

A novel method for dendritic spines detection based on directional morphological filter and shortest path



Ran Su^{a,b,*}, Changming Sun^a, Chao Zhang^{a,b}, Tuan D. Pham^c

^a CSIRO Computational Informatics, Locked Bag 17, North Ryde, NSW 1670, Australia

^b School of Engineering and Information Technology, The University of New South Wales, Canberra, ACT 2600, Australia

^c 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

ARTICLE INFO

Article history:

Received 4 December 2013

Received in revised form 18 June 2014

Accepted 28 July 2014

Keywords:

Dendritic spines

Dendrite

Backbone extraction

Boundary detection

Shortest path

ABSTRACT

Dendritic spines are tiny membranous protrusions from neuron's dendrites. They play a very important role in the nervous system. A number of mental diseases such as Alzheimer's disease and mental retardation are revealed to have close relations with spine morphologies or spine number changes. Spines have various shapes, and spine images are often not of good quality; hence it is very challenging to detect spines in neuron images. This paper presents a novel pipeline to detect dendritic spines in 2D maximum intensity projection (MIP) images and a new dendrite backbone extraction method is developed in the pipeline. The strategy for the backbone extraction approach is that it iteratively refines the extraction result based on 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. Touching spines will be split using a marker-controlled watershed algorithm. We present the results of our algorithm on real images and compare our algorithm with two other spine detection methods. The results show that the proposed approach can detect dendrites and spines more accurately. Measurements and classification of spines are also made in this paper.

© 2014 Elsevier Ltd. All rights reserved.

1. Introduction

Dendritic spines are small protrusions from neuron's dendrites. They have the function of receiving excitatory inputs and transmitting them to cell bodies. Normally, the dendrites of a single neuron can contain hundreds or thousands of spines. Spines typically have a large spine head, which connects to the dendrites via a membranous neck. According to their shapes, spines could be classified as stubby, mushroom or thin [1]. The variable spine shape and volume is thought to be correlated with the strength and maturity of each spine-synapse.

Studies in the past several decades have shown that the structural and electrical properties of dendritic spines are critical for local signal integration and molecular compartmentalization [2–4].

The motility, the changes of morphology and the number of spines have a great influence on brain function and development of human beings in motivation, learning, and memory [5,6]. For instance, schizophrenia and mental retardation are related to alterations of spine's morphology or density [7,8]; ischemia, trauma or epilepsy may all cause the increase in spines plasticity [9,10]; after a severe stroke, a rapid loss of spines and dendritic swelling is observed [4,11]. Analyzing the spines is helpful for diagnosing disease and developing drugs to treat or slow down these diseases.

Automatic detection of spines in images is very important, because it helps release biologists from the heavy burden of manual spines detection process. Through the analysis of dendritic spines images, spine shape and spine parameters are obtained, which can then be used for further biomedical research. In this paper, we propose a novel pipeline for the detection and measurement of dendrites and spines on the maximum intensity projection (MIP) images. 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

* Corresponding author. Current address: Bioinformatics Institute, Agency for Science and Technology Research, 30 Biopolis Street, 07-01 Matrix, Singapore 138671, Singapore. Tel.: +65 90563904.

E-mail addresses: suran.tju@gmail.com (R. Su), changming.sun@csiro.au (C. Sun), kone.zhang@gmail.com (C. Zhang), tdpham@u-aizu.ac.jp (T.D. Pham).

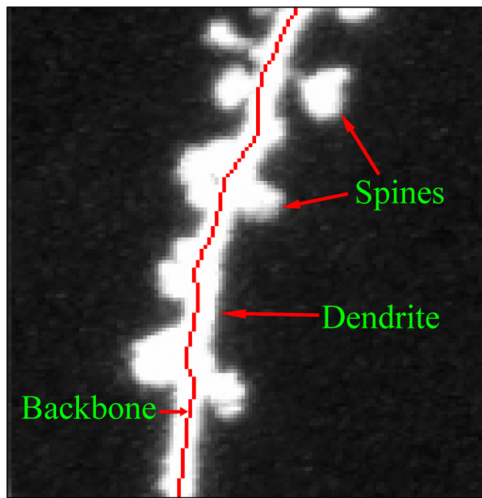


Fig. 1. An example showing spines, dendrites and backbone. This is the MIP image condensed from a 3D stack.

refine the backbone extraction results until satisfactory results are obtained. **The backbone extraction of the dendrites is a key step for spines detection as dendrites are the trunk where spines attach to.** However, backbone extraction is always affected by the huge number of small spines which usually disturb the backbone detection. Besides, a normal thinning algorithm is usually conducted on binary images, which depend heavily on thresholding algorithms. 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 any broken parts on the backbone. Moreover, touching spines can also be split through the proposed methods. Convincing results are obtained using our method. The spines, dendrites and backbone of dendrite are shown in Fig. 1. We will give a brief review of the existing spine detection methods in Section 2. The backbone extraction, boundary detection and spines extraction algorithms will be described in Sections 4 and 5. We show the experimental results in Section 6, followed by the discussion and conclusion of our work.

2. Related work

Great effort has been devoted to the automatic detection of spines in previous studies. In 1995, Watzel et al. proposed a spine detection approach which was based on the premise that only one dendrite was in the image [12]. Spines were extracted according to the medial axis spurs attached to the dendrites. This method cannot provide good results if multiple dendrites exist. Rusakov and Stewart described a method for quantification of spine length and distribution in [13]. Through binarization and skeletonization, lengths were computed through measuring skeletal branch lengths. Koh et al. detected spines based on their morphology [14]. The global thresholding method might lose many spines in the detection process. Weaver et al. further refined Koh et al.'s work in [15]. Xu et al. detected spines through two grassfire transforms to find the tips of spines and locate the boundary between spines and dendrites [16]. The adaptive local thresholding, voxel clustering and Rayburst sampling was used to analyze spines [17,18]. The Neuron-Studio is a software which can detect dendrites and spines [18,19]. A sampling core which can be used for the determination of the radial distance in each direction was generated through a Rayburst sampling algorithm. Zhou et al. reconstructed the dendrite with the representation of neuron surface [20]. In this method, the level set algorithm was used to segment dendrites. Then spines were

detected through a label-based thinning approach. Cheng et al. applied adaptive thresholding, SNR based method and morphology analysis to separate spines [21]. Attached spines and detached spines were detected separately. Bai et al. segmented the spines according to width-based criteria [22]. Spines were merged according to the distance and the orientation of spine head and base. They might not have a smooth backbone result by only using the thinning algorithm because spines in the binary image will affect the detection result. Fan et al. use a curvilinear structure detector to calculate the medial axis and maximum likelihood estimation was used to track spines [23]. Zhang et al. also used the curvilinear structure detector to extract the backbone and the boundary and used the linear discriminate analysis (LDA) classifier to further detect spines [24]. Janoos et al. reconstructed spines using surface representation and extracted skeletons based on the medial geodesic function [25]. Large spines were likely to be lost and spines shape was not well presented in these methods. They adopted another method which used local binary fitting (LBF) energy level set model with adaptive variances for the Gaussian kernel of each pixel to segment spines [26]. Lang et al. reconstructed neuron branches with detailed representation of spines from large-scale serial block-face scanning electron microscopy dataset [27]. Mukai et al. developed a novel method to extract spines based on their geometrical features and demonstrated the success of their method by analyzing effects elicited by androgen and estrogen on spinogenesis of hippocampal neurons [28].

