sect. 6.

2 Initial Classification

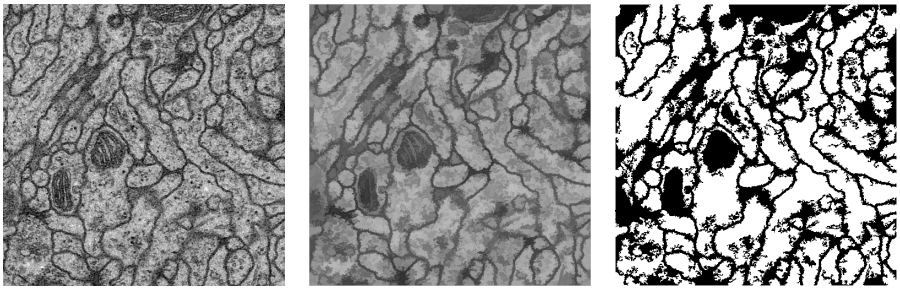
In this section, we detail our initial classification method, including: adaptive threshold classification using edge strength (ADTES), and dark blob location.

2.1 Adaptive Threshold Classification Using Edge Strength

Small dark regions in the non-membrane class may be incorrectly regarded as membrane, and small bright regions in dark blobs may impact the dark blob location step, which will be discussed later. We use attribute opening process [5] to eliminate these regions as well as noise before performing ADTES.

The ADTES starts with finding a measure of the edge strength in the input image using a 3×3 Sobel filter. For each pixel, the size of its local neighborhood is increased (from a 3×3 neighborhood up to a maximum of $W_{\mathrm{m}} \times W_{\mathrm{m}}$, W_{m} is the maximum width of the membrane structure) until the mean of the non-zero edges in the neighborhood of the Sobel image exceeds a global edge strength threshold. If this threshold is exceeded before W_{m} is reached, a local grey-scale threshold is then determined for this neighborhood in the input image, and the classification threshold for the pixel is set to the local grey-scale threshold. If the global edge strength threshold is not exceeded, the classification threshold is set to a global grey-scale threshold. Given the classification threshold, we classify a pixel as membrane if its grey level value is lower than the threshold; otherwise, as non-membrane.

The edge strength and grey-scale thresholds are derived as follows. The global mean, M_{G}, standard deviation, std, and edge strengths are calculated for all non-zero edges in the Sobel image. The global edge strength threshold is set to: $M_{\mathrm{G}} + \beta \times std$, where β is the weight on standard deviation used in thresholding edges. The global grey-scale threshold is set to the mean of the input grey-level image for all pixels which have edges strength greater than M_{G}. The local mean edge strength is calculated for all non-zero edges in the local neighborhood of the Sobel image. The local grey-scale threshold is set to the mean grey-level of all pixels in the local neighborhood of the input image which have edges with above local mean edge strength. An example of the ADTES result is given in Fig. 1.



(a) (b) (c)

Fig. 1. An example of the ADTES process. (a): Input image. (b): Result after attribute opening. (c): Classification result (black for membrane pixel, white for non-membrane pixel).

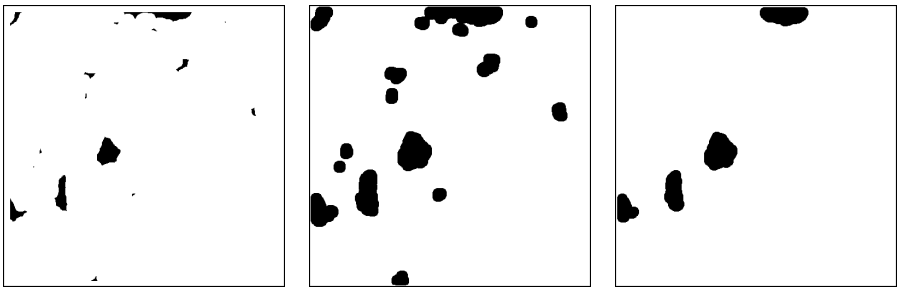
2.2 Dark Blob Location

In this step, we locate the dark blobs which are regarded as membrane and should be eliminated from the membrane class. As most neural structures are narrow, while the dark blobs are not, we first perform erosion to the membrane structure using a round mask with a radius of $\frac{M_G}{2}$; thus most neural structures will be eliminated from the membrane class. Each connected region in the result of erosion is called a blob core which is corresponding to a possible dark blob, and the dilation to the blob core with the same round mask is its blob region. Some thick neural structures may be incorrectly regarded as dark blob. It is hard to recognize them in a single image, but it will be easier when considering the image stack. Dark blobs have the following two properties: (1) they are close to each other in the neighboring images of the 3D stack; (2) the size of the dark blob change gradually in the neighboring images. We find dark blobs from all possible dark blobs based on these two properties.

