34th Annual International Conference of the IEEE EMBS
San Diego, California USA, 28 August - 1 September, 2012

Dendritic Spines Detection Based on Directional Morphological Filter and Shortest Path

Ran Su[1], Changming Sun[2] and Tuan D. Pham[3]

Abstract— **Dendritic spines are membranous protrusions from neuron's dendrites. They play a very important role in the nervous system. They are very small and have various shapes; hence it is very challenging to detect them in neuron images. This paper presents a novel method for detecting dendritc spines in 2D images. A new dendrite backbone or centerline extraction method is introduced herein which is based on an iterative process between smoothing and extraction. The proposed method iteratively refines the extraction result using both directional morphological filtering and improved Hessian filtering until a satisfactory extraction result is obtained. A shortest path method is applied along a backbone to extract the boundary of the dendrites. Spines are then segmented from the dendrites outside the extracted boundary. The proposed algorithm has been tested with many images and good results are achieved.**

I. INTRODUCTION

Dendritic spines are small protrusions from neuron's dendrites. Typically, spines have a large spine head, which connects to the dendrites via a membranous neck. The most notable classes of spine shapes are 'thin', 'stubby', 'mushroom', and 'branched'. A number of human diseases such as Schizophrenia and mental retardation are relevant to alterations of spine's morphology or density [1], [2]. Analyzing spines is helpful for developing drugs to treat or slow down these diseases. Automatic detection of spines in 2D images is very important, because it can help release biologists from the heavy burden of manual spines detection process.

There are some researches on automatic detection of spines. Herzog et al. use a parametric model to analyze the morphology of the dendrites and spines [3]. But this method needs to add the spines neck manually if they are too thin. Koh et al. detect spines based on their morphology [4]; Xu et al. detect spines through two grassfire procedures [5]; Rodriguez et al. use adaptive local thresholding, voxel clustering and Rayburst sampling to analyze spines [6]. However, these methods are based on segmentation methods using thresholding, so the results are sensitive to the threshold selection. Cheng et al. apply adaptive thresholding, SNR based method and morphology analysis to separate spines [7]. They may not have a smooth backbone result by just using the thinning algorithm as spines in the binary image will affect the detection result. Janoos et al. reconstruct spines using surface representation and extract skeletons based on the medial geodesic function [8]. Large spines are easy to be lost and spines shape are not well presented in this method.

Here we develop a new backbone extraction approach and apply a shortest path method to detect dendrite boundaries and isolate dendritic spines. A backbone is the centerline of the dendrite structure. The novelty of our backbone extraction approach is that it uses an iterative process which can smooth the dendrites and refine the backbone extraction results. The backbone extraction of the dendrites is a key step for spines detection as it is the place where spines attach to. However, backbone extraction is always affected by the huge number of small spines. Besides, normal thinning algorithm is usually conducted on the binary images, which depend heavily on the thresholding algorithm. A simple thresholding algorithm may make a backbone disconnected. Our backbone extraction approach can iteratively remove the small spines from the dendrites, making the backbone extraction result better and re-connect the broken parts on the backbone. Convincing results are obtained using our method. We will describe our method in Section II and Section III, and show experimental results in Section IV.

II. DENDRITES BACKBONE EXTRACTION

To improve the extraction results, we first use a median filter to reduce noise. The filtered image is denoted by I_M. Then to deal with spines which are too close to each other, which often become connected after operations such as the Hessian filter, we compute the gradient of the image, normalize the gradient magnitude to the range from 0 to 255, and use the difference between the image intensity and normalized gradient value to build a new image, denoted by I_l.

Frangi et al. present a multi-scale approach for the linear feature enhancement which is based on the eigenvalues of the Hessian matrix [9]. Assuming the two eigenvalues of the Hessian matrix are λ_1 and λ_2 and they are ordered as $|\lambda_1| \leq |\lambda_2|$. The following is used as a filter to enhance the linear feature:

$$H(s) = \begin{cases} 0 & \text{if } \lambda_2 > 0, \\ \exp(-\frac{{R_\beta}^2}{2\beta^2})(1-\exp(-\frac{S^2}{2c^2})) & \text{else.} \end{cases} \quad (1)$$

where $R_\beta = |\lambda_1| \,/\, |\lambda_2|$, which is a blobness measure, $S = \sqrt{\sum_{j\leq 2} \lambda_j^2}$, and β and c are constants set manually.

