

# A novel computational approach for automatic dendrite spines detection in two-photon laser scan microscopy

Jie Cheng<sup>a,c</sup>, Xiaobo Zhou<sup>a</sup>, Eric Miller<sup>e</sup>, Rochelle M. Witt<sup>d</sup>,  
Jinmin Zhu<sup>b</sup>, Bernardo L. Sabatini<sup>d</sup>, Steven T.C. Wong<sup>a,\*</sup>

<sup>a</sup> The Methodist Hospital Research Institute, Radiology Department, Houston, Texas 77030-2707, USA

<sup>b</sup> Department of Radiology, Brigham and Women's Hospital, 1 Brigham Circle, Boston, MA 02115, USA

<sup>c</sup> Department of Electrical and Computer Engineering, Northeastern University, Boston, MA 02115, USA

<sup>d</sup> Department of Neurobiology, Harvard Medical School, 220 Longwood Avenue, Boston, MA 02115, USA

<sup>e</sup> Department of Electrical and Computer Engineering, Tufts University, Medford, MA 02155, USA

Received 23 March 2007; received in revised form 17 May 2007; accepted 17 May 2007

## Abstract

**Background:** Recent research has shown that there is a strong correlation between the functional properties of a neuron and its morphologic structure. Current morphologic analyses typically involve a significant component of computer-assisted manual labor, which is very time-consuming and is susceptible to operator bias. The existing semi-automatic approaches largely reduce user efforts. However, some manual interventions, such as setting a global threshold for segmentation, are still needed during image processing.

**Methods:** We present an automated approach, which can greatly help neurobiologists obtain quantitative morphological information about a neuron and its spines. The automation includes an adaptive thresholding method, which can yield better segment results than the prevalent global thresholding method. It also introduces an efficient backbone extraction method, a SNR based, detached spine component detection method, and an attached spine component detection method based on the estimation of local dendrite morphology.

**Results:** The morphology information obtained both manually and automatically are compared in detail. Using the Kolmogorov–Smirnov test, we find a 99.13% probability that the dendrite length distributions are the same for the automatic and manual processing methods. The spine detection results are also compared with other existing semi-automatic approaches. The comparison results show that our approach has 33% fewer false positives and 77% fewer false negatives on average.

**Conclusions:** Because the proposed detection algorithm requires less user input and performs better than existing algorithms, our approach can quickly and accurately process neuron images without user intervention.

© 2007 Elsevier B.V. All rights reserved.

**Keywords:** Automatic dendritic spine detection; Adaptive thresholding; SNR

## 1. Introduction

Neuron morphology is illustrative of neuronal function and can be instructive in the dysfunction seen in neurodegenerative conditions such as Alzheimer's disease and Parkinson's disease (Human, 2001; Segal, 2005). Recently, researchers have shown great interest in studying dendritic spine morphology and density quantitatively, both statistically and dynamically (Anderson et al., 1995; Elston and Rockland, 2002; Elston et al., 1999;

Nimchinsky et al., 2002; Page et al., 2002; Seib and Wellman, 2003).

Modern fluorescence microscopy methods, such as confocal laser scanning microscopy (CLSM) and two-photon laser scanning microscopy (2PLSM), provide powerful tools to study dendritic spine structures. However, the analysis of neuron images generated by either modality has remained largely manual. Even with computer assistance, such analysis is still extremely time-consuming, and subject to user bias, i.e., results cannot be easily confirmed by other investigators (Cohen et al., 1994; Duan et al., 2002; Anderson et al., 1995; Elston et al., 1999; Feldman and Peters, 1979). Thus, automation is in great need for dendritic spine analysis.

\* Corresponding author at: The Methodist Hospital Research Institute, Radiology Department, Houston, Texas 77030-2707, USA.

E-mail address: wong@crystal.harvard.edu (S.T.C. Wong).

Recently, several semi-automatic dendritic spines analysis approaches have been proposed (Koh et al., 2002; Weaver et al., 2004; Xu et al., 2006). Although these methods can greatly help neurobiologists with their work, shortcomings still exist. Some manually determined globally applied thresholds are used during segmentation. In general, the proper determination of this threshold is largely dependent in the experience of human operator. More importantly, the final results in terms of dendritic morphology are quite sensitive to the thresholds, thereby making the reliable extraction of information difficult. Also, unbiased and rapid analysis of a large set of images by other detection methods is not feasible because of the amount of user input required. To solve the above problems, many locally adaptive thresholding methods have been proposed in the literature. Some are based on the local variance of the image intensity. More specifically, the threshold for each pixel is calculated according to the local mean and variance in a window of a pre-determined size (Niblack, 1986; Sauvola and Pietaksinen, 2000; Tier and Jain, 1995). The formulation is simple and straight forward; however, the results are easily corrupted by the presence of spurious local intensity maxima. Other methods use surface-fitting procedures (Chan et al., 1991; Shen and Ip, 1997; Yanowitz and Bruckstein, 1989). As gradient-based methods, they are sensitive to background noise. Kriging is also widely used in the literature (Oh and Lindquist, 1999; Weaver et al., 2004). It is basically a two-pass algorithm using two global thresholds. All the pixels whose gray values are smaller than the lower threshold are segmented as the background. Those with gray values greater than the higher threshold are segmented as the objects. The remaining undetermined pixels are left to the second pass, at which point these pixels are segmented by using the local covariance of the class indicators and the constrained linear regression technique called Kriging. In this paper, we propose a local contrast-based segmentation algorithm. It is easily implemented for automatic and consistent analysis of neuron morphology.

In addition to segmentation, other problems also exist for above mentioned dendritic spines analysis algorithms. Koh et al. propose a morphology-based algorithm for automatically detecting and quantifying the structure of dendritic spines (Koh et al., 2002). However, this is only a simple distance-based algorithm for detached spine head detection. Also, this geometric approach is sensitive to noise and cannot detect spines of all morphologies. To separate spines from the shafts of dendrites, Xu et al. propose a new attached spine component detection algorithm by using two grassfire propagations (Xu et al., 2006). Although their method is more robust to the noise and the irregular, rough surface of dendrites, it still has a problem similar to Koh's approach: the shape of the detected spine is not accurately described, i.e., the base of the spine protrudes into the dendrite, instead of stopping parallel to the surface of dendrite. Based on Koh's approach, Weaver et al. describe a package which is capable of morphometry of an entire neuron, by combining the spine detection algorithms with dendritic tracing algorithms (Weaver et al., 2004). Their algorithm, however, does not accurately represent the attached spine component either. In this paper we propose a signal-to-noise ratio (SNR)-based detached spine component detection algorithm that we show can considerably reduce the

poor detection of spine components with low intensity values. Our attached spine components detection algorithm, which is based on the morphology analysis of the local dendrite, is also presented. The method we describe is more robust to noise and rough dendrite boundaries, while also representing the shape of spines more accurately than previously existing detection methods.

## 2. Image acquisition

Neuronal morphology located within brain slices of intact animals can be determined at high resolution using 2-photon laser scanning microscopy (2PLSM) of cells expressing green fluorescent protein (Denk and Svoboda, 1997). The analyzed dataset included a variety of genotypes to ascertain how well our algorithm detected spines with a wide distribution of morphologies. In brief, 3D image stacks of neurons were collected using a custom-built two-photon laser scanning microscope (Carter and Sabatini, 2004) with an excitation wavelength of 910 nm. The objective that we use is Olympus LUMPlanFI/RI 60 $\times$ /0.90 NA, water immersion. Images of apical and basal dendrites of CA1 hippocampal pyramidal neurons were acquired at 0.8 $\times$  zoom (image field, 300  $\mu$ m  $\times$  270  $\mu$ m), whereas the spiny regions of basal and apical dendrites were imaged at 5 $\times$  magnification (image field, 42  $\mu$ m  $\times$  42  $\mu$ m). Optical sections were taken at 1.0  $\mu$ m spacing. The manual analysis of spine density, length, and width, as well as soma size, were measured using custom software (Trachtenberg et al., 2002) by observers who were blind to the genotype. Spine lengths were measured from the junction with the dendritic shaft to the tip. To determine head width and primary dendrite thickness, the fluorescence was measured in a line across each structure and the width of the distribution where fluorescent intensity fell to 30% of maximum was calculated. Measurements performed on 100 nm diameter yellow-green fluorescent microspheres (FluorSpheres, Molecular Probes) indicated that the point-spread function placed a lower limit on measurable widths of 550 nm. The apparent width is the convolution of the true fluorescence distribution and the point-spread function of the microscope and has a lower limit of about 550 nm. The 3D image stacks obtained are 16-bit gray-scale images.