With the first property, we check if the dark blobs are close to each other by the following. For each possible dark blob, we project its blob region onto the neighboring images in the stack, and then we check all pixels in the neighboring images within the projected region. All blob cores located in this region are denoted as the corresponding blob cores. If a dark blob has at least one corresponding blob core in each of the 4 sequential neighboring images, the dark blob will be kept; otherwise, it will not be regarded as a dark blob any more.

With the second property, we check the change ratio of the blob size by the following. For each possible dark blob, we project its blob region onto the two neighboring images, and then we calculate the size of the overlapping region between the projected blob region and the corresponding blob region in the two neighboring images. We denote the size of its blob region by S, and denote the larger value of the two overlapping sizes by O. If the value of $\frac{O}{S}$ is too small, i.e., smaller than a threshold, T_1, it indicates a sharp size change of a dark blob between two neighboring images, which is unlike to happen. In such a case, this

dark blob is more likely to be a thick neural structure and will not be regarded as a dark blob any more. On some occasions, structures which are near to the dark blob may be incorrectly connected by the nearby dark blob, and this will also result in a small $\frac{O}{S}$. To handle this, we introduce another threshold, T_2. If $\frac{O}{S}$ is higher than T_2, we accept the whole blob region. If the value of $\frac{O}{S}$ is smaller than T_2 and higher than T_1, we set the blob region to the maximum overlapping area and recalculate the blob core. We iteratively carry out the process described in the above paragraph and the process described in this paragraph until all possible dark blobs are regarded as dark blobs. Fig. 2 shows the result of dark blob location in an image. After finding the location of dark blobs, we label the pixels in the dark blobs as non-membrane (i.e., bright part in Fig. 1).

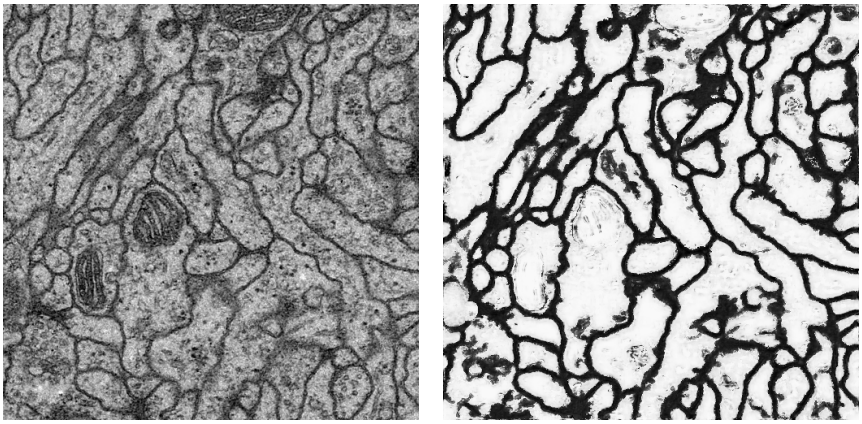


(a) (b) (c)

Fig. 2. The result of dark blob location. (a): Blob cores of all possible dark blobs. (b): Blob region of all possible dark blobs. (c): Dark blobs found using the process described in Sect. 2.2.

3 Further Classification Using SVM

In this step, we use a multiple scales scheme to extract features for each pixel. The features include: Laplace filter, Sobel filter, attribute opening filter [5], bilateral filter [7], non-linear anisotropic distribution, Hessian magnitude, the difference of Gaussians, the mean and variance of local neighborhood, the mean and roundness of mean-shift segmentation, and the distance transformation of the result obtained from the ADTES. We use the results from the ADTES together with all these features to train a SVM classifier. The number of non-membrane pixels is much larger than that of membrane pixels in the training data. But in order to obtain a good SVM classifier, the balance of the two classes in the training set is required. Therefore, we randomly select 50000 pixels (half membrane pixels and half non-membrane pixels) from all pixels of the training image sequence. We employ radius basis function (RBF) core in our SVM classifier and the parameters of RBF is trained according to the cross-validation accuracy [6]. An example of the classification result of test images is shown in Fig. 3.