[1]Ran Su is with School of Engineering and Information Technology, The University of New South Wales, Canberra, ACT 2600, Australia and CSIRO Mathematics, Informatics and Statistics, North Ryde, NSW 1670, Australia

[2]Changming Sun is with CSIRO Mathematics, Informatics and Statistics, North Ryde, NSW 1670, Australia

[3]Tuan D. Pham is with Aizu Research Cluster for Medical Engineering and Informatics, Research Center for Advanced Information Science and Technology, The University of Aizu, Aizu-Wakamatsu, Fukushima 965-8580, Japan

978-1-4577-1787-1/12/$26.00 ©2012 IEEE

However, the blob-like structures in dendritic spine images are suppressed too much if using this filter directly. The junctions where three or more dendrites meet, as well as extremely thick dendrites are sometimes suppressed, making these places hollow in the center. So we do not consider the blobness part of the filter. Using this improved filter, noise in I_l will be greatly reduced, and linear features will be enhanced, especially for the thin linear features.

To obtain a smooth and clean backbone, it is necessary to initially remove the spines along the dendrites. We use an attribute opening [10] operation on an inverted image and invert it back and use the attribute opening again to reduce the interference of small holes and small spines. Now we have obtained a much cleaner image with minimum noise, we can use thresholding to obtain a binary image. We use a thinning algorithm to obtain the skeleton and remove small branches of spines to trim the skeleton. Finally, we obtain the initial dendrite backbone. This is the start of our new backbone extraction method. Although there may be some small missing or broken parts on the backbone, this would not affect the final result.

From this initial backbone, we use an iterative process to refine the backbone extraction. Starting with the backbone extracted in the previous iteration, the operations we will discuss below will be repeated until a satisfactory backbone extraction result is obtained. First, all pixels on the initial backbone are traversed. At each pixel, its orientation using its neighbouring pixels is calculated. Then according to the local orientation of pixels along the backbone, smoothing based on directional morphological filtering [11] is applied to the dendrites. For each pixel p in the binary image which is used in the thinning algorithm, its closest backbone pixel's orientation is used as p's orientation. A discrete line is generated by a Bresenham line [12] in the center of p with p's orientation. Then p's intensity is replaced with the minimum value along this discrete line. This erosion operation along the main backbone direction will remove the attached spines which will affect the backbone extraction result. This makes the dendrites smoother. An improved Hessian filter is used on the eroded image. Then the backbone is extracted again using thinning and trimming.

However, some spines may be very close to each other, and they may become connected after the Hessian operation. After the above operations, they may appear as one backbone. Further operation is needed to disconnect this short backbone. The original image intensity along each line segment of the backbone is checked. The nearly touching spines have a gap between them, so there may be a sharp intensity change at this gap (Fig. 1). However, some dendrite regions may also have a similar intensity profile. The spines and dendrites which have similar intensity profile are differentiated using their shape information. The edges detected by a Canny edge detector [13] are different at these two different situations. The spines that are close to each other have two curves along the short backbone while the dendrite structure just has one linear structure (Fig. 2). The backbone will cross two edges when the two spines are close to each other but will not cross any edge for the dendrites. The backbone is disconnected if it has sharp intensity changes and crosses two edges. Through trimming, these disconnected backbone will be removed. After these operations, a backbone which is much better than the previously extracted one is obtained.

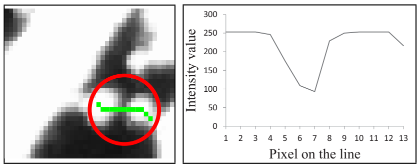

Fig. 1: The two spines which are close to each other. The green line is the backbone initially extracted and will be removed. The curve on the right represents the intensity profile along the green line.

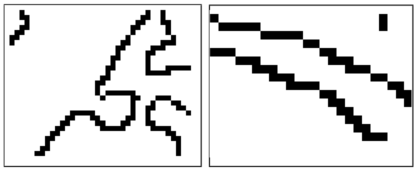

Fig. 2: Canny edges for spines and dendrites: the left is the Canny edges for the spines which are close to each other; the right is the edges for a dendrite which also has an intensity reduction along the structure.