3. Image acquisition

In our study, stimulated emission depletion (STED) microscope technique is used to image 3D dendritic spines of YFP-positive CA1 pyramidal neurons in living organotypic hippocampal slices, with spanning 140 μm in x , 140 μm in y , and 25–40 μm in z . There are 1024 \times 1024 pixels in xy and 0.5 μm step size in z to ensure that the significant parts of the dendrites are contained in the image stack. The images are acquired in the range of 0–10 μm above the surface of the coverslip without showing any decrease in resolution as may have been expected for STED imaging in optically dense tissue. The detailed information of the dataset can be found in [29]. Our experiments are conducted on 2D images condensed from 3D image stacks through a maximum intensity projection (MIP) process.

4. Dendrites backbone extraction

4.1. Preprocessing

To improve the extraction results, we first use a median filter with a small window size (usually 3 \times 3) to reduce noise. Then to deal with spines that are too close to each other, which often become connected after later 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 , which is obtained by**

$$I_l = I - \frac{I_g \cdot 255}{I_{g \max} - I_{g \min}} \quad (1)$$

where I_g is the gradient image; $I_{g \max}$ and $I_{g \min}$ are the maximum and minimum gradient values. We show the input image and I_l in Fig. 2. From the images, some structures that are almost touching become separated (as indicated by the red circles in the figure), because **this operation will reduce the effect of structures based on their directional change of magnitude in the intensity. The boundary has a larger gradient magnitude so its intensity values will be reduced**

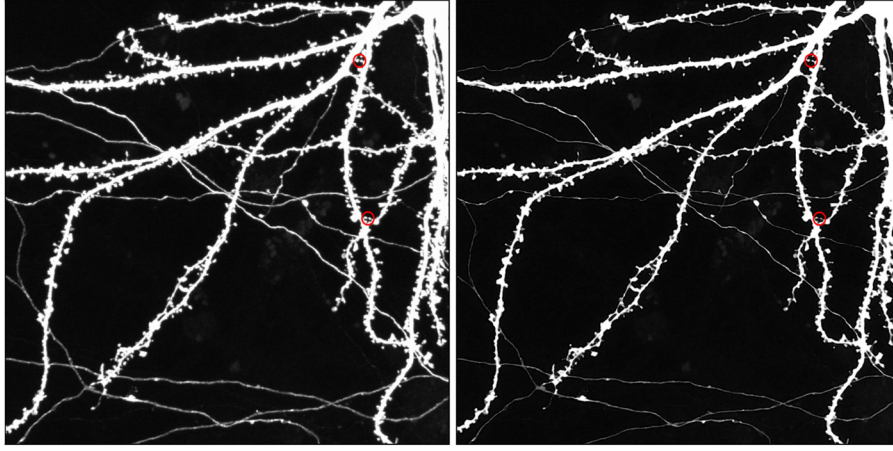


Fig. 2. One example showing I_l . The left image is the input image. The right image is the result after preprocessing. The spines in the red circles in I_l are more separable than that in the input image. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

more. Therefore, the touching spines become separable after this operation.

4.2. Hessian enhancement

Frangi et al. present a multi-scale approach for the linear feature enhancement which is based on the eigenvalues of the Hessian matrix [30]. 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 filter used to enhance the linear structures is given by

$$H(s) = \begin{cases} 0 & \text{if } \lambda_2 > 0 \\ \exp\left(-\frac{R^2}{2\beta^2}\right) \left(1 - \exp\left(-\frac{S^2}{2c^2}\right)\right) & \text{otherwise} \end{cases} \quad (2)$$

where $R = |\lambda_1|/|\lambda_2|$, which is a blobness measure, $S = \sqrt{\sum_{j=2}^n \lambda_j^2}$, and β and c are constants set manually. The scale s is used to fit with linear features with different widths by multiplying the Hessian matrix with s^2 before eigenvalue decomposition. s is set within a large range, which may cause thick structures suppressed, making them hollow in the center. Junctions where three or more branches intersect are also suppressed because junctions are thick and they are the locations where multiple branches meet [31]. In our proposed algorithm, two operations were conducted to solve the suppression problem.

Firstly, we do not consider the blobness measurement in $H(s)$. Because this measurement will suppress the blobs as well as the junctions. Therefore, we use $H(s) = 1 - \exp(-S^2/2c^2)$ instead of the one in Eq. (2). The output after the improved Hessian filter is denoted as I_{e1} . We give two examples of the improved Hessian filter in Fig. 3. From the figure, it shows that the new measurement can better enhance and preserve the structures. Secondly, in our previous work [32,33], the junctions are enhanced separately from linear features using Gabor functions or oriented anisotropic Gaussian functions. Here we also enhance junctions in a separate step with the Hessian method. We use the junction detection method we have developed to detect the junction center points and branch orientations [34]. It is based on Hessian information, correlation matrix and Gaussian template fitting. Candidate points are chosen according to a new measurement which combine the information of both Hessian information and a correlation matrix. Then the junction center points and branches are detected through a stick rotation around the possible candidate points. Then Gaussian template fitting is used to further determine the center points. After this, the junction regions in I_{e1} are extracted and divided into

n parts (n is the number of branches, with each part containing one branch). The dividing lines are angular bisectors between two neighboring branches. We enhance each part of the junction region using the improved Hessian filter along each branch separately. The

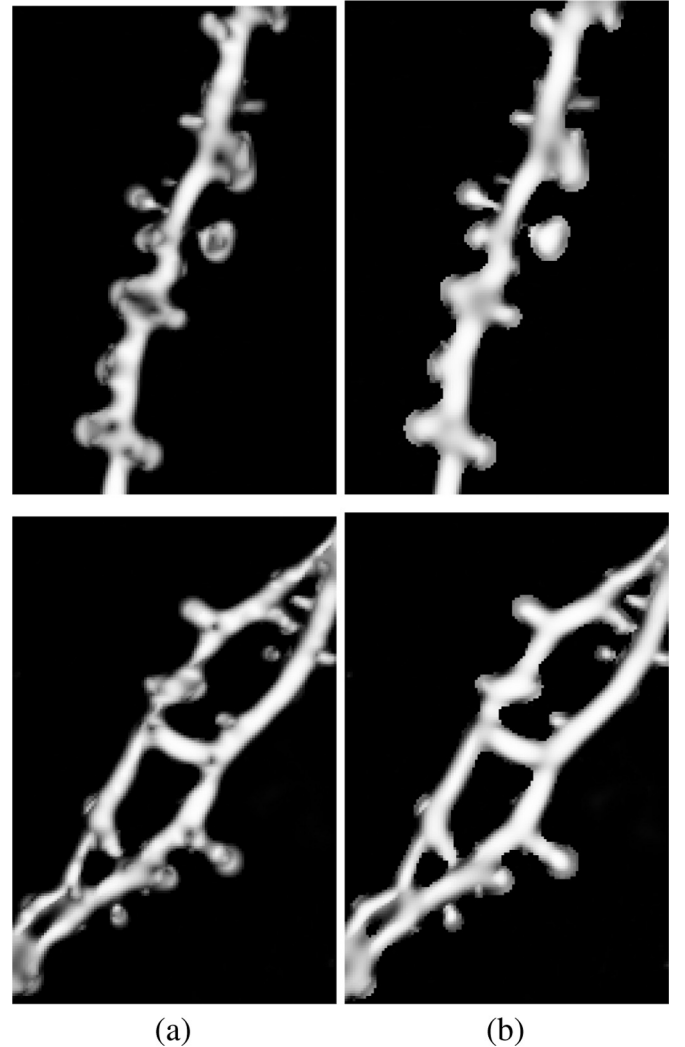


Fig. 3. Two examples showing the improved Hessian measurements. (a) The output using Frangi et al.'s filter; (b) the output using the improved filter.