(a) (b)

Fig. 3. (a): Original image. (b): The result of (a). Probability obtained from the SVM classification (from 0 (membrane) to 1 (non-membrane)).

4 Post-processing

In the two-class classification the threshold is often set to 0.5. However, some membrane pixels which have higher probability values than 0.5 will be regarded as non-membrane and this may cause the break of a membrane structure, such as the region pointed out by the red arrow in Fig.4(a). To reduce the break, we first set the classification threshold, T_C, higher than 0.5; and then we find the possible membrane by selecting the pixel whose probability value is lower than T_C. By doing this, the membrane pixels with higher probability values than 0.5 can be included into a possible connected membrane region. Some dark regions in the non-membrane region, such as the region pointed out by the red arrow in Fig.4(b), may be incorrectly selected as possible membranes. For each connected possible membrane region, we check if it has a circle structure which is the prominent property of a membrane structure. If it has no circle structure, it must has only one adjacent non-membrane region, and we set the pixels in this possible connected membrane region to its adjacent non-membrane region. For each connected region, we calculate the average probability value over all pixels in this region to obtain the final result. The results after post processing are shown in Fig.4(c) and (d).

5 Experimental Results

The parameters can be set empirically or trained from the training data set. In our experiment, the parameters are set as the following: $T_1 = 0.3$, $T_2 = 0.8$, $T_C = 0.6$, and $W_{\mathrm{m}} = 33$. The results of some images in the test image sequence are shown in Fig. 5. The quantitative evaluation to the method is carried out

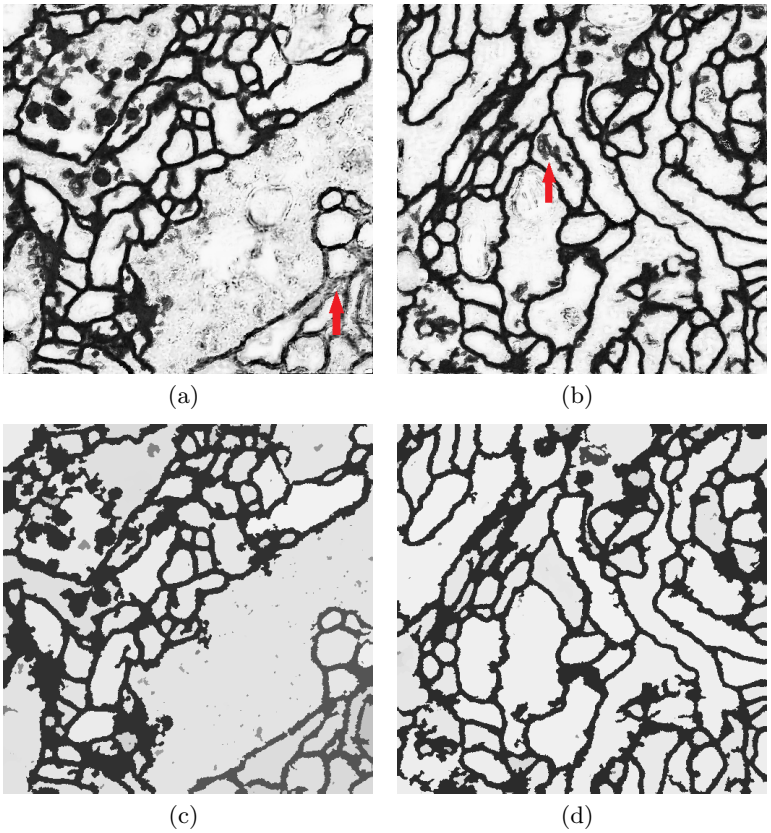


Fig. 4. (a) and (b): The probability value before post-processing (scaled to "0-255"). (c) and (d): The results of (a) and (b) after post-processing.

based on three metrics, i.e.: pixel error, Rand error [8], and warping error [9]. The evaluation result of our method is, Rand error: 0.15314, warping error: 0.00068, and pixel error: 0.08787.