Using the currently obtained backbone, these operations will be repeated: directional morphological filtering, improved Hessian filter, thinning, and trimming to obtain a better backbone extraction result. In each iteration, the backbone result is improved, so for the next iteration's morphological filtering, the dendrites becomes smoother. If one part of the backbone is lost in the initial extraction result, the morphological filtering and the improved Hessian filter will not be carried out on this part, and the intensities of this region will keep the intensities of the original image. Then this lost part will be detected in the backbone extraction iteration afterwards. Besides, this method will remove the wrongly detected backbones for spines which are too close to each other. The iterative process can be conducted until the backbone extraction result becomes stable. The iteration number can be set manually. In our experiment, two or three iterations are enough. We show the morphological filtering results in Fig. 3. The backbone extraction comparisons between the initial backbones and the results after our iterative backbone extraction are shown in Fig. 4. From these images, we can see that the backbones are smoother and lie in the center of the dendrites. Based on the finally extracted backbone, we use the directional line erosion again to obtain smooth dendrites, which will be used in the next step. We denote the smooth dendrites finally obtained with I_e.

III. BOUNDARY EXTRACTION AND SPINES DETECTION

In this step, we aim to extract the dendrite boundary and isolate the spines using the obtained boundary.

After the directional line erosion, most of the spines attached to the dendrites are removed from the dendrites.

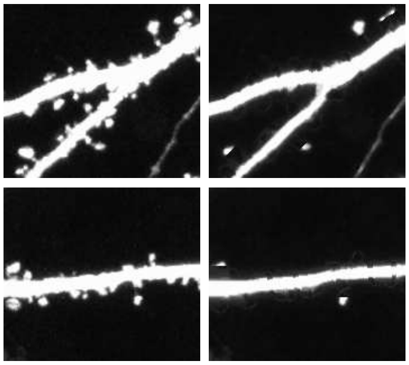

Fig. 3: Two examples showing the outputs after applying directional morphological filter.

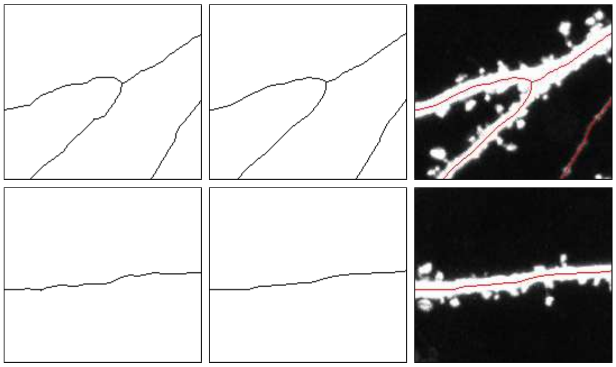

Fig. 4: Backbone extraction results and comparisons: The left column shows the initial backbone results, and the middle column shows the results after iterations. It can be seen that the iterated results are much smoother. The right column shows the final backbone extraction results overlaid on the original image.

Some spines may not be removed because the base of the spines is too wide. Besides, some dendrites may be eroded too much, making a small concave curve on the boundary. The dendrite boundary should be continuous and without those convex or concave regions. Therefore a shortest path approach is used here.

Gradient value is a good indicator for a boundary as the boundary is the place where bright and dark signals meet. We compute the gradient value of the eroded image I_e. In the gradient image, we take local regions along the orientation of the backbone to obtain the dendrite boundary. By doing so, the boundary extraction will only be carried out on pixels with dendrites and spines. To increase efficiency, we rotate the local region so that the backbone is roughly vertical. Next, we employ a shortest path boundary extraction method on one side of the backbone. Large bumps and concave regions will not be detected, but will be detected as a smooth boundary. Then we write the result back to the original image and obtain an image with dendrite boundary. We use the same method to obtain the boundary on the other side of the backbone. The examples for boundary extraction can be seen in Fig. 5.

After we have obtained the dendrite boundary, we smooth the boundary, check and connect broken parts. We set the boundary value to zero, then the spines in the original images are isolated from the dendrites. As the spines are now all isolated, we can use the attribute opening again to detect the spines.