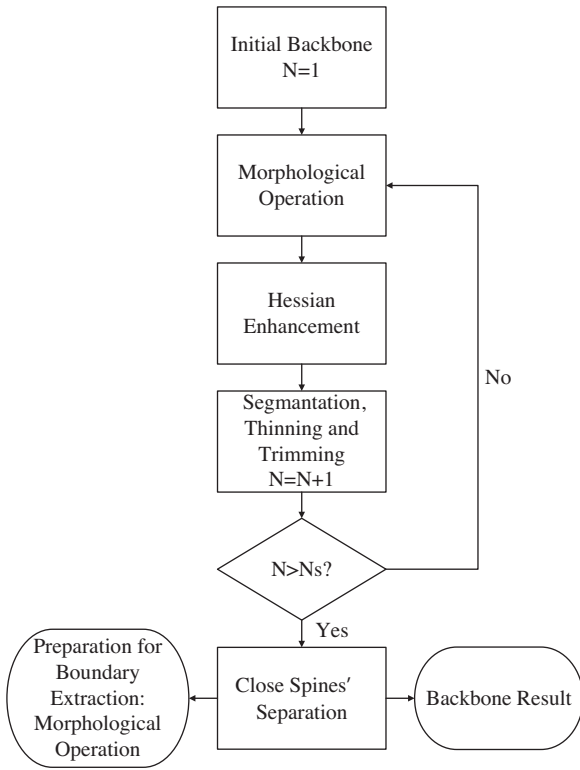


Fig. 4. The flow chart that shows the iterative backbone extraction method. N_s is the number of iterations.

branches are recombined after enhancement. Therefore, there will be no junction suppression problem, because branches of junctions are all enhanced as separate linear structures. For preprocessing purpose, we enhanced I_l and denote the output with I_{e2} . I_{e2} will be used as the input for the backbone extraction step. This improved Hessian enhancement filter will also be used in the next step, which is the iterative backbone extraction.

4.3. The novel backbone extraction procedure

The medial axis or skeleton is a compact representation of one shape while maintaining its topology [35]. Thinning algorithms have been widely used in skeletonization [36–38]. Distance transform is also an important method for skeleton detection [39,40]. Due to the disturbance of spines, it is not reasonable to use existing skeletonization approaches. Therefore another effective method is developed. The basic idea for the skeletonization method is to iteratively smooth the dendrite and obtain better skeleton extraction results. We provide an overview in Fig. 4 to show the novel iterative backbone extraction method and close-by spines separation method.

4.3.1. Initial backbone extraction

To minimize the interference of spines when detecting the initial backbone, it is necessary to remove the spines from the dendrites. White top-hat transform is used here to detect the small detached objects and black top-hat transform is used to detect any tiny holes in the images. Subtracting the original image with the small objects and holes can produce a cleaner image. The top-hat transform is defined as the following [41]

$$\begin{cases} T_w(I) = I - I \circ b & \text{white top-hat} \\ T_b(I) = I \bullet b - I & \text{black top-hat} \end{cases} \quad (3)$$

where $b(x)$ is the structuring element; \circ denotes the opening operation and \bullet denotes the closing operation. Here we use the attribute opening and closing to conduct the top-hat transform [42].

After enhancement and spine removal, now the output has been much cleaner compared with the input. We carry out a global thresholding segmentation to obtain a binary image. The binary image is denoted with I_b . We perform a skeletonization through a pseudo-Euclidian morphological thinning of the input [43]. The algorithm uses a good approximation of the Euclidean distance to achieve a higher efficiency. The thinning performs an erosion but without the change of object's topology. Next, the backbone is trimmed through removing all end pixels and then reconstructing chains from any remaining end pixels. Then the initial dendrite backbone is obtained. This is the start of our novel backbone extraction method. Although there may be some small missing or broken parts on the backbone, this would be corrected in the following steps.

4.3.2. Iterative backbone extraction

Based on the initial skeleton, the novel backbone extraction method is applied on the input image. This is a continuous refinement procedure. In each iteration, a better result compared with the last iteration will be obtained. 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 backbone are traversed. At each pixel on the backbone, its direction is calculated using its neighboring pixels. Therefore, a map showing the orientation of the backbone is obtained. Directional morphological erosion with a line structuring element [44] is applied to foreground pixels in I_b (assuming the foreground is the region containing spines and dendrites). For each pixel p in I_b (pixels on the skeleton will not be calculated), its closest backbone pixel's orientation is used as p 's orientation. The line structuring element $b(x)$ is centered at p . Along $b(x)$, p 's intensity is replaced by the minimum value along this discrete line. This erosion operation along the main backbone direction can remove the attached spines and make the dendrites smoother. We illustrate the algorithm procedure in Fig. 5. Two examples of the directional morphological erosion are shown in Fig. 6. The improved Hessian filter with junction enhancement is used on the eroded image followed with thresholding, thinning and trimming. The thinning and trimming have been described above. The iteration loop includes enhancement, thinning, trimming, and directional erosion. The number of iterations can be set empirically. We use a number N_s to control the number of refinement. If the number of iterations N is larger than N_s , the iterative process stops. N_s is set to 3 in all our experiments. Examples of the proposed backbone extraction method can be seen in Fig. 7. In this example, we compare the initial backbone extraction result, which is also the result using the normal thinning algorithm, with the result after two iterations. The results after two iterations are smoother than the initial result. It is because the thinning algorithm is greatly affected by the smoothness of the dendrites. If the dendrite has a rough surface or has many spines attached to it, the extracted backbone will be rather tortuous and has many spurs. Our algorithm will not be disturbed by the roughness and the spines in the way that we update and improve the backbone extraction in each iteration, which provides a better direction computation for morphological erosion. It works as a loop: better backbone extraction–smoother dendrite–better backbone extraction–smoother dendrite, etc.

4.3.3. Touching spine disconnection

However, some spines which lie head to head may be very close to each other, and they may become connected and appear as one backbone after the Hessian operation. Further operation is needed

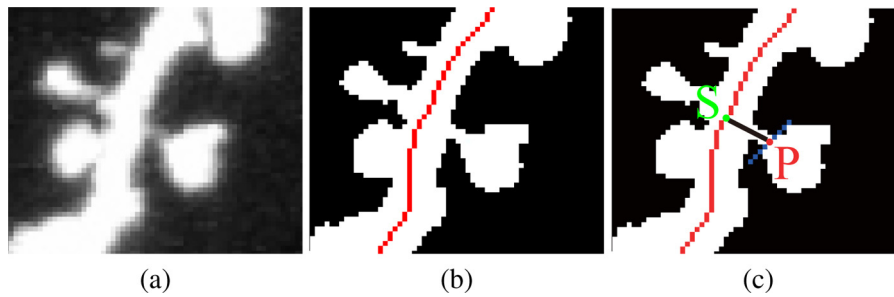


Fig. 5. The morphological erosion to smooth the dendrite surface. (a) the original input; (b) I_b and the skeleton; In (c), p is one pixel in I_b ; The skeleton pixel S , which has the shortest distance to p , is marked with green color; The blue pixels show the structure element, b_x , which orients in the direction of the green point. Here we use nine pixel length for b_x . The erosion is conducted on the blue structure and p . p 's intensity value will be replaced by the minimum along b_x . (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

to disconnect this short backbone. We check the intensity profile of I_l along each short skeleton. For a wrongly connected short skeleton, it starts and ends at pixels with three or four neighbor pixels because it consists of two small spines attached to the dendrites.

The nearly touching spines have a gap between them, so there should be a sharp intensity change at this gap (Fig. 8). However, due to the uneven distribution of intensity, 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 on I_l by a Canny edge detector [45] 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. The backbone will cross two edges when the two spines are close to each other but will not cross any edge for the dendrites (Fig. 9). The short backbone is disconnected if it has a sharp intensity change and crosses two edges when checking their intensity profile. Through

trimming, these disconnected backbones will be removed. After these operations, a smooth and clean backbone is obtained. We conduct the morphological erosion again to obtain a smooth surface of the dendrites, denoted by I_e , which will be used in the boundary extraction step.