## 3. An automatic approach for dendrite spine detection

In this section, we discuss the algorithms used in the proposed neuroinformatics system called neuron image quantifier (NeuronIQ), an integrated data processing pipeline for automatic dendrite spine detection, quantification, and analysis. The image processing pipeline for NeuronIQ is demonstrated in Fig. 1. In Section 3.2, we introduce an automatic adaptive segmentation algorithm. In Section 3.3, the backbone extraction algorithm is

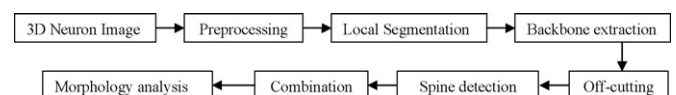


Fig. 1. Image processing pipeline for dendritic spine analysis.

presented. With the information obtained above, the detached and attached spine components are detected separately by using different approaches. The detached spine components are found by local SNR analysis, and the attached spine components are detected with using geometric properties of dendrites. The spine detection algorithms are discussed in Section 3.4. Lastly, certain post-processing algorithms, such as combination of broken spine components and measurement of spine density, are described in Section 3.5.

### 3.1. Preprocessing

There are two primary phenomena which cause the degradation of images obtained by optical microscopy. One is the shot noise introduced by the imaging mechanism of the photomultiplier tubes (PMT), which can generally be removed by median filtering without a loss in information concerning the neuronal spines and dendrites. The other mechanism for image degradation is the diffraction of light. The value of any voxel in the image actually is the convolution of intensities from its neighborhood. Numerous methods have been proposed to solve this problem with or without knowing the point spread function (PSF) of the system (Sonka et al., 1999). Based on our experience, deblurring algorithms can greatly enhance the contrast of the image, which makes it easier to detect weak spines. Nevertheless, since such processing can also amplify the noise, deconvolution sometimes degrades the segmentation results; thereby complicating later spine detection. Since our adaptive thresholding method can effectively segment the weak spines, we do not apply any deblurring algorithms during image preprocessing. As a result, the falsely detected spines are largely reduced, while the number of missed spines does not increase.

### 3.2. Adaptive threshold

The next step after denoising is to distinguish the spine or dendrite pixels from the background. The segmentation threshold can be set manually or automatically. Since there are no clear criteria to guide manual selection, the results are quite operator dependent. Furthermore, usually the manually set threshold is a global threshold, which causes problem in term of finding spine components with low intensities. To address these issues, an automatic local contrast-based segmentation method, which is suitable for high-content bioimage analysis, is presented in this paper.

In adaptive threshold methods, a threshold is set for each pixel. This threshold is then used to test against the pixel intensity value to produce binary image. The basic formulation of adaptive threshold for pixel  $p$  is given by

$$T_p = T(h(p), I(p)) \quad (1)$$

where  $I(p)$  is the gray level of point  $p(x, y)$  in the original image, and  $h(p)$  is a certain local property of this point  $p$ . For local contrast-based algorithms, usually the average or median value of intensity is used to calculate the local threshold. White and Rohrer (1983) compare the gray value of the pixel with the aver-

age of the gray values of neighboring pixels. The pixel will be recognized as the foreground pixel if its intensity is greater than the average intensity of all the pixels around. In the method proposed by Bensen (1986), the median value is selected as the threshold. Both methods work well at their cases. The width of the averaging window is set according to the quality of images and the general size of foreground objects. Basically, a larger window size would help suppress the noise in the image. However, it might also cause the loss of weak signals that are close to other strong signals. Based on Eq. (1), the adaptive thresholding algorithm is expressed as following:

$$s(x, y) = \begin{cases} 1, & \text{if } I(x, y) > \text{avg}\{I(x', y') | (x', y') \in \Omega(x, y)\} \\ 0 & \text{otherwise} \end{cases} \quad (2)$$

Here, '1' represents the foreground pixels, which are potentially spine or dendrite pixels and will be processed for later detection and analysis, '0' represents the background pixels, and  $I$  is the intensity value. The local region around a pixel  $(x, y)$ , i.e.,  $\Omega(x, y)$  is defined as  $\Omega(x, y) = \{(m, n) | |m - x| \leq (d - 1)/2, |n - y| \leq (d - 1)/2\}$ .  $d$  is the width of the window which decides the size of local region. The value of  $d$  is selected by considering the width of most spines. Normally,  $d$  can be set about one to two times of the spine width value. For our test images in Section 4, optimal results can be obtained for any number between 15 and 20 pixels. Generally, the segmentation results of spines are not sensitive to the window size, as long as enough foreground and background pixels are included. However, problems may occur for those small weak spines which are very close to the dendrites (whose intensities are higher than those of nearby spines). We will propose an efficient solution to this problem in the later part of this section.

Another potential problem for the above adaptive threshold algorithm is that some background pixels with relatively high intensity are prone to be segmented as foreground pixels. More information should be applied in order to distinguish this kind of local maximums from the real spine pixels. Besides the intensity difference, the exact value of each pixel's intensity should also be considered. Only those pixels whose intensities satisfy the minimal value requirement can be segmented as foreground pixels. For example, a pixel with intensity values as 1 should not be segmented as a foreground pixel if all its neighboring pixels' intensity is 0. This idea is similar to the lower threshold introduced by the Kriging method (Oh and Lindquist, 1999; Weaver et al., 2004). The minimal intensity value of a possible foreground pixel is selected as follows:

$$T_{\min} = T_{\text{low}} + \alpha \quad (3)$$

Here,  $T_{\text{low}}$  is the global lowest intensity value;  $\alpha$  is a control parameter. When  $\alpha$  is zero, the segmentation results are totally decided by the local contrast. With a bigger  $\alpha$ , local maximal intensity pixels are less likely to be segmented as foreground pixels. Generally, even a very small  $\alpha$  can obviously improve the segmentation results. This is because the high intensity noise, which is the shot noise, has already been removed by the median filter in the preprocessing stage. Based on the range of inten-

sity of spine pixels in 2PLSM images, the value of the control parameter  $\alpha$  is suggested as  $\alpha \in [5, 20]$ . The validation and comparison results in Section 4 are obtained with setting  $\alpha = 15$  for all images.

Besides the intensity, local region information should also be considered to improve the segment results. For example, a pixel has a higher probability to be a foreground pixel if most of its neighboring pixels are foreground pixels. This is based on the assumption that spines are blob-like objects. A simple approach for this idea is to only select those pixels whose most neighbors are segmented as foreground pixels. This can be easily realized by implementing a median filter to the originally segmented image  $s(x, y)$ .

For neuron images, a common problem for local thresholding is that the weak signal might be suppressed by nearby strong signals. For example for weak spines, low intensity pixels such as those at the spine neck are prone to be segmented as background pixels. This is because those pixels have lower intensity values compared with their neighboring average, which is caused by nearby dendrite pixels with very high intensity. To solve this problem, the high intensity values of nearby dendrite pixels should be adjusted, i.e., substitute the dendrite pixels with low intensity values. This is realized by two-step segmentation: dendrites are found after the first adaptive thresholding. The dendrites are defined as those blobs whose areas are larger than the largest possible size of a spine; then all pixels in the dendrites are replaced with lower intensity values in the original image; the adaptive thresholding is performed the second time after the intensity substitution. Compared with single adaptive thresholding, the two-step adaptive thresholding method can better detect the low intensity pixels at spine neck. Thus, the shape of spines is better represented. However, the revised intensity of dendrite tissues cannot be set arbitrarily low, otherwise some background pixels with relatively high intensity values will be wrongly segmented as foreground pixels. Therefore, the intensity adjustment should be carefully chosen with low bound being set.

The intensity adjustment can be linear or nonlinear. In our cases, we apply a simply linear transformation, i.e.  $I'_d(x, y) = \varepsilon I_d(x, y)$ . Here  $I_d(x, y)$  is the original intensity of the pixel in dendrite,  $I'_d(x, y)$  is the intensity value after adjustment,  $\varepsilon$  is the parameter which decides the suppression intensity. The upper

bound for  $\varepsilon$  is 1, which means no suppression is performed. The lower bound for  $\varepsilon$  is calculated as following:

$$\varepsilon_{\text{lower bound}} = \frac{\bar{I}_s}{\bar{I}_d} \quad (4)$$

Here  $\bar{I}_s$  is the average intensity of all pixels in spines and  $\bar{I}_d$  is the average intensity of all pixels in dendrites.

After segmentation, morphological filtering is performed to remove the noise, fill the holes, and smooth the boundaries. Some morphological processing, such as filling holes, can distort dendrite structures. To prevent these from happening, local intensity information is considered during the processing. Fig. 2 shows the comparison of segment results between global thresholding and adaptive thresholding. The latter one is obviously better, with fewer missed spine components and better segmented dendrite structure.