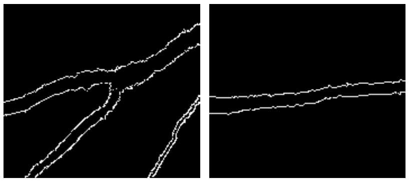

Fig. 5: Two examples of boundary extraction.

Algorithmic Procedure:

1) Preprocess the image to reduce noise.
2) Extract dendrites backbone:
   a) Apply Hessian filter and open-by-attribute operation.
   b) Extract the initial backbone using thinning and trimming algorithm.
   c) Use the iterative backbone extraction approach to obtain backbone.
   d) According to the final backbone result, smooth the dendrites again using directional morphological filter. The output will be used in the boundary extraction step.
3) Boundary extraction and spine detection:
   a) Subset and rotate small regions of the gradient images along the orientation of the backbone.
   b) Apply shortest path to detect the boundary of the dendrites.
   c) Obtain the boundary and isolate the spines from the dendrites.

IV. EXPERIMENTAL RESULTS

We tested our approach on real neuron images. We selected two examples from all the testing images and we show them in Fig. 6, Fig. 7. The detected spines are marked with red color. These images have complicated spines structures, and our method shows good detection results. Fig. 6 shows two subimages of the left image in Fig. 7. It can be seen that tiny spines can also be detected. Besides, there are no false positive spines detected on the thin dendrites which have no spines on it. From Fig. 7, we can see that the spines which are close to each other can also be detected. We ran our algorithm on a PC with an Intel Core i5 2.66GHz processor and 4GB RAM. It took about 95s to process the 1024×1024 image shown in Fig. 7. The parameters for our method can be empirically determined. For example, the attribute opening parameter can be decided by observing the largest blob or isolated spines.

Spine length, spine density or spine area can be estimated from the detected spines. For each spine, we measured its length using the Euclidean distance from dendrite boundary to the spine's farthest pixel from dendrites. The spines length distribution histogram can be seen in Fig. 8. From the graph, we can see that the spines have lengths between 0 to 31 pixels, and the spines' lengths have a larger distribution between 0 to 10 pixels. The comparison between manual visual detection and our method is shown in Table I. The statistics shows that our algorithm achieves a good result. The dendritic spine images have hundreds of spines, hence it will take a long time if bioscientists detect them manually. Our automatic spine detection approach will save time and labor for bioscientists.

TABLE I: Measurements of Dendrites and Spines

	Dendrite length (pixel)	Average spine length (pixel)	Spine density (pixel^{-1})
Manual	14343.16	10.23	0.0372
Our method	16580.73	9.12	0.0346

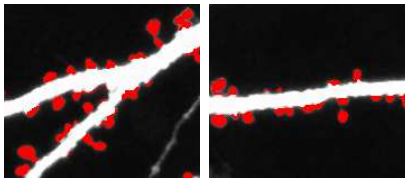

Fig. 6: Two subimages showing dendritic spines detection results.

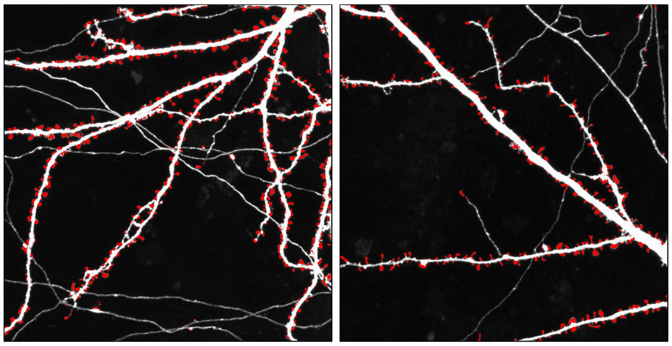

Fig. 7: Spines detection result: the red spots are the detected spines.

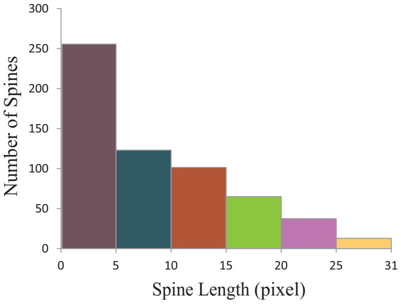

Fig. 8: Spine length distribution.