5. Boundary extraction and spines detection

5.1. Spines separation

The dendrite boundary and spines are extracted in this step. In the last section, after extracting the backbone, morphological line erosion is conducted again for a smooth dendrite surface. Therefore, on the dendrite boundary, most of the spines attached to the dendrites are removed from the dendrites. If the boundary of the dendrites is obtained and set to zero, the spines will become isolated and thus be extracted and measured. A

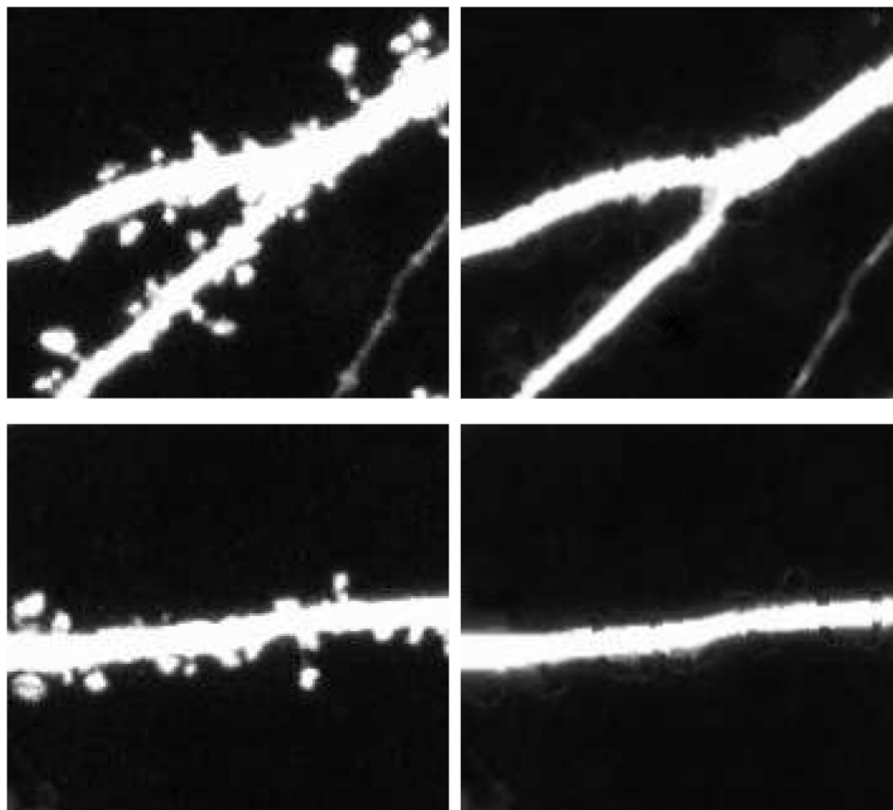


Fig. 6. Two examples showing the outputs after applying directional morphological filter. The left column is the input image and the right column is the result after applying the filter.

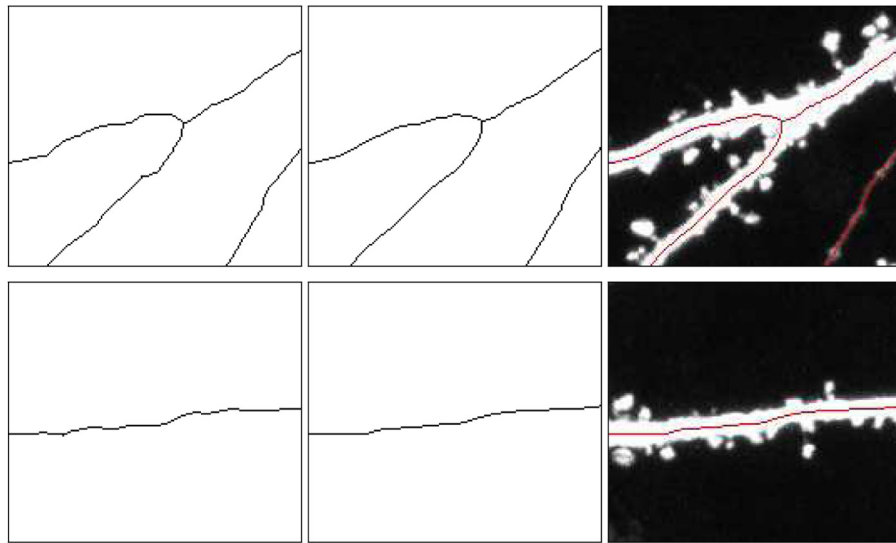


Fig. 7. Backbone extraction results and comparisons: the left column shows the initial backbone results, and the middle column shows the results after two iterations. It can be seen that the iterated results are much smoother. The right column shows the final backbone extraction results overlaid on the original image.

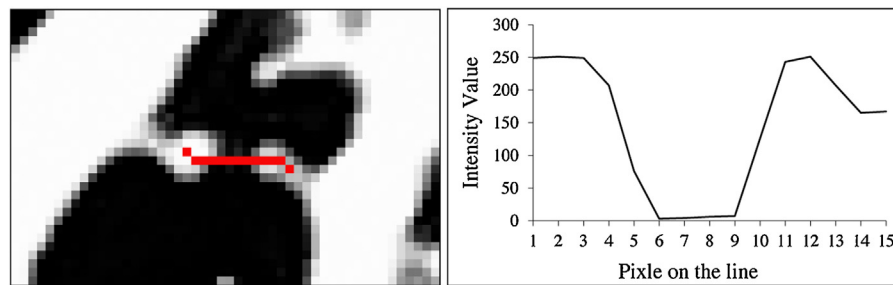


Fig. 8. The two spines which are close to each other. The red line is the backbone initially extracted and will be removed. The curve on the right represents the intensity profile along the red line. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

shortest path algorithm has been used for boundary detection in a number of publications [46,47]. Here we use a shortest path approach in a local window to extract the boundary of the dendrites.

Image gradient value is a good indicator for a boundary where bright and dark signals meet. We compute the gradient value of the image I_e . In the gradient image, we take a local window along the orientation of the backbone and extract the boundary in each local window. By doing so, 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. The rotation may bring some inaccuracy for low resolution images, but high resolution image will not be affected. Thus the rotation step will not be performed in low resolution images. Then

we employ a shortest path boundary extraction method on one side of the backbone. The extracted boundary is smooth as spines are already removed. After the backbone is traversed, we obtain an image with a dendrite boundary. We use the same method to obtain the boundary on the other side of the backbone. Examples for boundary extraction can be seen in Fig. 10.

After we have obtained the dendrite boundary, we smooth it, check and connect any broken parts. We overlay the detected boundary onto the original image. Therefore, the spines in the original images are isolated from the dendrites.

5.2. Connected spine splitting

Another problem may arise after the spines are segmented from the dendrites. Some spines which are next to each other along the dendrites may be connected after segmentation (Fig. 11(a)). We split the touching spines using a watershed algorithm. Watershed segmentation is widely used in round-object splitting [48–51]. Here we employ the marker-control watershed to conduct the segmentation [52], which can better handle the over-segmentation problem. The seeds are points with the largest distance to a local boundary (spines are foregrounds in our algorithm). The watershed segmentation are not applied to long (by the criteria of distance to boundary) spines with a large head and a long neck, as spines with this shape are similar to two touching blobs, which show a cutting line between the head and neck. We show examples of segmentation of touching spines in Fig. 11(b).