### 3.3. Backbone extraction

Dendrite structure is one of the most important geometric features in spine detection. Most existing spine detection algorithms are based on the successful extraction of backbones. There are two steps for backbone extraction: determining the medial axis by thinning algorithms and extracting the backbone from the medial axis by trimming branches. There are many thinning algorithms in the literature (Palagyi and Kuba, 1998; Lohou and Bertrand, 2002; Ma and Sonka, 1996; Ma and Wan, 2001; Nikolaidis and Pitas, 2001; Chuang et al., 2000) to find the medial axis in 3D. Basically a set of deleting templates are designed, which preserve the topology and geometry of the object. The object voxel under consideration is checked against the templates and is removed if its spatial arrangement is compliant to one of them. The medial axis or skeleton obtained by such thinning algorithms is sensitive to noise. As we can observe from Fig. 3, there are many spurs caused by noise (rough boundary). These spurs, in addition to the spurs caused by spines, should be removed to obtain a neat description of dendrite structures (backbones). The backbone is obtained by the trimming algorithms. In this section, we only discuss our trimming algorithm.

For some ideal situations, trimming can be very easy. All we need to do is to set a threshold for the physical length of the spines, and remove all spines whose length does not satisfy the

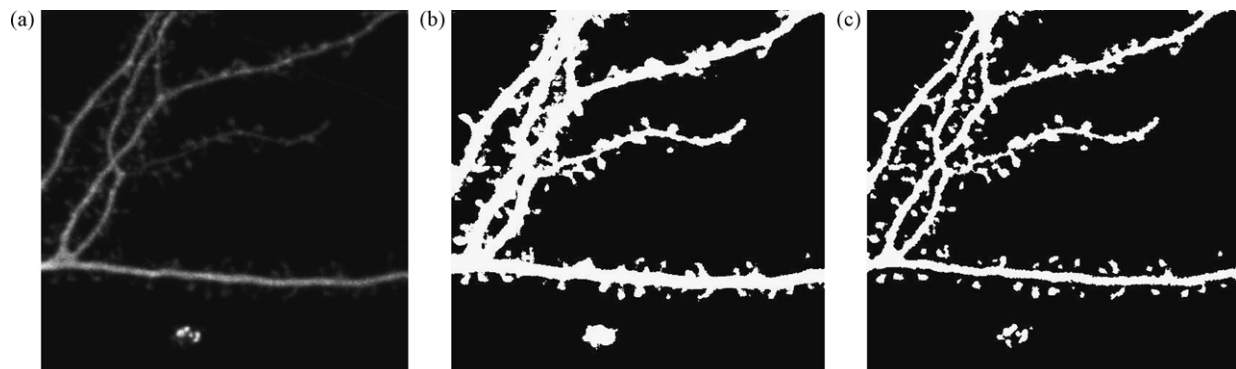


Fig. 2. Results of adaptive thresholding: (a) original images; (b) segmentation result with global threshold; (c) segmentation result with adaptive threshold.

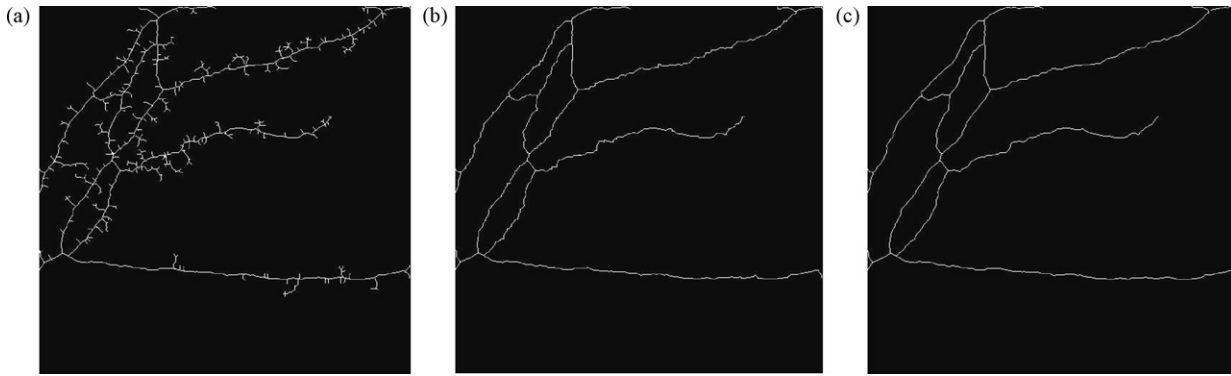


Fig. 3. (a) Medial axis of dendrites; (b) backbone after trimming; (c) backbone after removing bumps.

requirement. Or we can just pick up the longest pieces. This is the basic idea behind many of the skeleton pruning algorithms. An example of this kind of algorithm is proposed by Chen and Molloy (2003). They first try to find the starting point of the skeleton, by finding the line-end point with the maximal intensity value (or maximum diameter). From the starting point, the longest path  $P_{ij}$  is found.  $P_{ij}$  is then removed from the original skeleton image and another starting point is found. The procedures are repeated until the top  $N$  longest paths are found, which are assumed to be the dendrite backbone pieces. There are two potential problems for above mentioned method. One is how to select the starting point. While one could consider all line-end points as starting points, this will greatly increase the computation time. The other difficulty is how to ‘automatically’ set the value of  $N$ . As we can see from Fig. 3, the medial axis of a dendrite image can be very complex. This is caused by the complex structures in the big spines, or caused by the rough dendrite surface. If the spine density is high, the longest path-based trimming algorithm can be very time consuming. In this paper, we introduce a recursive trimming algorithm, which can deal with the above problems without knowing the starting point and number of dendrite pieces. The algorithm is summarized as following with the detailed listing of the steps provided in Table 1.

The basic idea is to keep removing end pieces (curves with end points) whose length is less than  $m$ . The value of  $m$  is iteratively changed in step (4). The process keeps repeated until the threshold  $m$  reaches the upper bound  $M$ , which is usually the longest possible length of a spine. Notice this algorithm is not simply removing all end pieces whose length is smaller than the threshold  $M$ , a process that would not only remove the medial axis of spines, but also will remove the end pieces of dendrites. This can be a serious problem when the threshold  $M$  is big. In our

approach, the dendrite pieces are restored at step (3). There are two situations in which the removed pieces cannot be restored, i.e.,  $l_i \neq m$ . (1) The length of the end piece is less than the temporary threshold  $m$ . In this case the short branches (medial axis of spines) are removed. (2) The length of the end piece is greater than the temporary threshold  $m$ . This happens when more than one connected branches are removed together. This is also the reason why the algorithm is implemented in an iterative way: if  $m = M$  at the very beginning, the end pieces of dendrites would not be restored, because there might be some spines attached to the end pieces of dendrites and are removed together. In such situation,  $l_i > m$  and the end pieces of dendrites cannot be restored. The trimming result is shown in Fig. 3.

After trimming, some correction to the backbone is performed. As we can see in Fig. 3(b), there are many small bumps along the backbone. These bumps are caused by the thinning algorithm during the process of obtaining the medial axis. The bumps may introduce the error for dendrite length estimation. They also cause trouble for spine length estimation and make length-based attached spine detection algorithm more difficult. Usually, the bumps locate at the place where there is a protrusion at the dendrite surface. For each protrusion, there is a piece of small branch in the medial axis. Based on above facts, here we propose a bump-removing algorithm as follows: (1) for each piece of branch along the backbone, find the branch point; (2) find local backbone pieces near the branch points, where are the possible locations of bumps; and (3) remove the bumps by connecting the two end points of above local backbone pieces.

### 3.4. Spine detection

From Fig. 2 we can find that there are some blobs which are not in a reasonable distance to the dendrites, i.e., the distance is larger than the longest possible length of a spine. This kind of blobs should be removed as non-spine blobs. The regions around the backbone should be defined before the detection, which will help reduce both the processing time and false positives. A local-region-cutting algorithm is designed for above purpose. Only blobs locate in the local regions are considered as potential spine components. Those who are not in the local regions are removed before spine detection. Although the maximal distance idea of the off-cutting algorithm is identical to Koh’s method (Koh et al., 2002), it is implemented in a more computation efficient way.

Table 1  
Recursive trimming algorithm

- 
- (1) Initialize  $m = 1$
  - (2) Remove all end points (who only has one neighbor) in the image; keep doing for  $m$  times
  - (3) Check the length of all the removed pieces  $l_i$ ; if  $l_i = m$ , restore the removed piece
  - (4) Let  $m = m + s$ , if  $m < M$ , go to step 1
  - (5) Finish trimming
-

Instead of calculating the distance from every detached blobs to the nearby dendrite backbone point, we define a local region around the backbone, which is simply obtained by dilating the backbone. The advantage becomes obvious if the spine density is high.