V. CONCLUSIONS

In this paper, we present a new approach which can be applied for dendritic spines detection. A novel approach for backbone extraction is proposed which iteratively refines the backbone extraction until a satisfactory result is obtained. It can give a smooth backbone extraction result, repair the broken parts in the backbone extraction process and also provide an isolation boundary for the spines detection. There are three main steps in our whole spines detection method: the first is to extract the dendrites backbone, the second is to detect the dendrite boundary, and the third step is to detect the spines. The method provides much information for researchers to analyze the spines, for example, the length, the density or the area. We will further our study in classifying the shapes of spines.

VI. ACKNOWLEDGEMENT

We would like to thank the anonymous reviewers for their helpful and constructive comments on this paper. We would also like to acknowledge and extend our gratitudes to the following persons: All the testing images were kindly provided by Prof. Valentin Nägerl, Neuroscience and Bioimaging, Avenir Group - Synaptic Plasticity and Superresolution Microscopy, Inserm U862/Universit Victor Segalen Bordeaux 2, France. This work was partly supported by China Scholarship Council and the University of New South Wales, Australia.

REFERENCES

[1] E. A. Nimchinsky, B. L. Sabatini, and K. Svoboda, "Structure and function of dendritic spines," *Annual Review of Physiology*, vol. 64, pp. 313–353, 2002.

[2] J. T. Trachtenberg, B. E. Chen, G. W. Knott, G. Feng, J. R. Sanes, E. Welker, and K. Svoboda, "Long-term *in vivo* imaging of experience-dependent synaptic plasticity in adult cortex," *Nature*, vol. 420, no. 6917, pp. 788–794, 2002.

[3] A. Herzog, G. Krell, B. Michaelis, J. Wang, W. Zuschratter, and K. Braun, "Three-dimensional quasi-binary image restoration for confocal microscopy and its application to dendritic trees," in *Proceedings of SPIE, Three-Dimensional Microscopy: Image Acquisition and Processing IV*, C. J. Cogswell, J. Conchello, and T. Wilson, Eds., pp. 146–157. April 1997.

[4] I. Y. Y. Koh, W. B. Lindquist, K. Zito, E. A. Nimchinsky, and K. Svoboda, "An image analysis algorithm for dendritic spines," *Neural Computation*, vol. 14, no. 6, pp. 1283–1310, June 2002.

[5] X. Xu, J. Cheng, R. M. Witt, B. L. Sabatini, and S. T. C. Wong, "A shape analysis method to detect dendritic spine in 3D optical microscopy image," in *Proceedings of the 3rd IEEE International Symposium on Biomedical Imaging*, Arlington, VA, USA, April 2006, pp. 554–557.

[6] A. Rodriguez, D. B. Ehlenberger, D. L. Dickstein, P. R. Hof, and S. L. Wearne, "Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images," *PLoS ONE*, vol. 3, no. 4, pp. e1997, April 2008.

[7] J. Cheng, X. Zhou, E. Miller, R. M. Witt, J. Zhu, B. L. Sabatini, and S. T. C. Wong, "A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy," *Journal of Neuroscience Methods*, vol. 165, no. 1, pp. 122–134, September 2007.

[8] F. Janoos, K. Mosaliganti, X. Xu, R. Machiraju, K. Huang, and S. T. C. Wong, "Robust 3D reconstruction and identification of dendritic spines from optical microscopy imaging," *Medical Image Analysis*, vol. 13, no. 1, pp. 167–179, February 2009.

[9] A. F. Frangi, W. J. Niessen, K. L. Vincken, and M. A. Viergever, "Multiscale vessel enhancement filtering," in *Proceedings of the 1st International Conference on Medical Image Computing and Computer-Assisted Intervention*, Cambridge MA, USA, 1998, vol. 1496, pp. 130–137.

[10] E. J. Breen and R. Jones, "Attribute openings, thinnings, and granulometries," *Computer Vision and Image Understanding*, vol. 64, no. 3, pp. 377–389, November 1996.

[11] P. Soille and H. Talbot, "Directional morphological filtering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 11, pp. 1313–1329, November 2001.

[12] J. E. Bresenham, "Algorithm for computer control of a digital plotter," *IBM Systems Journal*, vol. 4, no. 1, pp. 25–30, 1965.

[13] J. Canny, "A computational approach to edge detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 8, no. 6, pp. 679–698, November 1986.