The algorithm steps for the whole procedure are:

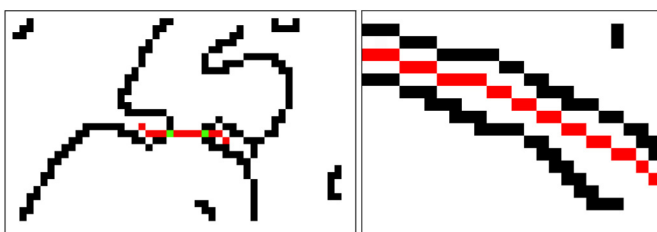


Fig. 9. Canny edges for spines and dendrites: the left is the Canny edges for the spines which are close to each other; the right is the edges for a dendrite which also has an intensity reduction along the structure. The backbone crosses two edges in the left image but the backbone does not cross any edges in the right image.

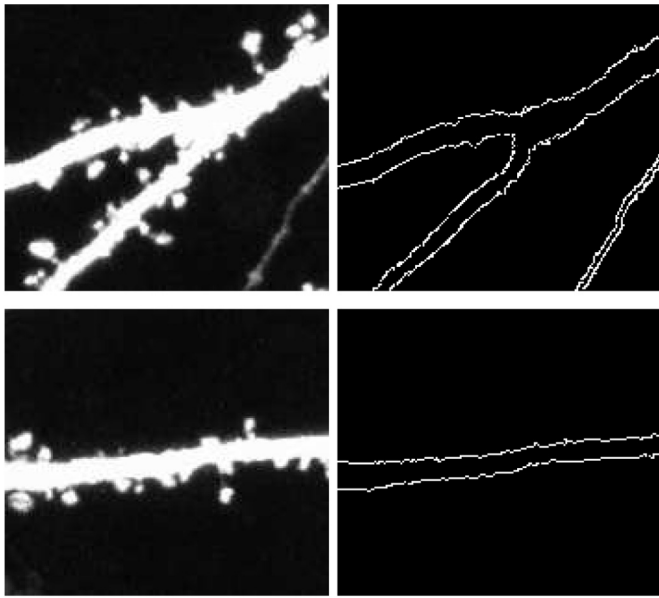


Fig. 10. Two examples of dendrite boundary extraction. The left column is the input image and the right column is the detected boundary.

- Dendrites backbone extraction:
 1. Preprocess to remove or reduce noise;
 2. Enhance image using Hessian information;
 3. Use the iterative backbone extraction approach to obtain backbones;
 4. According to the final backbone result, smooth the dendrites again using directional morphological filter. The output will be used in the boundary extraction step.
- Boundary extraction and spines detection:
 1. Extract and rotate small regions of the gradient images along the orientation of the backbone;

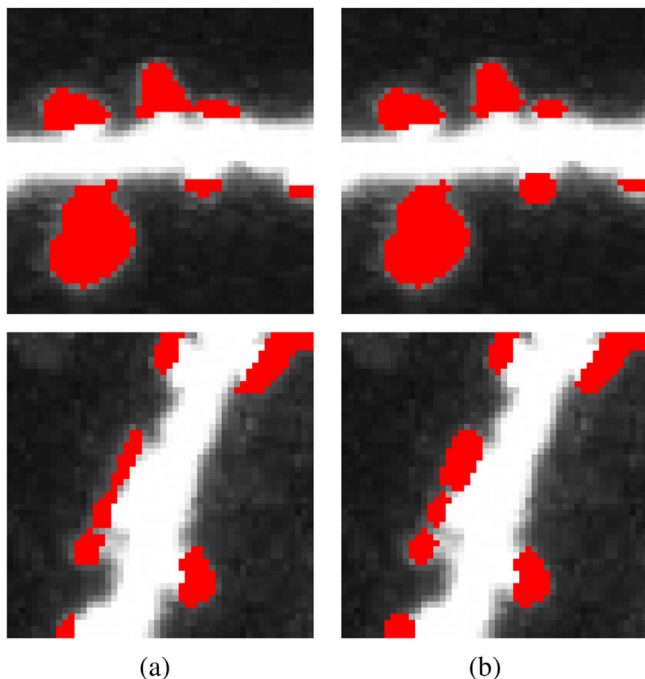


Fig. 11. The segmentation of touching spines. (a) The output after spines are separated from boundary; (b) the result after watershed segmentation.

2. Apply shortest path technique to detect the boundary of the dendrites;
3. Obtain the boundary and isolate the spines from the dendrites;
4. Split touching spines.

6. Experimental results

6.1. Performance of the proposed scheme

We evaluate the performance of the proposed method on twenty images which include a huge number of spines. We show some results in Fig. 12. The detected spines are marked with red color. These images have complicated spines structures, and our method shows good detection results. It can be seen that even tiny spines can also be detected and the number of missing spines is small in our method. The results show that spines close to each other can also be detected. We run our algorithm on a PC with an Intel Core i5 2.66 GHz processor and 4 GB RAM. It takes about 95 s to process the 1024×1024 pixel image shown in Fig. 12.

Dendrite length, spine length, spine density and spine area can be obtained from the detected spines. We make measurements of all the spines and their corresponding dendrites. We use a method proposed by Cheng et al. to measure the dendrite length [21]. In their method, n_0 is the total number of pixels of dendrite backbones. An (8-4) neighbor is defined as one pixel which is in the 8 neighborhood, but not in the 4-neighborhood. n_1 is the number

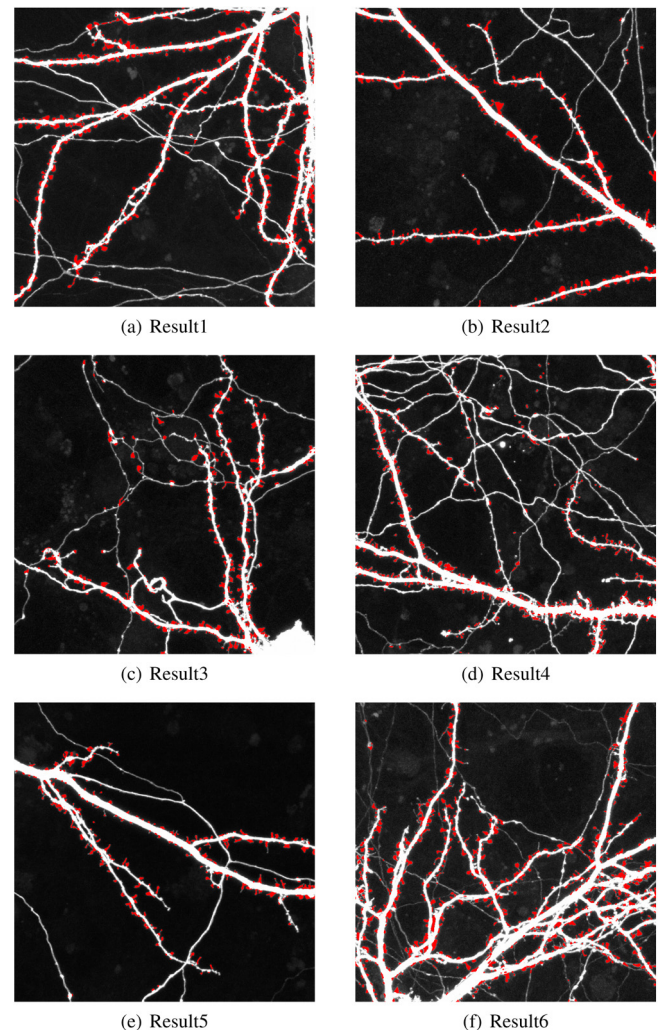


Fig. 12. Six examples showing the dendritic spines detection results.