6 Conclusion

In this paper, we proposed an algorithm for automatical extraction of membrane using two-step classification and post-processing. In the pixel classification step, we firstly build an initial classifier on pixels. The initial classifier is built by segmenting the input image into a binary image with an adaptive threshold based on edge strength and dark blob elimination. Then, we extract more features to train a SVM classifier for pixels. The post-processing step further improves the results based on the circle property of the membrane structure. The experimental results show that our method performs very well for the three metrics.

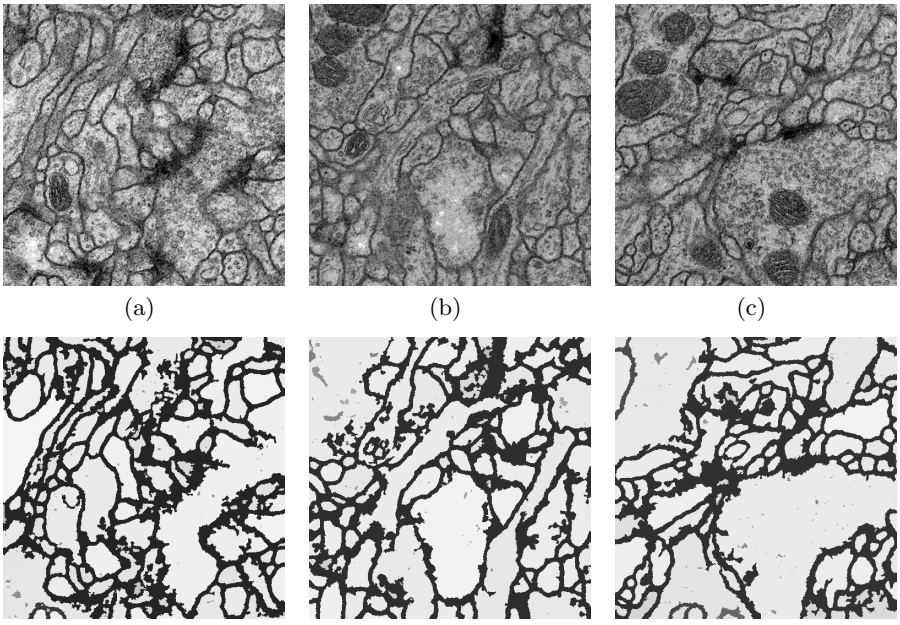


Fig. 5. Results of our method. (a)-(c): Images from test sequence. (d)-(f): Results obtained from our method.

References

1. Cardona A., Saalfeld S., Preibisch S., et al.: An integrated micro- and macroarchitectural analysis of the Drosophila brain by computer-assisted serial section electron microscopy. PLoS Biol 8(10): e1000502. doi:10.1371/journal.pbio.1000502 (2010)
2. FIALA, J.C.: Reconstruct: a free editor for serial section microscopy. Journal of Microscopy 218(1), 52–61 (2005)
3. Kremer, J.R., Mastronarde, D.N., McIntosh, J.R., et al.: Computer visualization of three-dimensional image data using IMOD. Journal of Structural Biology. 116(1), 71–76 (1996)
4. Mishchenko, Y.: Automation of 3D reconstruction of neural tissue from large volume of conventional serial section transmission electron micrographs. Journal of Neuroscience Methods. 176(2), 276–289 (2009)
5. Breen, E.J., Jones, R.: Attribute openings, thinnings, and granulometries. Computer Vision and Image Understanding. 64(3), 377–389 (1996)
6. Chang, C.C., Lin, C.J.: LIBSVM: A library for support vector machines. ACM Transactions on Intelligent Systems and Technology. 2(3), 27:1–27:27 (2011)
7. Tomasi, C., Manduchi, R.: Bilateral filtering for gray and color images. In: 6th ICCV, pp. 839–846. IEEE Press, Bombay (1998)
8. Unnikrishnan, R., Pantofaru C., Hebert M.: Toward objective evaluation of image segmentation algorithms. PAMI. 29(6), 929–944 (2007)
9. Jain, V., Bollmann, B., et al.: Boundary learning by optimization with topological constraints. In: CVPR, pp. 2488–2495, IEEE Press (2010)