There are two different approaches for the detection of attached spine components and detached spine components. The two detection algorithms are discussed in the following sections.

### 3.4.1. Detection of detached spines

The detached spine component detection algorithm is based on the local signal to noise ratio (SNR) analysis. The local region of each potential spine should be found before estimating the local SNR. Since the size of different spines may vary a lot, it is not feasible to choose a window with fixed size for local SNR analysis. The size of the window should be adjusted according to the size of each blob. Suppose the  $w(p)$  is the rectangle with the smallest area which can fully contain blob  $p$ , then the local region is defined as the following:

$$R_p = \{(x, y) | x \in [x_0 - \lambda, x_1 + \lambda], y \in [y_0 - \lambda, y_1 + \lambda]\} \quad (5)$$

Here  $x_0, x_1, y_0,$  and  $y_1$  are the least and largest values in  $x$ - and  $y$ -direction respectively for all pixels in the blob;  $\lambda$  is the window's size, which is introduced to include enough background pixels in the local region. Suppose the area of the blobs can be estimated with a rectangle  $a \times b$ , the area of local region is  $nab$ , then the value of  $\lambda$  can be obtained by

$$(a + 2\lambda)(b + 2\lambda) = nab \quad (6)$$

$$\lambda = \frac{\sqrt{(a+b)^2 + 4ab(n-1)} - (a+b)}{4} \quad (7)$$

For the purpose of implement,  $\lambda$  can be rounded to the nearest integer.

Once the local region for a spot is determined, we can then define the local SNR. In image processing, the local SNR for spot  $P$  is usually estimated as (Zhang et al., 2007):

$$\text{SNR}_p = \frac{I_{sp} - I_{bp}}{\sigma_{bp}} \quad (8)$$

Here,  $I_{sp}$  is the average intensity of the detected signals,  $I_{bp}$  is the average intensity of the background pixels,  $\sigma_{bp}$  is the standard deviation of the background intensity of the local region. The local SNR analysis is applied to all detached blobs in the maximal intensity projection (MIP) image of a neuron image for spine detection. If the SNR is higher than a threshold, which is obtained by clustering, the certain blob is recognized as a detached spine.

In some simple cases, this SNR-based algorithm can detect detached spines well. However, sometimes there are some small non-spine blobs, which arise from dying or imaging problems, that cannot be removed by simply increasing the SNR threshold. Nevertheless, the performance of SNR-based detection algorithm will be improved if intensity distribution information among neighboring slices can be used. For 3D neuron images, the resolution in  $z$ -direction is much lower compared with  $x$ - and  $y$ -direction. Most spines will only appear in two or three

slices. Thus, the section area of a spine changes obviously in the neighboring slices. In addition, the intensity for spine voxels in neighboring slices changes more obviously compared with the voxels in non-spine blobs. This is can be explained by the intensity difference among surface voxels and central voxels in a spine. Based on above observations, a new SNR which considers the 3D intensity properties of a spine is proposed:

$$\text{SNR}_p = \text{SNR}_{\text{normal}} f_1 = \frac{I_{sp} - I_{bp}}{\sigma_{bp}} \left(1 + \frac{N_{FG}}{A_{\text{local}}}\right)^\varepsilon \quad (9)$$

Here,  $A_{\text{local}}$  is the area of the blobs in the MIP (maximal intensity projection) image;  $N_{FG}$  is the number of foreground voxels:  $P_{\text{foreground}} \in \{P | \text{voxels whose intensity difference in adjacent slices} \geq n_0\}$ ;  $n_0$  is a threshold for intensity difference. If  $n_0 = 1$ , then all voxels with different intensity values among neighboring slices are considered as foreground voxels. For each blob, three slices (central slice and two neighboring slices) are considered. The central slice of a blob is defined as the one with the highest average intensity. According to our tests, the intensities of voxels in the small non-spine blobs have few changes compared with those spine voxels. The SNR-based detection algorithm works well even with a very small  $n_0$ . In our validation,  $n_0$  is set as 1;  $\varepsilon$  is the weight parameter with positive value. With a higher  $\varepsilon$ , we emphasize more on the change of intensity distributions among neighboring slices. In our cases,  $\varepsilon$  is set to 2. The number of the foreground pixels is normalized by the area of local regions.

The comparison of detection results based on normal SNR and revised SNR is shown in Fig. 4. As we can observe from the SNR distribution maps, there is no threshold which can totally separate the spine components and the non-spine blobs by using normal SNR. However, these two different classes are well separated if described by revised SNR. Nearly all non-spine blobs have obviously lower values compared with the spine components, e.g., the values are lower than the half of the value of the spine with the lowest revised SNR. We also notice that there are some non-spine blobs introduced by a crossing axon at the middle bottom of the image. This kind of non-spine blobs can also be recognized and discarded by using revised SNR.

Besides removing the false positives (non-spine blobs), our SNR analysis and adaptive thresholding-based algorithm can also detect weak detached spine components, which are possibly missed by global thresholding. From Fig. 5 we can see that four spines are missed because of low intensity values by using global thresholding. For these spine components detected by both approaches, the shape of spine components are much better represented by using our method, e.g., the volume of spine would not apparently shrink because of a relatively high global threshold, which may be introduced to suppress strong noise. The quantitative comparison of spine detection results between our method and Koh's method (Koh et al., 2002) are presented in the next section.

### 3.4.2. Detection of attached spines

The approach to detecting attached spine components along the dendrite is quite different from detached spine components

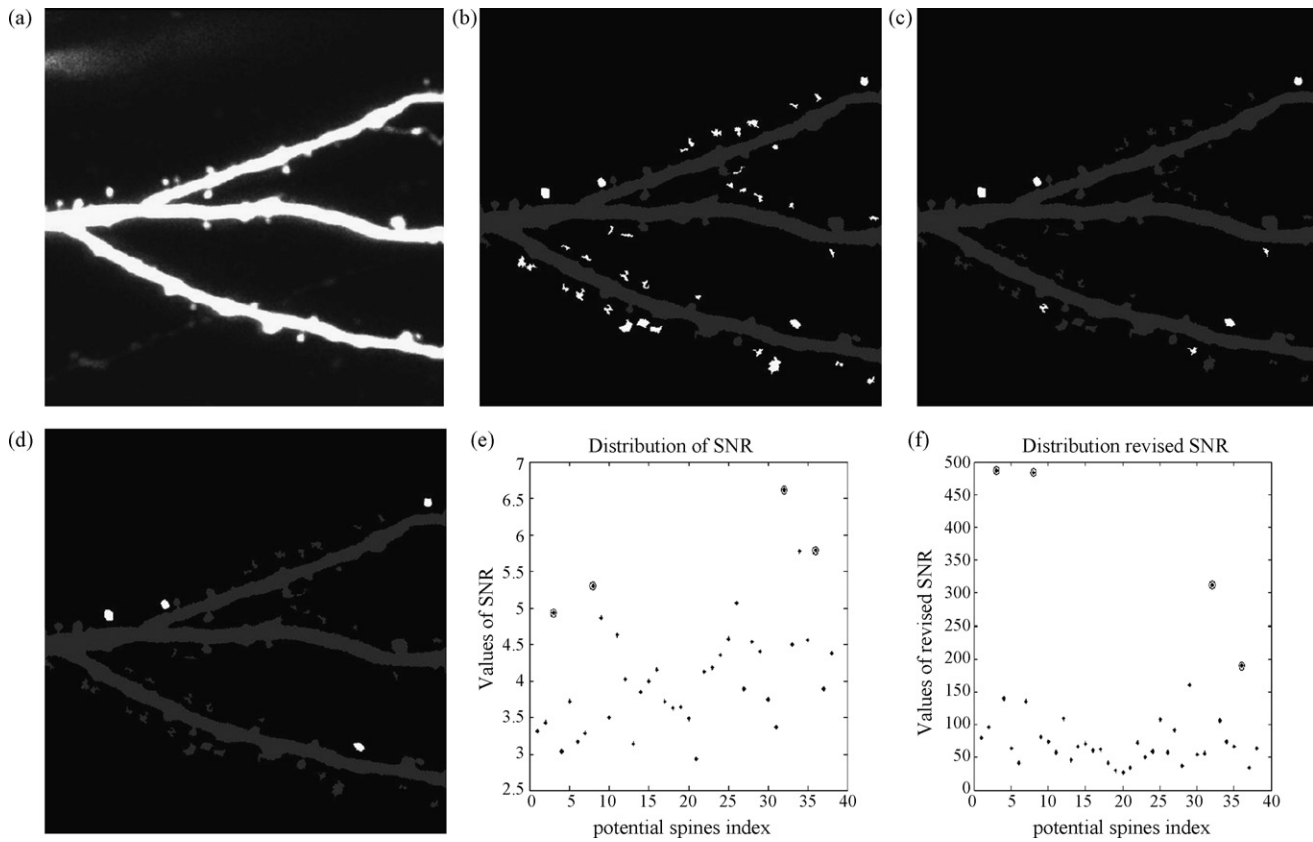


Fig. 4. Comparison of normal SNR and revised SNR for detached spine head detection. (a) Original images; (b) segmented image obtained from adaptive threshold algorithm, with all detached blobs (possible detached spine head) marked; (c) detached spine head detected by using normal SNR; (d) detached spine head detected by using revised SNR; distribution of (e) SNR and (f) revised SNR for each detached blob. The circles represent the manually detected spine head in 3D image.

detection. Unlike the local contrast and SNR-based approach, it is mainly based on the local morphological analysis.