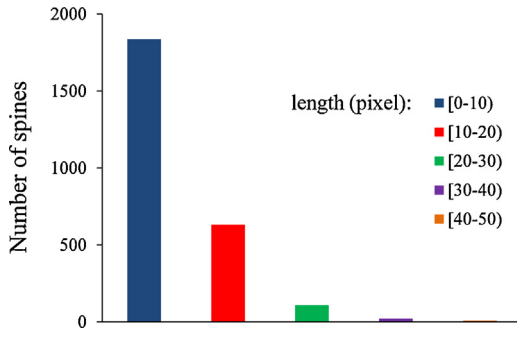


Fig. 13. The distribution of spine length.

Table 1

Measurements of dendrite backbone length, spine length and spine density.

	Dendrite length (pixel)	Average spine length (pixel)	Spine density (pixel ⁻¹)
Manual	40,116.11	9.17	0.060
Our method	42,874.07	8.79	0.064

Table 2

Classification criteria for spines: d_n is the spine neck diameter; d_h is the spine head diameter and L is the spine's total length.

Spine type	Stubby	Mushroom	Thin
Criteria	$d_n \approx L$	$d_h \gg d_n$, L should not be too large	$L \gg d_n$, d_h should not exceed d_n too much

of pixels which have one (8–4) neighbor. n_2 is the number of pixels which have two (8–4) neighbors. The length of dendrites can be calculated using $l = (n_0 - 1) + ((\sqrt{2} - 1)(n_1 + 2n_2))/2$. For each spine, we measure its length using the Euclidean distance from the dendrite boundary to the spine's farthest pixel from dendrites. The spines length distribution histogram can be seen in Fig. 13. From the graph, we can see that the spines have lengths from 0 to 55 pixels, and the spines' lengths have a larger distribution between 0 and 10 pixels. The comparison between manual detection and our method is shown in Table 1. In the table, spine density is computed using the following equation:

$$\text{Spine density} = \frac{\text{number of spines}}{\text{dendrite backbone length}} \quad (4)$$

We also calculated the average error values, 0.093, 0.050 and 0.081, for the dendrite backbone length, spine length and spine density compared with manual method. All the statistics shows that our algorithm achieves a good result which is in agreement with manual measurements. The dendritic spine images have a huge number 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 bioscience researchers.

Spines can also be classified according to its neck diameter (d_n), head diameter (d_h), and its total length (L). We classify the spines into three categories: stubby, mushroom, and thin according to [1]. The classification criteria is shown in Table 2. The specific numerical criteria we use is from [14]. We show some of the classification results in Fig. 14. As there is a continuum of shapes among the three categories of spines, it is difficult to establish a clear boundary between the three classes. Therefore, we compared the automated classification results with the manual results. They were produced from the examination of two skilled operators, who counted the number of spines in each class on the MIP images. The statistical results of classification is shown in Table 3. From the table, we can see that stubby types accounts for the largest part of the spines in our dataset.

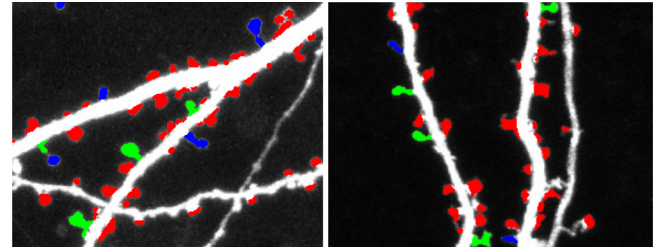


Fig. 14. Spine classification results: the red color shows the stubby ones; the green color represents the mushroom; and the blue color shows the thin spines. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Table 3

Spine classification statistical results. We also show the manual classification results from operator 1 and operator 2. From the table, it shows that the spines with stubby shape make up the vast majority of all the spines in our dataset.

Spine type	Stubby (%)	Thin (%)	Mushroom (%)
Automated	81	12	7
Operator 1	78	14	8
Operator 2	79	14	7

6.2. Comparison with other methods

Here we compare the proposed method with two other methods, Zhang et al.'s method [24] and the method from the NeuronStudio software [18,19]. Zhang et al. detect backbones and spines by calculating the second-order direction derivatives and then build a classifier to remove pseudo spines. NeuronStudio's backbone extraction is a semi-automatic method, which needs manual selection of seeds. The spines are detected by the means of voxel clustering. We randomly select five images which include approximately 1900 spines from the dataset and test the accuracy of each methods. We show six subimages of results using each approach here. It can be seen that Zhang et al.'s method loses more spines than the proposed method does and the NeuronStudio method sometimes detects one spine as two, breaking up from the neck. Our algorithm achieves better results (Fig. 15).

We also give statistical analysis of the comparison. Visual detection result is used as the ground truth. *Precision* and *Recall* are used here to evaluate the accuracy of different methods [53]. *Precision* can be seen as a measure of exactness or fidelity, whereas *Recall* is a measure of completeness. We use tp (true positives) to indicate the number of true spines detected; fn (false negatives) represents the number of missed spines; fp (false positives) is the number of falsely detected spines and $tp + fn$ should be equal to the spines which are detected manually. The definitions of these measures are as follows

$$\text{Precision} = \frac{tp}{tp + fp} \quad (5)$$

$$\text{Recall} = \frac{tp}{tp + fn} \quad (6)$$

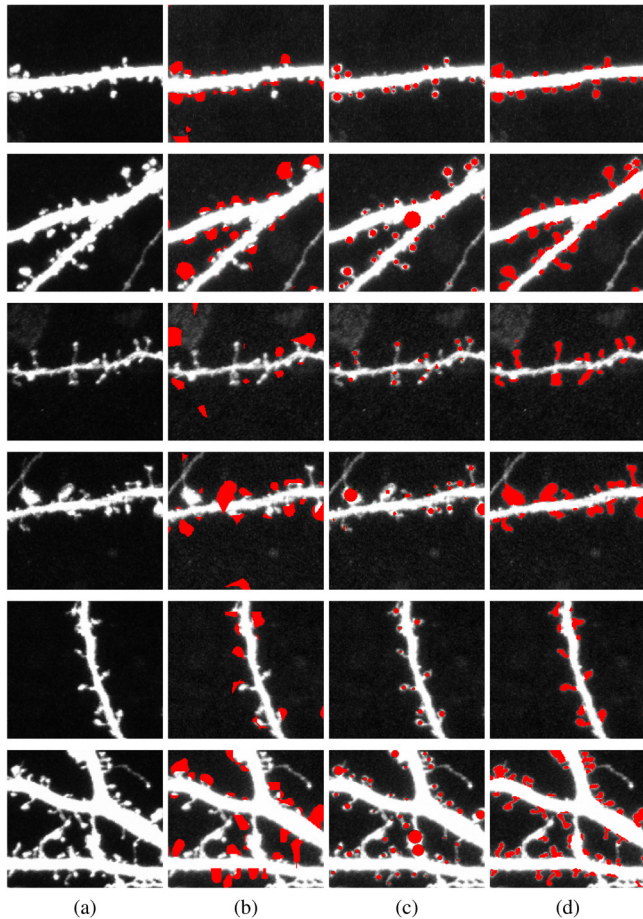
$$\text{Accuracy} = \frac{\text{Precision} + \text{Recall}}{2} \quad (7)$$

The overall true positive, false positive, false negative, average *Precision*, average *Recall* and average *Accuracy* values are listed in Table 4. From the table, it shows that the values for these two measurements for the proposed method are higher than that of the other two methods for the 1900 spines. The proposed method has an average *Accuracy* of 94.09%, which is higher than 83.44% for Zhang et al.'s method and 91.78% for the NeuronStudio method. Zhang et al.'s accuracy is lower than that of the proposed method and the NeuronStudio method. The NeuronStudio's approach

Table 4

Statistical comparison among three methods: Zhang et al.'s method, NeuronStudio method and the proposed method.