In the segmented image, there are some protrusions along the dendrite structures, which are potentially attached spine components. To find those spine components, one common approach is to find the tips for each protrusion, then locate each spine by local shape analysis (Koh et al., 2002). The tips are the pixels along the boundary, which have local maximal distance to the backbone pixels.

There are several problems for this kind of detection method. The location of the tip is noise sensitive: with rough boundary of the dendrite, there will be too many local maximal distance

pixels exist and it is hard to find the ‘ideal’ tip (the maximal distance pixel on the top of a protrusion). This problem can be partially solved by smoothing the dendrite with low-pass filters.

Another problem is that the shape of the spines is not well represented: the base of the spines is described as a piece of curve, instead of a straight line parallel to the nearby dendrite boundary. This is because the spine pixels are defined as the pixels those are within a certain range to the tip point of the protrusion. The local maximal distance-based algorithms also have difficulties to detect some very small spines or spines with irregular shapes.

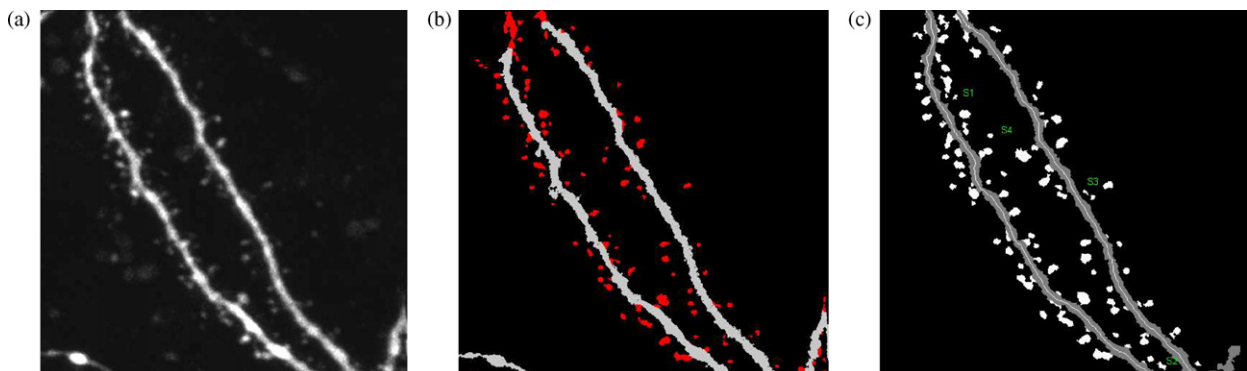


Fig. 5. Comparison of spine detection results: (a) original images; (b) Koh's approach; (c) our method.

Here we propose a medial axis-based attached spine detection algorithm, which can solve the above problems. In the medial axis image, there is a branch (small piece of line) at the location of each spine along the dendrite. Thus, the attached spine component can be located by referring the position of the branches. However, there also are many branches appear at the place where no spine presents. Some criteria are needed to locate the real spines. The algorithm is described as following:

- (1) Find all small branches according to the medial axis and backbone image—the small branches are found by removing backbone from the medial axis.
- (2) Estimate the local thickness of the dendrite in a neighboring region around each small branch.
- (3) Mark spine candidates based on the estimation of the local thickness and the distance map to backbone—all pixels in the local region that have a distance larger than the local thickness are marked as spine pixels.
- (4) Remove false spines based on the area and edge criterions.

The local thickness of the dendrite is estimated by calculating some local edge pixels' distance to backbone. Suppose  $p_i$  is the boundary pixel along the local region, and  $d_i$  is the distance to the nearest backbone pixel. The local thickness of the dendrite is estimated by

$$\text{Th} = \text{median}(\{d_i | d_i \leq \min(d_1, d_2, \dots, d_k) + \beta\}) \quad (10)$$

Here,  $\min(d_1, d_2, \dots, d_k)$  means the minimal distance among all  $k$  boundary pixels;  $\beta$  is a threshold which decides the range of distance to be considered. To be more robust, the local thickness is not set equal to the local minimal distance. Instead, a set of local distance values are considered and the median value is picked as the estimation of local dendrite thickness. So  $\beta$  should be greater than zero. Compared with the dendrite boundary pixels, the spine boundary pixels are much more far away from the backbone. For the purpose of accurate estimation of local dendrite thickness, as much as possible spine boundary pixels should be excluded. Thus,  $\beta$  should also have a higher bound. Usually, the value of parameter  $\beta$  is set according to the quality of the segmented results. If the quality of the image is very low,  $\beta$  should be set to a larger number to lower the probability of being trapped in the local minimums. For images with ordinary quality, small value of  $\beta$  can be chose to exclude more spine boundary pixels. For all the images demonstrated in this paper  $\beta=2$ . The reason that we choose the median value of local distances is based on the observation: most of the local non-protrusion boundary pixels have the same distance to the backbone, which is a good estimation of the local thickness of a dendrite.

As we mentioned before, not all spine candidates are actually a spine. Some criteria are set here to help to remove those false spines.

- Area criterion:

Only those candidates whose area is larger than a threshold can be a real spine:

$$C_i = \begin{cases} \text{is a spine,} & \text{if Area}(C_i) > A_0 \\ \text{is not a spine,} & \text{if Area}(C_i) \leq A_0 \end{cases} \quad (11)$$

Here  $C_i$  is the spine candidate found by above detection algorithms;  $A_0$  is the area threshold. This criterion is set to help remove the false positives along the rough dendrite boundary. Usually neurobiologists do not consider very small protrusions as spine components. This threshold can be manually set before the batch processing. Usually the value is related to the choice of different views. For example, in our experiments (where  $0.084 \mu\text{m}/\text{pixel}$ ), we ignore all small protrusions whose area is less than 5 pixels.

- Edge criterion:

Only those candidates that locate outside the trunk of the dendrite can be a real spine:

$$C_i = \begin{cases} \text{is a spine,} & \text{if } N'_{C_i} > N''_{C_i} \\ \text{is not a spine,} & \text{if } N'_{C_i} \leq N''_{C_i} \end{cases} \quad (12)$$

Here  $N'_{C_i}$  is the number of boundary pixels of a spine candidate, which are also the boundary pixels of a local dendrite;  $N''_{C_i}$  is the number of boundary pixels of a spine candidate, which are not the boundary pixels of the local dendrite (pixels inside the dendrite). This criterion is introduced to remove the false detected spine components which protrude deep into the dendrite.

For the purpose of easy implementation, the algorithm for attached spine detection is based on the projection of backbone and the image of whole stack. However, the idea of estimation of the thickness of local dendrite pieces and the criteria to remove the false positives can also be implemented in 3D.

Besides the advantage of being able to high-content batch process, our approach also has better detection results compared with the existing approaches. Fig. 6 shows the comparison of spine detection results between our method and Koh's method (Koh et al., 2002). Compared with the existing spine detection algorithms, our method has the following advantages:

- Better detection results for very weak spines. Either for attached spine or detached spine components.
- Fewer false positives caused by the broken dendrite part. This is due to our robust adaptive thresholding method.
- The shape of the spines is better represented, e.g., the bottom of the attached spine is a line close and parallel to the edge of dendrite. For Koh's approach, sometimes the position of the spine bottom cannot be found correctly, which means either only a part of the spine is detected or nearby dendrite tissue is segmented as the spine.