	True positive	False positive	False negative	Precision	Recall	Accuracy
Zhang et al.'s approach	1494	211	391	87.62	79.25	83.44
NeuronStudio approach	1725	154	160	91.80	91.75	91.78
The proposed method	1772	110	113	94.16	94.01	94.09

**Fig. 15.** The comparison between the proposed method and other methods: (a) the input subimages; (b) the results using Zhang et al.'s method; (c) the results processed with NeuronStudio; and (d) the results using the proposed method.

cannot give an exact shape of the spines, but only provide the size of the spine head. Besides, its backbone extraction result heavily depends on seeds selection. In order to extract all the skeletons in the images, sometimes three or more different seeds are required.

We use the default parameterizations for both Zhang et al.'s method and NeuronStudio method, for example, in Zhang et al.'s method, parameter 'Micron/Pixel' (ratio between micron and pixel) is set to 0.084 and 'Min Spine Length' (the minimal length of a spine to be counted) is set to 0.1 micron; For the NeuronStudio approach, minimum height for spine is set to 0.2 μm , maximal height is set to 3.0 μm . The parameters for our method can be empirically determined. For example, all the attribute opening parameters can be decided by observing the largest blob or isolated spines. The multiple scales for the Hessian filter are fixed constants: $\sqrt{2}/2$, $\sqrt{2}$, $3\sqrt{2}/2$ and $2\sqrt{2}$. The morphological erosion structure element in all our experiments is set as 7 pixels and does not need to be changed. The length of spines for determining the touching spines that will be used during watershed segmentation is automatically set through the computation of the average spine length.

7. Discussion

We build a novel pipeline to automatically detect and characterize the dendritic spines. Twenty images including a huge number spines were tested. Our results show that the spines have various lengths ranging from one pixel to more than fifty pixels. The measurements obtained from the proposed automated method are quite close to the manual measurement. In our dataset, the axons have similar structures as dendrites and sometimes they are detected as dendrite backbone and will disturb the measurement. We use the ratio between the number of spines and the backbone length to judge if it is an axon or dendrite. **If the ratio is small, it is considered as an axon.** We classified the spines into three groups in order to provide a more intuitive and geometrical view of the spines. We compare our method with two existing methods, with around 10.65% better than Zhang et al.'s method and 2.3% better than the NeuronStudio method in accuracy. Although the improvement over NeuronStudio's is not as significant as Zhang et al.'s, our method is better in the sense that it is a fully automated approach while NeuronStudio is a semi-automated method. This means no human intervention is required for our method but the NeuronStudio method needs the manual seed selection at the beginning of processing. Biologists have to choose one seed, observe the detection output and change or add seed points according to the output. Most of the time, in order to track all the skeletons in one image, more than three seeds are needed. However, **our method could do the segmentation automatically without the intervention of human beings.**

Our algorithm was tested on MIP images. It is possible to extend it to 3D images. In the backbone extraction step, the directional erosion, Hessian enhancement, thinning, and trimming can be calculated in 3D space. Three eigenvalues of the Hessian matrix will be used and one more measurement which distinguishes between plate-like and line-like structures is added. However, the proposed method is time consuming on 3D images since our algorithm is based on an iterative backbone extraction method. The thinning and Hessian filtering are algorithms that require repetitive work. If the iterative number is set to a large value, it may take a long time to process the data. In the future, we will focus on improving the speed of the algorithm and extend it to 3D images.

8. Conclusions

In this paper, we present a new pipeline for dendritic spines detection. There are three main steps in our whole spine detection method: **the first is to extract the dendrite backbones, the second is to detect the dendrite boundary, and the third step is to segment the spines.** 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 any broken parts in the backbone extraction process and also provide an isolated boundary for spines detection. Moreover, touching spines can be segmented through the proposed method. We measure the lengths for dendrites and spines and depict the length distribution in a graph. The classification of spines is also made in our work according to their head, neck length and total length. We compare our detection algorithm with two other methods, Zhang et al.'s approach and NeuronStudio's method.

The obtained images and graphs show that our method can achieve more accurate detection results. The proposed automatic detection method gives biologists statistical analysis of spines, which can be used as the diagnosis of some diseases.

Acknowledgements

We would like to acknowledge and extend our gratitudes to the following persons: Xiao Tan at the University of New South Wales, Canberra, Australia, and Katarina Mele of CSIRO for their discussions and comments on this paper; Prof. Valentin Nägerl, Neuroscience and Bioimaging, Avenir Group – Synaptic Plasticity and Superresolution Microscopy, Inserm U862/Universit Victor Segalen Bordeaux 2, France, for providing all the test images. We thank the anonymous reviewers for their helpful and constructive comments on this paper. The preliminary conference version of this paper was presented in [54].

References

- [1] Harris KM, Jensen FE, Tsao B. Three-dimensional structure of dendritic spines and synapses in rat hippocampus (CA1) at postnatal day 15 and adult ages: implications for the maturation of synaptic physiology and long-term potentiation. *J Neurosci* 1992;12(7):2685–705.
- [2] Yuste R, Bonhoeffer T. Morphological changes in dendritic spines associated with long-term synaptic plasticity. *Annu Rev Neurosci* 2001;24:1171–89.
- [3] Alvarez VA, Sabatini BL. Anatomical and physiological plasticity of dendritic spines. *Annu Rev Neurosci* 2007;30:79–97.
- [4] Bhatt DH, Zhang S, Gan WB. Dendritic spine dynamics. *Annu Rev Physiol* 2009;71:261–82.
- [5] Kasai H, Fukuda M, Watanabe S, Hayashi-Takagi A, Noguchi J. Structural dynamics of dendritic spines in memory and cognition. *Trends Neurosci* 2010;33(3):121–9.
- [6] Roberts TF, Tschida1 KA, Klein ME, Mooney R. Rapid spine stabilization and synaptic enhancement at the onset of behavioural learning. *Nature* 2010;463(7283):948–52.
- [7] Nimchinsky EA, Sabatini BL, Svoboda K. Structure and function of dendritic spines. *Annu Rev Physiol* 2002;64:313–53.
- [8] Trachtenberg JT, Chen BE, Knott GW, Feng G, Sanes JR, Welker E, et al. Long-term in vivo imaging of experience-dependent synaptic plasticity in adult cortex. *Nature* 2002;420(6917):788–94.
- [9] Kolb B, Gibb R. Possible anatomical basis of recovery of function after neonatal frontal lesions in rats. *Behav Neurosci* 1993;7(9):799–811.
- [10] Hasbani MJ, Schlieff ML, Fisher DA, Goldberg MP. Dendritic spines lost during glutamate receptor activation reemerge at original sites of synaptic contact. *J Neurosci* 2001;21(7):2393–403.
- [11] Zhang S, Boyd J, Delaney K, Murphy TH. Rapid reversible changes in dendritic spine structure in vivo gated by the degree of ischemia. *J Neurosci* 2005;25(22):5333–8.
- [12] Watzel R, Braun K, Hess A, Scheich H, Zuschratter W. Detection of dendritic spines in 3-dimensional images. In: *Deutsche Arbeitsgemeinschaft für Mustererkennung symposium*. 1995. p. 160–7.
- [13] Rusakov DA, Stewart MG. Quantification of dendritic spine populations using image analysis and a tilting disector. *J Neurosci Methods* 1995;60(1–2):11–21.
- [14] Koh IYY, Lindquist WB, Zito K, Nimchinsky EA, Svoboda K. An image analysis algorithm for dendritic spines. *Neural Comput* 2002;14(6):1283–310.
- [15] Weaver CM, Hof PR, Wearne SL, Lindquist WB. Automated algorithms for multiscale morphometry of neuronal dendrites. *Neural Comput* 2004;16(7):1353–83.
- [16] Xu X, Cheng J, Witt RM, Sabatini BL, Wong STC. A shape analysis method to detect dendritic spine in 3D optical microscopy image. In: *Proceedings of the 3rd IEEE international symposium on biomedical imaging*. 2006. p. 554–7.
- [17] Rodriguez A, Ehlenberger DB, Hof PR, Wearne SL. Rayburst sampling, an algorithm for automated three-dimensional shape analysis from laser scanning microscopy images. *Nat Protoc* 2006;1(4):2152–61.
- [18] Rodriguez A, Ehlenberger DB, Dickstein DL, Hof PR, Wearne SL. Automated three-dimensional detection and shape classification of dendritic spines from fluorescence microscopy images. *PLoS ONE* 2008;3(4):e1997.
- [19] Wearne SL, Rodriguez A, Ehlenberger DB, Rocher AB, Henderson SC, Hof PR. New techniques for imaging, digitization and analysis of three-dimensional neural morphology on multiple scales. *Neuroscience* 2005;136(3):661–80.
- [20] Zhou W, Li H, Zhou X. 3D dendrite reconstruction and spine identification. In: *Proceedings of medical image computing and computer-assisted intervention-MICCAI*, vol. 5242. 2008. p. 18–26.
- [21] Cheng J, Zhou X, Miller E, Witt RM, Zhu J, Sabatini BL, et al. A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy. *J Neurosci Methods* 2007;165(1):122–34.
- [22] Bai W, Zhou X, Ji L, Cheng J, Wong STC. Automatic dendritic spine analysis in two-photon laser scanning microscopy images. *Cytometry A* 2007;71(10):818–26.
- [23] Fan J, Zhou X, Dy JG, Zhang Y, Spires TL, Hyman BT, et al. An automatic method for spine detection and spine tracking in in vivo images. In: *Life science systems and applications workshop*. 2007. p. 233–6.
- [24] Zhang Y, Zhou X, Witt RM, Sabatini BL, Adjeroh D, Wong ST. Dendritic spine detection using curvilinear structure detector and LDA classifier. *Neuroimage* 2007;36(2):346–60.
- [25] Janoos F, Mosaliganti K, Xu X, Machiraju R, Huang K, Wong STC. Robust 3D reconstruction and identification of dendritic spines from optical microscopy imaging. *Med Image Anal* 2009;13(1):167–79.
- [26] Fan J, Zhou X, Dy JG, Zhang Y, Wong STC. An automated pipeline for dendrite spine detection and tracking of 3D optical microscopy neuron images of in vivo mouse models. *Neuroinformatics* 2009;7(2):113–30.
- [27] Lang S, Drouvelis P, Tafaj E, Bastian P, Sakmann B. Fast extraction of neuron morphologies from large-scale SBFSEM image stacks. *J Comput Neurosci* 2011;31(3):533–45.
- [28] Mukai H, Hatanaka Y, Mitsuhashi K, Hojo Y, Komatsuzaki Y, Sato R, et al. Automated analysis of spines from confocal laser microscopy images: application to the discrimination of androgen and estrogen effects on spinogenesis. *Cereb Cortex* 2011;21(12):2704–11.
- [29] Nägerl UV, Willig KI, Hein B, Hell SW, Bonhoeffer T. Live-cell imaging of dendritic spines by STED microscopy. *Proc Natl Acad Sci U S A* 2008;105(48):18982–7.
- [30] Frangi AF, Niessen WJ, Vincken KL, Viergever MA. Multiscale vessel enhancement filtering. In: *Proceedings of the 1st international conference on medical image computing and computer-assisted intervention*, vol. 1496. 1998. p. 130–7.
- [31] Truc PTH, Khan MAU, Lee YK, Lee S, Kim TS. Vessel enhancement filter using directional filter bank. *Comput Vis Image Understand* 2009;113(1):101–12.
- [32] Su R, Sun C, Zhang C, Pham TD. Linear feature enhancement based on morphological operation and Gabor function. In: *Proceedings of the 27th image and vision computing*. 2012. p. 91–6.
- [33] Su R, Sun C, Zhang C, Pham TD. A new method for linear feature and junction enhancement in 2D images based on morphological operation, oriented anisotropic Gaussian function and Hessian information. *Pattern Recogn* 2014;47(10):3193–208.
- [34] Su R, Sun C, Pham TD. Junction detection for linear structures based on Hessian, correlation and shape information. *Pattern Recogn* 2012;45(10):3695–706.
- [35] Hassouna MS, Farag AA. Robust centerline extraction framework using level sets. In: *Proceedings of the 2005 IEEE computer society conference on computer vision and pattern recognition*. 2005. p. 458–65.
- [36] Zhang TY, Suen CY. A fast parallel algorithm for thinning digital patterns. *CACM* 1984;27(3):236–9.
- [37] Pudney C. Distance-ordered homotopic thinning: a skeletonization algorithm for 3D digital images. *Comput Vis Image Process* 1998;72(3):404–13.
- [38] Petrosino A, Salvi G. A two-subcycle thinning algorithm and its parallel implementation on SIMD machines. *IEEE Trans Image Process* 2000;9(2):277–83.
- [39] Helman JL, Hesselin L. Visualizing vector field topology in fluid flows. *IEEE Comput Graph Appl* 1991;11(3):36–46.
- [40] Niblack C, Gibbons PB, Capson DW. Generating skeletons and centerlines from the distance transform. *CVGIP: Graph Models Image Process* 1992;54(5):420–37.
- [41] Breen EJ, Jones R. Attribute openings, thinnings, and granulometries. *Comput Vis Image Understand* 1996;64(3):377–89.
- [42] Serra J. *Image analysis and mathematical morphology*. Academic Press; 1982. ISBN 0-12-637240-3.
- [43] Danielsson PE. Euclidean distance mapping. *Comput Graph Image Process* 1980;14(3):227–48.
- [44] Soille P, Talbot H. Directional morphological filtering. *IEEE Trans Pattern Anal Mach Intell* 2001;23(11):1313–29.
- [45] Canny J. A computational approach to edge detection. *IEEE Trans Pattern Anal Mach Intell* 1986;8(6):679–98.
- [46] du Buf H, Bayer M, Droop S, Head R, Juggins S, Fischer S, et al. Diatom identification: a double challenge called ADIAC. In: *Proceedings of the international conference on image analysis and processing*. 1999. p. 734–9.
- [47] Sun C, Pallottino S. Circular shortest path in images. *Pattern Recogn* 2003;36(3):709–19.
- [48] Vincent L, Soille P. Watersheds in digital spaces: an efficient algorithm based on immersion simulations. *IEEE Trans Pattern Anal Mach Intell* 1991;13(6):583–98.
- [49] Najman L, Schmitt M. Geodesic saliency of watershed contours and hierarchical segmentation. *IEEE Trans Pattern Anal Mach Intell* 1996;18(12):1163–73.
- [50] Couprie C, Grady L, Najman L, Talbot H. Power watershed: a unifying graph-based optimization framework. *IEEE Trans Pattern Anal Mach Intell* 2011;33(7):1384–99.
- [51] Zhang C, Sun C, Pham TD. Segmentation of clustered nuclei based on concave curve expansion. *J Microsc* 2013;251(1):57–67.
- [52] Meyer F, Beucher S. Morphological segmentation. *J Vis Commun Image Represent* 1990;1(1):21–46.
- [53] Pedrosa GV, Barcelos CAZ. Anisotropic diffusion for effective shape corner point detection. *Pattern Recogn Lett* 2010;31(12):1658–64.
- [54] Su R, Sun C, Pham TD. Dendritic spines detection based on directional morphological filter and shortest path. In: *Proceedings of the 34th annual international conference of the IEEE engineering in medicine and biology society*. 2012. p. 5343–6.