### 3.5. Post-processing

There are mainly two purposes for post-processing. One is to adjust the numbers of detected spines, and the other one is to



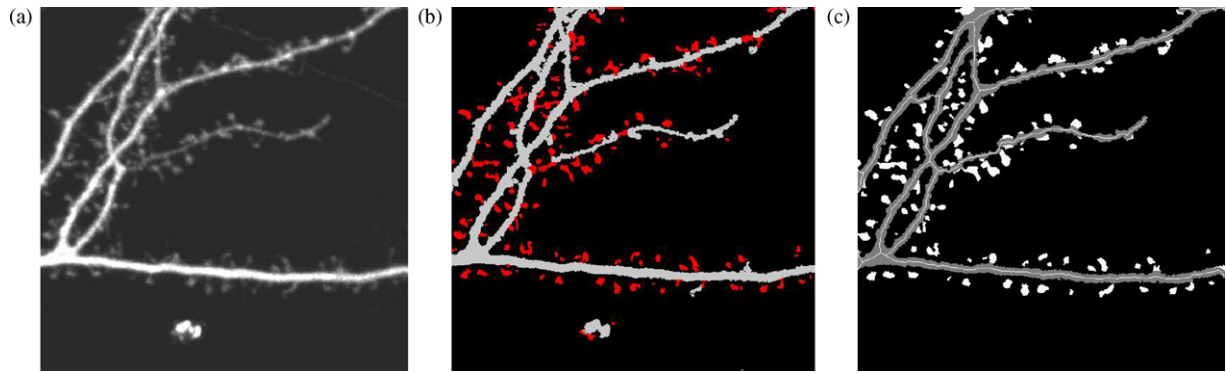


Fig. 6. Comparison of spine detection results between Koh's approach and our approach: (a) original images; (b) detection results by using global thresholding and Koh's approach; (c) detection results of our fully automatic approach.

provide measurements about the spines and dendrites. The number of blobs detected as spines is not actually the real number of spines. In some cases, a spine can be broken into several parts in the image. One reason is that the signal around the neck of some thin spines is weak so that the spine is broken into the detached head and attached base components after segmentation.

A merging algorithm is performed to combine those detached and attached parts. The merging algorithm considers every attached spine or spine base, checking for possible merges with other detached components in local region. It is possible that several detached components are recognized as the parts of a single spine. This happens when the detached components are combined with the same attached spine base.

There are two criteria for the merging algorithm: the separated components should be close enough, and the separated components should satisfy certain relative orientation requirement. It is easy to implement the first criteria. For the second criterion, suppose  $P_1P_2$  is the cutting line of the attached spine base on the dendrite;  $C_1C_2$  is the connection line of the centers of attached component and detached component;  $\theta$  is the angle between  $P_1P_2$  and  $C_1C_2$ . If  $\theta \geq \theta_0$ , then the separated components are considered to be the parts of the same spine. A suggested range for the threshold  $\theta_0$  is:  $\theta_0 \in [30^\circ, 45^\circ]$ . In our cases,  $\theta_0 = 40^\circ$ . Compared to Koh's method which requires the accurate position of the tip of each protrusion, this method is more robust to noise and bad segmentation.

During post-processing, some important measurement of spines and dendrites are acquired. NeuronIQ provides the function to measure dendrite length, spine length, spine density, spine volume, spine section area, spine section perimeter, spine neck width, etc. For the purpose of comparison with manual results, only spine length and spine density are discussed in this paper.

Spine length and spine density are two of the most important properties for neurons. The spine density is defined as:  $d = n/l$ , where  $l$  is the length of a dendrite, and  $n$  is the total number of spines on that dendrite.

The length of a dendrite is calculated based on the backbone image. One common approach for this problem is to first code the backbone by using some chain coding algorithms, e.g. Freeman Chain Coding (Freeman, 1974) or Primitives Chain Coding (O'Gorman, 1992). The total length of the dendrites

is calculated afterwards: for 2D images, when two pixels are in 4-neighborhood to each other, the total length is increased by 1; when two pixels are in 8-neighborhood but not in 4-neighborhood to each other, the total length is increased by 1.414.

The problem for the coding-based algorithm is that the codes may become complex when the structure of the dendrites is complex. The coding becomes complex because it tries to store the exact location information for each pixel. However, we do not need to know the exact location of each pixel of the dendrite in order to figure out the total length of the dendrite. Based on this idea, we propose a fast and efficient algorithm to estimate the length of a dendrite:

- (1) Get the total number of pixels  $n_0$ .
- (2) Get the total number of pixels which have one (8–4)-neighbor, say  $n_1$ .
- (3) Get the total number of pixels which have two (8–4)-neighbors, say  $n_2$ .
- (4) Calculate the length of dendrite:  $l = (n_0 - 1) + ((\sqrt{2} - 1)(n_1 + 2n_2))/2$

A (8–4)-neighbor is defined as a pixel which is in the 8-neighborhood, but not in the 4-neighborhood at the same time. The number of (8–4)-neighbor for each backbone pixel can be easily obtained by filtering backbone image. The filtering window is defined as

$$w = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

Actually,  $n_1$  and  $n_2$  can be obtained by one time filtering. This method can be easily extended to 3D images. We just need to find the total number of (26–18) neighbors in the similar way.

By applying the same method, the length of each spine is obtained by calculating the length of branch relative to the spine. The branch is obtained from the medial axis image and backbone image.

**4. Results and discussions**

*4.1. Results analysis based on single image*

To validate our algorithms, the results obtained by NeuronIQ are compared with the manual results image by image. The results for spine numbers, spine density, and spine length distribution are compared in detail in this section. The manual detection is performed without knowing the automated detection results. An expert will observe several neighboring slices before mark each spine. The whole image stack is divided into four regions. The expert can zoom into each region during observation. The manual detection and measurement are performed in 3D. Fig. 7(a) is just a 2D demonstration of the manual results.

Fig. 7 shows the manually and automatically detected spines for one image sample. There are totally 69 spines detected both

manually and automatically in this image. Compared with the manual results: there is one spine missed (spine number 70), and there are four small spines not manually detected (spine number 73, 74, 75, and 76). The detailed comparison between the automate and manual results are shown in Table 2. The reason of missing in automatic detection is mainly because the intensity of the spine is very low compared with the nearby dendrite. As for manual detection, those small size attached spine components are most likely to be ignored by human.

The reason that the manually measured dendrite length is smaller than the automatic result is that the manual result is only an estimation of the actual dendrite length: estimate curve with several connected lines. The detailed spine length distribution can be found in Fig. 8.

The mean square error (MSE) for the manually and automatically measured spine lengths is 0.0292. The length distributions

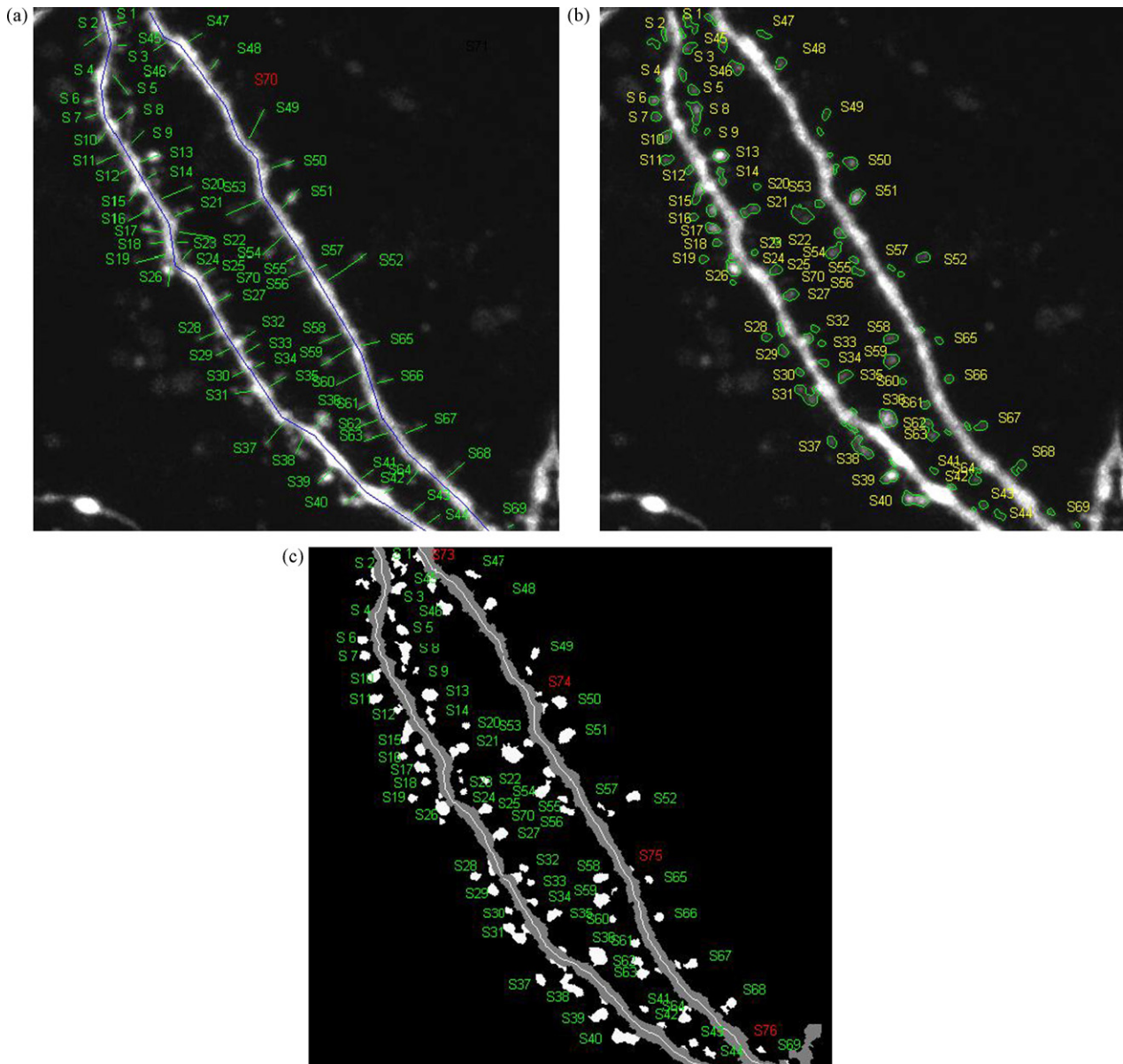


Fig. 7. Comparison of (a) manual results; (b) automated spine detection results with edges marked on the original image; (3) automated results on the segmented image.

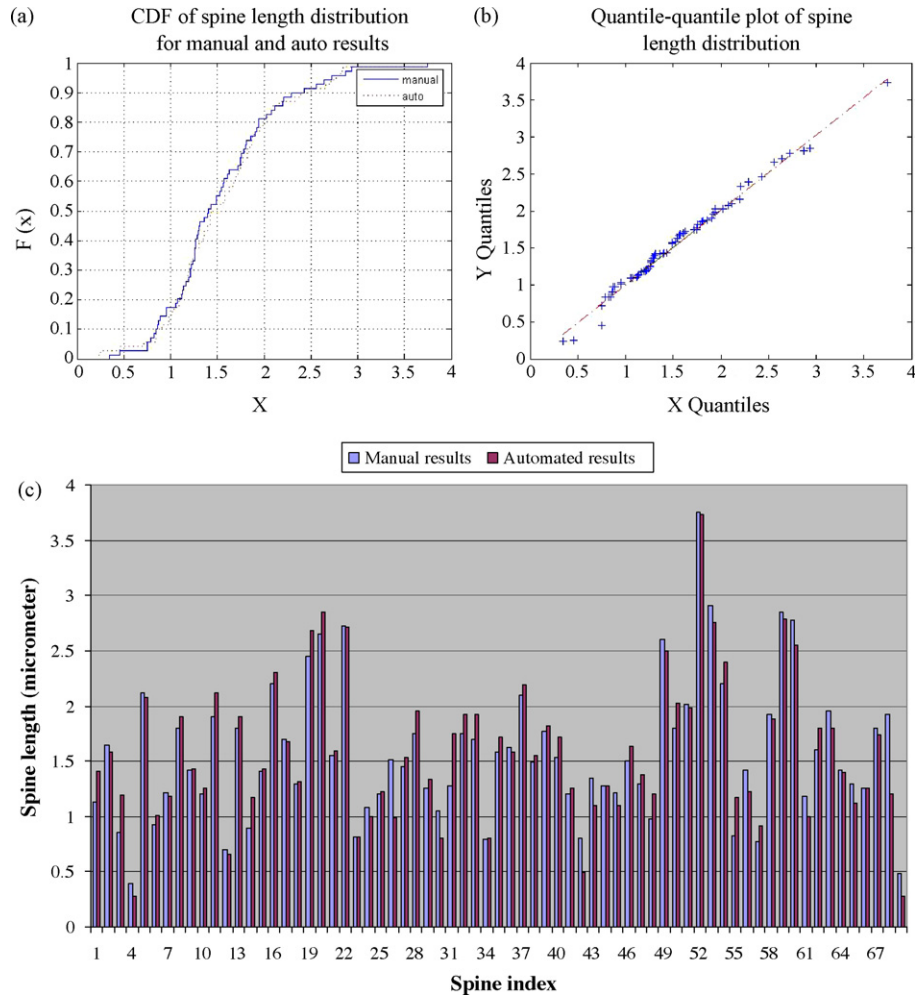


Fig. 8. (a) CDF of spine length distribution for manual and automatic results; (b) empirical quantile–quantile plot (called QQplot) for two distributions; (c) comparison of spine lengths.

for manual and automatic results are also tested by two-sample Kolmogorov–Smirnov test. The null hypothesis that the two distributions are the same is not rejected, which means that there is no obvious difference between the two distributions. The probability that the dendrite length distributions of manual and automate results are the same is 99.13%. The biggest difference between these two distributions is 0.075.

#### 4.2. Results analysis based on image set

In addition to the single image comparisons, the manual and automatic results are also compared in image set. We randomly select 16 images in our image dataset (Zhou et al., 2006) for validation purpose. The spine densities are compared first. As we can see from the result shown in Fig. 9, there is obvious linear

relation between the manually and automatically measured spine densities. Two reasons cause the difference between manually measured spine density and the automatic measured results: one is the difference of dendrite length measured, and the other is that the very small spines are likely to be ignored manually.

Besides comparing with the manual results, we also compare the results obtained by our approach with those obtained by applying Koh's method (Koh et al., 2002). Spine detection results of five neuron dendrite images are compared (Table 3). The five image stacks are five views from five different neurons. The false positives and false negatives are decided by comparing with the manual results. We can see clearly from the results that our method has less wrong detection and missing compared with Koh's method. The missed spines are obviously reduced by using our method.

Table 2  
Comparison of manual and automatic results

Method	Num of spine detected	Spine density (number/micron)	False positive	False negative
Manual	70	$70/104.675 = 0.669$	0	4
Automatic	73	$73/110.8388 = 0.659$	0	1

Table 3  
Comparison of spine detection results

Image index	Manually detected spine number	Our Method		Koh's method	
		False positives (wrong detection)	False negatives (missing)	False positives (wrong detection)	False negatives (missing)
1	70	4	1	5	8
2	32	3	1	4	4
3	28	3	0	2	4
4	63	1	0	1	4
5	36	1	3	6	2

4.3. Discussion

From the results shown above we can see clearly that our approach has obviously better performance in spine detection compared with other existing methods. The spine length distribution and spine density obtained are also very similar to the manual results. Besides being automatic, our approach also has the following advantages:

- Robust local thresholding can obviously reduce the possibility of missing spines. Spines with relatively low intensity values will not be segmented as background according to a globally set threshold. At the same time, the false positives are reduced because of the control parameter  $\alpha$  stated in Eq. (3).
- The measurement of spine components is more consistent. This is a very important property for the neuroscientists, who are interested in comparing the measurement in different conditions. Nevertheless, it is not the case for the global threshold-based algorithms (Koh et al., 2002; Xu et al., 2006): the size of the spines are sensitive to the global threshold.
- The shapes of both attached and detached spine parts are better represented. The base of attached spine components will not protrude into the dendrite. This can be observed from the comparison between our method and Koh's method, in both Figs. 5 and 6.

- For the detection of attached spine components, our method is more robust to noise and the rough dendrite boundary compared with local maximal distance (tips of a protrusion) based algorithms, such as Koh and Xu's approaches (Koh et al., 2002; Xu et al., 2006). This is because these kinds of methods are prone to be trapped in the local maximums because of the noise. On the contrary, our local dendrite thickness estimation method, which relies on the median values of more than 10 pixels, is much more robust.

There remain unsolved problems in our approach however. We can successfully solve the problem to combine an attached spine component with its detached components. In cases that there are only two or more detached spine components in the image and without a relative attached spine component, we cannot find an efficient way to combine those detached spine parts without wrongly combining the parts from different spines. This is the problem we need to address in our future work.

5. Conclusions

In this paper we discuss the algorithms implemented in our automatic spine detection software: NeuronIQ. The comparison between manual results and automatic results is also presented. It shows that our algorithms have a high spine detection rate with both low false positives and false negatives. The manual and automatic measurements of spines show close similarity. No obvious difference is found according to our tests.

Besides the satisfying results, another attractive property of our approach is that it is able to batch process a large number of images. This means that no human interference is required during preprocessing, segmentation, backbone extraction, spine detection and measurement, although the parameter adjustments, such as the size of filtering windows, might be needed before the processing of a new dataset. This highly automatic approach provides a pragmatic solution for enabling high throughput screening in the research of neuron science.

With robust segmentation results and highly accurate spine detection results, we are interested in more complex and specified analysis of the dendrite properties. Our future works will include spine classification and dendrite reconstruction. We believe that highly automate image processing approaches would greatly help the neuroscientists improve the productivity and efficacy of their research, especially for some manually time-consuming tasks such as quantitative morphological analysis about neuron images.

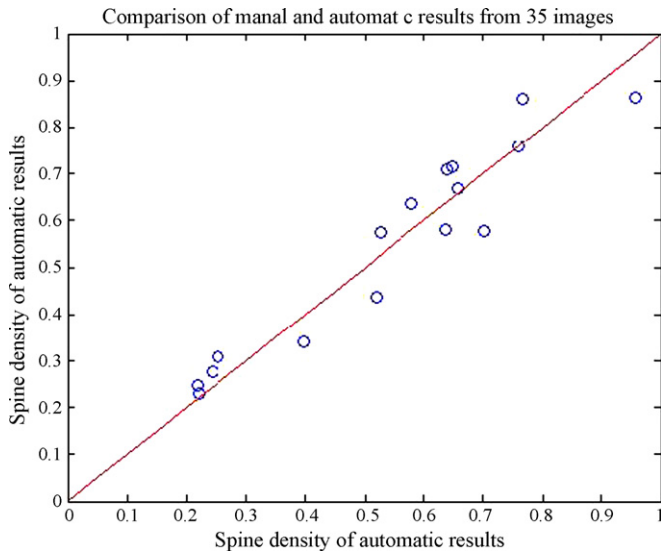


Fig. 9. Comparison of manual and automatic spine density results from 16 images.

## Information sharing statement

Some of the images for validation are available online from <http://neuroinformatics.harvard.edu> once this paper is formally accepted. NeuronIQ software package will be available upon request.

## Acknowledgments

The authors would like to thank research members of the Life Science Imaging Group of the Center for Bioinformatics, Harvard Center for Neurodegeneration and Repair (HCNR), Harvard Medical School. This research is funded by the HCNR Center for Bioinformatics Research Grant, Harvard Medical School and a NIH R01 LM008696 Grant to STCW.

## References

- Anderson SA, Classey JD, Conde F, Lund JS, Lewis DA. Synchronous development of pyramidal neuron dendritic spines and parvalbumin-immunoreactive chandelier neuron axon terminals in layer III of monkey prefrontal cortex. *Neuroscience* 1995;67:7–22.
- Bersen J. Dynamic thresholding of gray level images. In: *ICPR'86: Proc. Intl. Conf. Patt. Recog.*; 1986. p. 1251–5.
- Carter AG, Sabatini BL. State dependent calcium signaling in dendritic spines of striatal medium spiny neurons. *Neuron* 2004;44(3):483–93.
- Chan FHY, Lam FK, Zhu H. Adaptive thresholding by variational method. *IEEE Trans Image Process* IP-7 1991:468–73.
- Chen Z, Molloy S. Automatic 3D vascular tree construction in CT angiography. *Comput Med Imag Graph* 2003;27:469–79.
- Chuang JH, Tsai CH, Ko MC. Skeletonization of three-dimensional object using generalized potential field. *IEEE Trans PAMI* 2000;22(11):1241–51.
- Cohen AR, Roysam B, Turner JN. Automated tracing and volume measurements of neurons from 3-D confocal fluorescence microscopy data. *J Microsc* 1994;171:103–14.
- Denk W, Svoboda K. Photon upmanship: why multiphoton imaging is more than a gimmick. *Neuron* 1997;18:351–7.
- Duan H, Wearne SL, Morrison JH, Hof PR. Quantitative analysis of the dendritic morphology of corticocortical projection neurons in the macaque monkey association cortex. *Neuroscience* 2002;114:349–59.
- Elston GN, Rockland KS. The pyramidal cell of the sensorimotor cortex of the macaque monkey: phenotypic variation. *Cereb Cortex* 2002;12:1071–8.
- Elston GN, Tweedale R, Rosa MG. Cellular heterogeneity in cerebral cortex: a study of the morphology of pyramidal neurons in visual areas of the marmoset monkey. *J Comp Neurol* 1999;415:33–51.
- Feldman ML, Peters A. Technique for estimating total spine numbers on Golgi-impregnated dendrites. *J Comp Neurol* 1979;118:527–42.
- Freeman H. Computer processing of line drawing images. *Comput Surv* 1974;6(1):57–98.
- Human BT. Molecular and anatomical studies in Alzheimer's disease. *Neurologia* 2001;16(3):100–4.
- Koh IYY, Brent Lindquist W, Zito K, Nimchinsky EA, Svoboda K. An image analysis algorithm for dendritic spines. *Neural Comput* 2002;14:1284–310.
- Lohou C, Bertrand G. In: A new 3D 6-Subiteration thinning algorithm based on P-simple points, discrete geometry for computer imagery: 10th international conference, DGCI 2002; 2002.
- Ma CM, Sonka M. A fully parallel 3D thinning algorithm and its applications. *Comput Vision Image Understanding* 1996;64(3):420–33.
- Ma CM, Wan S. A medial-surface oriented 3D two-subfield thinning algorithm. *Pattern Recogn Lett* 2001;22(13):1439–46.
- Niblack W. An introduction to image processing. Englewood Cliffs, NJ: Prentice-Hall; 1986. pp. 115–6.
- Nikolaïdis N, Pitas I. 3D image processing algorithm. London: Wiley; 2001.
- Nimchinsky EA, Sabatini BL, Svoboda K. Structure and function of dendritic spines. *Annu Rev Physiol* 2002;64:313–52.
- O'Gorman L. Primitives chain code. In: Rosenfeld A, Shapiro LG, editors. *Progress in computer vision and image processing*. San Diego: Academic; 1992. p. 167–83.
- Oh W, Lindquist B. Image thresholding by indicator kriging. *IEEE Trans Pattern Anal Mach Intell PAMI-21* 1999:590–602.
- Page TL, Einstein M, Duan H, He Y, Flores T, Rolshud D, et al. Morphological alterations in neurons forming corticocortical projections in the neocortex of aged Patas monkeys. *Neurosci Lett* 2002;317:37–41.
- Palagyi K, Kuba A. A 3D 6-subiteration thinning algorithm for extracting medial lines. *Pattern Recogn Lett* 1998;19:613–27.
- Sauvola J, Pietaksinen M. Adaptive document image binarization. *Pattern Recogn* 2000;33:225–36.
- Segal M. Dendritic spines and long-term plasticity. *Nat Rev Neurosci* 2005;6:277–84.
- Seib LM, Wellman CL. Daily injections alter spine density in rat medial prefrontal cortex. *Neurosci Lett* 2003;337:29–32.
- Shen D, Ip HHS. A Hopfield neural network for adaptive image segmentation: An active surface paradigm. *Pattern Recogn Lett* 1997;18:37–48.
- Sonka M, Hlavac V, Boyle R. *Image processing, analysis, and machine vision*. 2nd ed. Brooks/Cole Publishing Company; 1999.
- Tier OD, Jain AK. Goal-directed evaluation of binarization methods. In: *IEEE Trans. Pattern Anal. Mach. Intell. PAMI-17*; 1995. p. 1191–201.
- 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;420:788–94.
- Weaver CM, Hof PR, Wearne SL, Brent Lindquist W. Automated algorithms for multiscale morphometry of neuronal dendrites. *Neural Comput* 2004;16:1353–83.
- White JM, Rohrer GD. Image thresholding for optical character recognition and other applications requiring character image extraction. *IBM J Res Dev* 1983;27(4):400–11.
- Xu X, Cheng J, Witt RM, Sabatini BL, Wong STC. A shape analysis method to detect dendritic spine in 3D optical microscopy image, *Biomedical Imaging: Macro to Nano*, 2006. In: *Proceedings of the 3rd IEEE International Symposium on April 6*; 2006. p. 554–7.
- Yanowitz SD, Bruckstein AM. A new method for image segmentation. *Comput Graph Image Process* 1989;46:82–95.
- Zhang Y, Zhou X, Degterev A, Lipinski M, Adjeroh D, Yuan J, et al. A novel tracing algorithm for high throughput imaging screening of neuron based assays. *J Neurosci Methods* 2007;160:149–62.
- Zhou X, Zhu J, Liu KY, Sabatini BL, Wong STC. Mutual information-based feature selection in studying perturbation of dendritic structure caused by TSC2 inactivation. *Neuroinformatics* 2006;4(1):81–94.